

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA  
E TECNOLOGIA FLUMINENSE**

**PROGRAMA DE PÓS-GRADUAÇÃO EM  
SISTEMAS APLICADOS À ENGENHARIA E  
GESTÃO**

**João Paulo Seixas Alves e Silva**

**IDENTIFICAÇÃO DE PARTÍCULAS SUBATÔMICAS EM  
ALTAS ENERGIAS MEDIANTE ALGORITMOS DE  
MACHINE LEARNING**

**Campos dos Goytacazes/RJ**

**2020**

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA  
E TECNOLOGIA FLUMINENSE

PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS APLICADOS À ENGENHARIA E  
GESTÃO

JOÃO PAULO SEIXAS ALVES E SILVA

IDENTIFICAÇÃO DE PARTÍCULAS SUBATÔMICAS EM ALTAS ENERGIAS  
MEDIANTE ALGORITMOS DE MACHINE LEARNING

Cristine Nunes Ferreira  
(Orientadora)

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação do Instituto Federal de Educação, Ciência e Tecnologia Fluminense, no Curso de Mestrado Profissional em Sistemas Aplicados à Engenharia e Gestão (MPSAEG), como parte dos requisitos necessários à obtenção do título de Mestre em Sistemas Aplicados à Engenharia e Gestão.

Campos dos Goytacazes/RJ

2020

Biblioteca Anton Dakitsch  
CIP - Catalogação na Publicação

S586i

Silva, João Paulo Seixas Alves e  
IDENTIFICAÇÃO DE PARTÍCULAS SUBATÔMICAS EM ALTAS  
ENERGIAS MEDIANTE ALGORITMOS DE MACHINE LEARNING /  
João Paulo Seixas Alves e Silva - 2020.  
138 f.: il. color.

Orientador: Cristine Nunes Ferreira

Dissertação (mestrado) -- Instituto Federal de Educação, Ciência e  
Tecnologia Fluminense, Campus Campos Centro, Curso de Mestrado  
Profissional em Sistemas Aplicados à Engenharia e Gestão, Campos dos  
Goytacazes, RJ, 2020.

Referências: f. 93 a 95.

1. Aprendizado de Máquina. 2. Experimentos em Altas Energias. 3.  
Física de Partículas. I. Nunes Ferreira, Cristine, orient. II. Título.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA FLUMINENSE  
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS APLICADOS À ENGENHARIA E  
GESTÃO

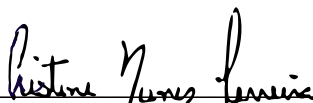
João Paulo Seixas Alves e Silva

IDENTIFICAÇÃO DE PARTÍCULAS SUBATÔMICAS EM ALTAS ENERGIAS  
MEDIANTE ALGORITMOS DE MACHINE LEARNING

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação do Instituto Federal de Educação, Ciência e Tecnologia Fluminense, no Curso de Mestrado Profissional em Sistemas Aplicados à Engenharia e Gestão (MPSAEG), como parte dos requisitos necessários à obtenção do título de Mestre em Sistemas Aplicados à Engenharia e Gestão.

Aprovado(a) em 29 de maio de 2020.

Banca Examinadora:



---

Cristine Nunes Ferreira, D.Sc  
Centro Brasileiro de Pesquisas Físicas (CBPF)  
(Orientadora)

ROGERIO ATEM DE  
CARVALHO:12067988824

Assinado de forma digital por ROGERIO  
ATEM DE CARVALHO:12067988824  
Dados: 2021.01.12 14:11:54 -03'00'

---

Rogério Atem de Carvalho, D.Sc  
Universidade Estadual do Norte Fluminense (UENF)

DALESSANDRO SOARES  
VIANNA  
dalessandrovianna@id.uff.br:0  
3703513713

Assinado de forma digital por  
DALESSANDRO SOARES VIANNA  
dalessandrovianna@id.uff.br:03703513  
713  
Dados: 2021.01.12 16:59:45 -03'00'

---

Dalessandro Soares Vianna, D.Sc  
Pontifícia Universidade Católica (PUC-Rio)

## DEDICATÓRIA

Dedico esta dissertação ao meu amado pai que nos deixou de forma inesperada e ainda no percurso deste estudo, o senhor nos faz enorme falta. Carrego em mim seus princípios e valores e a realização do seu trabalho de vida, nos direcionar aos estudos. O conhecimento é algo que ninguém pode nos tirar, sempre nos disse.

## AGRADECIMENTOS

Agradeço a Deus por me guiar e abrir portas dos seus caminhos a mim, serei eternamente grato. À minha família, amor e amigos que sempre dividiram e viram meus passos nesta jornada. A todos que ajudam manter o IFF, esta instituição tão importante na vida de muitos. A todos os professores da minha vida que me cederam parte do seu conhecimento para que eu construísse o meu, de todo meu coração, muito obrigado!

*“O homem científico não pretende alcançar um resultado imediato. Ele não espera que suas ideias avançadas sejam imediatamente aceitas. Seus trabalhos são como sementes para o futuro. Seu dever é lançar as bases para aqueles que estão por vir e apontar o caminho.*

*O dia em que descobriremos exatamente o que é a eletricidade, isso irá marcar um evento provavelmente maior, mais importante que qualquer outro na História da Humanidade. Então, será apenas uma questão de tempo para que o Homem consiga ligar suas máquinas diretamente à própria natureza.*

*Imagine o que está por vir...”*

Nikola Tesla

## RESUMO

*Machine Learning*, ou, aprendizado de máquina, é uma das áreas de grande destaque na ciência da computação atualmente. Nessa área, existem técnicas muito promissoras para trabalhar com um número muito grande de dados que podem ser usadas em uma gama de problemas incluindo a comprovação dos modelos de física de partículas. Os grandes empreendimentos envolvendo colaborações internacionais para o estudo das partículas fundamentais da natureza buscam por respostas para as grandes questões. Por este motivo, se construiu um grande colisor, que opera numa energia equivalente a energia de criação de partículas, importante não só para validação do modelo padrão, mas também para encontrar nova física, pois se acredita que o modelo padrão seja somente um modelo efetivo. Nesse sentido, o objetivo deste trabalho é a classificação das partículas geradas nesta colisão usando dois tipos de modelos do aprendizado de máquina: árvore de decisão e rede neural. O método utilizou base de dados vindas do LHCb passando por seleção das melhores *features* e otimização dos modelos para alcançar maior precisão. Um dos principais resultados alcançados é de que as redes neurais obtiveram uma precisão similar às árvores de decisão e que a utilização de todas as *features* mostrou-se mais proeminente nos modelos de árvores de decisão. Espera-se com esse trabalho possa elucidar o funcionamento dos principais modelos de *machine learning* e que sirva de referência futura para utilização de métodos computacionais a física de altas energias.

**Palavras-chave:** Aprendizado de Máquina, Experimentos em Altas Energias, Física de Partículas.



## **ABSTRACT**

Machine Learning is one of the areas of great prominence in computer science today. In this area, there are very promising techniques for working with many data that can be used in a range of problems including the verification of models of particle physics. Large enterprises involving international collaborations to study the fundamental particles of nature seek answers to major questions. For this reason, a large collider was built, which operates in an energy equivalent to the energy of particle creation, important not only for validating the standard model but also for finding new physics, as it is believed that the standard model is only an effective model. In this sense, the objective of this work is to classify the particles generated in this collision using two types of machine learning models: decision tree and neural network. The method used a database coming from the LHCb through the selection of the best features and optimization of the models to achieve greater precision. One of the main results achieved is that the neural networks obtained an accuracy similar to the decision trees and that the use of all features was more prominent in the decision tree models. It is hoped that this work can elucidate the functioning of the main models of machine learning and that it will serve as a future reference for the use of computational methods in high energy physics.

**Keywords:** Machine Learning, High Energy Physics Experiments, Particle Physics.

## LISTA DE FIGURAS

<b>Figura 1</b> - Modelo Padrão.....	20
<b>Figura 2</b> - Localização do LHC. ....	24
<b>Figura 3</b> - A configuração do LHCb. ....	26
<b>Figura 4</b> - Visão Geral do VELO LHC.....	27
<b>Figura 5</b> - Seção transversal no plano (x, z) dos sensores de silicose VELO, em $y = 0$ , com o detector na posição totalmente fechada. ....	28
<b>Figura 6</b> - Imagem à esquerda: layout esquemático do detector LHCb RICH1. A aceitação de $\pm 250$ mrad é indicada. Imagem à direita: vista esquemática de um HPD do sistema LHCb RICH.....	30
<b>Figura 7</b> - (a) Exemplo de característica de baixo nível: a soma da energia transversal ausente ( $\sum ET$ ). ....	37
<b>Figura 8</b> - (b) Exemplo de característica de alto nível: as massas dos candidatos de Higgs calculadas por um algoritmo de ajuste baseado em hipóteses (M $\pi\pi$ , MMC). ....	38
<b>Figura 9</b> - Ilustração da importância da permutação fracionária dos recursos de alto nível estimada usando a importância média em cinco modelos de Floresta Aleatória. ....	38
<b>Figura 10</b> - Ilustração da dependência de recursos e importância de recursos associados usando a regressão RandomForest. ....	39
<b>Figura 11</b> - Ilustração do cronograma de evolução da taxa de aprendizagem e momento usados durante o teste. ....	43
<b>Figura 12</b> - ROC 1º momento. ....	45
<b>Figura 13</b> - ROC 2º momento. ....	45
<b>Figura 14</b> - Resultados para energia dos neutrinos.....	47
<b>Figura 15</b> - Curvas ROC de eficiência de sinal vs. eficiência de fundo para a (esquerda) $\gamma$ vs. $\pi^0$ e (à direita) e vs. $\pi$ classificador. Os pontos vermelhos marcam o ponto de trabalho BDT escolhido. ....	48
<b>Figura 16</b> - Resolução de energia para fótons, elétrons, íons neutros e carregados em comparação com o modelo CNN vs. Linear.....	49
<b>Figura 17</b> - Comparação da largura do chuveiro transversal (esquerda) e largura do chuveiro longitudinal (direita) para Simulação GAN vs. Geant de elétrons com energias de 200 a 300 GeV.....	49
<b>Figura 18</b> - Componentes de um experimento “tradicional” de física de partículas. .	51
<b>Figura 19</b> - Exemplo da tabela com os dados experimentais do LHCb. ....	52

<b>Figura 20</b> - Exibição de evento LHCb.....	52
<b>Figura 21</b> - Comportamento das partículas no Imã (5) da Figura 21, em branco. ....	54
<b>Figura 22</b> - Partícula carregada de velocidade $v$ , passando através de um determinado meio (nesse caso um gás denso).....	55
<b>Figura 23</b> - Ilustração da recuperação de bremsstrahlung no LHCb. ....	59
<b>Figura 24</b> - Sistema completo de subdetectors LHCb. ....	60
<b>Figura 25</b> - Cascata eletromagnética.....	63
<b>Figura 26</b> - Foto-cintilador.....	64
<b>Figura 27</b> - Cascata Hadrônica.....	66
<b>Figura 28</b> - Ilustração de uma árvore de decisão. ....	72
<b>Figura 29</b> - Entropia de um lançamento de moeda.....	74
<b>Figura 30</b> - Entropia de um lançamento de moeda, fórmula e cálculo.....	74
<b>Figura 31</b> - Ganho de informação.....	75
<b>Figura 32</b> - Exemplo de rede neural .....	80
<b>Figura 33</b> - Esquema da Rede neural.....	81
<b>Figura 34</b> - Recorte da matriz de correlação. ....	86

## LISTA DE TABELAS

<b>Tabela 1:</b> Dados e decaimento das partículas.....	22
<b>Tabela 2:</b> Decaimento Múon.....	23
<b>Tabela 3:</b> Glossário de recursos de alto nível (derivados) usados para treinamento, juntamente com seu agrupamento.....	36
<b>Tabela 4:</b> Glossário de recursos de nível inferior (primário) usados para treinamento, juntamente com seu agrupamento.....	37
<b>Tabela 5:</b> Ilustração da importância da permutação fracionária dos recursos de alto nível estimada usando a importância média em cinco modelos de floresta aleatória.....	40
<b>Tabela 6:</b> Comparação de explorações de simetria de dados em termos de métricas de otimização e impacto no tempo.....	41
<b>Tabela 7:</b> Comparação das várias abordagens avançadas de montagem.....	42
<b>Tabela 8:</b> AUC 1º Momento.....	44
<b>Tabela 9:</b> AUC 2º Momento.....	45
<b>Tabela 10:</b> Parâmetros ligados aos subdetetores de rastreamento.....	54
<b>Tabela 11:</b> Os dados vindos do RICH.....	56
<b>Tabela 12:</b> Delta log (DLL) do RICH.....	57
<b>Tabela 13:</b> Parâmetros ligados ao Bremsstrahlung.....	60
<b>Tabela 14:</b> Informações dos cintiladores.....	62
<b>Tabela 15:</b> O Calorímetro Eletromagnético.....	64
<b>Tabela 16:</b> O Delta Log do Calorímetro Eletromagnético.....	65
<b>Tabela 17:</b> Controle de qualidade dos clusters do calorímetro.....	65
<b>Tabela 18:</b> O Calorímetro Hadrônico.....	66
<b>Tabela 19:</b> O Delta Log do Calorímetro Hadrônico.....	67
<b>Tabela 20:</b> Parâmetros do Detetor de Múons.....	68
<b>Tabela 21:</b> Ranking SelectKBest.....	84
<b>Tabela 22:</b> Ranking Extra Tree.....	84
<b>Tabela 23:</b> Ranking LightGBM.....	87
<b>Tabela 24:</b> Resultados da Random Forest.....	89
<b>Tabela 25:</b> Resultados da Gradient Boost Classifier.....	89
<b>Tabela 26:</b> Resultados da XGBoost.....	89
<b>Tabela 27:</b> Resultados da Rede Neural.....	89

## LISTA DE SIGLAS

**ALICE** - A Large Ion Collider Experiment  
**AMS** - Approximate Median Significance  
**ATLAS** - A Toroidal LHC Apparatus  
**AUC** - Area Under the Curve  
**BDT** - Boosted Decision Tree  
**CERN** - Conseil Européen pour la Recherche Nucléaire  
**CMS** - Compact Muon Solenoid  
**CNN** - Convolutional Neural Networks  
**CP** - Carga Paridade  
**DELPHI** - Detector with Lepton, Photon and Hadron Identification  
**DER** - Derived  
**DIRC** - Detection of Internally Reflected Cherenkov  
**DLL** - Delta Log Likelihood  
**DNN** - Deep Neural Networks  
**ECAL** - Electromagnetic Calorimeter  
**GAN** - Generative Adversarial Networks  
**GEM** - Gas Electron Multiplier  
**GeV** - Giga Electron Volt  
**HCAL** - Hadron Calorimeter  
**HEP** - High Energy Physics  
**HPD** - Hybrid Photon Detector  
**ID3** - Iterative Dichotomiser 3  
**IT** - Inner Tracker  
**LED** - Light Emitting Diode  
**LEP** - Large Electron–Positron  
**LHCb** - Large Hadron Collider Beauty  
**LR** - Learning Rate  
**MAE** - Mean Absolute Error  
**MAPA** - Mean Averaged Public AMS  
**MAPMTs** - Multi Anode Photomultipliers  
**MMVA** - Mean Maximal Validation AMS  
**MSE** - Mean Squared Error

**MVAC** - Mean Validation MAS at Cut  
**MWPC** - Multi Wire Proportional Chambers  
**NN** - Neural Networks  
**NOvA** - NuMI Off-axis ve Appearance  
**OT** - Outer Tracker  
**PID** - Particle Identification  
**PRI** - Primary  
**PRS** - Preshower  
**PV** - Primary Vertex  
**RICH** - Ring Imaging Cherenkov  
**ROC** - Receiver Operating Characteristic  
**SPD** - Scintillator Pad Detector  
**SPD/PS** - Scintillator Pad Detector Preshower  
**ST** - Silicon Tracker  
**SVM** - Support Vector Machine  
**TT** - Tracker Turicensis  
**VELO** - Vertex Locator  
**WLS** - Wave-Length Shifting

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>16</b>
1.1	OBJETIVO GERAL	17
1.2	OBJETIVO ESPECÍFICO	17
1.3	JUSTIFICATIVA	18
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>19</b>
2.1	O FUNCIONAMENTO DA FÍSICA DE PARTÍCULAS	19
2.2	MÉSONS E SEUS DECAIMENTOS	22
2.3	O GRANDE COLISOR DE HÁDRONS E O GRANDE VOLUME DE DADOS	24
2.4	O LHCb	25
2.4.1	<i>Sistema Localizador de Vértice (VELO)</i>	27
2.4.2	<i>Ring Imaging Cherenkov (RICH)</i>	29
2.4.3	<i>PDS/PRS</i>	31
2.4.4	<i>Calorímetros</i>	32
2.5	BÓSON DE HIGGS E APRENDIZADO DE MÁQUINA	33
2.5.1	<i>Métricas, base de dados e préprocessamento</i>	34
2.5.2	<i>Features Selection</i>	35
2.5.3	<i>Modelos, arquitetura e performance</i>	40
2.5.4	<i>Técnicas, hiperparâmetros, resultados e discussão</i>	41
2.6	REDES NEURAIS E ÁRVORES DE DECISÃO	43
2.6.1	<i>Redes neurais profundas, rasas e convolucionais</i>	46
2.6.2	<i>Redes neurais profundas, convolucionais, generativas e modelos lineares</i>	47
<b>3</b>	<b>METODOLOGIA</b>	<b>50</b>
3.1	A FÍSICA ENVOLVIDA NAS MEDIÇÕES	50
3.1.1	<i>O papel dos diferentes tipos de interação na identificação de partículas</i>	50
3.1.2	<i>Sistema de rastreamento</i>	53
3.1.3	<i>O detetor RICH</i>	55
3.1.4	<i>A emissão via efeito Bremsstrahlung</i>	58
3.1.5	<i>Calorímetros</i>	60
3.1.6	<i>SPD/PS</i>	62
3.1.7	<i>Calorímetro Eletromagnético</i>	63
3.1.8	<i>Calorímetro Hadrônico</i>	65
3.1.9	<i>Câmaras de Múons</i>	67
3.2	PID POR DETERMINAÇÃO DE MASSA	68
3.3	IDENTIFICAÇÃO DE PARTÍCULAS	69
3.3.1	<i>Base de Dados</i>	70
3.3.2	<i>Ferramentas e Bibliotecas</i>	70
3.3.3	<i>Seleção das colunas da tabela de dados (Features Selection)</i>	71
3.4	ÁRVORES DE DECISÃO	71
3.4.1	<i>Random Forest</i>	77
3.4.2	<i>Alguns hiper parâmetros</i>	77
3.5	REDE NEURAL	79
3.5.1	<i>Arquitetura</i>	82
3.5.2	<i>Otimização</i>	82
<b>4</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>83</b>

4.1	TRANSFORMAÇÃO DOS DADOS .....	83
4.2	FEATURES SELECTIONS .....	83
4.2.1	<i>SelectKBest</i> .....	83
4.2.2	<i>Extra Tree</i> .....	84
4.2.3	<i>Matriz de correlação</i> .....	85
4.2.4	<i>LightGBM Selection</i> .....	86
4.3	RESULTADOS PRINCIPAIS .....	87
4.4	MATRIZ DE CONFUSÃO E RELATÓRIO .....	90
<b>5</b>	<b>CONCLUSÕES .....</b>	<b>91</b>
	<b>REFERÊNCIAS .....</b>	<b>93</b>
	<b>APÊNDICE A .....</b>	<b>96</b>
	<b>ANEXO A.....</b>	<b>98</b>



# 1 INTRODUÇÃO

*Machine Learning*, ou, aprendizado de máquina, é uma das áreas de grande destaque na ciência da computação atualmente. A premissa de se gerar conhecimento a partir de técnicas de computação e matemática aplicada impulsiona esta tecnologia para aplicação em múltiplas áreas do conhecimento. Ferramentas como *Machine Learning* possuem a capacidade de lidar com grandes volumes de dados e gerar análise, conhecimento e conclusões para fins de classificação, regressão, agrupamento entre outras (NATARAJAN, 2014). O mundo passa por uma produção de dados nunca vista, gerada majoritariamente pela grande rede de computadores. A produção dessas quantidades massivas de dados advém de inúmeros tipos de dispositivos eletrônicos e, ao mesmo tempo, grandes iniciativas de pesquisas científicas internacionais também o fazem (COHEN; FREYTSIS; OSTDIEK, 2018).

Problemas iniciados na física de partículas procuram por soluções através de aprendizado de máquina afim de tratar enormes quantidades de dados geradas após colisões de prótons. O *Conseil Européen pour la Recherche Nucléaire* (CERN) é um dos maiores laboratórios para pesquisas em *High Energy Physics* (HEP), ou, física em altas energias. Atualmente o CERN produz cerca de 25 *petabytes* anualmente, o que torna seus desafios ainda maiores (KUO et al., 2014). O problema central de experimentos em altas energias é a correta identificação de partículas através de dados coletados por detetores. Para isto, a reconstrução de trajetórias, agrupamentos, chuveiros (*showers*), jatos (*jets*), anéis (*rings*) e outros, são feitos através de gravações das interações das partículas nos detetores onde depositam energias, direções, tamanho e outras propriedades após colisões de matérias.

Existem duas potenciais fontes de falhas na classificação de partículas: falhas na construção das propriedades dos dados brutos acarretando falhas na categorização dos eventos físicos (erro nos modelos de física teórica); falhas na determinação das propriedades utilizadas para caracterização dos eventos. Muitas vezes as informações utilizadas são limitadas, não oferecendo substancial teórico suficiente para a tarefa. Métodos que utilizam múltiplas variáveis, como *Fisher discriminates* e redes neurais, vem sendo utilizados em física de altas energias desde 1990 em experimentos no Fermilab<sup>1</sup> e CERN com notável sucesso. Porém, mais recentemente, com avanços em *Machine Learning*, novas ferramentas como *Boosted Decision*

---

<sup>1</sup> Fermilab é um laboratório especializado em física de partículas de alta energia do Departamento de Energia dos Estados Unidos.

*Tree* (BDT), *Support Vector Machines* (SVM), *Generative Neural Network* (GNN) dentre outras foram desenvolvidas e estão sendo utilizados para identificação de partículas em diversos experimentos (ADAM-BOURDARIOS et al., 2015).

A utilização de redes neurais e BDT para classificação de partículas demonstram bastante eficiência e são utilizadas com frequência em diversas pesquisas. Conforme Zoe et al. (2005), a procura por neutrinos em experimento no Fermilab obteve resultados satisfatórios com ambos algoritmos, porém quando são utilizadas menos variáveis as redes neurais performam ligeiramente piores do que os métodos de impulsionamento (*boosted*). Em trabalho mais recente, Aurisiano et al. (2016) descreveram uma aplicação de *Convolutional Neural Network* (CNN), ou redes neurais convolucionais, para identificação de interações de neutrinos em calorímetros. Com uma reconstrução mínima do evento, o algoritmo conseguiu de forma excelente separar elétrons e múons dos neutrinos, classificando assim os neutrinos.

## 1.1 Objetivo geral

Este trabalho objetiva a classificação de partículas a partir de uma base de dados higienizada, ou seja, todo o processo de preparação de dados como extração, transformação e limpeza já feita. A base de dados possui dados coletados no detetor LHCb (*Large Hadrons Collider beauty*) que possui subdetetores para interagir com a matéria colidida em seu centro.

## 1.2 Objetivo específico

- Entender a síntese dos dados vindos do LHCb bem como seus detetores.
- Comparar performance dos modelos de aprendizado de máquina
- Avaliar técnicas de seleção de *features* via matriz de confusão e seus relatórios.
- Otimização dos modelos e seus hiperparâmetros afim de alcançar melhor precisão.
- Classificar partículas subatômicas, são elas: píons, káons, elétrons, múons e prótons.

### 1.3 Justificativa

A classificação de partículas em experimentos de altas energias é uma tarefa que inclui uma grande quantidade de ferramentas, dependendo da solução proposta os meios são diversos. Trabalhar com experimentos deste porte ajuda na evolução das técnicas e modelos de aprendizado de máquina. É importante também interligar várias áreas da ciência que podem se complementar e comprovar eficácia de ambos os lados, neste caso, física e computação. Isto faz com que os resultados vindos da física sejam interpretados de forma mais eficiente. Com isso cria-se ferramenta para validar modelos teóricos a partir de dados experimentais podendo descartar modelos não pertinentes. Sendo um modelo padrão um modelo efetivo que não inclui a gravitação serve também para se encontrar nova física a partir de novas partículas.

Essa dissertação foi construída com a seguinte organização: No Capítulo 1 encontra-se a introdução com a justificativa, objetivo gerais e específico. O referencial teórico está no Capítulo 2. O método e os materiais utilizados para a construção da metodologia seguem no Capítulo 3. Os resultados principais da classificação das partículas estão no Capítulo 4 e a conclusão destes na Capítulo 5. Em seguida as referências e por fim o Apêndice A e Anexo A com os resultados detalhados.

## 2 REFERENCIAL TEÓRICO

O objetivo deste capítulo é estudar os principais métodos usados para classificar inúmeros processos envolvendo muitos dados. Este trabalho se concentrou nos métodos para classificação de partículas do grande acelerador de partículas em inglês *Large Hadron colider* (LHC). Também será tratado os principais resultados sobre as vantagens de se trabalhar com o aprendizado de máquina para a identificação de partículas elementares. Partindo do pressuposto que para aplicar uma ferramenta a problemas de física há de saber um pouco sobre o tema em si. Tal conteúdo também será discutido brevemente abordando os principais elementos da física de partículas necessários para o entendimento dos processos de classificação de partículas e o entendimento dos dados experimentais.

### 2.1 O Funcionamento da Física de Partículas

A Física de partículas nasceu com a própria curiosidade humana de compreender a natureza da matéria. Teve a participação de vários cientistas e inventores ao longo dos anos. Pode-se citar Joseph John Thomson (1856-1940), que descobriu o elétron em 1897 e ganhou o Prêmio Nobel de Física de 1906. Em 1886, Ernest Rutherford (1871-1937) e Elgen Goldstein (1850-1930) descobriram os prótons. Durante o experimento Goldstein detectou um feixe de luz no sentido oposto ao dos elétrons, ele propôs a existência de partículas com carga oposta (positiva) no sistema. No entanto até então não se sabia o modelo atômico, somente em 1911, Ernest Rutherford o descobriu. Esse modelo foi mais tarde aperfeiçoado por Niels Bohr, provando a sua estabilidade, admitindo certas características impostas que só seriam elucidadas com o advento da Mecânica Quântica. Apesar da descoberta de um núcleo positivo feita em 1911, somente em 1932 o físico inglês James Chadwick descobriu o nêutron ganhando o prêmio Nobel de 1935 por essa descoberta. Desta forma, para saber a quantidade de nêutrons que um átomo possui, basta fazer a subtração entre o número de massa ( $A$ ) e o número atômico ( $Z$ ) (ALVES, 2008). Apesar de todo esse desenvolvimento, ainda existia uma pergunta na cabeça dos cientistas, sabendo da interação eletromagnética, que dentro do núcleo só existia até o momento prótons e nêutrons e que prótons tem carga positiva e nêutrons não tem carga, porque

então o núcleo atômico não se repelia. Essa questão só foi entendida com o advento da Mecânica Quântica e o estudo das radiações nucleares e das colisões entre partículas (CALVO, 2015). Foi descoberto que o núcleo atômico possuía mais do que prótons e nêutrons. As descobertas de novas partículas feitas a partir do século XX e a ideia de que seriam partículas elementares ocorreram por volta de 1950, com um novo ramo da Física denominado Modelo Padrão (LOPES,1993). Apesar dos trabalhos com o modelo padrão, esses eram trabalhos profundamente teóricos, somente em 1964 o modelo do quark foi proposto de forma independente pelos físicos Murray Gell-Mann e George Zweig em 1964 (HODDESON, BROWN, RIORDAN, DRESDEN, 1997). Os quarks foram introduzidos como parte de um esquema de organização dos hádrons, e havia pouca evidência de sua existência física até os experimentos de Espalhamento Inelástico Profundo no Centro de Aceleração Linear de Stanford em 1968 (CHEKALINA, RATNIKOV,2008; MODEL GYM, 2019 ). Experimentos com os aceleradores forneceram evidências para todos os seis sabores de quarks. O quark top, observado pela primeira vez no Fermilab em 1995, foi o último a ser descoberto (HODDESON, BROWN, RIORDAN, DRESDEN, 1997). A descoberta em 1975 do méson que veio a ser denominado de  $J/\psi$  levou ao reconhecimento que ele seria composto de um quark charmoso e um antiquark. O quark inferior (*down*) foi descoberto em 1980 e o quark superior (*up*) em 1996 no acelerador de partículas Tevatron, do Fermilab (HODDESON, BROWN, RIORDAN, DRESDEN, 1997).

O modelo padrão das partículas elementares prevê a existência de uma série de partículas elementares como mostra a tabela da Figura 1:

mass →	≈2.3 MeV/c <sup>2</sup>	≈1.275 GeV/c <sup>2</sup>	≈173.07 GeV/c <sup>2</sup>	0	≈126 GeV/c <sup>2</sup>
charge →	2/3	2/3	2/3	0	0
spin →	1/2	1/2	1/2	1	0
	<b>u</b> up	<b>c</b> charm	<b>t</b> top	<b>g</b> gluon	<b>H</b> Higgs boson
<b>QUARKS</b>	≈4.8 MeV/c <sup>2</sup>	≈95 MeV/c <sup>2</sup>	≈4.18 GeV/c <sup>2</sup>	0	
	-1/3	-1/3	-1/3	0	
	1/2	1/2	1/2	1	
	<b>d</b> down	<b>s</b> strange	<b>b</b> bottom	<b>γ</b> photon	
	0.511 MeV/c <sup>2</sup>	105.7 MeV/c <sup>2</sup>	1.777 GeV/c <sup>2</sup>	91.2 GeV/c <sup>2</sup>	
	-1	-1	-1	0	
	1/2	1/2	1/2	1	
	<b>e</b> electron	<b>μ</b> muon	<b>τ</b> tau	<b>Z</b> Z boson	
<b>LEPTONS</b>	<2.2 eV/c <sup>2</sup>	<0.17 MeV/c <sup>2</sup>	<15.5 MeV/c <sup>2</sup>	80.4 GeV/c <sup>2</sup>	
	0	0	0	±1	
	1/2	1/2	1/2	1	
	<b>ν<sub>e</sub></b> electron neutrino	<b>ν<sub>μ</sub></b> muon neutrino	<b>ν<sub>τ</sub></b> tau neutrino	<b>W</b> W boson	
				<b>GAUGE BOSONS</b>	

**Figura 1**– Modelo Padrão.

Fonte: MISSMJ (2019).

Nessa Figura 1, tem-se as partículas elementares que, pelo modelo padrão, estão ligadas as três forças fundamentais: eletromagnética, nuclear forte e nuclear fraca. Por incompatibilidade de escala e pelo fato de não se ter uma teoria quântica para a gravitação, não se pode unificar a força gravitacional a física de partículas. Os quarks mais usuais, são, os quarks “*up*” e “*down*”, eles são os constituintes dos prótons e nêutrons. O próton é constituído por dois quarks “*up*” e um “*down*” e o nêutron de dois quarks “*down*” e um “*up*”. Essas partículas que são compostas de quarks são chamadas de Hádrons. O quark *bottom* ou quark *b* é um quark de terceira geração, de carga  $-1/3$ . Ele corresponde ao segundo quark mais pesado, com uma massa de 4,2 GeV que é quatro vezes o próton (ABDALA, 2006). O decaimento do quark *bottom* ocorre através da interação fraca em  $\pm 10^{-12}$  segundos, sendo a partícula mais instável, e ele decai num quark *up* ou um quark *charm* *c*, da tabela, por esse motivo existe um detector somente para ele, que é o LHCb. Com o estudo dos raios cósmicos, que são partículas altamente energéticas muitas vezes vinda de explosões de supernovas ou outros cataclismas existentes no universo, foi revelado a existência de um novo bárion, que apresentava um tempo de vida previsto de  $10^{-23}$  segundos, no entanto tal bárion sobreviveu por  $10^{-10}$  segundos revelando um tempo de desintegração maior que o esperado para essas partículas, Delta D. Desta forma os cientistas atribuíram a essa anomalia, o nome de estranheza. A partícula portadora desta estranheza, foi apelidada de quark estranho. Outra espécie de partículas elementares, ainda com spin semi-inteiro, denominada de férmions são os léptons, são partículas que não contém quarks, entre elas temos o elétron, conhecida por todos, o múon, que tem uma massa entre o elétron e o próton, um tempo de decaimento grande o bastante para chegarem ao nível do mar. Outra partícula interessante são os neutrinos, que tem de três tipos, os neutrinos eletrônicos, cuja a maior fonte é o sol; os neutrinos tauônicos e muônico. Esses outros foram importante para o estudo da massa do neutrino, pois a partir do sol, com nenhuma outra fonte detectada, partiam neutrinos eletrônicos, mas chegava a Terra uma quantidade grande de neutrinos muônicos e tauônicos, sugerindo que o neutrino estivesse oscilando, esse problema foi denominado “O problema do Neutrino Solar”, e ocasionou mudanças drásticas no modelo padrão das partículas elementares, onde o neutrino era considerado sem massa. Hoje, considera-se o neutrino com massa, existem experimentos que detectaram essa oscilação, o Kamiokande e Super Kamiocande. Esse assunto ainda é ponto de pesquisa. Tendo esgotado as partículas elementares de spin semi-inteiro e importante falar das partículas de spin inteiro, chamada de bósons. A

principal e mais conhecida partícula bosônica é o fóton, ela é a mediadora da interação eletromagnética, não possui nem carga nem massa de repouso, tendo somente massa relativística. Tem-se ainda os glúons que são mediadores da força forte e temos os bósons W e Z que são os mediadores da interação fraca e por último tem-se o famoso bóson de Higgs. Esse bóson é o responsável pelas quebras espontâneas de simetria responsável pela massa das partículas, uma atenção que se tem que ter é que a massa é uma propriedade fundamental da matéria, não resolve o problema da gravitação quântica, e sua interação não é considerada no modelo padrão das partículas elementares (HAUZEN, MAERTIN,1984).

## 2.2 Mésons e seus decaimentos

Nesta seção é importante revisar os decaimentos das partículas que serão trabalhadas neste documento, que são os decaimentos do méson Pi ( $\pi$ ) (ou pión) e o méson Kappa (K) (ou káon) , que pode ser visto na Tabela 1.

**Tabela 1**– Dados e decaimento das partículas.

<b>Partícula</b>	$\pi^+$	$\pi^0$	$K^+$	$K^+$	$K^0$	$K^0$
<b>Anti Partícula</b>	$\pi^-$	$\pi^0$	$K^-$	$K^-$	$K^0$	$K^0$
<b>Lê-se</b>	pi +	Pi 0	Kappa-	Kappa+	Kappa0	Kappa0
<b>Carga</b>	+1	0	+1	+1	0	0
<b>Spin</b>	0	0	0	0	0	0
<b>Estranheza</b>	0	0	+1	+1	+1	+1
<b>Charme</b>	0	0	0	0	0	0
<b>Beleza</b>	0	0	0	0	0	0
<b>(MeV)</b>	140	135	494	494	498	498
<b>MeiaVida (s)</b>	$2,6 \times 10^{-8}$	$8,52 \times 10^{-17}$	$1,24 \times 10^{-8}$	$1,24 \times 10^{-8}$	$0,90 \times 10^{-10}$	$0,90 \times 10^{-10}$
<b>Decaimentos</b>	$\mu^+ \nu_\mu$	$2 \gamma$	$\mu^+ \nu_\mu$	$\pi^+ \pi^0$	$\pi^0 \pi^0$	$\pi^+ \pi^-$
<b>Predominância</b>	99,99%	98,80%	63,44%	20,92%	30,69%	69,20%

Fonte: Elaboração própria (TANABASHI, 2018).

Pode ser visto que existem mais de uma possibilidade, que conserva a carga. Pode ser observado também que o tempo de vida dessas partículas é baixo. Os káons tem meia vida

muito baixa comparada com a do pión, por este e outros motivos, que para classificar direito essas partículas temos que remover o fundo de pions que podem coexistir.

O méson  $\pi$ , ou pión, foi descoberto por Cesare Mansueto Giulio Lattes que deu o prêmio Nobel a Cecil Frank Powell em 1950. Essa descoberta, incentivou muitos pesquisadores nos estudos dos chamados raios cósmicos, basicamente composto de prótons, que são partículas, cuja principal fonte são as explosões de supernovas que chegam a Terra e colidem com as partículas do ar. Dessas colisões, produz-se uma profusão de partículas hoje observadas com mais detalhes no LHC, são basicamente: pions, káons etc., que decaem no final em múons, elétrons e fótons estes últimos podendo ser observados a nível do mar. O méson K pode ser de três tipos  $K^+$ ,  $K^-$  e  $K^0$ . São os mésons mais leves que possuem um quark s (ou antiquark s). Possuem spin nulo (bósons, portanto). Descobertos em 1964, pelo Laboratório Nacional de Brookhaven.

Essas partículas como os mésons B violam a simetria CP, pois os anti- $K^0$  transformam-se em  $K^0$  com uma frequência um pouco menor que o inverso.

A simetria CP, é uma simetria quase exata das leis da natureza sobre o efeito da transformação entre partículas em anti-partículas, a assim chamada conjugação de Carga, e a inversão das coordenadas espaciais, a Paridade. Como no exemplo da imagem, um elétron UP torna-se um pósitron Down a simetria CP inverte todos os eixos espaciais e transforma partículas em anti-partículas.

A ideia da simetria CP surgiu quando da descoberta da violação da paridade em certas reações de radioatividade nos anos 1950, mas só foi realmente estabelecido em 1964 que a interação fraca violava esta simetria. É a isto que se chama a violação da simetria CP e a descoberta do decaimento do méson neutro K. Os méritos foram a James Cronin e a Val Fitch o Prêmio Nobel de Física em 1980 (TAVARES, 2018).

O múon também decai conforme pode-se ver na Tabela 2, mas como pode ser visto a meia vida é muito alta e pode atravessar o detector sem decair. As colunas “Anti”, “MeV” e “Pred” referem-se respectivamente a anti-partícula, energia de repouso e a predominância.

**Tabela 2** – Decaimento Múon.

Partícula	Anti	Lê-se	Carga	Spin	MeV	Meia vida	Decaimentos	Pred
$\mu^+$	$\mu^-$	Muon	+1	+1/2	105,66	$2,20 \times 10^{-6}$	$e^+ \nu_e \nu_\mu$	100%
$e^-$	$e^+$	elétron	-1	+1/2	0,511	$4,6 \times 10^{26}$	$2\gamma$	100%

Fonte: Elaboração própria, informação retirada de (TANABASHI, 2019).



Ainda nesta Tabela 2 pode ser visto que o elétron tem uma meia vida notavelmente maior que todas essas partículas.

Além dessas possibilidades existem outras extremamente raras que podem ser testadas no acelerador, classificando esses decaimentos como *ghosts*. Os métodos de classificação serão revisados na seção 3.

### 2.3 O Grande Colisor de Hádrons e o grande volume de dados

A compreensão a respeito da constituição fundamental da matéria obteve evolução significativa nos últimos anos devido a comprovações, resultantes de experimentos de física de altas energias. O Grande Colisor de Hadrons (*Large Hadron Collider - LHC*) (FIYNN,2015) é o maior acelerador de partículas em operação atualmente situado no CERN (Organização Europeia para a Pesquisa Nuclear) (KARAVAKIS, 2014).

O objetivo do LHC é analisar a estrutura fundamental da matéria, investigar as propriedades das partículas fundamentais propostas no Modelo Padrão e também buscar por fenômenos desconhecidos (KARAVAKIS, 2014). Para tanto, o LHC conta com alguns detectores de partículas como ATLAS (*A Toroidal LHC ApparatuS*), ALICE (*A Large Ion Collider Experiment*), CMS (*Compact Muon Solenoid*) e LHCb (*Large Hadron Collider beauty*).

O laboratório localiza-se em um túnel de 27 km de circunferência, com colisões ocorrendo a taxa de até  $40 \times 10^6$  vezes por segundo bem como a 175 metros abaixo do nível do solo na fronteira franco-suíça, próximo a Genebra, Suíça.



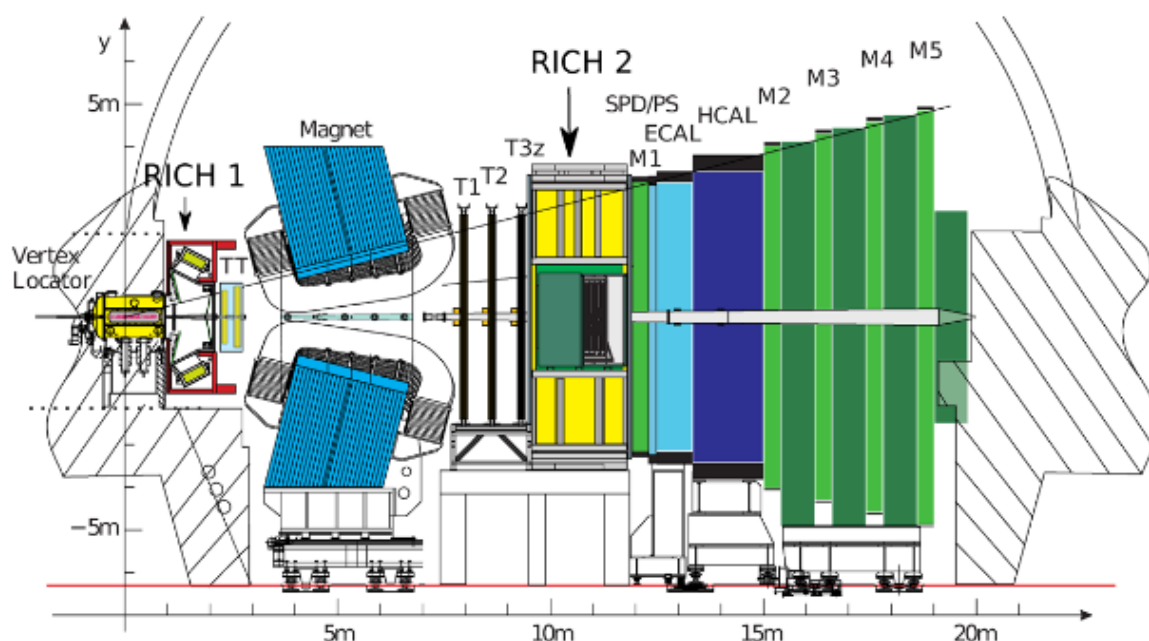
**Figura 2 - Localização do LHC.**

Fonte: <https://mc2h2o.blogspot.com/2009/02/o-maior-e-mais-complexo-instrumento.html>

## 2.4 O LHCb

O LHCb é um experimento dedicado ao estudo da física dos sabores para partículas pesadas (*heavy flavour*) no LHC (EVANS, 2008; ANDRONIC, WESSELS, 2012). Seu objetivo principal é procurar evidências indiretas de nova física em violações de carga e paridade (CP), e decaimentos de hádrons beleza e charme. Os recentes resultados, em física de sabor pesado obtidos nas fábricas B e em Tevatron (segundo maior acelerador já construído) são, até agora, totalmente compatíveis com o CKM mecanismo cujo objetivo é conter informações acerca das probabilidades de mudança de sabor de um quark. Por outro lado, o nível de violação de CP nas interações fracas do Modelo Padrão não pode explicar a quantidade de matéria no universo. Uma nova fonte de violação de CP além do modelo padrão é, portanto, necessária para resolver esse quebra-cabeça. Com muita precisão aprimorada, o efeito de uma nova fonte pode ser visto na física dos sabores pesados. De fato, muitos modelos de nova física produzem contribuições que alteram as expectativas das fases de violação do CP, frações raras de ramificação do decaimento e podem gerar modos de decaimento proibidos no Modelo Padrão.

Do ponto de vista da máquina, o Grande Colisor de Hádrons b em inglês, *Large Hadron Collider b* (LHCb) é um espectrômetro de braço único com uma cobertura angular direta de aproximadamente 10 mrad a 300 (250) mrad no plano de flexão (sem flexão). A escolha da geometria do detector é justificada pelo fato de que, em altas energias, ambos os b- e b-hádrons são predominantemente produzidos no mesmo cone para frente ou para trás. O *layout* do espectrômetro LHCb é mostrado na Figura 3. O sistema de coordenadas destre adotado apresenta os eixos ao longo da viga e os eixos na vertical. Ponto de interseção 8 do LHC, usado anteriormente pelo experimento DELPHI durante a LEP foi alocado para o detector LHCb. Uma modificação na ótica do LHC, deslocando o ponto de interação por 11,25 m do centro, permitiu o máximo uso da caverna existente para os componentes do detector LHCb.



**Figura 3** - A configuração do LHCb.

Fonte: LIPPMANN (2012).

O LHCb é composto por vários módulos o primeiro na Figura 3 é o sistema localizador de vértices que inclui um contador de vetor acumulado (VELO). Apresenta também dois anéis foto-cintilantes de Cherenkov (RICH1 e RICH2) usando aerogel  $C_4F_{10}$  e  $CF_4$  como radiadores, para obter uma perfeita separação do méson ( $\pi$ ) com o méson chamado Kaon (K). Essa separação ocorre na faixa de momento de 2 a 100 GeV/c. Logo após do RICH1 tem-se um sistema de rastreamento feito de um *Tracker Turicensis* (TT) que é um detector de micro faixas de silicone, que volta a se repetir depois do imã e antes do RICH2, como três faixas de rastreamento  $T_1$ ,  $T_2$  e  $T_3$  que também é um detector feito com micro faixas de silicone nas partes internas (IT) e no exterior, o detector (OT) é feito de canudos Kapton/Al. Entre o RICH1 e o *Tracker Turicensis*(TT), tem-se um espectrômetro magnético que é um imã configurado como dipolo, quente e que fornece um campo integrado de 4 Tm. Os detectores híbridos de fóton. O sistema calorimétrico composto por um detector cintilador e pré-banho (SPD/PS). Um calorímetro eletromagnético (tipo *shashlik*) (ECAL) e um calorímetro hadrônico (telhas de Fe e cintilador) (HCAL). O sistema de detecção de múons composto por MWPC (exceto na região de taxa mais alta, onde são utilizados os GEM triplos) (LIPPMANN, 2012).

### 2.4.1 Sistema Localizador de Vértice (VELO)

O sistema de rastreamento do LHCb é um localizador de vértices (VELO) e quatro estações de rastreamento: o *Tracker Turicensis* (TT) a montante do ímã dipolo e T1-T3 a jusante do ímã. O VELO e o TT usam detectores de micro-tira de silicone. Em T1-T3, são utilizadas micro-tiras de silicone na região próxima ao tubo do feixe (*Inner Tracker*, IT), enquanto os tubos de palha são empregados na região externa das estações (*Outer Tracker*, OT). O TT e o TI foram desenvolvidos em um projeto comum chamado de *Silicon Tracker* (ST). Na Figura 4, tem-se uma visão geral do VELO.

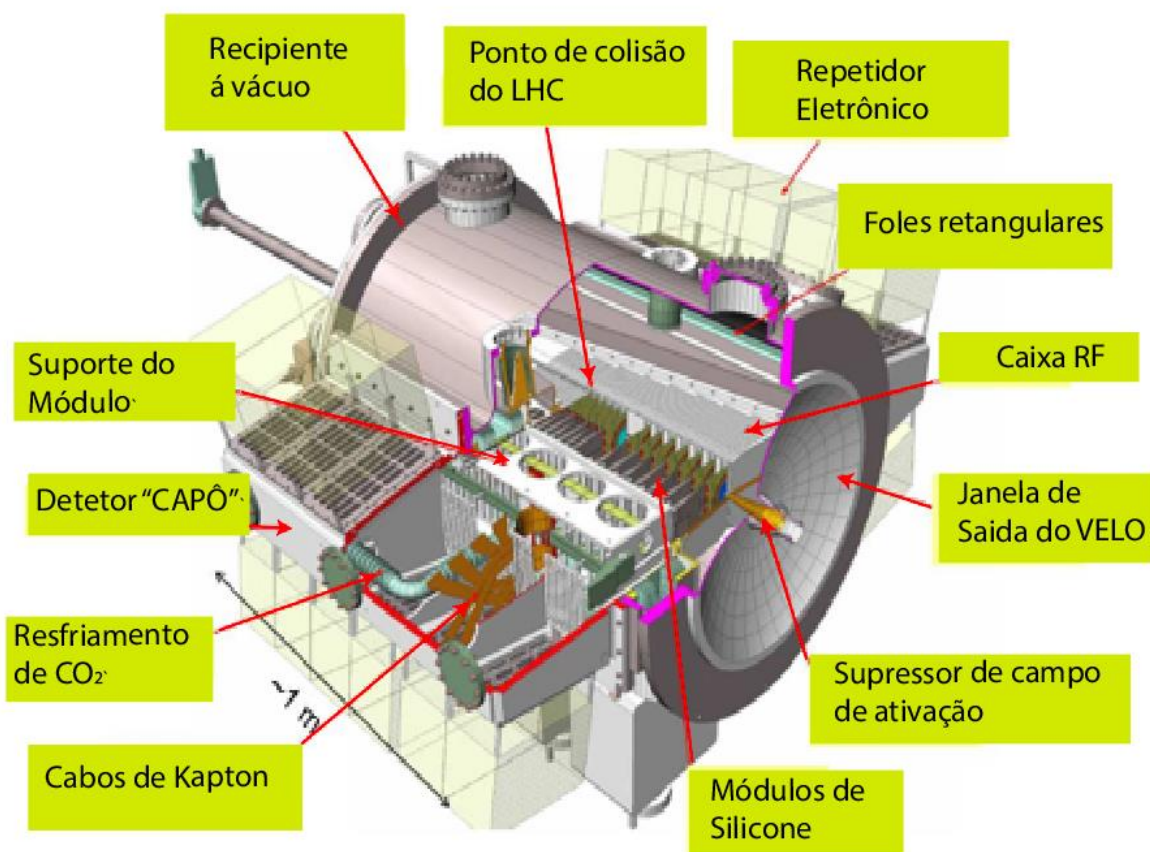


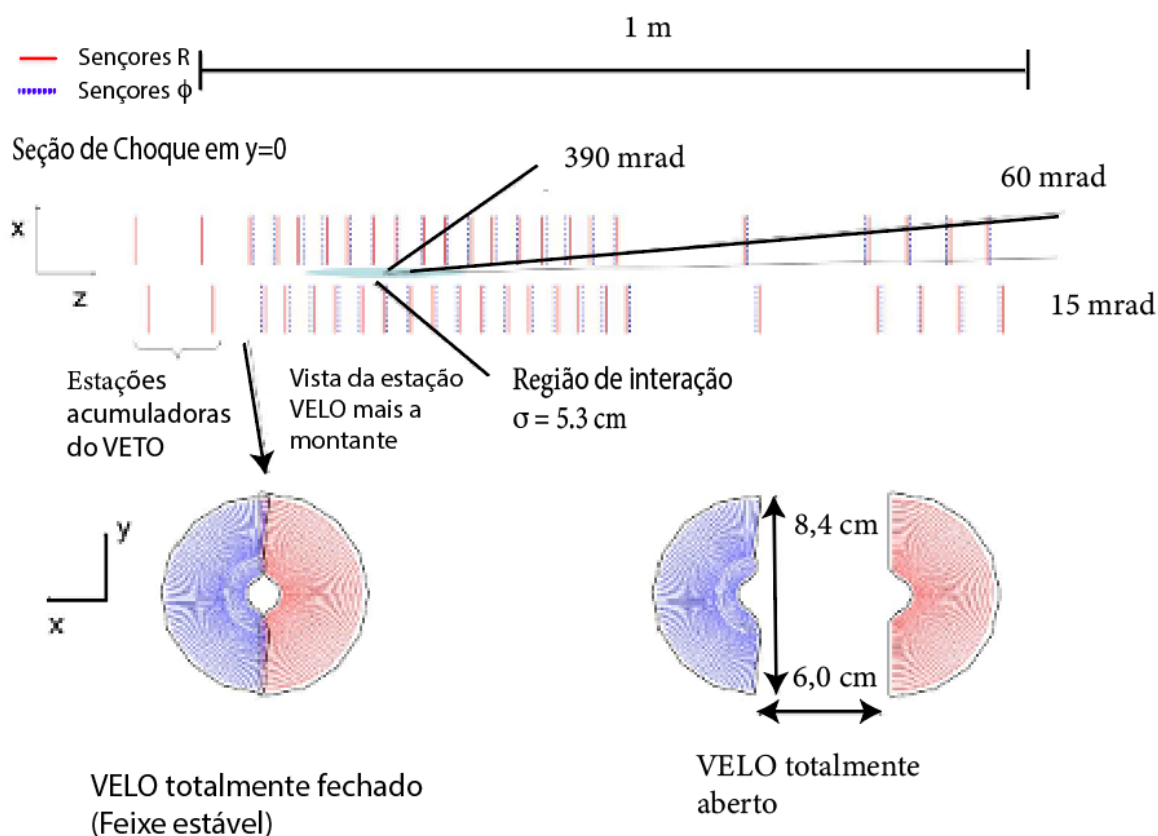
Figura 4 - Visão Geral do VELO LHC.

Fonte: Adaptado de Evans, 2008 e usado em Andronic e Wessels, 2012.

O VERTex LOcator (VELO) faz medições das coordenadas da pista próximas da região de interação, esse arranjo é necessário para identificar os vértices secundários deslocados, que são

uma característica distintiva dos decaimentos dos hádrons envolvendo os quarks “b” e quarks “c” onde o rótulo b (quark beleza) e c (quark charme), como visto na seção 2.1.

O VELO é composto de uma série de módulos de silício, cada um fornecendo uma medida das coordenadas  $r$  e  $\phi$  dispostas ao longo da direção do feixe como mostra a Figura 5.



**Figura 5** - Seção transversal no plano  $(x, z)$  dos sensores de silício VELO, em  $y = 0$ , com o detector na posição totalmente fechada.

Fonte: Adaptado de Navarro, 2012

A face frontal dos primeiros módulos também é ilustrada nas posições fechada e aberta. As duas estações de veto estão localizadas a montante dos sensores VELO.

Dois planos perpendiculares à linha do feixe e localizados a montante dos sensores VELO são chamados de sistema de estacas de veto-up. Os detectores são montados em um recipiente que mantém o vácuo ao redor dos sensores e são separados do vácuo da máquina por uma folha de alumínio corrugada de paredes finas. Isso é feito para minimizar o material atravessado por uma partícula carregada antes de cruzar os sensores.

O VELO deve abranger a aceitação angular dos detectores a jusante, ou seja, detectar partículas com uma pseudo-rapidez na faixa  $21,6 < \eta < 4,9$  e emergir dos vértices primários na faixa  $|z| < 10,6$  cm.

#### 2.4.2 Ring Imaging Cherenkov (RICH)

Em 1958, P. A. Cherenkov, I. Y. Tamm e I. M. Frank ganharam o prêmio Nobel em Física devido a descoberta e interpretação do efeito Cherenkov. A radiação de Cherenkov é uma onda de choque resultante de uma partícula carregada, que se move através um material, com uma velocidade maior que a velocidade da luz no meio. Em geral, os detectores Cherenkov contêm dois elementos principais: um radiador através do qual as partículas carregadas passam (um meio dielétrico transparente) e um detector de fótons (LIPPMANN, 2012).

Como a radiação de Cherenkov é uma fonte fraca de fótons, a transmissão da luz, a coleta e a detecção deve ser o mais eficiente possível. Existem três tipos diferentes de contadores de Cherenkov que podem ser distinguidos:

1. Os contadores de limiar medem a intensidade da radiação de Cherenkov e são utilizados para detectar partículas com velocidades superiores ao limiar  $\beta_\tau$ . Uma estimativa grosseira da velocidade da partícula acima do limiar é dada por um forte pulso que é medido no detector de fótons.
2. Os contadores diferenciais focam apenas os fótons de Cherenkov com uma determinada emissão ângulo no detector e, assim, detectar partículas em um intervalo estreito de velocidades.
3. Os detectores de imagem Cherenkov utilizam ao máximo as informações disponíveis (Ângulo de Cherenkov e número de fótons) e pode ser dividido em dois categorias principais: RICH (*Ring Imaging Cherenkov*) e DIRC (Detecção de Dispositivos de luz Cherenkov refletidos internamente).

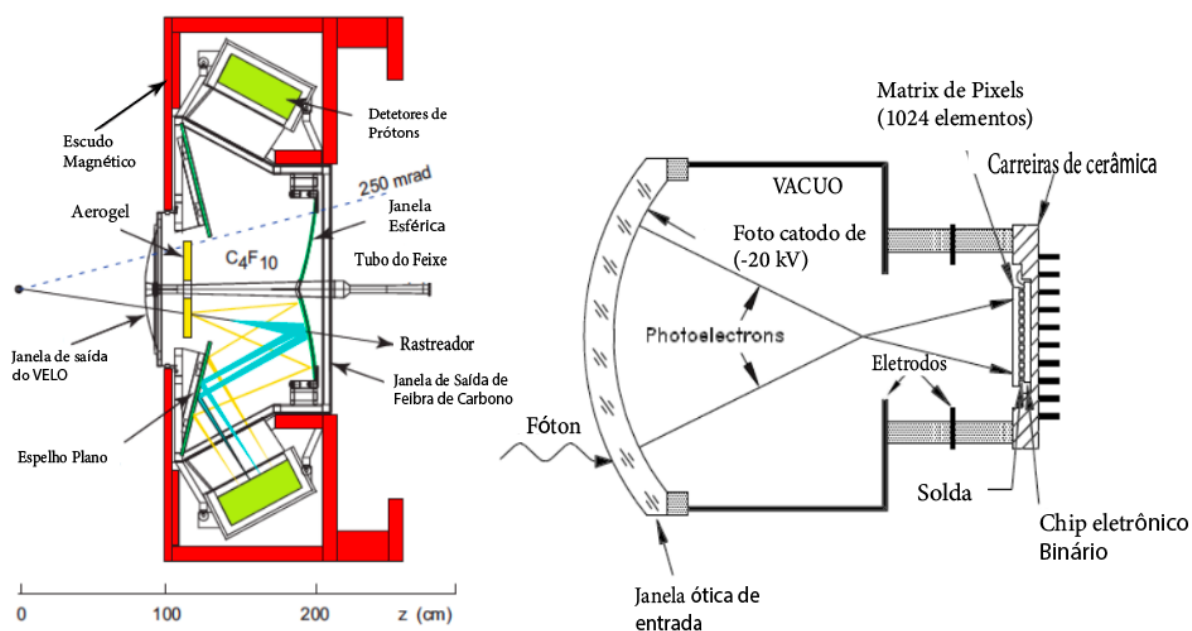
A partir do conhecimento do momento da partícula  $p$  e do ângulo  $\Theta_c$  de Cherenkov, uma determinação da massa da partícula carregada é possível.



Em um dispositivo RICH, a radiação Cherenkov é emitida no radiador e coletada por um detector de fótons, geralmente após serem transmitidos por meios ópticos. O primeiro RICH foi desenvolvido por A. Roberts em 1960 (ROBERTS, 1960).

O detector RICH foi projetado para medir velocidades em um intervalo de momento especificado usando um radiador Cherenkov com índice de refração ( $n$ ) escolhido de forma que o ângulo de Cherenkov varie com a velocidade, do limiar ao momento ( $p$ ) mais alto esperado.

Os detectores RICH1 e RICH2 da Figura 6 são caracterizados por um design que coloca os fotodetectores fora da aceitação do espectrômetro LHCb, a fim de limitar a degradação da resolução dos sistemas de rastreamento devido à interação com o material detector. Um conjunto de espelhos esféricos e planos projeta a luz Cherenkov no plano do detector. Os espelhos esféricos são colocados dentro da aceitação do espectrômetro e foram tomados cuidados especiais para minimizar a fração do comprimento da radiação e, ao mesmo tempo, garantir a integridade mecânica da óptica. O espelho esférico de RICH1 é feito de fibra de carbono. O layout esquemático do RICH1 é mostrado na Figura 6.



**Figura 6** - Imagem à esquerda: layout esquemático do detector LHCb RICH1. A aceitação de  $\pm 250$  mrad é indicada. Imagem à direita: vista esquemática de um HPD do sistema LHCb RICH.

Fonte: Ambos os números são adaptados da referência Augusto e Alves, 2008.

Nos dois detectores, são utilizados radiadores gasosos. No RICH1, um segundo radiador mais denso feito de sílica aerogel também é instalado. Juntos, os três radiadores cobrem a grande faixa de momento necessária. Os ângulos de Cherenkov para os três materiais diferentes e para diferentes partículas em função do momento como elétrons, káons, píons, múons e prótons. Na metodologia, será detalhado os parâmetros físicos medidos que serão conectados com o modelo de classificação utilizado neste trabalho.

### 2.4.3 SPD/PRS

Os calibradores de LHC consistem em várias camadas: o *Scintillating Pad Detector* (SPD), o *Pre-Shower Detector* (PRS) seguido pelo 'shashlik' - Calorímetro eletromagnético do tipo 'shashlik' (ECAL) e a placa de ferro cintilante Calorímetro Hadrônico (HCAL).

A descrição detalhada do projeto do detector já foi publicada e pode ser encontrada em outros lugares como páginas do CERN. Em resumo, o SPD e o PRS consistem em blocos cintilantes com uma espessura de 15 mm. A coleta de luz é feita por fibras de deslocamento de comprimento de onda (WLS). Quase quatro voltas de fibra são inseridas e coladas na ranhura redonda feita no esquadro. Ambas as extremidades da fibra WLS são usadas para transmitir a luz aos fotomultiplicadores de múltiplos ânodos (MAPMTs) localizados na periferia do detector. O número total de pastilhas no SPD ou PRS é de cerca de 6000. Cada pastilha é equipada com um diodo emissor de luz (LED) incorporado. Eles são acionados pela rede pulsante distribuída pela superfície do detector (AMATO et al, 2003).

Para ECAL, foi implementada uma estrutura de amostragem shashlik com placas cintilantes e placas de chumbo alternadas. O tamanho da célula varia de 4 X 4 cm<sup>2</sup> na parte interna do detector a 6 X 6 e 12 X 12 cm<sup>2</sup> nas partes central e externa. A granularidade celular corresponde à do SPD / PRS, visando a separação combinada do uso. As dimensões gerais do detector são 7.76 X 6.30 m<sup>2</sup>. A luz é detectada pelos multiplicadores fotográficos de 10 estágios Hamamatsu R7899-20 com uma base de alta tensão regulada individualmente do tipo Cockcroft-Walton.

O HCAL de 5.6 λ de espessura segue o ECAL. Sua estrutura interna consiste em finas chapas de ferro intercaladas com ladrilhos cintilantes dispostos ao longo da viga Collider. Novamente, as partes interna e externa têm dimensões celulares diferentes de 13 X 13 e 26 X 26 cm<sup>2</sup>, respectivamente. A estrutura escolhida do HCAL tem a seguinte característica: na direção lateral, os ladrilhos são espaçados com 1 cm de ferro (correspondendo ao comprimento



de radiação X0); ao mesmo tempo, na direção longitudinal, o comprimento dos ladrilhos e dos espaçadores de ferro corresponde à interação do hádron no comprimento do ferro. A luz nessa estrutura é coletada por fibras de deslocamento do comprimento de onda que percorrem o detector em direção à parte traseira

#### 2.4.4 Calorímetros

O sistema de calorimetria do LHCb (CALO) assume a forma clássica de um calorímetro eletromagnético (ECAL) seguido de um calorímetro de hádrons (HCAL) e está localizado a jusante do RICH-2. O ECAL é um do tipo calorímetro de amostragem, composto por 66 camadas alternadas de absorvedor de chumbo e cintilador. O HCAL, também é um detector do tipo de amostragem, composto de camadas alternativas de ferro e cintilador. Para ajudar a distinguir  $\pi^0$  do fundo avassalador de mésons  $\pi^0$  e  $\pi^\pm$ , é necessário a separação longitudinal dos chuveiros EM. Isso é conseguido usando dois detectores adicionais em frente ao ECAL: um detector de cintilador (SPD) e pré-chuveiro (PRS) descritos na seção anterior.

O sistema de calorímetro do LHCb realiza várias tarefas, fornecendo ao gatilho de primeiro nível candidatos de fótons de alto momento transversal, elétrons e hádrons, medindo suas energias e posições e realizando a separação entre fótons, elétrons e hádrons. O LHCbECAL baseia-se na tecnologia Shashlik de ladrilhos cintilantes e placas de chumbo alternados, precedidos por um pré-chuveiro (PS) e contém 1536/1792/2688 células em suas regiões interna / média / externa, respectivamente. Quando a célula ECAL possui um excesso de deposição de energia (comparado às células adjacentes), originará um cluster. Depósitos de energia na ECAL as células são agrupadas aplicando um padrão de células  $3 \times 3$  em torno do máximo local da célula semente de deposição de energia. Consequentemente, as células-semente dos aglomerados reconstruídos são sempre separadas por pelo menos uma célula. É necessário que a energia transversal da célula-semente seja maior que 50 MeV. Os agrupamentos neutros são identificados como aqueles que não correspondem às faixas carregadas extrapoladas para a superfície do calorímetro. Para cada par de trilho-cluster,  $\chi^2_{2D}$  is obteve levando em consideração: a posição do ponto de intersecção da trilha extrapolada a partir do calorímetro, a covariância-matriz dos parâmetros da trilha, a posição do baricentro do cluster e a matriz dos segundos do cluster. A assinatura de um decaimento do  $\pi^0$  no ECAL depende da cinemática dos dois fótons. O momento baixo  $\pi^0$  produz dois aglomerados separados. Tais  $\pi^0$  são classificados como  $\pi$  resolvidos e sua reconstrução é baseada na massa

invariável do par de fótons. O momento alto  $\pi^0$  produz um cluster único e é classificado como um mesclado 0. A região de transição entre o momento alto e o baixo  $\pi^0$  ocorre em torno de 2 GeV/c. Os recursos discriminantes que separam o fóton e os clusters  $\pi^0$  mesclados são baseados na forma de cluster. Espera-se que os agrupamentos  $\pi^0$  mesclados sejam alongados e assimétricos devido ao deslocamento residual entre dois fótons, enquanto se espera que os agrupamentos de fótons genuínos sejam mais simétricos. A descrição completa das variáveis usadas para a discriminação entre fótons e mésons  $\pi^0$ , que chamamos de abordagem baseada em forma (ou linha de base), pode ser encontrada em (DZHELYADIN, 2007).

O sistema LHCb Múon é composto por cinco estações de Câmaras Proporcionais Multi-Wire (MWPCs), rotuladas M1-M5, posicionadas em torno do eixo do feixe. As estações M2 a M5 estão localizadas a jusante dos calorímetros, com placas de ferro separadas de 80 cm de espessura, intercaladas entre cada uma. Essas placas agem como absorvedores para reduzir qualquer fundo hadrônico que sobreviva além dos calorímetros. A primeira estação, M1, fica imediatamente em frente aos calorímetros. Devido aos altos fluxos de partículas experimentados em torno da região interna desta estação a montante, é utilizada uma tecnologia com longevidade estendida: detectores de triplo GEM (Gas Electron Multiplier).

## 2.5 Bóson de Higgs e aprendizado de máquina

Modernas técnicas de aprendizado de máquina têm sido relevantes para as áreas de pesquisa em física experimental de altas energias ao redor do mundo. A construção de soluções que abordam as 4 forças elementares da natureza, eletromagnética, gravitacional, fraca e forte possuem forte apelo a tais técnicas. Um ponto de virada para a adoção de aprendizado de máquina em muitos experimentos deve-se a descoberta do bóson de Higgs que utilizou não menos que 4 árvores de decisão na procura pelo bóson em decaimentos de fótons nos colisores ATLAS e CMS do CERN (STRONG, 2020).

Desde 2009 redes neurais também são utilizadas em experimentos no LEP e Tevatron embora seu entendimento ainda seja baixo e serem consideradas “caixas-pretas” quando estão performando. Porém, nesta década o aprimoramento das redes neurais se tornou mais compreensíveis e confiáveis ao ponto de serem uma das técnicas mais utilizadas para experimentos em altas energias. Pesquisas relacionadas a reconstrução de objetos, colisões, simulação em detetores e identificação de partículas são alguns destes exemplos. Contudo,

novas descobertas e melhorias se dão na maioria das vezes em outras áreas não relacionadas a Física (STRONG, 2020).

Em estudo publicado por Strong em fevereiro 2020 relacionado ao uso de uso de redes neurais em desafio lançado pela plataforma de ciência de dados chamada Kaggle®, faz uma análise do custo computacional e de tempo gasto e diversos experimentos com redes neurais e árvores de decisão.

O trabalho desenvolvido por Strong (2020) propõe uma análise coerente do quão se pode melhorar a performance de redes neurais e árvores de decisão utilizando técnicas precisas ao invés de procurar gasto computacional extensivo.

O desafio lançado na popular plataforma de ciência de dados é identificar eventos classificados como decaimentos de Higgs em dois taus. A análise se dá comparando este sinal contra o *background*, porém este sinal é muito fraco e de difícil descoberta.

O princípio deste desafio se dá na colisão de prótons que produz inúmeras partículas que decaem em outras perdendo energia e interagindo com os detetores. O resultado dessas colisões é obtido com a ajuda de detetores que produzem interações com essas partículas possibilitando o estudo comparativo aos modelos teóricos com a finalidade de gerar um melhor entendimento da natureza do universo.

Para identificar sinais de partículas variadas dentre inúmeras colisões são utilizados curvas ROC que comparam o sinal e *background*. Sinal seria a suposta partícula ou evento que se pretende conhecer através da comparação de tudo aquilo que já se sabe, o *background*, ou ruído de energia do sistema. Porém na comparação entre sinal e *background* ainda se faz necessário a correção do que pode ser ruído, e não pertencer necessariamente ao modelo teórico proposto, e a energia resultando do sistema. Isto pode ser feito através de colisões reais ou também adicionando dados simulados através de processo de Monte Carlo.

### 2.5.1 Métricas, base de dados e préprocessamento

Para tal experimento foi utilizado uma métrica *Approximate Median Significance* (AMS) disponibilizada em código *python* pelo próprio desafio. É uma métrica derivada e foi proposta para o desafio por se tratar de uma classificação binária dentre outros fatores. Porém, o autor conferiu outras métricas a partir desta afim de melhorar a otimização, a saber:

. *Mean Maximal Validation* AMS (MMVA): O máximo de AMS alcançado nos dados de validação.

. *Mean Validation* MAS at Cut (MVAC): O AMS nos dados de validação em um corte escolhido.

. *Mean Averaged Public* AMS (MAPA): O valor de AMS na parte pública dos dados de teste.

De todo o *dataset*, foi separado para treino cerca de 250.000 eventos ou linhas e para teste foi reservado 550.000 eventos. Ambos possuem as duas classes, sinal e *background* com 30 colunas ou *features* cada base. Cada evento é caracterizado por dois tipos de *features*:

. Primária – com baixo nível de informações

. Derivada – Alto nível de informações calculada a partir de combinações lineares das *features* de baixo nível.

É comum em experimentos de muitas *features* a redução da dimensionalidade afim de gerar dados mais precisos e que evitem o modelo de aprendizado de máquina sobreajustar. Toda a base de dados foi normalizada e passou transformações substituindo valores faltantes por zero. A divisão entre *training set* e *validation set* dentro da original base de treino foi de 80:20 respectivamente com inicializações aleatórias em repetidos treinos.

Todo o processo de treino passou por validação cruzada com 10  *folds* via *random stratified splitting* em cada classe (sinal e *background*) sendo que as *features* foram transformadas para ter média zero e desvio padrão 1. A base de teste passo também pela mesma validação cruzada com uma única diferença, *simple random splitting*.

### 2.5.2 Features Selection

A seleção das melhores *features* ficou a cargo de modelos de *Random Forest* pela sua praticidade e facilidade de uso além da possibilidade de não precisar transformar os dados para classificação. Todas as *features* são elegíveis para utilização dos modelos, porém devido a importância das *features* de alto nível o trabalho seguiu classificando sua importância em uma escala através de 5 modelos diferentes de *random forest*. Nas Figuras 7 e 8 estão descritas as características das *features* de alto e baixo nível retiradas diretamente do trabalho de Strong.

**Tabela 3:** Glossário de recursos de alto nível (derivados) usados para treinamento, juntamente com seu agrupamento.

Feature name	Description	Grouping
DER_mass_MMC	Mass of the Higgs bóson estimated by a hypothesis based fitting technique.	Mass, Higgs.
DER_mass_transverse_met_lep	Transverse mass of the lepton and $PT_{\text{miss}}$	Mass, Higgs.
DER_mass_vis	Invariant mass of the lepton and the tau a naive estimate of the mass of the Higgs bonson.	Mass, Higgs.
DER_pt_h	Transverse momenta of the vector sum of the lepton, tau, and $PT_{\text{miss}}$ .	3-momenta, Higgs.
DER_deltaeta_jet_jet	Absolute difference in pseudorapidity of the leading and subleading jets (undefined for less than two jets).	Angular, Jet.
DER_mass_jet_jet	Invariant mass of the leading and subleading jets (undefined for less than two jets).	Mass, Jet.
DER_prodeteta_jet_jet	Product of the pseudorapidities of the leading and subleading jets (undefined for less than tow jets).	3-momenta, Jet.
DER_deltar_tau_lep	Separation in $n = 0$ space of the lepton and the tau.	Angular, Final-state.
DER_pt_tot	Transverse momentum of the vector sum of the transverse momenta of the lepton, tau, the leading and subleading jets (if presente), and $PT_{\text{miss}}$ .	Sum,Final-state.
DER_sum_pt	Sum of the transverse momenta of the lepton, tau, and all jets.	Sum, global event.
DER_pt_ratio_lep_tau	Transverse momenta of the lepton divided by the transverse momenta of the tau.	3-momenta Final-state..
DER_met_phi_centrality	Centrality of the azimuthal angle of $PT_{\text{miss}}$ realtive to the lepton and the tau.	Angular, Final-state.
DER_lep_eta_centrality	Centrality of the pseudorapidity of the lepton relative to the leading and subleading jets (undefined for less than two jets).	Angular, Jet.

Fonte: STRONG, 2020.

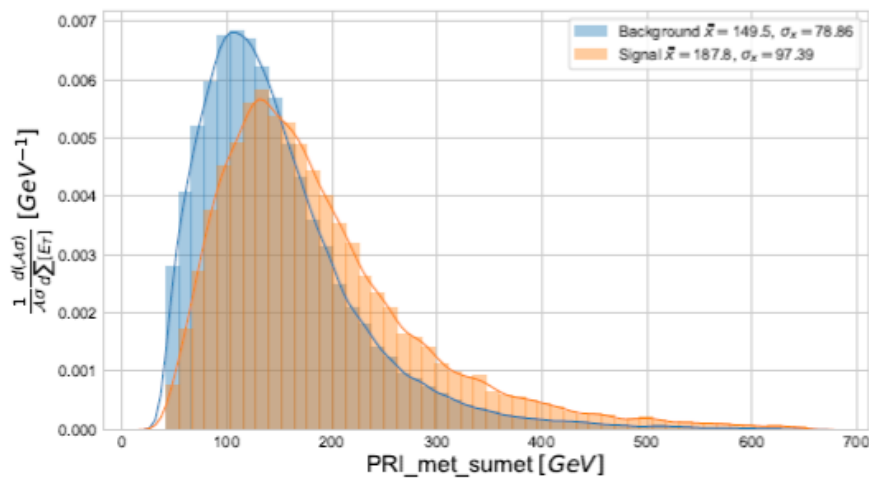
**Tabela 4:** Glossário de recursos de nível inferior (primário) usados para treinamento, juntamente com seu agrupamento.

Feature name	Description	Grouping
PRI tau [px/py/pz]	3-momenta of the tau in Cartesian coordinates.	3-momenta Final-state.
PRI lep [px/py/pz]	3-momenta of the lepton in Cartesian coordinates.	3-momenta Final-state.
PRI met [px/py]	Components of the vector of missing transverse momentum in Cartesian coordinates.	3-momenta Final-state.
PRI met	Modulus of the vector of missing transverse momentum in Cartesian coordinates.	3-momenta Final-state.
PRI met sumet	Sum of all transverse energy.	Energy, Final-state.
PRI jet num	Number of jets in event.	Multiplicity, Jet.
PRI jet leading [px/py/pz]	3-momenta of the leading jet in Cartesian coordinates (undened if no jets presente).	3-momenta, Jet.
PRI jet subleading [px/py/pz]	3-momenta of the subleading jet in Cartesian coordinates undened if < than two jets present).	3-momenta, Jet.
PRI jet all pt	Sum of transverse momenta of all jets Cartesian coordinates.	3-momenta, Jet.

Fonte: STRONG, 2020.

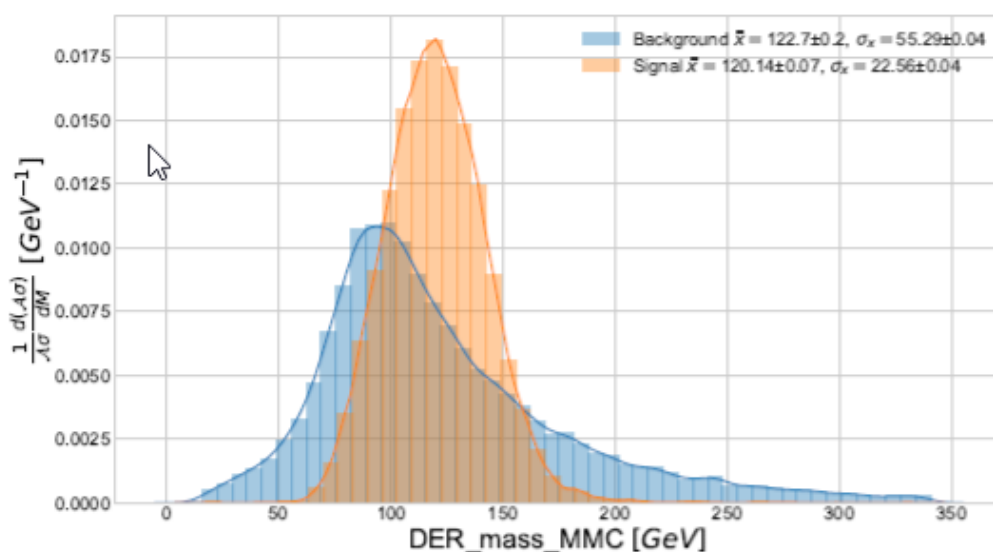
Note-se que o PRIjet pode ser diferente da soma dos de jatos principais e subconjuntos, pois pode haver eventos que contêm mais de dois jatos.

Em seguida, nas Figuras 9 e 10 mostra-se a distribuição do sinal e *background* dentro de ambas *features*, baixo e alto nível respectivamente.



**Figura 7 - (a)** Exemplo de característica de baixo nível: a soma da energia transversal ausente ( $\Sigma ET$ ).

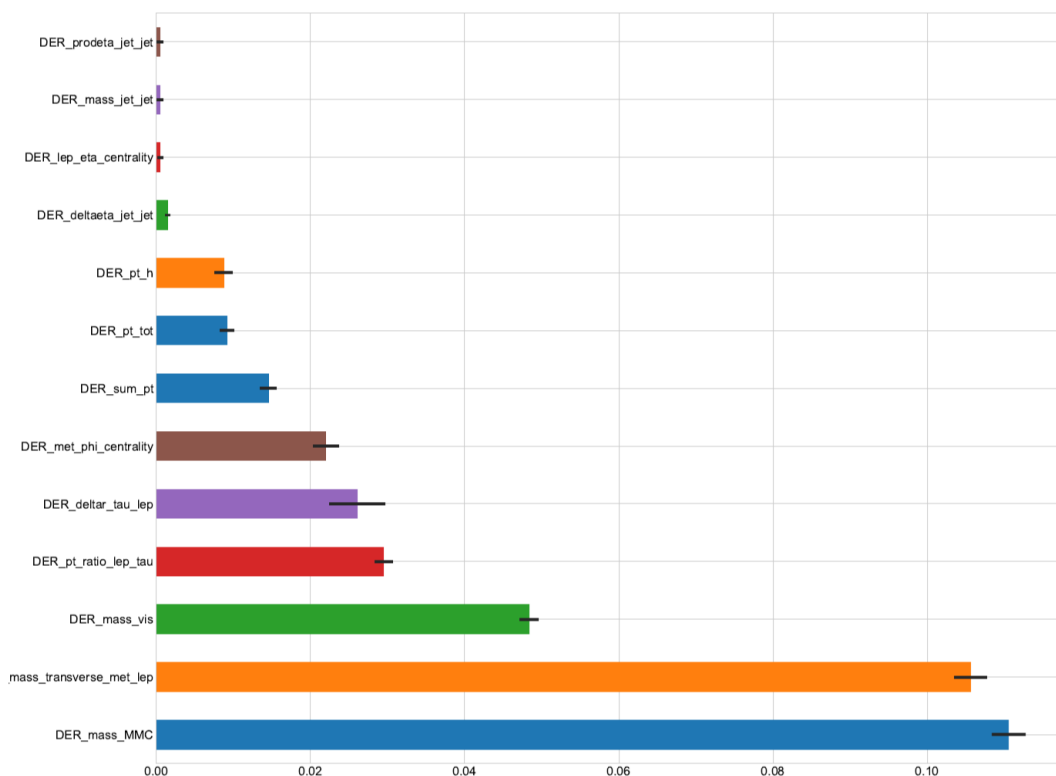
Fonte: STRONG, 2020.



**Figura 8 - (b)** Exemplo de característica de alto nível: as massas dos candidatos de Higgs calculadas por um algoritmo de ajuste baseado em hipóteses ( $M_{\tau\tau}$ , MMC).

Fonte: STRONG, 2020.

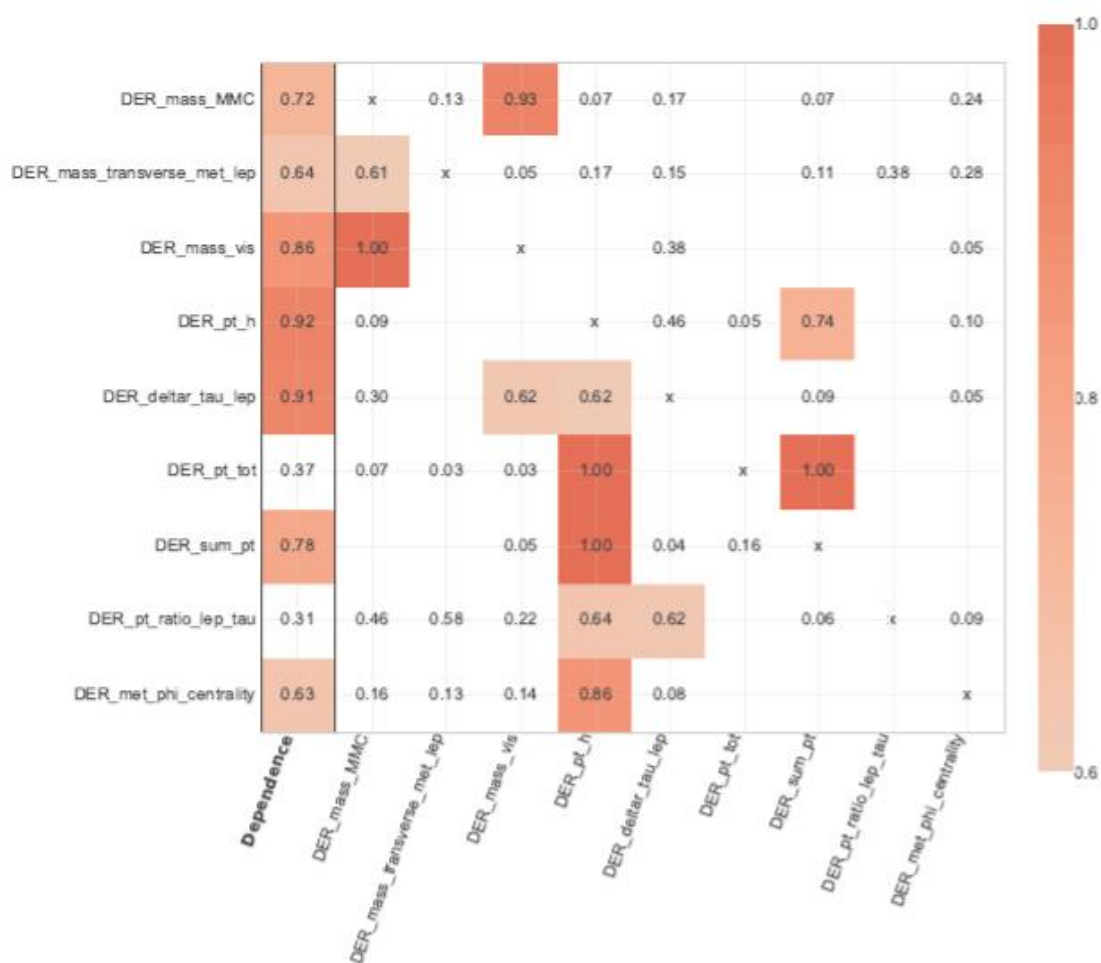
A Figura 9 mostra a importância de cada *features* de alto nível através da permutação da base de dados.



**Figura 9 -** Ilustração da importância da permutação fracionária dos recursos de alto nível estimada usando a importância média em cinco modelos de Floresta Aleatória.

Fonte: STRONG, 2020.

Por fim, na Figura 12 encontra-se um *heatmap* que ilustra a correlação de cada *features* de alto nível entre si. Pode-se ser considerado como um fator de dependência.



**Figura 10** - Ilustração da dependência de recursos e importância de recursos associados usando a regressão RandomForest.

Fonte: STRONG, 2020.

Considerando cada linha separadamente, o recurso na linha de eixo é aquele que está sendo registrado. O valor na coluna "Dependência" é o coeficiente de determinação ( $R^2$ ); mais alto significa mais fácil prever os valores do recurso. As colunas restantes indicam quanto cada recurso no eixo contribui para o desempenho do regressor; mais alto significa mais importante. Como exemplo: DER\_pt\_h é o recurso mais fácil de prever (dependência de 0,92), e o resumo é moderadamente importante para o regressor.



### 2.5.3 Modelos, arquitetura e performance

Para criação dos modelos e utilização de técnicas de otimização foi desenvolvido código a partir da biblioteca PyTorch. Foi criado um modelo base de rede neural com 4 camadas *fully-connected* com 100 neurônios cada uma e inicializando os pesos com *He-normal*. Funções de ativação foi escolhida a ReLU seguindo pelas camadas *fully-connected*. A camada de saída possui apenas 1 neurônio por ser uma classificação binária tendo como função de ativação *sigmoid*. A inicialização dos pesos desta camada se dá através do *Glorot-normal*.

Toda a atualização e otimização dos pesos e *bias* é feita através da técnica de *backpropagation* como de comum, estimando a perda através da função de perda *binary cross-entropy*. O tamanho do *minibatch* é 256 linhas. Como otimizador foi utilizado o ADAM com seus respectivos parâmetros padrões,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$  e  $\epsilon = 1 \times 10^{-8}$ . A taxa de aprendizado ou *learning rate* (LR) é 0,002 cujo treino é de 50 *epochs*.

Em um segundo momento foi testado uma arquitetura de camadas densamente conectadas ou *densely connected networks* com a finalidade de descobrir a performance entre diferentes tipos de conexões entre camadas.

Para a primeira performance foram desenvolvidos um modelo de árvore de decisão simples e um outro *ensemble*. Ambos os modelos treinaram 9 de 10 *training folds* usando o décimo para validação e monitoramento da perda. Uma vez terminado foram expostas as bases de validação e teste para apurar suas métricas. Tal processo passou por 6 testes em 6 hardwares diferentes. A Tabela 13 mostra os resultados.

**Tabela 5:** Ilustração da importância da permutação fracionária dos recursos de alto nível estimada usando a importância média em cinco modelos de floresta aleatória.

Setup	MMVA	MVAC	MAPA	Fractional Training	Time-Increase Inference
Single Model	$3.43 \pm 0.04$	$3.38 \pm 0.04$	$3.44 \pm 0.02$	-	-
<b>Ensemble</b>	<b><math>3.70 \pm 0.05</math></b>	<b><math>3.64 \pm 0.04</math></b>	<b><math>3.664 \pm 0.007</math></b>	$10.4 \pm 0.6$	$5.9 \pm 0.9$

Fonte: STRONG, 2020.

Como modelo escolhido, o método *ensemble*, apresentou maior performance para continuar na futura comparação com as redes neurais.

#### 2.5.4 Técnicas, hiperparâmetros, resultados e discussão

O resultado das redes *fully-connected* não obteve resultado satisfatório e isto é potencialmente devido ao fato de as camadas não conseguirem repassar as representações obtidas nas camadas anteriores perdendo assim informação importante e útil para as próximas camadas. O tamanho do *dataset* também pode ter influenciado a performance do modelo pôr a profundidade da rede ser determinada por este fator. Por outro lado, isso não ocorre nas redes de camadas densamente conectadas pelo motivo contrário.

Uma das formas de se melhorar a performance e evitar o sobre ajuste dos modelos é o aumento da sua base de dados original através de técnicas como *data augmentation* e *data fixing*. Ambas são para redirecionar os dados de forma simétrica sem que se perca o conteúdo da fonte e que se forme mais um evento não repetido, porém com informações simuladas. O problema se encontra em dados que dificilmente possuem simetria e para isso é mais indicado a técnica de *data fixing* pois mantém pelo menos um dos vetores fixos para gerar novos dados a partir deste. Na tentativa de escolher um ou outro a técnica de *data augmentation* gerou dados que performaram melhor o modelo. Resultados na Tabela 6.

**Tabela 6:** Comparação de explorações de simetria de dados em termos de métricas de otimização e impacto no tempo.

Setup	MMVA	MVAC	MAPA	Fractional Training	Time-Increase Inference
Fixing	3.90 ± 0.04	3.83 ± 0.05	3.76 ± 0.01	-	-
<b>Ensemble</b>	<b>3.96 ± 0.04</b>	<b>3.89 ± 0.05</b>	<b>3.79 ± 0.01</b>	1.11 ± 0.05	13 ± 5

Fonte: STRONG, 2020.

Os melhores valores para cada métrica são mostrados em negrito e a configuração escolhida também é indicada em negrito.

Para atingir o melhor hiperparâmetro dos modelos é comum se estabelecer um range de valores possíveis e adequados para que sejam testados e escolhidos. Uma arquitetura muito utilizada para descoberta destes valores chama-se *Random Search* que é facilmente encontrada em outras bibliotecas. É através desta estrutura que é facilmente descoberto um modelo cujo valores dos hiperparâmetros obtém a melhor performance e de forma automatizada. A seguir estão alguns dos ranges de hiperparâmetros utilizados.

- Depth [2; 9) em Z
- Width [33; 101) em Z
- Growth [-0:2; 1) em R
- L2 {0, 10<sup>-2</sup>, 10<sup>-3</sup>, 10<sup>-4</sup>, 10<sup>-5</sup>, 10<sup>-6</sup>} x 10<sup>-5</sup>
- Dropout {0, 0.05, 0.1, 0.25, 0.5}

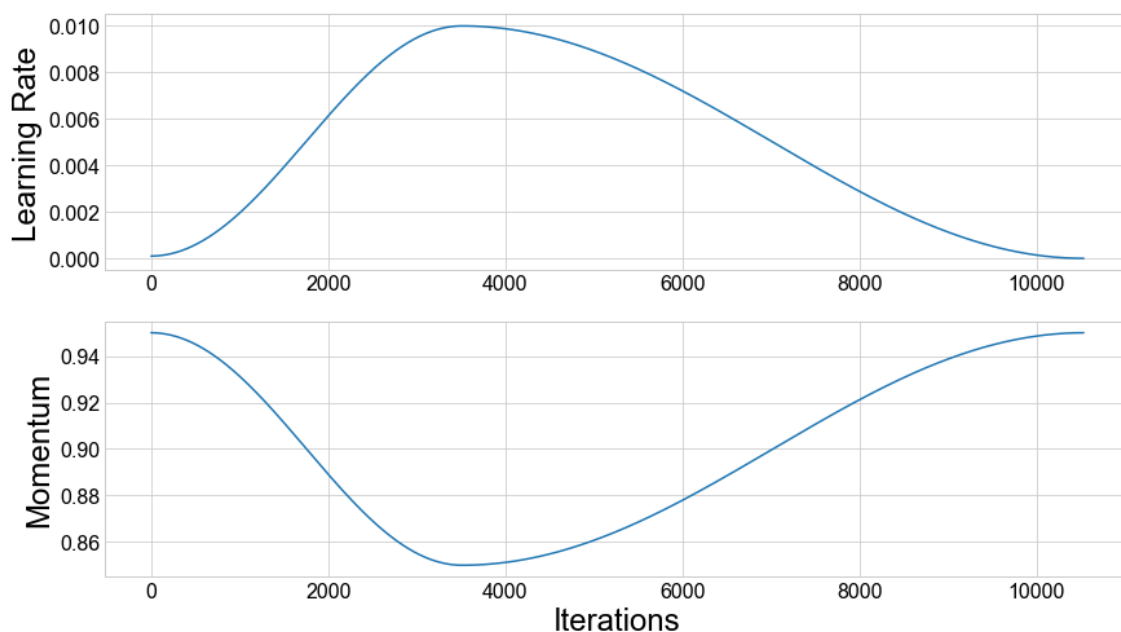
A seguir, em negrito estão os índices e modelos que melhor performaram. Não houve melhora significativa quando utilizadas.

**Tabela 7** - Comparação das várias abordagens avançadas de montagem.

Setup	MMVA	MVAC	MAPA	Fractional Training	Time-Increase Inference
Baseline	3:70 ± 0:05	3:64 ± 0:04	3:664 ± 0:007	-	-
+Embedding	3:78 ± 0:05	3:70 ± 0:06	3:71 ± 0:02	0:13 ± 0:03	0:08 ± 0:04
+Augmentation	3:96 ± 0:04	3:89 ± 0:05	3:79 ± 0:01	1:45 ± 0:08	14 ± 5
+SGDR	3:94 ± 0:06	<b>3:90 ± 0:04</b>	3:80 ± 0:02	2:12 ± 0:07	14 ± 5
+Swish	3:96 ± 0:06	3:86 ± 0:05	3:81 ± 0:02	2:32 ± 0:07	17 ± 4
-SGDR + SWA	3:96 ± 0:06	3:88 ± 0:05	3:81 ± 0:02	2:03 ± 0:07	17 ± 4
-SWA + 1cycle	<b>4:04 ± 0:06</b>	3:85 ± 0:04	3:81 ± 0:02	0:66 ± 0:04	17 ± 4
<b>+Dense</b>	3:95 ± 0:04	3:89 ± 0:05	<b>3:82 ± 0:02</b>	0:81 ± 0:08	17 ± 5

Fonte: STRONG, 2020.

Uma técnica que obteve significante redução do tempo de treino e causou uma *super convergence* foi a utilização do *learning rate* com *momentum*. A estratégia aqui é utilizar ambos hiperparâmetros em direções opostas, enquanto *learning rate* aumenta o *momentum* diminui. Isso faz com que ambos se balanceiem permitindo alto valor de LR sem que a rede neural divirja. A Figura 11 é o resultado.



**Figura 11:** Ilustração do cronograma de evolução da taxa de aprendizagem e momento usados durante o teste.

Fonte: STRONG, 2020.

Com resultado dos testes voltados para mensuração da performance através de fatias de tempo utilizada, uso do componente computacional, utilização de hardware diferentes dentre outras chegou-se à conclusão de que as soluções propostas pelo autor não melhoraram as métricas já alcançadas pelos participantes vencedores da competição. Porém, em termos de tempo gasto e esforço computacional tais técnicas demonstradas aqui incrementam neste sentido. Ou seja, é possível chegar ao mesmo resultado em menor quantidade de tempo e com menos uso dos recursos computacional.

## 2.6 Redes neurais e árvores de decisão.

Colisões em altas energias são objetos de estudo que geram grande impacto na ciência, não somente devido a sua tentativa de explicação da síntese da matéria universal, como também todo desafio tecnológico afim de gerar experimentações. Classificação de partículas é a diferenciação do que é partícula, a isso dá-se o nome de *signal*, de tudo aquilo que é percebido (detectado) e aquilo que não é identificado por alguma similaridade teórica chamando-se *background*. A trabalho de identificar partículas é basicamente diferenciar o que é *signal* e o

que é *background*, para isso inúmeras soluções foram propostas. Pelo fato de novas partículas serem difíceis de observar em seu momento inicial, o estudo de identificação de partículas é focado no decaimento que acontecem posteriormente, esses detectados com maior propriedade pelos detetores (BALDI; SADOWSKI; WHITESON, 2014). O recente sucesso das redes neurais profundas, ou *Deep Neural Networks* (DNN), trazem uma nova perspectiva frente aos métodos já empregados em HEP que em partes falham em capturar inúmeras informações disponíveis dos experimentos. Em Baldi et al. (2014), foi demonstrado que DNNs podem melhorar as métricas de classificação em *benchmark datasets*, mesmo comparadas com outros algoritmos como *Boosted Decision Tree* (BDT). O estudo foi dividido em dois momentos: o primeiro avança na distinção entre um *signal*, proveniente de uma nova teoria do bóson de Higgs, e *background*, com produtos de decaimento idênticos mas características cinemáticas distintas; o segundo momento tem a tarefa de diferenciar novas partículas supersimétricas levando em consideração seu estado final no qual existem partículas detectáveis e outras invisíveis. O *dataset* foi dividido em 3 níveis de características (*features*); *low level*, *high level* e *complete*. Ambas as divisões foram treinadas por 3 algoritmos diferentes por 5 vezes e manteve-se as configurações escolhidas, são eles: uma rede neural rasa, ou *shallow neural network* (NN), uma DNN e uma BDT. Assim, como de comum, experimentos envolvendo física de altas energias e aprendizado de máquina, a performance dos modelos são descritas pelas curvas ROC (*Receiver Operating Characteristic*) e a acurácia é medida pela área sob a curva ROC, ou *area under ROC* (AUC). Nas tabelas 8 e 9 estão os resumos das experimentações e nas Figuras 12 e 13 são exemplos de curvas ROC do primeiro momento (BALDI; SADOWSKI; WHITESON, 2014).

**Tabela 8:** AUC 1º Momento.

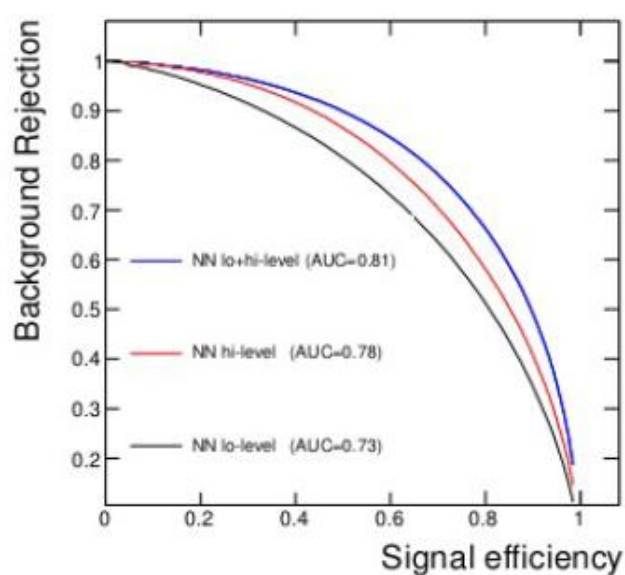
Algoritmo	Low Level	High Level	Complete
BDT	73%	78%	81%
NN	73,3%	77,7%	81,6%
DNN	<b>88%</b>	<b>80%</b>	<b>88,5%</b>

Fonte: Elaboração própria, dados de (BALDI, SADOWSKI, WHITERTSON, 2014).

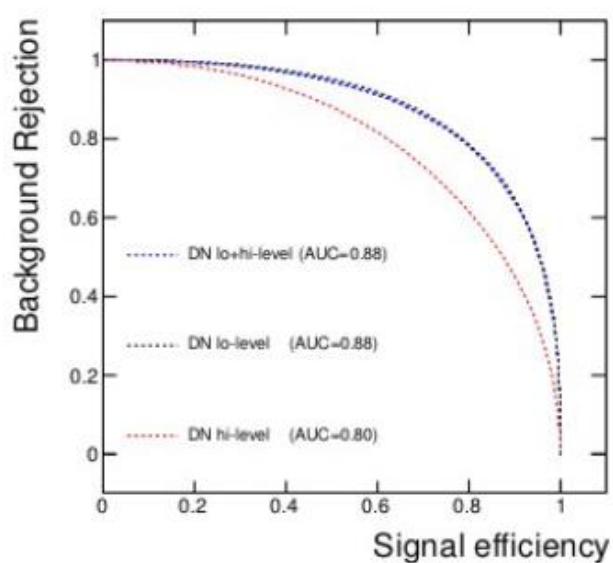
**Tabela 9:** AUC 2º Momento.

Algoritmo	Low Level	High Level	Complete
BDT	85%	83,5%	86,3%
NN	86,7%	86,3%	87,5%
NN <i>Dropout</i>	<b>88,6%</b>	85,9%	87,3%
DNN	87,2%	86,5%	87,6%
DNN <i>Dropout</i>	87,6%	<b>86,9%</b>	<b>87,9%</b>

Fonte: Elaboração própria, dados de (BALDI, SADOWSKI, WHITERSON, 2014).

**Figura 12 - ROC 1º momento.**

Fonte: BALDI; SADOWSKI, WHITERSON, 2014.

**Figura 13 - ROC 2º momento.**

Fonte: BALDI; SADOWSKI, WHITERSON, 2014.

No primeiro momento, os algoritmos que utilizaram *high level features* performaram abaixo do esperado em comparação daqueles que utilizaram todo o *dataset(Complete)*, sugerindo que apesar de um alto nível de representatividade das características o modelo demonstra dificuldade de capturar *insights* (BALDI;SADOWSKI; WHITESON, 2014).

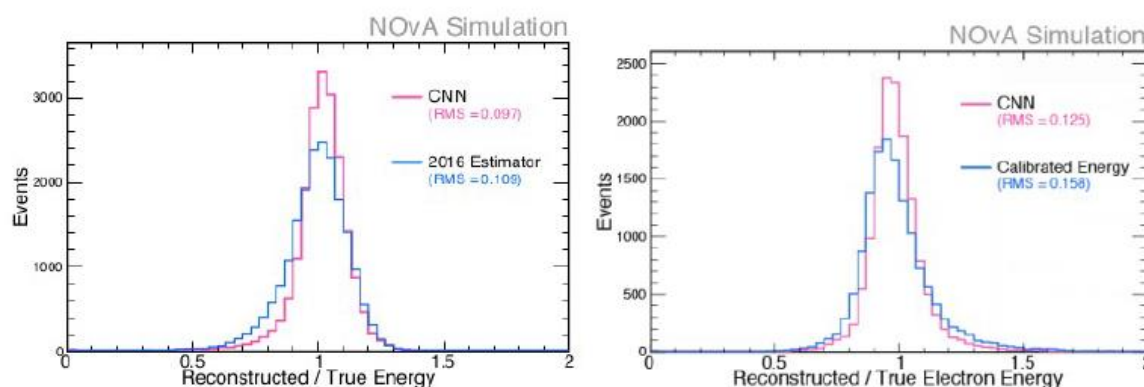
Já no segundo momento uma nova configuração para as redes neurais foi formada, otimizando-as com regularização do tipo *dropout*. Em ambos os experimentos as redes neurais se destacaram, principalmente aquelas que possuíam camadas ocultas (DNN). Aplicando-as em diferentes conjuntos de dados, observou-se que as redes neurais conseguem abstrair as características necessárias para uma classificação com satisfatória acurácia sobrepujando assim as árvores de decisão (BDT), esta última performa de forma semelhante apenas quando comparada as redes neurais rasas (NN).

As redes neurais utilizaram configuração de hiperparâmetros os seguintes valores: tangente hiperbólica como função de ativação, inicialização dos pesos através de distribuição normal com zero *mean* e desvio padrão de 0.1 na primeira camada, 0.001 na camada de saída e 0.05 em todas as camadas ocultas. Os gradientes foram computados com tamanho de *minibatches* igual a 100. O modelo foi treinado por 200 épocas com taxa de aprendizado de 1.000002. Quando submetidos a otimização de *dropout*, o valor é de 50% (BALDI; SADOWSKI; WHITESON, 2014).

### 2.6.1 Redes neurais profundas, rasas e convolucionais

Em experimento desenvolvido por Hertel et al. (2017), foram utilizadas redes neurais convolucionais cuja finalidade é de classificação através de imagens. O trabalho objetivou o estudo de oscilações dos neutrinos baseado na estimação da energia de elétrons neutrinos e elétrons *showers* em detectores do NOvA alcançando resultados de estado da arte. Neutrinos não possuem massas e raramente interagem com alguma matéria e por isso são de difíceis detecções. O experimento observa a oscilação dos neutrinos pois são uma função da sua energia, e isso consegue-se através de líquido cintilante posto dentro dos calorímetros. Ao interagir com o líquido forma-se uma imagem 3D dividida em duas direções,  $x - z$  e  $y - z$ . Toda a base de dados gerada a partir da experimentação é dividida em *training set*, *validation set* e *testing set*. Para processamento da base de dados foram utilizados três métodos de normalização: *mean zero unit variance standardization*, *log transformation* e *constant scaling*. Todos os métodos obtiveram

resultados similares (HERTEL et al., 2017). A *convolutional layer* foi configurada com 32 *filters*, 200 *units* e *Root Mean Square* (RMS) para avaliar a performance. Para *training loss* foi utilizado *Mean Squared Error* (MSE) e *Mean Absolute Error* (MAE), o algoritmo foi otimizado de duas formas, com *dropout* e *L2 weight penalty regularization*. Os modelos são treinados com gradiente estocástico descendente usando *ADAM algorithm*. Foi escolhido o tamanho de *batch size* igual a 128 com *initial rate* 1e-3 e foi treinado por 100 épocas dentre outros hiperparâmetros. Como resultado, anteriormente tinha-se como base uma precisão de 84%, com regularização L2 aumentaram para 85% e por fim 87% utilizando *dropout*. Na Figura 14 estão resultados para energia de neutrinos de elétrons (esquerda) e energia de chuva de elétrons (direita). Para cada parcela, o método de reconstrução tradicional é mostrado em azul e a curva rosa é o método baseado em CNN (HERTEL et al., 2017).



**Figura 14** – Resultados para energia dos neutrinos.

Fonte: HERTEL et al, 2017.

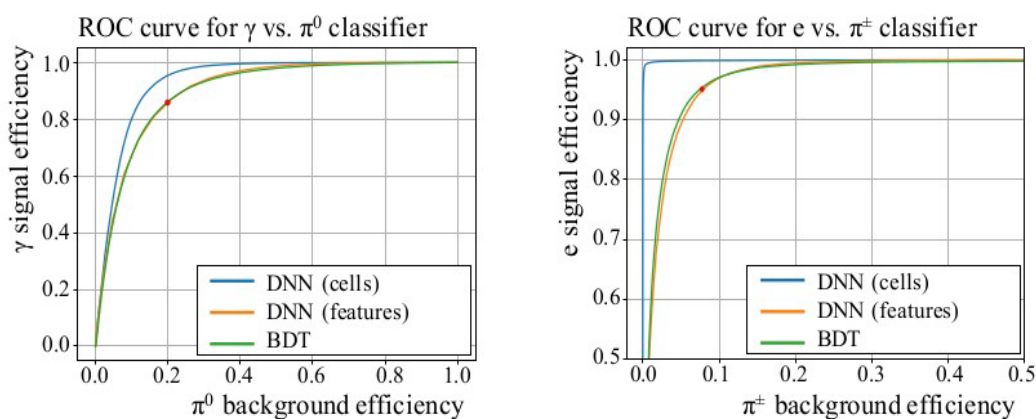
O modelo proposto mostrou-se proeminente em predição da energia elétron neutrino e energia de chuva de elétrons, porém, um desafio futuro é de fazer predição de múon neutrino *energy* e múon *energy* pelo fato de múons não ter trajetória de fácil detecção necessitando assim uma grande quantidade de imagens (HERTEL et al., 2017).

## 2.6.2 Redes neurais profundas, convolucionais, generativas e modelos lineares

Em trabalho desenvolvido por Carminati et al. (2017), uma aplicação de redes neurais profundas, convolucionais e generativas foram utilizadas para classificação, regressão de



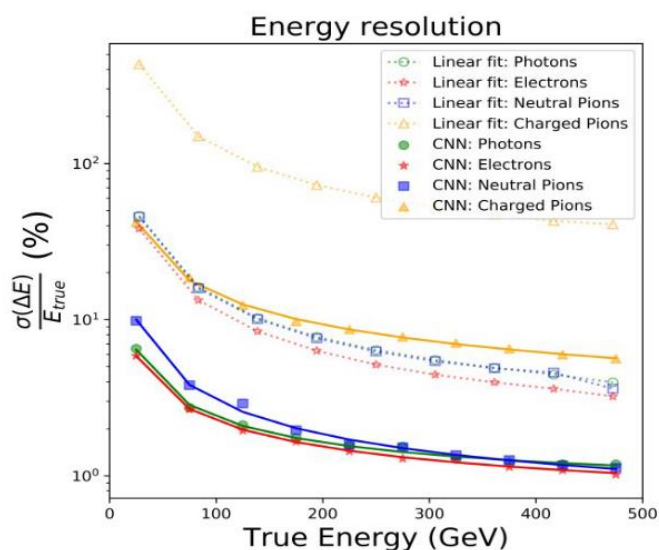
energia e simulação de partículas produzidas em colisões de altas energias. Para classificação foi utilizada DNN com 4 camadas ocultas com 256 neurônios cada e função de ativação ReLU, ao mesmo tempo que otimizada com *dropout* 0.5 (50%) e treinada com gradiente estocástico Adam com taxa de aprendizado (*learning rate*) de 0.001. A base de dados foi dividida em 80% de *training set* e 20% destinada a *testing set*. Em paralelo foi construída BDT com 400 *estimators*, com profundidade 5 e taxa de aprendizado 0.5. Figura 15 demonstra os resultados.



**Figura 15** - Curvas ROC de eficiência de sinal vs. eficiência de fundo para a (esquerda)  $\gamma$  vs.  $\pi^0$  e (à direita)  $e$  vs.  $\pi^\pm$  classificador. Os pontos vermelhos marcam o ponto de trabalho BDT escolhido.

Fonte: (CARMINATI et al., 2017).

Tratando-se de regressão, foram construídas duas CNN dedicadas para dados do *Electromagnetic Calorimeter* (ECAL) e *Hadrons Calorimeters* (HCAL). A saída de ambas as ramificações é linearizada e mesclada, seguida por uma camada totalmente conectada (*fully connected layer*) com 1000 neurônios. O neurônio da camada de saída tem uma função de ativação linear e a função de perda é calculada pelo *Mean-Squared Error* (MSE). A distribuição da base de dados deu-se 50% *training set*, *validation set* 12,5% e *testing set* com 37,5%. Foi feita comparação com modelos lineares e chegando a conclusão que a rede DNN performa melhor como pode ser vista na Figura 16, que compara a dependência energética da resolução do calorímetro para cada tipo de partícula com a rede neural e os modelos de regressão linear simples (CARMINATI et al., 2017).

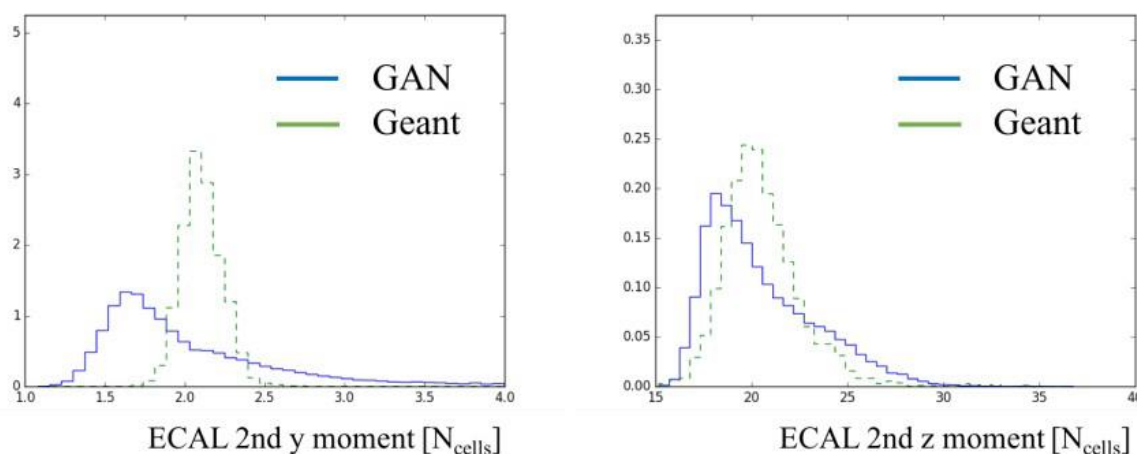


**Figura 16** - Resolução de energia para fótons, elétrons, íons neutros e carregados em comparação com o modelo CNN vs. Linear.

Fonte: (CARMINATI et al., 2017).

Continuando o trabalho explorando as potencialidades das redes neurais, foi desenvolvido através de redes generativas, ou *Generative Adversarial Networks* (GAN) simulações de partículas como uma prova de conceito afim de testar a capacidade de gerar novas simulações.

O resultado gerado pela GAN é razoável quando comparado ao sistema Geant, porém necessita de alguns ajustes para modelar *showers*. O resultado encontra-se na Figura 17.



**Figura 17** - Comparação da largura do chuveiro transversal (esquerda) e largura do chuveiro longitudinal (direita) para Simulação GAN vs. Geant de elétrons com energias de 200 a 300 GeV.

Fonte: (CARMINATI et al., 2017).

### 3 METODOLOGIA

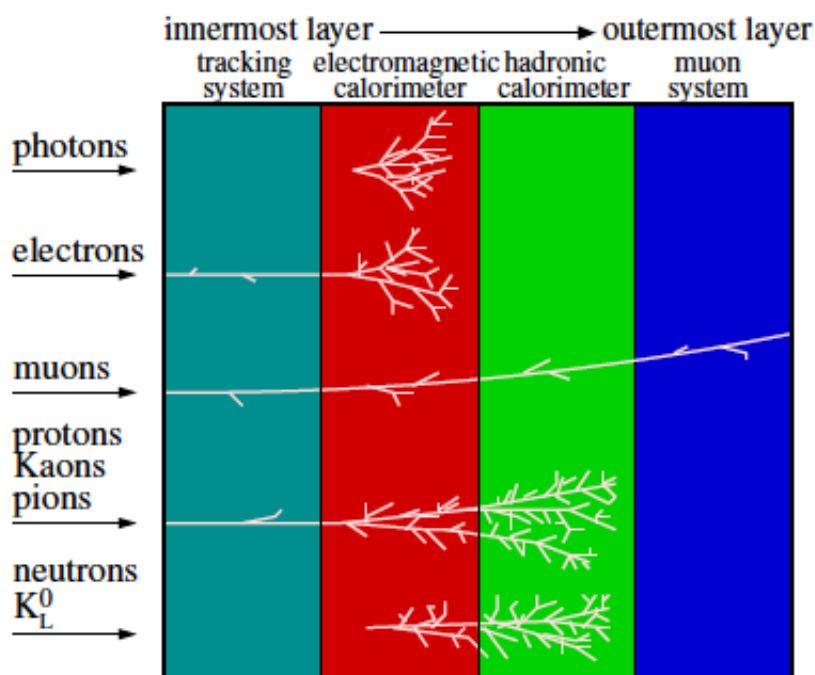
A metodologia empregada nesse trabalho leva em conta diversos aspectos. Primeiramente que os dados utilizados neste trabalho foram disponibilizados pelo LHC<sub>b</sub>, discutido na revisão de bibliografia seção 2. A planilha de dados está exemplificada na Figura 20 e possui 50 colunas correspondendo aos dados dos detectores que podem dar informações sobre o comportamento das partículas. Esses dados correspondem a cada detector e estão dispostos em seções onde detalhou-se a física envolvida e o detector. Esses são os dados utilizados para fazer os modelos de classificações.

#### 3.1 A Física envolvida nas medições

No capítulo 2, foi revisado a física que se tem disponível para o estudo do comportamento das partículas quando interagem com o meio onde elas passam. Nesta seção o objetivo é estudar a conexão da Física com o que se observa nos detectores e o que pode ser observado na tabela. Além do rastreamento e da calorimetria, a identificação de partículas (PID) é um aspecto crucial da maioria dos experimentos de física de partículas. A identificação de partículas estáveis é alcançado analisando a maneira como eles interagem ou determinando sua massa. A diferença na interação é usada principalmente para identificação de lepton e fóton.

##### 3.1.1 O papel dos diferentes tipos de interação na identificação de partículas

Em um experimento “tradicional” de física de partículas, são identificadas partículas (elétrons e suas antipartículas e fótons) ou, pelo menos, atribuídos às famílias ou hádrons neutrons), pelas assinaturas características que eles deixam no detector. O experimento é dividido em alguns componentes principais, como mostra a Figura 18, onde cada componente testa um conjunto específico de propriedades das partículas.



**Figura 18** - Componentes de um experimento “tradicional” de física de partículas.

Fonte: (LIPPMAN 2003).

Cada partícula tem sua própria assinatura no detector. Por exemplo, se uma partícula for detectada somente no calorímetro eletromagnético, é bastante certo que é um fóton. Esses componentes são empilhados em camadas e as partículas atravessam as camadas sequencialmente a partir do ponto de colisão para o exterior: primeiro um sistema de rastreamento, depois um eletromagnético (EM) e um calorímetro hadrônico e um sistema de múons. Todas as camadas são incorporadas em um magnético campo, a fim de defletir as faixas de partículas carregadas para a determinação do momento e sinal de carga.

A base de dados utilizada é proveniente do curso *Addressing Large Hadron Collider Challenges by Machine Learning* da plataforma de cursos online Coursera© e não passou por nenhum processo de preparação de dados devido ao fato de estar para pronta aplicabilidade no curso. Tal conjunto de dados possui 1.2 milhões de linhas e 50 (cinquenta) colunas cujas descrições serão apresentadas nas próximas seções.

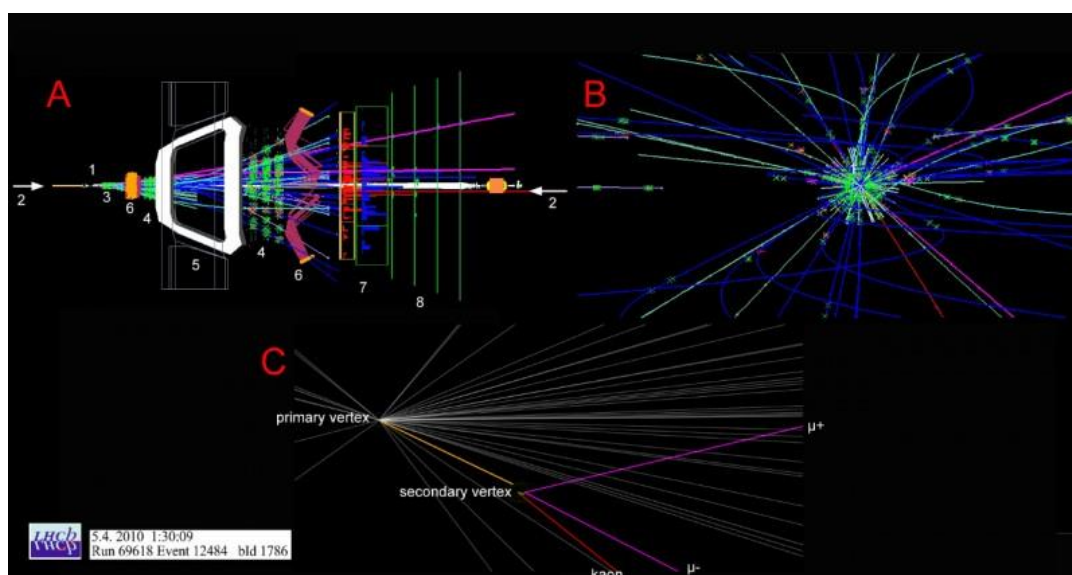
Nessa base de dados está identificado numericamente quais detectores correspondem as colunas da planilha. Uma pequena amostra destes dados está demonstrada na Figura 19, um recorte do *output* utilizando a biblioteca Pandas.

	TrackP	TrackNDoFSubdetector2	BremDLLbeElectron	MuonLooseFlag	FlagSpd	SpdE	EcalDLLbeElectron
0	74791.156263	15.0	0.232275	1.0	1.0	3.2	-2.505719
1	2738.489989	15.0	-0.357748	0.0	1.0	3.2	1.864351
2	2161.409908	17.0	-999.000000	0.0	0.0	-999.0	-999.000000
3	15277.730490	20.0	-0.638984	0.0	1.0	3.2	-2.533918
4	7563.700195	19.0	-0.638962	0.0	1.0	3.2	-2.087146

**Figura 19** - Exemplo do quadro com os dados experimentais do LHCb.

Fonte: Elaboração própria.

Na Figura 20 está uma ilustração das colisões e produção dos dados.



**Figura 20** - Exibição de evento LHCb.

Fonte: Imagem cortesia da colaboração do LHCb. Acessado em 26/01/2020.

<https://www.symmetrymagazine.org/breaking/2011/02/07/lhcb-event-display-decoded>

Todas as quatro visualizações da exibição do evento mostram dados da mesma colisão.

- 1 -> Ponto de colisão: marca o local onde os prótons dos 2 feixes se chocam.
- 2 -> Linha de luz: As setas mostram os caminhos dos feixes de prótons.
- 3 -> Localizador de vértices
- 4 -> Detetores de rastreamento
- 5 -> Ímã
- 6 -> detetores de imagens em anel Cherenkov

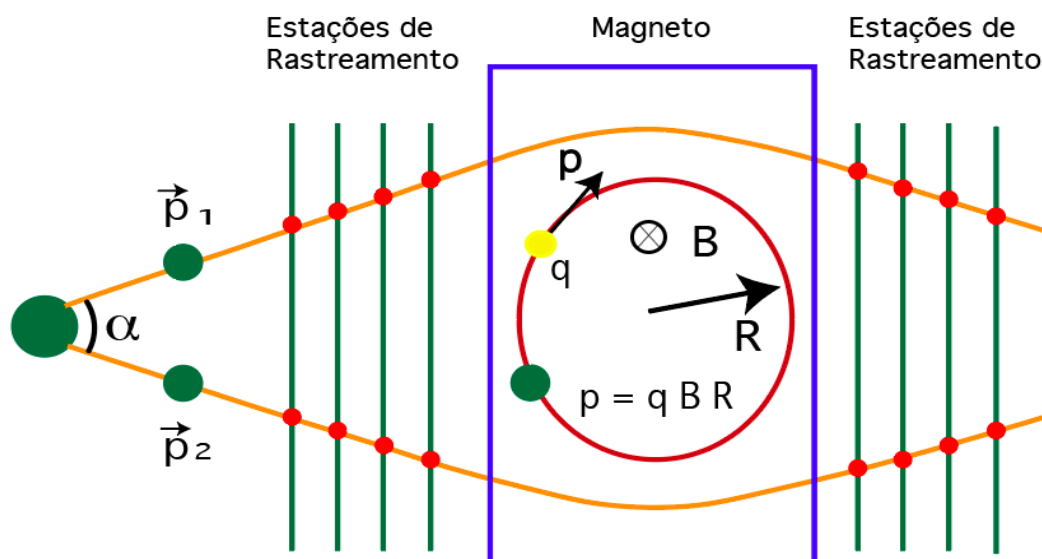
7 -> Calorímetros

8-> detetores de múon

### 3.1.2 Sistema de rastreamento

O sistema de rastreamento determina se as partículas são carregadas. Esse sistema consiste em um conjunto com um campo magnético, que mede o sinal da carga e o momento da partícula. Os fótons podem converter-se em um par elétron-pósitron e, nesse caso seja detectado no sistema de rastreamento. Além disso, decaimentos do kaon carregados podem ser detectados em um sistema de rastreamento de alta resolução através de sua característica “torção” topologia: por exemplo  $K^\pm \rightarrow \mu^\pm \nu$  (64%) e  $K^\pm \rightarrow \pi^\pm \pi^0$  (21%). O pai carregado (kaon) decai em uma filha neutra (não detectada) e uma filha carga de mesmo sinal. Portanto, o processo de identificação do kaon se reduz à descoberta de torções no sistema de rastreamento. A cinemática da topologia destas torções permite separar o decaimento do Kaon do fundo de decaimento de pions carregados (MIRONOV, 2006) .

Na Figura 20 o rótulo A é a vista superior do detector. Essa visualização nos fornece uma imagem completa da colisão, mostrando dados de todas as camadas do detector LHC<sub>b</sub>. Os feixes de prótons colidem à esquerda e as partículas, rótulo B, voam perto do caminho dos feixes de prótons. Essa propriedade das partículas, no rótulo B, é refletida no projeto do experimento: os sub-detetores ficam lado a lado ao longo do caminho do feixe, como livros em uma estante gigante. Na Figura 20, rótulo A, estão as grandes bobinas do ímã do LHC<sub>b</sub>, mostradas em branco [5]. O campo magnético é perpendicular à página e curva os caminhos das partículas carregadas no plano da página conforme mostra a Figura 21.



**Figura 21** - Comportamento das partículas no Ímã (5) da Figura 21, em branco.

Fonte: Adaptado de Cousera, 2019.

Nos dois lados do ímã estão detectores de rastreamento como mostra a Figura 21 que medem as posições das partículas à medida que passam. Os sinais que as partículas abandonam são mostrados como cruces. Não há sub-detektors dentro do ímã, então os caminhos das diferentes partículas que saem da colisão são reconstruídos em uma versão sofisticada de "conectar os pontos" usando dados de todas as camadas do detector LHC<sub>b</sub>.

Dos 50 sub-detektors 8 deles são destinados ao rastreamento da partícula, eles estão listados na Tabela 10.

**Tabela 10:** Parâmetros ligados aos subdetektors de rastreamento.

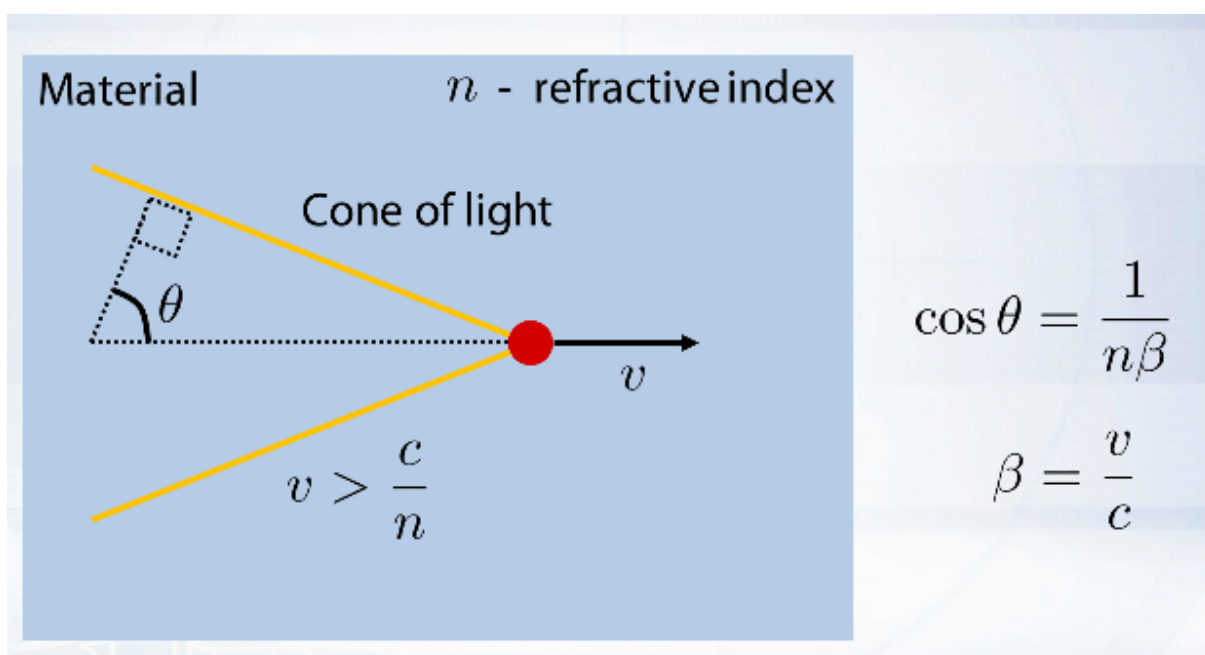
Sub-detektors	Característica da Medição
TrackQualitySubdetector1	Qualidade $\chi^2$ do ajuste da trilha usando hits (incidências) no sub-detektor de rastreamento 1 em 4 da Figura 21.
TrackNDoFSubdetector 1	Número de graus de liberdade para o ajuste da faixa usando hits no sub-detektor de rastreamento 1.
TrackQualitySubdetector2	Qualidade $\chi^2$ do ajuste da trilha usando hits (incidências) no sub-detektor de rastreamento 2 em 4 da Figura 21.
TrackNDoFSubdetector2	Número de graus de liberdade para o ajuste da faixa usando hits no sub-detektor de rastreamento 2.

TrackNDoF	Número de graus de liberdade para ajuste da faixa usando hits em todos os sub-detecores de rastreamento.
-----------	--

Fonte: Elaboração própria.

### 3.1.3 O detetor RICH

As partículas da colisão também passam pelos detetores Ring Imaging Cherenkov, que são usados para identificação de partículas. Eles trabalham medindo as emissões de radiação Cherenkov. Esse fenômeno, frequentemente comparado ao *boom* sônico produzido por uma aeronave que quebra a barreira do som, ocorre quando uma partícula carregada passa através de um determinado meio (neste caso, um gás denso) mais rápido que a luz no meio. Enquanto viaja, a partícula emite um cone de luz, que os detetores Cherenkov refletem em uma série de sensores usando espelhos, Figura 23.



**Figura 22** - Partícula carregada de velocidade  $v$ , passando através de um determinado meio (nesse caso um gás denso).

Fonte: Adaptado de Cousera, 2019.



O momento da partícula relativístico pode ser revisado em (LIPPMAN (2003) e escrito como:

$$p = \frac{m c \beta}{\sqrt{1 - \beta^2}}, \quad (1)$$

então

$$\beta = \frac{p}{\sqrt{p^2 + m^2 c^2}}. \quad (2)$$

O ângulo de emissão Cherenkov tem a forma:

$$\cos \theta = \frac{1}{n \beta} = \frac{\sqrt{p^2 + m^2 c^2}}{n p}. \quad (3)$$

Pode-se verificar na equação (3) que o ângulo de emissão Cherenkov pode ser obtido facilmente em função do momento dado pela tabela de parâmetros ligados ao detetor RICH. A Tabela 11 apresenta um número de sub-detectors ligado ao RICH 1 e 2.

**Tabela 11:** Os dados vindos do RICH.

Sub-detetores	Característica da Medição
TrackP	Momento das partículas.
Flag RICH 1	Flag (0 ou 1), se a pista reconstruída passa pelo primeiro detector RICH.
RICHpFlagElectron	flag (0 ou 1) se o momento for maior que o limite para que os elétrons produzam luz Cherenkov.
RICHpFlagProton	flag (0 ou 1) se o momento for maior que o limite para que os prótons produzam luz Cherenkov.
RICHpFlagPion	flag (0 ou 1) se o momento for maior que o limite para que os píons produzam luz Cherenkov.
RICHpFlagKaon	flag (0 ou 1) se o momento for maior que o limite para os káons produzirem luz Cherenkov.

RICHpFlagMuon	flag (0 ou 1) se o momento for maior que o limite para que os múons produzam luz Cherenkov.
FlagRICH2	flag (0 ou 1), se a pista reconstruída passa pelo segundo detector RICH.

Fonte: Elaboração própria.

Após a classificação das partículas e separação dos dados de káons, píons, elétrons, prótons e múons é possível calcular o ângulo de emissão Cherenkov em função do momento. A massa que aparece na equação 3 é a massa da partícula como mostrado na Tabela 1 e 2 do referencial teórico.

Os detetores RICH podem ser usados para identificar partículas carregadas mais massivas como os prótons e os káons. O RICH assume uma identidade para o rastreamento calculando a probabilidade global para a distribuição observada das batidas sejam consistentes com a distribuição esperada considerando várias hipóteses de identificação. O algoritmo interage com cada rastro e recalcula a probabilidade considerando várias hipóteses de identificação que muda se a partícula é um elétron, múon, káon ou próton. Para elétrons e múons informações adicionais do calorímetro e detetores de múons são usadas. As hipóteses que maximiza a probabilidade direciona o rastreamento. Para quantificar a qualidade da identificação é usado hipóteses sobre os píons que são usados como ponto de referência e a probabilidade de uma determinada identificação é dada em termos da diferença do logaritmo da probabilidade (*likelihood*) da partícula ser de um tipo comparado com a hipóteses de ser píon. Essa variável é chamada de  $\Delta \log(\mathcal{L})$  (DLL), para o káon temos  $\log(\mathcal{L}_k) - \log(\mathcal{L}_\pi)$ . Na Tabela 12 está relacionado os DLL que vamos usar neste trabalho para o RICH.

**Tabela 12:** Delta log (DLL) do RICH.

DLL do RICH	Característica da Medição
RICH_DLLbeElectron	Probabilidade delta de log para que um candidato a partícula seja elétron usando informações de RICH.
RICH_DLLbeMuon	Probabilidade delta de log para que um candidato a partícula seja múon usando informações de RICH.
RICH_DLLbeProton	Probabilidade delta de log para um candidato a partícula ser próton usando informações de RICH.

RICH_DLLbeKaon	Probabilidade de log delta para um candidato a partícula ser káon usando informações de RICH.
RICH_DLLbeBCK	Probabilidade delta de log para que um candidato a partícula seja em segundo plano usando informações de RICH.

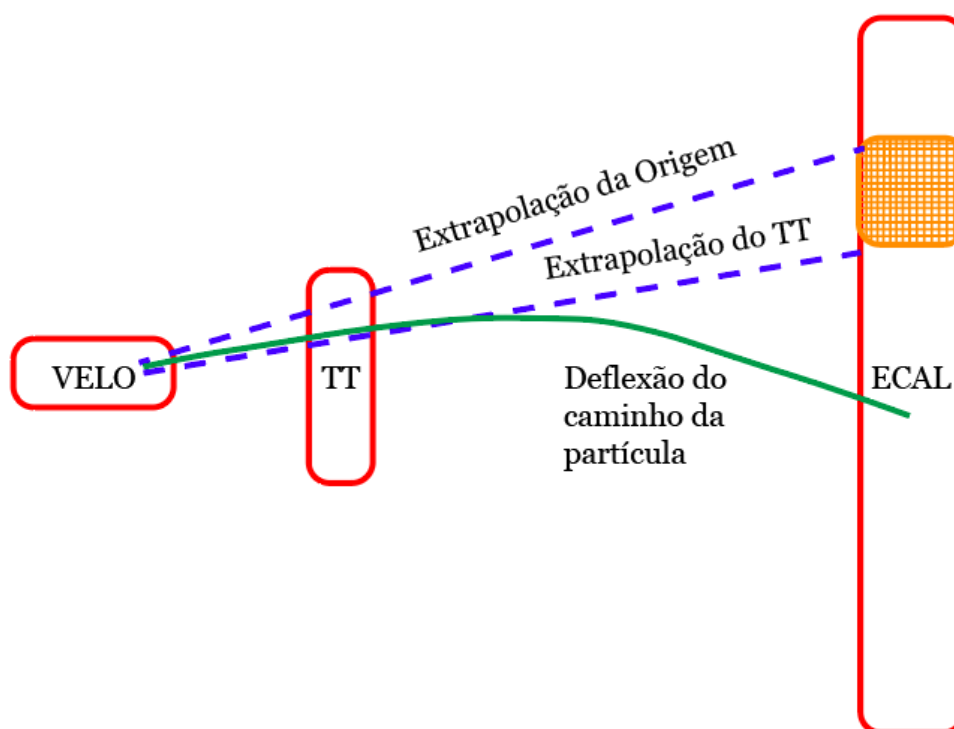
Fonte: Elaboração própria.

### 3.1.4 A emissão via efeito Bremsstrahlung

O Bremsstrahlung é emitido por partículas carregadas quando a direção do voo é alterada ou a velocidade é reduzida devido à interação eletromagnética. Isso acontece, por exemplo, em interações materiais ou em campos magnéticos. Os fótons resultantes são emitidos na direção do voo e a energia que uma partícula perde devido a *bremsstrahlungis* inversamente proporcional à sua massa em relação à quarta potência. Portanto, o efeito é significativamente mais distinto para elétrons do que para múons. De fato, no experimento do LHC, as perdas de energia devido à emissão de *bremsstrahlung* são desprezíveis para os múons. Os fótons são detectados pelo ECAL, pelo que é necessário um momento transversal maior que 75 MeV/c. Por razões técnicas, como espaço em disco limitado, os fótons em dados simulados são armazenados apenas se o momento for maior que 100 MeV/c. O desempenho da reconstrução de fótons é estudado com base em uma simulação de invasão eletrônica do detector de LHCb. Aqui, são distinguidos três casos que diferem na região em que o elétron emite o fóton. Em primeiro lugar, o fóton é emitido após a partícula atravessar a região do ímã, e a coordenada  $z$  da origem do fóton é para  $z > 7$  m. Esses fótons são detectados principalmente na mesma célula calorimétrica do elétron correspondente, tornando os dois indistinguíveis. No entanto, isso não compromete a medição do momento do elétron. A razão disso é que o momento é medido a partir da curvatura da trilha, que não é alterada pela emissão de fótons após a região do ímã, pois aqui a trilha não é dobrada. O segundo caso consiste em fótons emitidos na região do ímã, por exemplo,  $3 < z < 7$  m. Isso acontece tão raramente que é insignificante. O cenário mais importante é que o elétron emite uma explosão antes de ser dobrado pelo ímã, por exemplo para  $z < 3$  m, porque aqui a energia é perdida antes que o momento seja medido. Cerca de 50% dos fótons nos dados simulados (por exemplo, que ultrapassam o limiar de momento) atendem ao requisito de momento transversal, de modo que sejam reconstrutíveis. Desses, outros 50% são reconstruídos no ECAL se emitidos antes da região do ímã. Para reconstrução bem-sucedida de

um fóton, ele deve estar dentro da aceitação do ECAL e não deve compartilhar uma célula de calorímetro com outra partícula.

O procedimento para adicionar a energia dos fótons reconstruídos à faixa de elétrons correspondente é ilustrado na Figura 23. O esboço mostra uma vista superior dos componentes detectores envolvidos, que são o VELO, o TT e o ECAL. Uma janela de busca nesta última, indicada pela região hachurada, é definida extrapolando linearmente a trilha do elétron considerado uma vez a partir do seu vértice de origem, que é o decodificador davertex estudado nesta tese e uma vez a partir do ponto de interseção com o TT. Uma linha no plano-proxy é construída conectando os pontos onde as duas extrapolações indicam o ECAL. Todos os aglomerados de fótons a uma distância de  $2\sigma$  dessa linha são considerados como bremsstrahlung emitidos pela trilha de elétrons. Aqui, é definida como a incerteza combinada da extrapolação da trilha e a localização do baricentro do cluster. Como o ECAL mede apenas a energia depositada, a direção do momento é deduzida assumindo que o fóton se origina do PV. O resultado de quatro momentos é então adicionado ao elétron.



**Figura 23:** Ilustração da recuperação de bremsstrahlung no LHCb.

Fonte: (ALVES,2008)

Os parâmetros gerados nessa detecção estão na Tabela 13:

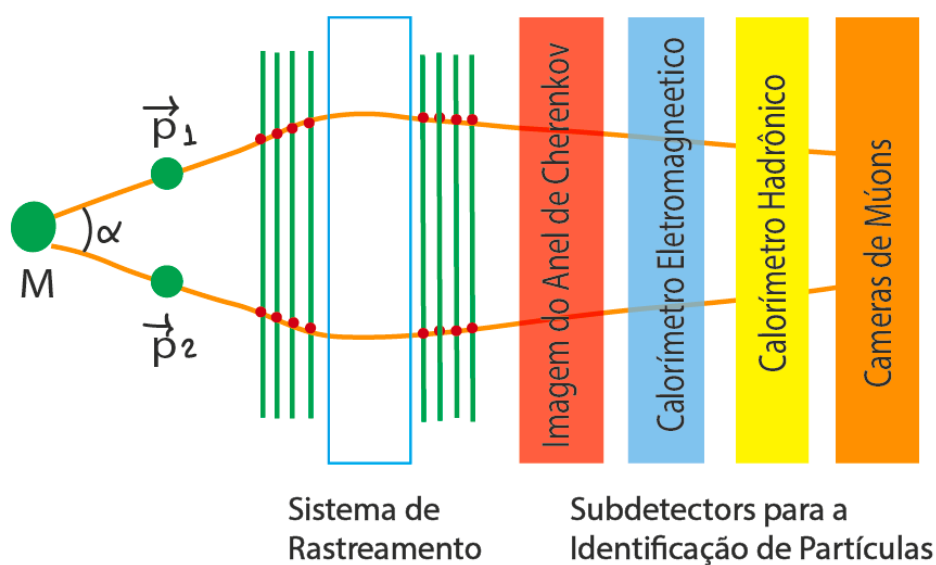
**Tabela 13:** Parâmetros ligados ao Bremsstrahlung.

Sub-detetores do Rastreamento	Característica da Medição
TrackPt	Momento transversal das partículas.
TrackDistanceToZ	Distância da trajetória da partícula até o eixo Z.
FlagBrem	Flag (0 ou 1), se a pista reconstruída passa por Brem.
BremDLLbeElectron	Probabilidade delta de log para que um candidato a partícula seja elétron usando informações de Brem.

Fonte: Elaboração própria.

### 3.1.5 Calorímetros

Em seguida, as partículas são paradas pelos calorímetros e a energia que depositam é representada na exibição do evento por histogramas. As barras vermelhas representam a energia de partículas mais leves (como fótons e elétrons), enquanto as barras azuis mostram a energia do grupo de partículas mais pesadas (como prótons). Na Figura 24 pode-se visualizar o sistema completo.

**Figura 24 -** Sistema completo de subdetectors LHCb.

Fonte: Adaptado de Cousera, 2019.

Os calorímetros podem detetar partículas neutras, medem a energia das partículas e determinam se eles têm interações eletromagnéticas ou hadrônicas. EM e hadron calorimetria no LHC é descrita em detalhes em (BROWN: COKERILL, 2011). Todas as partículas, exceto múons e neutrinos, depositam toda sua energia no sistema calorímetro por produção de chuveiros eletromagnético ou hadrônico. A resolução relativa com a qual a energia é depositada pode ser escrita como:

$$\left(\frac{\sigma_E}{E}\right)^2 = \left(\frac{a}{\sqrt{E}}\right)^2 + \left(\frac{b}{E}\right)^2 + c^2 \quad (4)$$

O primeiro termo leva em consideração as flutuações estocásticas e limita o desempenho a baixa energia. O segundo termo é devido ao ruído eletrônico. O terceiro termo é um termo constante que leva em consideração as uniformidades e erros do detetor, durante a calibração. Este termo limita o desempenho do calorímetro à altas energias.

Os valores dos parâmetros  $a$ ,  $b$  e  $c$  foram em todos os casos determinados por ajustes aos dados dos testes de feixe e são dados no descrições dos diferentes experimentos, nesse trabalho usar-se-á somente os dados do LHCb. Em caso dos calorímetros hadrônicos do ATLAS e CMS, as resoluções de todo o são mostrados sistemas que combinam EM e calorímetros hadrônicos (LIPPMAN, 2003).

Os calorímetros podem ser classificados em um de dois tipos: calorímetros de amostragem e calorímetros homogêneos. Os calorímetros de amostragem consistem em camadas de um material absorvente passivo denso intercalado com camadas detetoras ativas. Em homogêneo calorímetros, por outro lado, o absorvedor também atua como a detecção médio.

Fótons, elétrons e pósitrons depositam toda a sua energia no calorímetro EM. Seus chuveiros são indistinguíveis, mas um elétron pode ser identificado pela existência de uma faixa no sistema de rastreamento associado ao chuveiro. Nesse caso o depósito de energia deve corresponder ao momento medido no sistema de rastreamento. Hadrons, por outro lado, depositam a maior parte de sua energia no calorímetro hadrônico (parte dele também é depositada no calorímetro EM). No entanto, os membros individuais das famílias de hádrons carregados e neutros não podem ser distinguidas calorímetro.

Resumindo, o sistema de calorímetro consiste em várias camadas: o Detector de Almofadas Cintilantes em inglês “*Scintillating Pad Detector* (SPD), o Detector de Pré-Banho em inglês “*Pre-Shower Detector* (PRS)”, o Calorímetro Eletromagnético (ECAL) do tipo 'shashlik' e a Calorímetro Hadron (HCAL) da placa de ferro cintilante

### 3.1.6 SPD/PS

O SPD determina se as partículas que atingem o sistema do calorímetro são carregadas ou neutras, enquanto o PRS indica o caráter eletromagnético da partícula (isto é, se é um elétron, se carregado, ou um fóton, se neutro). Eles são usados no nível do gatilho em associação com o ECAL para indicar a presença de elétrons, fótons e pions neutros.

O SPD e o PRS consistem em blocos cintilantes com uma espessura de 15 mm, intercalados com um conversor de condutor de  $2,5 X_0$ . A luz é coletada usando fibras de deslocamento de comprimento de onda (WLS). Quase quatro voltas de fibra são inseridas e coladas na ranhura redonda feita no bloco quadrado, e as duas extremidades da fibra WLS são usadas para transmitir a luz aos fotomultiplicadores de múltiplos ânodos (MAPMTs) localizados na periferia do detetor. O SPD e o PRS contêm aproximadamente 6000 blocos, cada um deles equipado com um diodo emissor de luz (LED) incorporado. As almofadas são acionadas pela rede pulsante distribuída por toda a superfície do detetor.

As variáveis que da tabela que fornecem informações destes cintiladores são elencadas na Tabela 14.

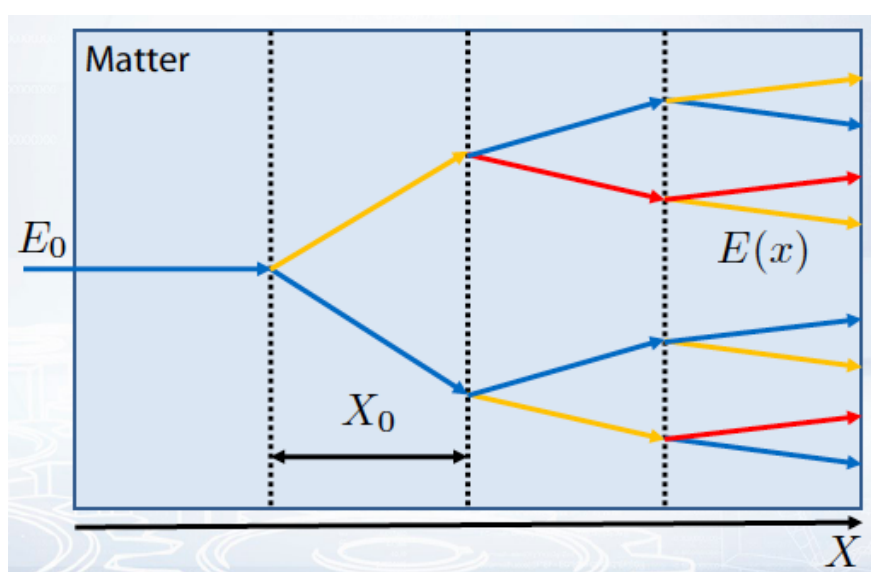
**Tabela 14:** Informações dos cintiladores.

SPD/PRS	Característica da Medição
FlagSpd	Flag (0 ou 1), se a trilha reconstruída passa pelo Spd.
SpdE	Depósito de energia associado à pista no Spd.
SpdDLLbeElectron	Probabilidade delta de log para que um candidato a partícula seja elétron usando informações de Spd.
FlagPrs	Flag (0 ou 1), se a pista reconstruída passa por Prs.
PrsE	Depósito de energia associado à pista no Prs.
PrsDLLbeElectron	Probabilidade delta de log para que um candidato a partícula seja elétron usando informações de Prs.

Fonte: Elaboração própria.

### 3.1.7 Calorímetro Eletromagnético

O Calorímetro eletromagnético é responsável pela medição da energia de elétrons e fótons (DZHELYADIN, 2007). Na interação com a matéria um elétron emite um fóton e um fóton produz um par elétron pósitron. Esse processo cria uma cascata eletromagnética. A cascata eletromagnética pode ser visualizada na Figura 25.



**Figura 25** - Cascata eletromagnética.

Fonte: (COUSERA, 2019).

A relação entre a energia final e inicial da cascata eletromagnética é dada por:

$$E(x) = E_0 e^{-\frac{x}{X_0}}. \quad (5)$$

O chuveiro eletromagnético cresce enquanto a energia das partículas está acima do valor crítico  $E_c$ . O tamanho do chuveiro  $X_{max}$  pode ser estimado da seguinte forma:

$$X_{max} \approx X_0 \ln \frac{E_0}{E_c}. \quad (6)$$

O número total de partículas no chuveiro é estimado como:



$$N \sim \frac{E_0}{E_C}. \quad (7)$$

Medir o número de partículas permite determinar a energia da partícula recebida. Este número é medido por contadores de cintilação.



**Figura 26:** Foto-cintilador.

Fonte: CMS Ecal / <https://cms-docdb.cern.ch/cgi-bin/PublicDocDB/ShowDocument?docid=12030>

As informações extraídas da foto cintilador Figura 26 estão listadas na Tabela 15, para o calorímetro eletromagnético posicionado na Figura 25.

**Tabela 15:** O Calorímetro Eletromagnético.

<b>Ecal</b>	<b>Característica da Medição</b>
FlagEcal	Flag (0 ou 1), se a pista reconstruída passa por Ecal.
EcalE	Depósito de energia associado à pista no Ecal.
EcalShowerLongitudinalParameter	Parâmetro longitudinal do chuveiro Ecal.

Fonte: Elaboração própria.

Nesse detector também se faz a medição via  $\Delta \log$ , os elementos da Tabela 16 que são variáveis usadas no modelo para classificação no caso do Calorímetro Eletromagnético (Ecal).

**Tabela 16:** O Delta Log do Calorímetro Eletromagnético.

<b>DLL do Ecal</b>	<b>Característica da Medição</b>
EcalDLLbeElectron	Probabilidade delta de log para que um candidato a partícula seja elétron usando informações de Ecal.
EcalDLLbeMuon	Probabilidade de log delta para que um candidato a partícula seja múon usando informações de Ecal.

Fonte: Elaboração própria.

Pode ser verificado na Tabela 16 que se comparar com a Tabela 11 não existe nem káons, nem prótons e nem píons, de fato, e o calorímetro eletromagnético oferece os léptons, mais precisamente múon e o elétron. Na Tabela 17, temos os parâmetros que revelam o controle de qualidade.

**Tabela 17:** Controle de qualidade dos clusters do calorímetro.

<b>Quality</b>	<b>Característica da Medição</b>
Calo2dFitQuality	Qualidade do 2º ajuste dos clusters no calorímetro.
Calo3dFitQuality	Qualidade do ajuste 3d no calorímetro com a suposição de que a partícula era elétron.

Fonte: Elaboração própria.

No feixe teste será usado a resolução da energia relativa escrita como:

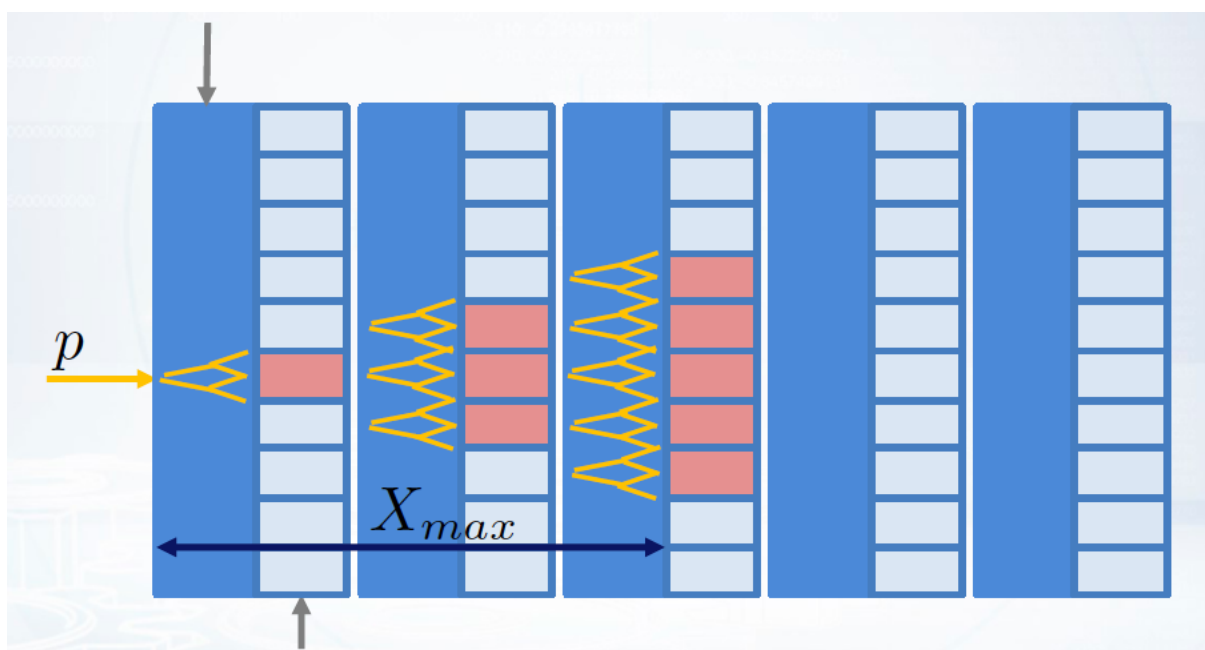
$$\left(\frac{\sigma_E}{E}\right)^2 = \left(\frac{0,094}{\sqrt{E(GeV)}}\right)^2 + \left(\frac{0,145}{E(GeV)}\right)^2 + (0,0083)^2, \quad (8)$$

considerando que o calorímetro EM é uma parede retangular construída com placas de chumbo e cintilador de azulejos. A espessura total corresponde a  $2,5X_0$  como pode ser visto na Figura 27.

### 3.1.8 Calorímetro Hadrônico

O calorímetro hadronico é responsável pela medida da energia de protons, neutrons e outras partículas contendo quarks (AKCHURIN;WIGMANS, 2011). Este calorímetro tem as seguintes características:

- é similar ao calorímetro eletromagnético.
- produz um chuveiro hadrônico devido as interações com os núcleos dos átomos da matéria.
- o chuveiro consiste em um grande número de partículas diferentes tipos.



**Figura 27 - Cascata Hadrônica.**

*Fonte: (COUSERA, 2019).*

Na figura 27 as células azuis é a matéria que cria o chuveiro hadrônico e as células azuis claro são os contadores de cintilação, cujo objetivo é contar o número de partículas do chuveiro. Das células destes sub-detectors até atingir o alcance máximo pode se extrair as seguintes variáveis que também são levadas em conta em nosso trabalho dados na Tabela 18 e 19.

**Tabela 18: O Calorímetro Hadrônico.**

Hcal	Característica da Medição
FlagHcal	Flag (0 ou 1), se a pista reconstruída passa por Hcal.
HcalE	Depósito de energia associado à pista no Hcal.

Fonte: Elaboração própria.

e os dados vindos do  $\Delta \log$ , podem ser visualizados na Tabela 19.

**Tabela 19:** O Delta Log do Calorímetro Hadrônico.

DLL do Hcal	Característica da Medição
HcalDLLbeElectron	Probabilidade delta de log para que um candidato a partícula seja elétron usando informações de Hcal.
HcalDLLbeMuon	Probabilidade de log delta para um candidato a partícula estar usando informações de Hcal.

Fonte: Elaboração própria.

O calorímetro hadrônico consiste em ferro e cintilador telhas com uma resolução de energia relativa de

$$\left(\frac{\sigma_E}{E}\right)^2 = \left(\frac{0,69}{\sqrt{E(\text{GeV})}}\right)^2 + (0,09)^2, \quad (9)$$

medido com um protótipo em um teste de feixe.

### 3.1.9 Câmaras de Múons

A posição dos detetores de múons, mais afastada do ponto de colisão à direita, é mostrada pelas linhas verdes verticais. As faixas deixadas pelos dois múons criados nesta colisão são coloridas em magenta na Figura 21.

- Uma câmara de múon é preenchida com gás e tem um fio dentro.
- A tensão é aplicada entre o fio (ânodo) e as paredes da câmara (cátodo).

O múon difere do elétron apenas por sua massa, que é cerca de um fator 200 maior. Como consequência, a energia crítica  $E_c$  (a energia para a qual em um determinado material as taxas perda de energia por ionização e bremsstrahlung são iguais) é muito maior para múons: é de cerca de 400 GeV para múons em cobre, enquanto para elétrons em cobre é apenas cerca de 20MeV. Como consequência, os múons geralmente não produzem chuviscos eletromagnéticos e, portanto, podem ser facilmente identificados por sua presença no detetores mais externos, pois todas as outras partículas carregadas são absorvidas no sistema do calorímetro.

O conjunto de variáveis para o detetor de múons é dado pela Tabela 20.

**Tabela 20:** Parâmetros do Detetor de Múons.

<b>Detetor de muons</b>	<b>Característica da Medição</b>
FlagMuon	Flag (0 ou 1), se a pista reconstruída passa pelas estações de múons (Muon).
MuonFlag	Bandeira de múon (é esse múon de faixa) que é determinado a partir das estações de múon.
MuonLooseFlag	É determinado a partir das estações de múon usando critérios mais flexíveis.
MuonLLbeBCK	Probabilidade de log para um candidato a partícula não ser múon usando informações de estações de múons.
MuonLLbeMuon	Probabilidade de log para um candidato a partícula ser múon usando informações de estações de múons.

Fonte: Elaboração própria.

As tabelas também contém os rótulos:

- ID - valor de ID para faixas (presente apenas no arquivo de teste para fins de envio).
- Label - string com valor observável denotando tipos de partículas. Pode levar os valores "Elétron", "Muon", "Kaon", "Proton", "Pion" e "Ghost". Esta coluna está ausente no arquivo de teste.
- GhostProbability - probabilidade de um candidato a partícula ser trilha fantasma. Essa variável é uma saída do modelo de classificação usado no algoritmo de rastreamento.

### **3.2 PID por determinação de massa PID por determinação de massa**

Os três hádrons carregados mais importantes (pions, kaons e prótons) e suas antipartículas têm interações idênticas em uma configuração experimental como a mostrada na Figura 1 (depósito de carga no sistema de rastreamento e chuveiro hadrônico no calorímetro). Além disso, todos eles são efetivamente estáveis. Contudo, sua identificação pode ser crucial, em

particular para o estudo de decaimentos hadrônicos. A possível melhoria na relação sinal / fundo ao usar PID é demonstrado na Figura 4, usando o exemplo da decadência  $\phi \rightarrow K^+K^-$ .

Na física da Figura 21 B, o estudo de hádrons contendo o quark da beleza, diferentes decaimentos geralmente existem modos, e suas propriedades individuais só podem ser estudadas com uma identificação de hádrons, o que melhora a relação sinal / fundo (a maioria das faixas são píons de várias fontes). O PID é igualmente importante na física de íons pesados. Um exemplo é a medição da partícula charme (e da partícula beleza), que permite investigar os mecanismos para a produção, propagação e hadronização de quarks pesados no meio quente e denso produzido na colisão de íons pesados.

### 3.3 Identificação de Partículas

A classificação de partículas em experimentos de altas energias é uma tarefa que inclui uma grande quantidade de ferramentas, dependendo da solução proposta os meios são diversos. Este experimento concentra-se em utilização de algoritmos de aprendizado de máquina para identificação de partículas em bases de dados provenientes do colisor LHC. Os dados são gerados a partir de colisões próton-próton dentro dos detectores e selecionados a partir de um trigger com a finalidade de filtrar quais melhores eventos ocorridos (DI MEGLIO et al., 2018).

A atribuição dada a este trabalho é de programação, foi desenvolvido dois algoritmos (classificadores), uma árvore de decisão e uma rede neural. Os classificadores identificam o tipo de partícula. Existem seis tipos de partículas: elétron, próton, múon, káon, pión e aquilo que foi chamado de “ghost”. O “ghost” é uma partícula com outro tipo que não os primeiros cinco ou um ruído do detector. Diferentes tipos de partículas permanecem diferentes respostas nos sistemas de detecção ou subdetectores. São cinco sistemas: sistema de rastreamento (*Tracking System*), detector Cherenkov de imagem em anel (RICH), calorímetros eletromagnéticos e hadrônico (ECAL e HCAL), e a câmara de múons (*Múon System*) (LIPPMANN, 2012). As ferramentas e bibliotecas utilizadas para desenvolvimento do código, manipulação da base de dados, compilação dentre outras, serão detalhadas abaixo.

### 3.3.1 Base de Dados

A base de dados utilizada para treinamento dos modelos foi extraída do curso *Addressing Large Hadron Collider Challenges by Machine Learning* da plataforma Coursera® e está disponível em <https://github.com/hse-aml>. Suas dimensões são 1.200.000 instâncias e 50 *features* (colunas). Algumas colunas podem ser descritas como: reconstrução de trajetória, energia depositada associada a trajetória, delta *loglikelihood* para partícula candidata, momentum da partícula, momentum transverso e outras. A divisão entre *training set* e *testing set* foi definida em 80% e 20%, respectivamente. Para a utilização em redes neurais toda base de dados passou por pré-processamento transformando seus valores literais (coluna “*Label*”) em numéricos. Esta é uma técnica conhecida como *one hot encoding* cujo propósito é transformar variáveis categóricas em valores numéricos. Ou seja, foi transformado as classificações das partículas, coluna *Label*, em número de 0 a 5. Para melhorar o desempenho da rede neural valores foram normalizados.

### 3.3.2 Ferramentas e Bibliotecas

As ferramentas para elaboração e compilação do código foram respectivamente Jupyter Notebook e Colaboratory (Colab). O trabalho foi desenvolvido utilizando a nuvem de aplicativos Google Drive que é detentora do aplicativo Colab construído para criação de *script* inteiramente em *cloud*.

Para elaboração dos algoritmos foram utilizadas 6 (seis) bibliotecas de código aberto em linguagem Python, a saber:

- Numpy: Utilizado para computação científica com recursos Álgebra Linear, transformação de Fourier, números aleatórios, matrizes N-dimensionais e inúmeros cálculos numéricos.
- Pandas: Biblioteca para manipulação e análise de base de dados através da criação de *data frame*.
- Matplotlib: Objetivo é a produção de figuras e gráficos em formato interativo através de várias plataformas. Pode-se fazer *histograms*, *scatterplots*, *bar charts*, *power spectra* dentre outros tipos de visualizações.
- Sci-kit Learn: Inclui vários algoritmos de classificação, regressão, agrupamento, árvores de decisão dentre tantos outros cuja finalidade é a criação de aprendizado de máquina.

- Tensorflow: Criada com foco em criação e treinamento de redes neurais, porém com uma grande variedade de aplicações. Foi utilizada em *backend*.
- Keras: Desenvolvida para ser utilizada *on top* ou no *frontend* do Tensorflow e outras plataformas. Tem como propósito a criação de redes neurais profundas de forma mais *user-friendly*, ou seja, intuitiva.

### 3.3.3 Seleção das colunas da tabela de dados (Features Selection)

A seleção de características, ou em inglês, *feature selection* é uma técnica cuja finalidade é estabelecer as melhores *features* dos dados com o intuito de diminuir o *overfitting*, ou sobreajuste. As *features* nada mais são do que as colunas do *dataset* que representa a natureza original do problema. Desta forma, selecionando as colunas que possuem maior significância para a resolução do problema aumentamos nosso êxito.

Foram escolhidas 3 técnicas de seleção de características apresentadas a seguir:

*Univariate Selection*: Teste estatístico que mede o quão forte é a relação da *feature* com a variável de saída, ou *target*. Está na biblioteca Scikit-Learn pela classe `SelectKBest`.

*Feature Importance*: É uma classe construída dentro de todos os classificadores baseados em árvores de decisão cuja propriedade é fornecer um *score* para cada *feature*. Quanto maior a pontuação maior a relação com a variável de saída.

*Correlation Matrix with Heatmap*: Demonstra através de uma matriz de mapa de calor qual a relação que as *features* tem entre si em com o *target*.

## 3.4 Árvores de decisão

Árvores de decisão tem se tornado cada vez mais populares dentre os algoritmos de *machine learning* quando se trata de aprendizado supervisionado. O objetivo da utilização de uma árvore de decisão é prever ou classificar classes ou valores. Sua estrutura é composta pelos seguintes termos:

*Root Node* (Nó Raiz): Representa toda a amostra e é dividido em dois ou mais conjuntos.

*Splitting*: É o processo de dividir um nó em dois ou mais subnós.



*Decision Node*: No momento em que um nó se divide em outros subnós dá se o nome de nó de decisão ou *decision node*.

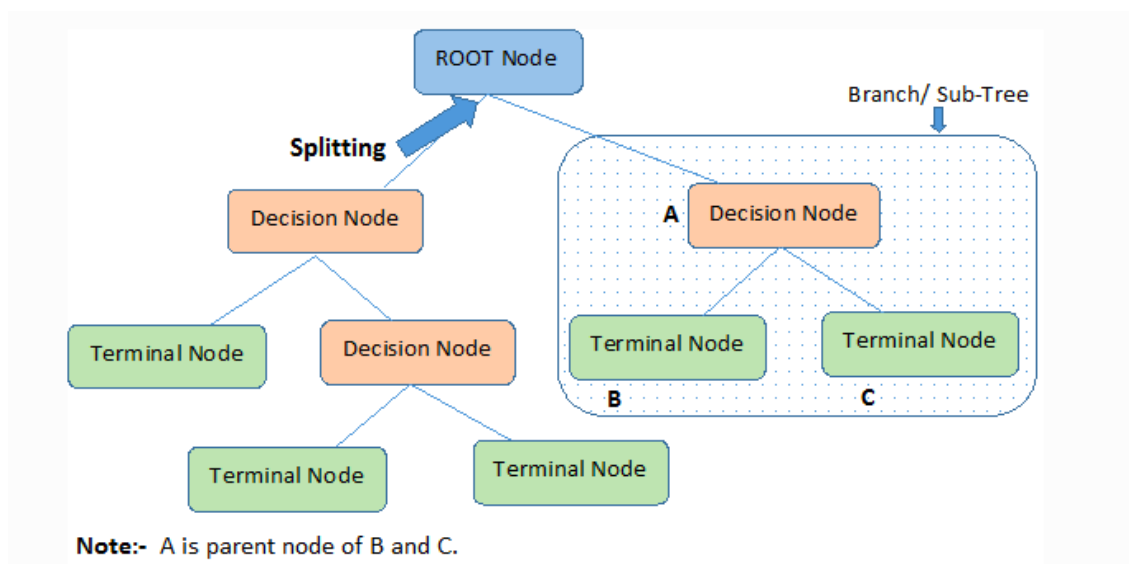
*Leaf/Terminal Node*: São os nós que se dividem.

*Prunning*: Processo oposto ao de divisão, é quando o nó é retirado ou podado da árvore.

*Branch/Sub-Tree*: Uma subseção de toda árvore.

*Parent and Child Node*: O nó que é dividido em subnós é o nó pai e suas subdivisões o nó filho.

A Figura 28 é uma breve ilustração.



**Figura 28** – Ilustração de uma árvore de decisão.

Fonte: (CHAUHAN,2019)

Toda árvore de decisão começa estabelecendo seu nó raiz ou somente raiz (*root*). A partir deste ponto os dados são subdivididos em direção as folhas ou nó fins. Cada nó corresponde a um teste de *feature* em que cada nova subdivisão corresponde a uma melhor resposta para o caso. O processo se repete recursivamente a cada raiz de subárvore (*subtree*) até um novo nó.

No início toda base de dados é raiz e os valores contidos em cada *features* são preparadas por categoria. Para isso é utilizado alguma abordagem estatística. O primeiro desafio é estabelecer qual será o atributo ou *feature* raiz.

A decisão de dividir os nós criando assim subárvores é um processo que tem grande impacto na performance do modelo. Existem alguns algoritmos que se reservam a fazer tal tarefa que nada mais são do que criar a própria árvore de decisão, em seguida:

ID3: Dicotomizador Iterativo é a primeira implementação das árvores de decisão que se seguiram em número de 3.

C4.5: Versão seguinte do ID3 implementando novas funcionalidades aceitando recursos contínuos e discretos, dados incompletos, resolve sobreajuste com técnica de *bottom-up* (pode ser entendido como uma técnica de *prunning*(poda), entre outros.

CART: É o acrônimo de árvore de classificação e regressão, porém muito se assemelha ao c4.5. Diferença é encontrada quando se baseia em critério de divisão numérica utilizando recursividade dos dados.

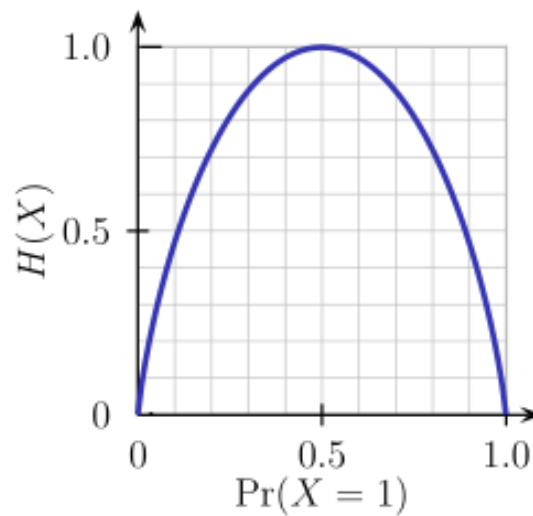
CHAID: Detector de interação automática  $\chi^2$  em *multi-level* splits, antecede o ID3.

MARS: *Splines* de regressão multi-adaptativa cuja função relevante pode-se encontrar em bibliotecas em R como *polymars*.

Para fins de exemplificação pode-se citar o ID3 que começa selecionando o nó raiz e em seguida se calcula a entropia e ganha de informações de cada *feature*. Cada *feature* selecionado deve ter a menor entropia e o maior ganho de informação. O nó raiz é então dividido por estas *features* selecionados para produzir uma nova subárvore ou uma menor base de dados. O algoritmo continua desta forma até que nenhuma nova *feature* seja utilizada para ser subdividida.

De fato, é complexo a seleção do ponto de partida, ou melhor, qual *feature* será utilizada como *root*. Isso não somente para o nó raiz como os próximos nós. Fazendo uma seleção meramente aleatória com certeza irá produzir resultados não desejados e de baixa acurácia. Para isso, utiliza-se técnicas de seleção desses critérios, a saber:

**Entropy:** É a medida de aleatoriedade da informação que está sendo processada. Quando maior a entropia maior a dificuldade de chegar-se a uma conclusão da informação. O lançar de moedas é um exemplo de como existe alto grau de entropia ou alto grau de aleatoriedade, ilustração na Figura 29 abaixo.



**Figura 29-** Entropia de um lançamento de moeda.

Fonte: CHAUHAN (2019).

A Figura 29 acima demonstra a entropia de um lançamento de moeda. A entropia é zero quando a probabilidade é 0 ou 1, ou seja, a entropia é máxima quando a probabilidade é de 0.5 ou 50%. Simplificando, quando existe probabilidade de 50% de cada lado ser atingido isso significa que a aleatoriedade está perfeita e não existe chance para que um evento tenha tendência de ser selecionado. Figura 30 é um exemplo do cálculo de entropia.

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$$

Play Golf	
Yes	No
9	5

$$\begin{aligned} \text{Entropy}(\text{PlayGolf}) &= \text{Entropy}(5,9) \\ &= \text{Entropy}(0.36, 0.64) \\ &= -(0.36 \log_2 0.36) - (0.64 \log_2 0.64) \\ &= 0.94 \end{aligned}$$

**Figura 30:** Entropia de um lançamento de moeda, fórmula e cálculo.

Fonte: Fonte: CHAUHAN (2019).

De forma resumida,  $S$  é o estado corrente e  $P_i$  a probabilidade de um evento  $i$  de estado  $S$  ou porcentagem da classe  $i$  no nó de estado  $S$ . Para múltiplas *features* pode-se entender a partir da figura X abaixo:

Onde  $T$  é o estado corrente e  $X$  é a *feature* selecionada.

**Information Gain:** É uma propriedade estatística que mede o quão bem separado está uma *feature* dos exemplos dados numa base de treino de acordo com a classificação pretendida. Construir uma árvore de decisão é encontrar todas as *features* que retornam o maior ganho de informação e a menor entropia. O ganho de informação computa a diferença entre a entropia antes da divisão e a média de entropia após a divisão. Matematicamente pode ser visto na equação abaixo:

$$\text{Information Gain} = \text{Entropy}(\text{before}) - \sum_{j=1}^K \text{Entropy}(j, \text{after}) \quad (10)$$

Onde *before* é a base de dados antes da divisão, o  $K$  é o número de *subsets* (porções menores da base de dados original) gerados pela divisão e  $(j, \text{after})$  é o *subset* de  $j$  após a divisão.

Um exemplo mais prático é dado na Figura 31:

$$\text{Information Gain}(T, X) = \text{Entropy}(T) - \text{Entropy}(T, X)$$

$$\begin{aligned} \text{IG}(\text{PlayGolf}, \text{Outlook}) &= E(\text{PlayGolf}) - E(\text{PlayGolf}, \text{Outlook}) \\ &= 0.940 - 0.693 \\ &= 0.247 \end{aligned}$$

**Figura 31:** Ganho de informação.

Fonte: CHAUHAN (2019).

**Gini Index:** Pode-se entender como uma função de custo utilizada para calcular as divisões na base de dados. Calcula-se subtraindo a soma das probabilidades ao quadrado de cada uma das classes. Abaixo a fórmula:

$$Gini = 1 - \sum_{i=1}^c (p_i)^2 \quad (11)$$

Gini Index trabalha exclusivamente com divisões binárias de variáveis categóricas como “Certo” ou “Errado”. Quanto maior for o valor do *Gini Index* maior será a homogeneidade. Calcula-se o *Gini* através formula acima para *certo*( $p$ ) e *errado*( $q$ ) ( $p^2+q^2$ ). Em seguida para cada divisão usa-se o *Gini Score* de cada nó que dividiu.

**Gain Ratio:** Ganho de informação é muito tendencioso para escolher features com grandes quantidades de valores, preferindo as *features* com grandes quantidades de números de valores distintos. O *gain ratio* é uma modificação do ganho de informação com a tentativa de reduzir este viés e normalmente é uma opção melhor. Abaixo sua fórmula.

$$Gain\ Ratio = \frac{Information\ Gain}{SplitInfo} = \frac{Entropy(before) - \sum_{j=1}^k Entropy(j,after)}{\sum_{j=1}^k w_j \log_2 w_j} \quad (12)$$

**Reduction in Variance:** São utilizadas mais em problemas de regressão de variáveis contínuas. A divisão com menor variância é selecionada como critério de divisão. O algoritmo utiliza fórmula padrão de variância, segue abaixo.

$$Variance = \frac{\sum (x - \bar{x})^2}{n} \quad (13)$$

Onde o  $\bar{x}$  é a média dos valores, o  $x$  é o atual e  $n$  é o número de valores. São duas etapas, a primeira calcula a variância de cada nó e a segunda calcula a variância de cada divisão como a média ponderada de cada variância do nó.

**Chi-Square:** É um dos métodos mais antigos de classificação em árvores de decisão. Ele descobre a significância estatística entre as diferenças entre o subnós e o nó pai. Mede-se pela soma dos quadrados das diferenças padronizadas entre as frequências observadas e esperadas da variável pretendida. Funciona como uma variável destino categórica (*certo* ou *errado*). Quanto maior o valor do  $\chi^2$  maior a significância estatística. Matematicamente pode ser entendido conforme equação abaixo:

$$X^2 = \sum \frac{(O-E)^2}{E} \quad (14)$$

Onde  $O$  é o *score* observado e o  $E$  é o *score* esperado.

O cálculo se divide em duas partes. A primeira calculando o  $\chi^2$  para o nó individual e em seguida o  $\chi^2$  da divisão.

### 3.4.1 Random Forest

Uma *Random Forest* é um algoritmo de *machine learning* de fácil utilização e compreensão. É utilizado tanto para tarefas de regressão quanto para predição ou classificação. Na sua maioria, as *random forests* produzem resultados bem satisfatórios sem que se necessite o ajuste intenso de seus hiper parâmetros.

Em grande parte dos casos são utilizados o método de treinamento *bagging* que resumidamente se utiliza da combinação de várias árvores de decisão para aumentar a performance.

As árvores de decisão são algoritmos de aprendizado supervisionado que possui inúmeras vertentes como *Gradient Boost Machine*, *XGBoost* dentre outras como a própria *Random Forest*. Este algoritmo é um dos pilares do aprendizado de máquina.

Um algoritmo de floresta aleatória adiciona aleatoriedade ao modelo afim de se criar vários modelos diferentes. Ao invés de procurar uma melhor configuração de uma única árvore o algoritmo tenta por um melhor subconjunto aleatório das características. Portanto, criar uma árvore aleatória é criar um modelo de árvore satisfatória, mas com a intenção de escolher melhor suas características, ou, *features*.

Existe uma grande importância ao se escolher bem as *features* para o modelo pois é de senso comum que em aprendizado de máquina muitas *features* sobreajustam o modelo, chama-se *overfitting*. E o contrário também é verdadeiro, modelos que são treinados com poucas *features* subajustam ou *underfitting*.

### 3.4.2 Alguns hiper parâmetros

Para aumentar o poder preditivo é adicionado o valor do *n\_estimators* cuja função determina o número de árvores construídas antes de fazer a predição. Com o aumento da quantidade de árvores aumenta-se consequentemente a necessidade de performance computacional o que em certos casos pode-se gerar lentidão ou travamento.

Um segundo hiper parâmetro de grande importância é o *max\_features* ou o número máximo de *features* que serão utilizadas pelo modelo. Bem como a quantidade de features importa para uma melhor performance do modelo também é preciso se ater ao tipo de função que medirá a impureza ou ganho de informação. Para tal, *criterion* é quem define o tipo de entropia, é comum se utilizar o *Gini Index* por padrão.

Outro hiperparâmetro a ser considerado é o *min\_sample\_leaf* que indica o número mínimo de folhas a ser utilizado em cada árvore. Como o número máximo de folhas também a profundidade das árvores é importante, o *max\_depth*.

Alguns hiperparâmetro que devem ser levados em consideração para uma melhor performance computacional pode-se destacar o *n\_jobs* cuja função é determinar a quantidade de processadores irão ser utilizados para a tarefa. O *random\_state* ou estado randômico serve para que o resultado seja replicável caso seja utilizado os mesmos valores nos dados de treinamento. Há de se citar o *oob\_score* que é um método de validação cruzada para *random forests* cuja função determina que um terço dos dados não sejam utilizados no treinamento, mas que pode ser usado em uma posterior avaliação da performance.

Para se designar o mínimo de samples requeridos para dividir um nó interno é utilizado o *min\_samples\_split*. Um nó irá se dividir caso a divisão diminua a impureza o estabeleça o mesmo valor, para isso utiliza-se o hiperparâmetro *min\_impurity\_decrease*. Para limitar o crescimento da árvore indefinidamente é necessário estabelecer um valor de *threshold* para que o nó não se divida caso a impureza esteja acima do valor de *threshold*, de outra forma será uma folha, este é o *min\_impurity\_split*.

Um ponto muito importante para trabalhar grande conjuntos de dados é a utilização de balanceamento das classes para que não ocorra prejuízo no aprendizado de nenhuma e que isso se dê de forma mais harmônica possível. Neste ponto, tem-se o hiperparâmetro *class\_weight* com esta finalidade.

Outros hiperparâmetros existem e são aplicados caso a caso e não serão abordados aqui por uma questão de praticidade.

A *random forest* utilizada no estudo foi configurada com critério de cálculo de impureza a *entropy* ou entropia. A profundidade máxima de suas árvores num valor de 9. O número máximo de *features* utilizadas pelo modelo. O número mínimo de folhas e divisões estão na ordem de 3 unidades. A quantidade de árvores construídas para a predição do modelo será de 125. Será utilizado o método semelhante ao de validação cruzada, o *oob\_score* e será utilizado

todos os núcleos do processador para criação do modelo. O modelo será acompanhado por mensagens em tempo real de seu aprimoramento.

### 3.5 Rede Neural

Atualmente as redes neurais estão na vanguarda nos algoritmos de *machine learning* possuindo uma denominação específica dentro do tema, a *deep learning*. Dentre vários algoritmos de aprendizado de máquina as redes neurais se destacam por atingir inúmeros estados da arte em pesquisas e desempenharem performance acima dos outros modelos já consagrados, como é o caso das árvores de decisão.

Uma discussão que se estende tanto pelo meio acadêmico quanto na indústria é o fato de as redes neurais serem consideradas verdadeiras “caixas-pretas” no que tange ao que acontece em suas camadas ocultas, ou *hidden layers*. Entendendo melhor o que está acontecendo nas camadas ocultas é um dos meios pelos quais a construção das redes neurais e sua otimização torna-se eficaz.

A inspiração para as redes neurais, também chamadas de redes neurais artificiais, é o cérebro humano, ou o que se sabe até o momento da sua estrutura e funções. Por definição básica, pode-se entender as redes neurais como sistema de computação que conjuga elementos interconectados chamados de nó organizando-os em camadas cujo finalidade é calcular dados que chegam pela camada de entrada, computá-los de diversas formas até que terminem pela camada de saída com significado requerido.

Basicamente redes neurais são excelentes em tarefas de reconhecimento de padrões e possui em sua estrutura elementos chamados de *weights* e *bias*, ou em português *peso* e *viés*. Ainda de forma sucinta pode-se definir o funcionamento de uma rede neural como uma entrada de dados na camada de entrada que irá calcular a soma dos pesos adicionando o viés e como resultado irá passar por uma função de ativação cuja função é definir se esta informação irá continuar a se propagar pela rede ou não, ou seja, irá ativar ou não. O movimento que leva as informações de camada a camada é chamado de *forward pass*. Por fim a camada de saída irá ter como resultado a informação desejada. Figura 32 e 33 é um exemplo de uma rede neural comum.



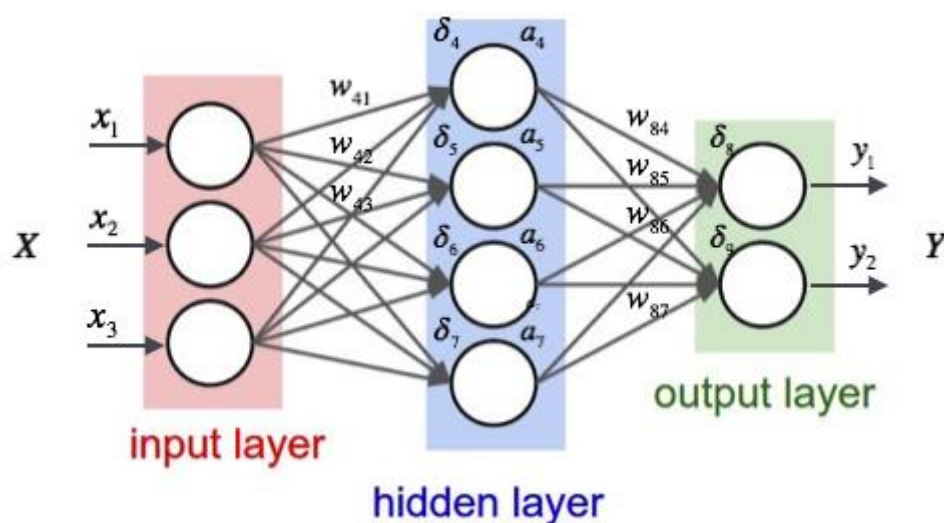


Figura 32: Exemplo de rede neural

Fonte: VALKOV (2017)

Para entender um pouco melhor como as redes neurais funcionam faz-se necessário explicar por onde iniciou-se e qual sua arquitetura mais básica.

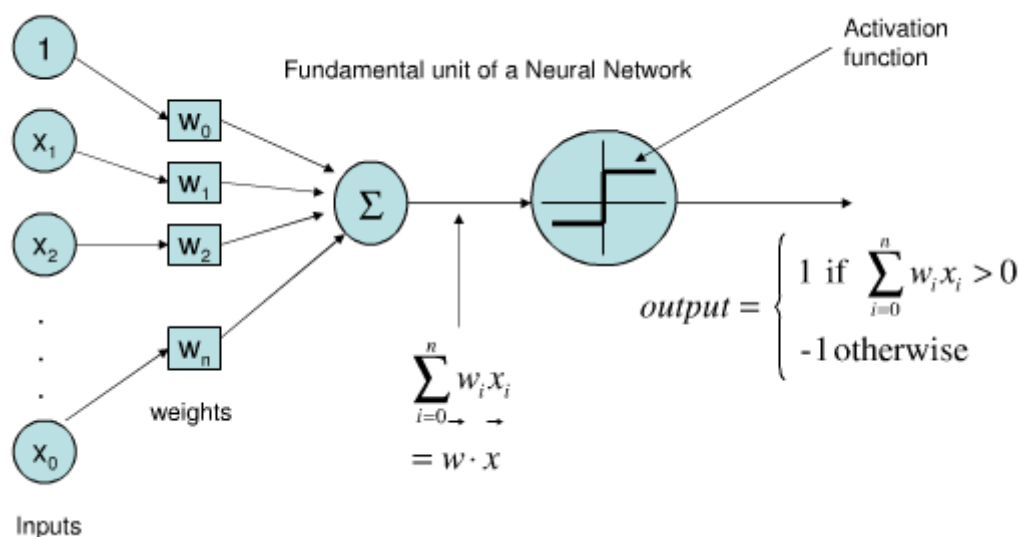
As redes neurais começam em 1943 com os pesquisadores Warren McCullock e Walter Pitts descrevendo o primeiro conceito da célula cerebral simplificada, foi chamado de neurônio McCullock-Pitts (MCP). Descreveu-se que a célula cerebral era uma simples porta lógica com saída binária. A dinâmica observada do fluxo do sinal em uma célula cerebral seria a base para a formação da rede neural artificial. Em suma, múltiplos sinais chegam aos dendritos e são levados até o corpo da célula e caso esse acúmulo ultrapasse um certo limite (*threshold*) então é gerado uma saída para esse sinal que irá passar pelo axônio. Uma breve comparação entre as redes neurais cerebrais e as artificiais pode-se considerar da seguinte forma:

Os neurônios biológicos seriam os neurônios artificiais, o núcleo da célula o nó ou unit (unidade/neurônio), os dendritos a camada de entrada (*input*), as sinapses os *weights* ou conexões e o axônio a camada de saída ou simplesmente *output*.

Sua arquitetura mais básica foi desenvolvida em 1957 por Frank Rosenblatt chamada de algoritmo *Perceptron* cujo usabilidade era para aprendizado supervisionado de classificadores binários.

Conceitualmente a arquitetura *perceptron* multiplica todas as entradas pelos seus pesos (*weights*) cuja interpretação pode ser, o quão importante esta entrada é para a saída. Em seguida soma-se todos os pesos como uma soma ponderada e aplica-se a função de ativação que determina se a soma dos pesos irá ser ativada ou não dependendo do valor de *threshold*. Se o

valor da soma dos pesos for maior que a do *threshold* então atribui-se 1 e 0 ou -1 caso seja menor.



**Figura 33:** Esquema da Rede neural.

Fonte: VALKOV (2017)

O funcionamento de todo neurônio da rede se dá basicamente desta forma, porém quando a saída de uma camada é entrada para outra é chamado de *feedforward* e a informação passa somente uma vez. Os pesos e os vieses normalmente são inicializados aleatoriamente. Porém, somente esta estrutura não explica como uma rede neural vai aprender ou reconhecer padrões. Portanto, o primeiro passo para que o que está sendo predito pela rede neural é de fato o que está classificado na base de dados se dá através do cálculo de erro ou melhor definido como *cost function*, função de custo. A propriedade da função de custo é calcular tal diferença através de métricas, dentre elas a mais comum *mean squared error*, erro quadrado médio. A questão que se põe é que pequenos erros não seriam perceptíveis aos vieses e então não mudaria de forma significativa a saída predita, portanto o fato de ser elevado ao quadrado isto maximiza o erro tornando-o agora perceptível.

Para que o erro diminua gradativamente e aumente assim a precisão do que está sendo predito usa-se a ferramenta de gradiente descendente, ou *gradient descent*.

Sabe-se através do cálculo que funções possuem máximos e mínimos globais que podem ser alcançados através de derivadas e derivadas parciais. Portanto, uma maneira de saber esses pontos, ou se chegar o mais perto possível são através de derivadas.

Para alcançar valores perto dos mínimos globais precisa-se de um número a ser utilizado como pequenos passos em sua direção. É chamado *learning rate*, ou taxa de aprendizado, cuja função é ser o tamanho deste “passo” em direção ao mínimo global. Caso essa taxa seja muito pequena o modelo irá demorar muito para convergir e caso contrário pode ultrapassar o mínimo e nunca conseguir encontrá-lo. O propósito é encontrar o melhor valor.

Após escolhido este melhor valor é possível então encontrar melhor pesos e vieses que criem um modelo de rede neural mais preciso. De forma resumida, o gradiente descendente computa a derivada repetidas vezes corrigindo os pesos e vieses com a finalidade de encontrar o menor valor para a função de custo. Desta forma que se dá o aprendizado da rede neural.

Todo este processo que é repedido várias vezes até que a rede neural tenha sua melhor performance, ou seja tenha seu valor predito mais próximo possível do valor real, e isso se consegue através da técnica chamada *backpropagation*.

*Backpropagation* é um dos pilares do aprendizado da rede neural pois é através deste método que toda rede atualiza seus pesos e vieses a fim de melhorar performance. Ao final de todo passo de *feedforward* ocorre a propagação de volta, agora com os erros calculados que irão atualizar os pesos e vieses e o valores das derivadas parciais mais precisos.

### 3.5.1 Arquitetura

A arquitetura da rede neural começa pela camada de entrada contendo 200 neurônios computando 49 *features* e como saída a função de ativação ReLU. A camada oculta possui 600 neurônios cuja função de ativação é ReLU. A camada de saída representa o que queremos distinguir, neste caso 6 tipos de partículas, ou seja, 6 classes, portanto 6 neurônios. Estas mesmas classes necessitam de uma função de ativação que consiga interpretar multiclases, sendo assim, a *softmax* é a mais indicada.

### 3.5.2 Otimização

A otimização da rede neural inicia-se com a escolha do número de neurônio em cada camada bem como sua função de ativação. A escolha de se ter ou não uma ou mais camadas varia de caso a caso, neste utilizamos camadas ocultas. A escolha determina uma maior abstração dos dados utilizados.

Após escolha da arquitetura base da rede neural faz-se necessário a otimização de alguns hiperparâmetros como *batch size*, *batch normalization*, regularizadores L2 bem como *Dropout*., O tamanho do batch é de 512 samples, regulador L2 assume valor de 0.01 e *Dropout* de 0.1.

As otimizações seguem utilizando Adam para aperfeiçoar a taxa de aprendizado com valores padrões dos seus índices. Taxa de aprendizado é de 0.01 e a métrica utilizada para avaliar o modelo final é a F1-score através da matriz de confusão. O link para o código utilizado se encontra no Apêndice A.

## 4 RESULTADOS E DISCUSSÕES

### 4.1 Transformação dos dados

A transformação ou preparação dos dados é etapa fundamental para o êxito dos algoritmos. Diminuir o sobreajuste e aumentar a acurácia sobre novos dados é tarefa que pode ser alcançada com a transformação dos dados.

Parte fundamental de toda criação de aprendizado de máquina, a preparação dos dados, acumula cerca de 80% do tempo dedicado a construção do modelo. É comum se utilizar de diversas técnicas para melhorar o conjunto de dados. O resultado da utilização de algumas dessas está na Seção 4.2.

### 4.2 Features Selections

#### 4.2.1 SelectKBest

A seleção das *features* é um momento importante dentro de todo *pipeline* do aprendizado de máquina. É uma tarefa particular de cada autor e designada ao problema em específico.

Neste trabalho foi utilizado a classe *SelectKBest* da biblioteca *Scikit-learn* conforme mencionado na metodologia. Dentro das possibilidades de configuração desta classe, o parâmetro de referência é o *k*. Este estabelece qual será a quantidade de *features* a serem medidas. Esta classe necessita da métrica que será utilizada para classificação. Para esta tarefa foi utilizada *f\_classif* cuja função é classificar classes categóricas. O resultado está na Tabela 21 abaixo:

**Tabela 21:** Ranking SelectKBest.

Rank	Feature	Score	Rank	Feature	Score
1	MuonFlag	22.218.972.743	26	TrackNDoFSubdetector1	375.634.168
2	MuonLLbeMuon	17.436.981.251	27	FlagBrem	363.034.014
3	MuonLLbeBCK	17.381.647.601	28	DLLproton	360.525.034
4	MuonLooseFlag	17.155.008.704	29	DLLkaon	354.843.911
5	GhostProbability	13.367.591.884	30	FlagMuon	284.717.332
6	TrackQualityPerNDoF	4.781.059.022	31	FlagRICH1	271.959.151
7	EcalE	2.048.503.093	32	DLLmuon	161.442.777
8	RICHpFlagKaon	1.715.691.535	33	TrackNDoFSubdetector2	142.756.071
9	RICHpFlagProton	1.387.135.400	34	DLLelectron	141.369.789
10	TrackPt	1.370.497.843	35	TrackQualitySubdetector2	123.533.528
11	RICH_DLLbeKaon	1.102.593.072	36	TrackQualitySubdetector1	99.861.172
12	RICH_DLLbeProton	1.091.646.700	37	Calo2dFitQuality	89.751.816
13	RICH_DLLbeBCK	1.084.231.689	38	HcalDLLbeMuon	87.504.847
14	RICH_DLLbeMuon	1.056.829.335	39	HcalDLLbeElectron	85.578.089
15	RICHpFlagPion	994.245.150	40	SpdE	75.345.852
16	RICH_DLLbeElectron	991.540.423	41	PrsDLLbeElectron	67.639.263
17	TrackP	906.235.209	42	FlagHcal	66.901.497
18	Calo3dFitQuality	893.224.588	43	PrsE	61.445.299
19	FlagRICH2	879.764.352	44	FlagSpd	60.614.916
20	TrackP	0.018811	45	FlagEcal	0.002703
21	EcalDLLbeElectron	0.017886	46	FlagPrs	0.002653
22	TrackNDoF	0.017333	47	RICHpFlagMuon	0.002556
23	TrackDistanceToZ	0.016935	48	FlagSpd	0.002343
24	EcalE	0.016927	49	RICHpFlagElectron	0.001710
25	TrackNDoFSubdetector2	0.016236	###	###	###

Fonte: Elaboração própria.

#### 4.2.2 Extra Tree

A classe *feature\_importances\_* é utilizada para descartar *features* possivelmente irrelevantes para o modelo. Esta é uma classe que estão presentes em árvores de decisão, neste caso foi utilizada a árvore *extra tree* e pode ser encontrada na biblioteca *Scikit-learn*. Na Tabela 22 está o resultado.

**Tabela 22:** Ranking Extra Tree

Rank	Feature	Score	Rank	Feature	Score
1	DLLelectron	0.078848	26	TrackQualitySubdetector2	0.016122
2	GhostProbability	0.077083	27	TrackQualitySubdetector1	0.015760

3	DLLkaon	0.044451	28	TrackNDoFSubdetector1	0.015495
4	RICH_DLLbeKaon	0.041319	29	EcalDLLbeMuon	0.014612
5	RICH_DLLbeElectron	0.040032	30	BremDLLbeElectron	0.014360
6	DLLproton	0.037865	31	RICHpFlagKaon	0.012414
7	MuonLLbeMuon	0.035772	32	Calo2dFitQuality	0.011582
8	RICH_DLLbeProton	0.035604	33	HcalDLLbeMuon	0.011041
9	RICH_DLLbeBCK	0.035475	34	HcalDLLbeElectron	0.010948
10	TrackQualityPerNDoF	0.034119	35	HcalE	0.010931
11	MuonLLbeBCK	0.033552	36	EcalShowerLongitudinalParameter	0.010746
12	MuonFlag	0.032529	37	RICHpFlagProton	0.008007
13	DLLmuon	0.031125	38	SpdE	0.007800
14	MuonLooseFlag	0.029231	39	FlagRICH2	0.006793
15	PrsE	0.022708	40	RICHpFlagPion	0.005832
16	TrackPt	0.022612	41	FlagMuon	0.004450
17	RICH_DLLbeMuon	0.021691	42	FlagBrem	0.004269
18	PrsDLLbeElectron	0.021111	43	FlagRICH1	0.003867
19	Calo3dFitQuality	0.020736	44	FlagHcal	0.003014
20	TrackP	0.018811	45	FlagEcal	0.002703
21	EcalDLLbeElectron	0.017886	46	FlagPrs	0.002653
22	TrackNDoF	0.017333	47	RICHpFlagMuon	0.002556
23	TrackDistanceToZ	0.016935	48	FlagSpd	0.002343
24	EcalE	0.016927	49	RICHpFlagElectron	0.001710
25	TrackNDoFSubdetector2	0.016236	###	###	###

Fonte: Elaboração própria.

#### 4.2.3 Matriz de correlação

A matriz de correlação demonstra o quão relacionada é uma *feature* específica as demais dos conjuntos de dados ou da própria variável a ser predita. A correlação admite valores dentre -1 a 1 sendo que quanto mais próximo ao negativo menos correlação existe e vice versa. A Figura 34 ilustra uma pequena parcela desta matriz e a imagem completa encontra-se no Apêndice A.



Figura 34: Recorte da matriz de correlação.

Fonte: Elaboração própria.

#### 4.2.4 LightGBM Selection

*LightGBM* é um *gradient boost* de aprendizado em árvore. A diferença deste para outros *gradient boost* está no seu crescimento vertical da árvore enquanto outros fazem-no de forma horizontal. Pode ser entendido também como um crescimento em ordem das folhas ao invés de nível. A classificação das *features* se deu conforme Tabela 23.

**Tabela 23:** Ranking LightGBM.

<b>Rank</b>	<b>Feature</b>	<b>Score</b>	<b>Rank</b>	<b>Feature</b>	<b>Score</b>
<b>1</b>	TrackP	3984	<b>26</b>	Calo2dFitQuality	1850
<b>2</b>	RICH_DLLbeElectron	3799	<b>27</b>	Calo3dFitQuality	1706
<b>3</b>	TrackPt	3773	<b>28</b>	HcalDLLbeMuon	1506
<b>4</b>	PrsDLLbeElectron	3696	<b>29</b>	TrackQualitySubdetector2	1263
<b>5</b>	DLLkaon	3567	<b>30</b>	MuonLLbeBCK	1251
<b>6</b>	RICH_DLLbeKaon	3531	<b>31</b>	RICHpFlagKaon	1244
<b>7</b>	RICH_DLLbeBCK	3331	<b>32</b>	EcalShowerLongitudinalParameter	1174
<b>8</b>	DLLproton	3298	<b>33</b>	SpdE	1028
<b>9</b>	RICH_DLLbeProton	3198	<b>34</b>	MuonLLbeMuon	1017
<b>10</b>	GhostProbability	3175	<b>35</b>	RICHpFlagProton	922
<b>11</b>	DLElectron	3067	<b>36</b>	FlagRICH1	914
<b>12</b>	PrsE	2919	<b>37</b>	FlagRICH2	877
<b>13</b>	RICH_DLLbeMuon	2903	<b>38</b>	RICHpFlagPion	846
<b>14</b>	EcalE	2662	<b>39</b>	TrackQualitySubdetector1	814
<b>15</b>	TrackQualityPerNDoF	2362	<b>40</b>	MuonLooseFlag	769
<b>16</b>	TrackNDoFSubdetector1	2354	<b>41</b>	FlagPrs	733
<b>17</b>	TrackNDoFSubdetector2	2302	<b>42</b>	MuonFlag	723
<b>18</b>	DLLmuon	2262	<b>43</b>	FlagBrem	674
<b>19</b>	TrackDistanceToZ	2168	<b>44</b>	FlagEcal	664
<b>20</b>	TrackNDoF	2155	<b>45</b>	FlagMuon	626
<b>21</b>	EcalDLLbeElectron	2070	<b>46</b>	FlagSpd	610
<b>22</b>	EcalDLLbeMuon	2027	<b>47</b>	RICHpFlagElectron	583
<b>23</b>	HcalE	1873	<b>48</b>	RICHpFlagMuon	516
<b>24</b>	HcalDLLbeElectron	1863	<b>49</b>	FlagHcal	495
<b>25</b>	BremDLLbeElectron	1856	<b>###</b>	<b>###</b>	<b>###</b>

Fonte: Elaboração própria.

### 4.3 Resultados principais

No princípio do levantamento do método apenas fora escolhido dois algoritmos sem quaisquer técnicas de seleção de *features*. O foco principal até então era de somente aprimorar ambos os modelos para gerar melhor precisão e assim classificar melhor as partículas. Porém ao se tentar inúmeras otimizações com diferentes tipos de arquiteturas os modelos não tinham melhora significativa quando testados em novos dados (*testing data*). Com isso, suspeitando uma possível exacerbação de dados ruidosos ou dificuldade de encontrar melhorias nos próprios algoritmos teve-se como solução incrementar novos modelos e selecionar *features* com técnicas diferentes.



O primeiro método escolhido foi o *SelectKBest* cuja função principal é escolher as  $k$  *features* de maior *score*. O número de  $k$  *features* foi determinado através de uma variável de corte que corresponde a 1% do maior *score* arbitrariamente. Ou seja, qualquer *feature* que obteve menos que 1% do valor da *feature* com maior *score* é descartada. Desta forma, a número de *features* escolhido foi 31 (trinta e um).

Em seguida foi utilizado o atributo *feature\_importance\_* do metatransformador *SelectFromModel* que se utilizou do modelo de árvore *ExtraTreeClassifier*. Tal atributo possui parâmetro chamado *treshold* que determina o coeficiente de corte podendo ser uma média, mediana ou um número *float* arbitrário. O *threshold* escolhido foi de 10% do valor do maior *score*. Este também de forma arbitrário e como resultado apenas 36 *features* foram selecionadas. Tal método foi chamado aqui de *ExtraTree*.

Já em um terceiro momento foi definido como método para seleção uma matriz de correlação (*Corr Matrix*) cuja resultado dentre as *features* fora maior que 0.9 uma delas será desconsiderada. Vale lembrar que os índices variam de -1(menos um) para total falta de correlação e 1(um) para o contrário. Sendo assim foram escolhidas 30(trinta) *features*.

O quarto método utilizado deriva também do mesmo meta transformador e atributo *feature\_importance\_* do segundo método porém se utiliza do modelo *LightGBM* para gerar seus *scores*. Como limítrofe para escolha das *features* foi utilizado a média dentre os *scores*. Todas *features* que obtiveram *scores* acima da média foram selecionadas restando um total de 22(vinte e duas). Por fim foi definido aqui como *LightGBM*.

Por fim foi chamado de *No Selection* quando o modelo se utiliza de todas as *features* do *dataset*, ou seja 49.

Nas tabelas 24, 25, 26 e 27 estão os resultados provenientes de cada modelo sobre cada seleção diferente das *features*. As métricas utilizadas foram *logloss* e *accuracy*, esta última representada em valores de porcentagem adquiridas na base de treino (*Training Set*) e teste (*Testing Set*). A melhor indicação de *logloss* será sempre quando este for o mais baixo, porém já a *accuracy* estabelece-se o contrário.

Em se tratando de tempo de execução foi registrado que cada modelo precisou de 60 a 180 minutos para obtenção dos resultados.

**Tabela 24:** Resultados da *Random Forest*

<b>Método</b>	<b>Logloss</b>	<b>Training Set</b>	<b>Testing Set</b>
<i>No Selection</i>	<b>0,622</b>	<b>81,4</b>	<b>75</b>
<i>SelectKBest</i>	0,650	80,2	74,1
<i>ExtraTree</i>	0,652	80,2	74,1
<i>Corr Matrix</i>	0,763	76,6	68
<i>LightGBM</i>	0,666	79,8	73,5

Fonte: Elaboração própria.

**Tabela 25:** Resultados da *Gradient Boost Classifier*.

<b>Método</b>	<b>Logloss</b>	<b>Training Set</b>	<b>Testing Set</b>
<i>No Selection</i>	<b>0,563</b>	<b>78,6</b>	<b>77,2</b>
<i>SelectKBest</i>	0,599	77,7	75,7
<i>ExtraTree</i>	0,599	77,7	75,7
<i>Corr Matrix</i>	0,711	71,6	70,0
<i>LightGBM</i>	0,625	76,2	74,7

Fonte: Elaboração própria.

**Tabela 26:** Resultados da *XGBoost*

<b>Método</b>	<b>Logloss</b>	<b>Training Set</b>	<b>Testing Set</b>
<i>No Selection</i>	<b>0,584</b>	<b>76,7</b>	<b>76,4</b>
<i>SelectKBest</i>	0,615	75,5	75,2
<i>ExtraTree</i>	0,615	75,5	75,2
<i>Corr Matrix</i>	0,73	69,8	69,4
<i>LightGBM</i>	0,637	74,6	74,3

Fonte: Elaboração própria.

**Tabela 27:** Resultados da Rede Neural.

<b>Método</b>	<b>Logloss</b>	<b>Training Set</b>	<b>Testing Set</b>
<i>No Selection</i>	<b>0,571</b>	<b>76</b>	<b>76</b>
<i>SelectKBest</i>	0,606	75	75

<i>ExtraTree</i>	0,625	70,0	69,0
<i>Corr Matrix</i>	0,717	70,0	69,0
<i>LightGBM</i>	0,660	73,0	73,0

---

Fonte: Elaboração própria.

Em análise é possível notar que nenhum dos modelos em nenhuma seleção de *features*, ou na falta dela, obtiveram resultado significativamente diferentes. Porém, quando não ocorre nenhuma seleção (*No Selection*) obtém-se o melhor resultado.

#### 4.4 Matriz de Confusão e Relatório

Os resultados das matrizes de confusão e relatórios de cada modelo utilizando em cada seleção de *features* diferentes tanto nas bases de treino como teste estão dispostas em maiores detalhes no Anexo A.

Para início de análise faz-se necessário identificar o resultado da técnica *one-hot encoding*. A transformação dos dados categóricos em numéricos foram da seguinte forma:

**{'Electron' = 0, 'Ghost' = 1, 'Kaon' = 2, 'Muon' = 3, 'Pion' = 4, 'Proton' = 5}**

Em análise generalista foi possível perceber que existe grande conexão entre a performance dos modelos de aprendizado de máquina com a proporção de acertos sejam na matriz de confusão ou no relatório com outras métricas envolvidas.

De forma decrescente e observando de forma resumida nota-se que as partículas múons possuem maior facilidade de aprendizado com cerca de 90% de precisão seguida dos elétrons com 85%, ghosts 75%, píons 70% e entre 60% e 65% os prótons e kaons.

Uma explicação plausível tanto para o mundo da física quanto para os dados é de que do ponto de vista da física faz-se necessário entender que os múons além de terem um detector só para esse tipo de partícula também apresentam um livre caminho médio longo sendo detectamos diretamente na câmara de múons. Este fato corrobora com a precisa elevada em comparação as outras partículas. Pelo lado dos dados existe uma correspondência estrutural e proporcional aonde os recolhimentos das interações do detetor se transforma numa quantidade de características dos dados.

## 5 CONCLUSÕES

O método se mostrou eficaz na proposta de classificação de partículas diante da complexidade do tema e na melhora da precisão dos algoritmos. Foram encontradas algumas dificuldades durante o processo de otimização dos modelos como tempo de processamento e ajuste dos hiperparâmetros.

De volta a hipótese, verificou-se que não houve diferença significativa na eficácia da rede neural sobre a árvore de decisão. Após exaustivas tentativas de averiguar razão estima-se que haja limitação na natureza dos dados ou na relação dentre eles que possam concluir melhor a classificação de partículas. É certo que o LHC possui inúmeras outras interações das partículas em seus detetores e que a base de dados utilizadas neste trabalho é apenas um recorte do que é possível. Porém, não se pode negar o êxito em se utilizar técnicas de aprendizado de máquina em problemas de física de partículas pois, ao se ter uma grande quantidade de dados cria-se material fértil para aplicação de métodos computacionais.

É notável a limitação de ambos modelos de exceder a precisão satisfatória em bases de testes. De forma arbitrária poder-se-ia pensar em precisão acima de 80%.

Embora a árvore de decisão *gradient boost classifier* tenha superado em 1,2% a rede neural não se pode declarar vantagem significativa, ou melhor, não expressa uma significância estatística.

Chama a atenção os resultados derivados das matrizes de confusão. Pode-se concluir que existe a necessidade por mais interações, ou seja mais dados, e isso apoia-se no fato das partículas múons obterem maior precisão na classificação. Os múons, ao se decaírem durante todo o processo de trajetória geram informações e detecções que por si levam a mais características a serem analisadas. Portanto, um passo para trabalhos futuros pode ser encontrado na obtenção de mais e diferentes dados da natureza do processo das colisões.

Ainda acerca dos resultados relacionados às partículas das matrizes de confusão, impressiona o fato de ter sido classificado com 80% de precisão as partículas chamadas aqui por “*ghosts*”. Isso traz à tona uma dúvida quanto à natureza dos dados gerados pelos detetores. Por qual motivo teria tanta precisão dados que não correspondem a partículas algumas. Seriam apenas ruídos do experimento ou algo a mais?

Como resposta para esta dúvida coloca-se aqui um processo para trabalhos futuros que visam investigar mais precisamente a construção dos dados gerados pelos detetores e uma relação às teses físicas sobre estes dados das partículas “*ghosts*”.

Não somente haja interesse pela revisão teórica da física de partículas e a natureza da síntese dos dados pelos detetores, há ainda necessidade de pesquisa e melhora nos modelos de aprendizado de máquina. Todo processo de otimização dos modelos ainda se faz necessário forte análise e conhecimento dos seus hiperparâmetros e não somente utilização de código de programação. Alimentar os resultados com ilustrações mais precisas e condizentes com aspectos científicos é alvo futuro também.

Por fim, todo o código gerado por esta metodologia está disponível no repositório GitHub e o endereço encontra-se no Apêndice A.

## REFERÊNCIAS

AAIJ, R.; et al, LHCb collaboration, J. Instrum. 3, S08005, 2008; Phys. Rev. Lett. 108, 251802, 2012; Phys. Lett. B 712, 203, 2012; Phys. Rev. D 85, 112013, 2012; Phys. Rev. Lett. 108, 201601, 2012; J. High Energy Phys. 10, 037, 2012; Eur. Phys. J. C 72, 2022, 2012.

ABDALLA, M. C. “O Charme das Partículas Elementares” 2006.

ADAM-BOURDARIOS, C. et al. The Higgs Machine Learning Challenge. Journal of Physics: Conference Series 664, 072015, 2015.

AKCHURIN, N.; WIGMANS, R. “Hadron Calorimetry”, Nuclear Instruments and Methods in Physics Research A 666, 80, 2012.

ALVES JR, A. A. et al. (LHCb collab.) The LHCb detector at the LHC, J. Instrum. 3, S08005, 2008.

AMATO, S. et al. (LHCb collab.) LHCb Calorimeters Technical Design Report (CERN LHCC 2000-0036), 2003

ANDRONIC, A.; WESSELS, J. P., Transition radiation detectors, Nucl. Instr.& Meth. A 666, 130, 2012.

AUGUSTO, A.; ALVES JR. et al., The LHCb Detector at the LHC, JINST 3, S08005, 2008.

AURISANO, A. et al. A Convolutional Neural Network Neutrino Event Classifier. Journal of Instrumentation 11, P09001, 2016.

BALDI, P.; SADOWSKI, P.; WHITESON, D., Searching for Exotic Particles in High-Energy Physics with Deep Learning. Nature Communications 5, 4308, 2014.

BROWN, R. M.; COCKERILL, D. J. A., Electromagnetic Calorimetry, Nucl. Instr. And Meth in Phys. Res. A 666, 47, 2012.

CALVO, M.; COGNERAS, E. E.; DESHAMPS, O.; HOBALLAH M., A tool for  $\gamma/\pi^0$  separation at high energies, LHCb-PUB-2015-016.

CARMINATI, F. et al. Calorimetry with Deep Learning: Particle Classification, Energy Regression, and Simulation for High-Energy Physics. p. 6, 2017.

CHAUHAN, NAGESH SINGH. Decision Tree Algorithm, Explained. Hyderabad, Andhra Pradesh, Índia, 24 de dez. de 2019. Disponível em: <<https://towardsdatascience.com/decision-tree-algorithm-explained-83beb6e78ef4>>. Acesso em: 15 de jan 2020.

CHEKALINA, V.; RATNIKOV, F, Machine Learning approach to  $\gamma/\pi^0$  separation in the LHCb calorimeter, Phys. Conf. Ser. 1085, 042036, 2018.

COHEN, T.; FREYTSIS, M.; OSTDIEK, B., (Machine) learning to do more with less. *Journal of High Energy Physics* 2018, 34, 2018.

DZHELYADIN, R., The LHCb calorimeter detectors, *Nucl. Instr. Meth. Phys. Res. A.* 581, 55, 2007.

EVANS, L, et al. (Editors), LHC machine, *JINST* **3**, S08001, 2008.

FLYNN, J 2015 Computing Resources Scrutiny Group Report Tech. Rep. CERN-RRB-2015-014 CERN Geneva, 2015. Disponível em <https://cds.cern.ch/record/2002240>

HALZEN, F.; MARTIN, A. D.; “Quarks & Leptons: Introductory Course in Modern Particle Physics”, John Wiley & Sons, Inc.; Canada, 1984.

HERTEL, L. et al. Convolutional Neural Networks for Electron Neutrino and Electron Shower Energy Reconstruction in the NOvA Detectors, *5*, 2017.

HODDESON, L.; BROWN, L. M.; RIORDAN, M.; DRESDEN, M., *The Rise of the Standard Model: A History of Particle Physics from 1964 to 1979*, CAMBRIDGE UNIVERSITY PRESS, New York, 1997.

LIPPMANN, C.; Particle identification, *Nucl. Instrum. Meth. A*, 666, 148, 2012.

LOPES, J. L., *A Estrutura Quântica da Matéria: Do átomo pré-socrático às partículas elementares*. Ed. UFRJ, 2ª ed, p. 820, 1993.

KARAVAKIS, E. et al., *Journal of Physics: Conference Series* 513, 062024, 2014.

KUO, M. H. et al. Health big data analytics: current perspectives, challenges and potential solutions. *International Journal of Big Data Intelligence* 1, 114, 2014.

MIRONOV, C., Studying nuclear matter created in p+p, d+Au and Au+Au collisions using charged kaons, *J. Phys.: Conf. Ser.* 50, 31, 2006.

MISSMJ, CUSH. Standard model of elementary particles: the 12 fundamental fermions and 5 fundamental bosons. Internet, 17 set. De 2019. Disponível em [https://commons.wikimedia.org/wiki/File:Standard\\_Model\\_of\\_Elementary\\_Particles.svg](https://commons.wikimedia.org/wiki/File:Standard_Model_of_Elementary_Particles.svg). Acesso em 14 de nov. 2019.

MODEL GYM, Project 2017 Model Gym [software] Available from <https://github.com/yandexdataschool/modelgym> [accessed 2019-05-14]; XGBoost: A Scalable Tree Boosting System. arXiv:1603.02754 [cs.LG], March 2016.

NAVARRO, A. P.; “First measurements of radiative B decay in LHCb”, report number CERN-THESIS-2012-025, Universidade de Barcelona, 2012.

NATARAJAN, Balas K. *Machine learning: a theoretical approach*. Elsevier, 2014.

ROBERTS, A. A new type of Cherenkov detector for the accurate measurement of the particle velocity and direction, *Nucl. Instr. Meth.* 9, 55, 1960.

STRONG, G.C. Strong, On the impact of modern deep-learning techniques to the performance and time-requirements of classification models in experimental high-energy physics, 2002. arXiv:2002.01427.

TANABASHI, M *et al.* (Particle Data Group), *Phys. Rev. D* 98, 030001 (2018) and 2019 update. Acesso em <http://pdg.lbl.gov> 23 de março de 2020.

TrackML Particle Tracking Challenge. Acesso de <https://www.kaggle.com/c/trackml-particle-identification/data> em 23 de março 2020.

TAVARES, O., *Ciência e Sociedade*, CBPF, 3, p. 1-42, 2018.

VALKOV, V., Creating a Neural Network from Scratch — TensorFlow for Hackers (Part IV). Internet, 19 maio de 2017. Disponível em <<https://medium.com/@curiously/tensorflow-for-hackers-part-iv-neural-network-from-scratch-1a4f504dfa8>>. Acesso em 21 de dez. 2019.

ZHANG, L., Muon Reconstruction Performance of the ATLAS Detector at the LHC at  $\sqrt{s} = 13 \text{ TeV}$ , 2020. doi /10.1142/9789811207402\_0045



## APÊNDICE A

Este apêndice possui o link para visualização do jupyter notebook com código em python e sua compilação. Porém, é necessário reforçar que existe somente um bloco de código para cada método de classificação e foi descrito durante a metodologia que foram utilizado 5 diferentes seleções de *features*. Portanto, para se utilizar cada base de dados foram instanciadas respectivos dataframes que deram entrada nos modelos de forma separada e em momentos distintos. Sendo assim, o que se segue nesta amostra de Código e compilação somente de uma única seleção das *features*, ou seja de uma única base de dados.

Também parte deste apêndice está a matriz de correlação completa.

### **CÓDIGO COM TODOS OS MÉTODOS, MODELOS E TÉCNICAS.**

[https://github.com/jpseixasesilva/LHC-ML/blob/master/Particles\\_Identification.ipynb](https://github.com/jpseixasesilva/LHC-ML/blob/master/Particles_Identification.ipynb)



## ANEXO A

### TÉCNICA DE SELEÇÃO DE FEATURES:

-NO SELECTION

### MODELO DE CLASSIFICAÇÃO:

-RANDOM FOREST

### TEMPO DE EXECUÇÃO:

-48 MINUTOS

-----TRAINING DATA-----

Matriz de Confusão training\_data :

```
[[142637 10795 1211 1156 2787 1232]
 [ 10415 131340 2920 2447 9199 3669]
 [ 1963 6471 117901 2658 14763 16374]
 [ 967 5046 1368 146093 5659 1134]
 [ 3166 10561 5872 5342 129738 4980]
 [ 1935 7361 20869 1864 14122 113985]]
```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.89	0.89	0.89	159818
1	0.77	0.82	0.79	159990
2	0.79	0.74	0.76	160130
3	0.92	0.91	0.91	160267
4	0.74	0.81	0.77	159659
5	0.81	0.71	0.76	160136
accuracy		0.81	960000	

```

macro avg    0.82    0.81    0.81    960000
weighted avg 0.82    0.81    0.81    960000

```

-----TESTING DATA-----

Matriz de Confusão testing\_data :

```

[35067 3324 330 346 789 326]
[ 3107 31457 855 748 2747 1096]
[ 512 1754 23945 806 4044 8809]
[ 270 1421 423 35644 1626 349]
[ 890 3013 1833 1559 31378 1668]
[ 495 1958 10460 513 3939 22499]

```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.87	0.87	0.87	40182
1	0.73	0.79	0.76	40010
2	0.63	0.60	0.62	39870
3	0.90	0.90	0.90	39733
4	0.70	0.78	0.74	40341
5	0.65	0.56	0.60	39864

```

accuracy                0.75  240000
macro avg    0.75    0.75    0.75  240000
weighted avg 0.75    0.75    0.75  240000

```

**TÉCNICA DE SELEÇÃO DE FEATURES:**

-NO SELECTION

**MODELO DE CLASSIFICAÇÃO:**

-GRADIENT BOOST CLASSIFIER

**TEMPO DE EXECUÇÃO:**

-133 MINUTOS

-----TRAINING DATA-----

Matriz de Confusão training\_data :

```

[[143538 10322 1266 1053 2337 1302]
 [ 10592 130149 3352 2297 9219 4381]
 [ 1667 6259 104413 2420 14618 30753]
 [ 815 3809 1416 147841 5347 1039]
 [ 2854 10169 7499 5124 127850 6163]
 [ 1654 7499 35464 1588 13546 100385]]

```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.89	0.90	0.89	159818
1	0.77	0.81	0.79	159990
2	0.68	0.65	0.67	160130
3	0.92	0.92	0.92	160267
4	0.74	0.80	0.77	159659
5	0.70	0.63	0.66	160136
accuracy			0.79	960000
macro avg	0.78	0.79	0.78	960000
weighted avg	0.78	0.79	0.78	960000

-----TESTING DATA-----

Matriz de Confusão testing\_data :

```
[[35792 2879 314 297 583 317]
 [ 2994 31901 815 691 2453 1156]
 [ 420 1603 25048 681 3678 8440]
 [ 219 1023 409 36383 1423 276]
 [ 753 2662 2039 1364 31904 1619]
 [ 387 1857 9428 407 3426 24359]]
```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.88	0.89	0.89	40182
1	0.76	0.80	0.78	40010
2	0.66	0.63	0.64	39870
3	0.91	0.92	0.91	39733
4	0.73	0.79	0.76	40341
5	0.67	0.61	0.64	39864
accuracy			0.77	240000
macro avg	0.77	0.77	0.77	240000
weighted avg	0.77	0.77	0.77	240000

**TÉCNICA DE SELEÇÃO DE FEATURES:**

-NO SELECTION

**MODELO DE CLASSIFICAÇÃO:**

-XGBOOST

**TEMPO DE EXECUÇÃO:**

-88 MINUTOS

-----TRAINING DATA-----

Matriz de Confusão training\_data :

```

[[141404 11918 1337 1138 2687 1334]
 [ 11577 128929 3209 2321 9699 4255]
 [ 1841 6972 100125 2680 16758 31754]
 [ 840 5155 1451 145920 5896 1005]
 [ 3210 11739 7089 5322 126346 5953]
 [ 1856 8237 38996 1818 15912 93317]]

```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.88	0.88	0.88	159818
1	0.75	0.81	0.77	159990
2	0.66	0.63	0.64	160130
3	0.92	0.91	0.91	160267
4	0.71	0.79	0.75	159659
5	0.68	0.58	0.63	160136
accuracy			0.77	960000
macro avg	0.77	0.77	0.76	960000
weighted avg	0.77	0.77	0.76	960000

-----TESTING DATA-----

Matriz de Confusão testing\_data :

```
[[35561 3030 306 297 660 328]
 [ 3015 32002 793 611 2493 1096]
 [ 458 1763 24590 740 4155 8164]
 [ 230 1334 396 35984 1522 267]
 [ 807 2891 1858 1353 31884 1548]
 [ 425 2003 9792 449 3901 23294]]
```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.88	0.88	0.88	40182
1	0.74	0.80	0.77	40010
2	0.65	0.62	0.63	39870
3	0.91	0.91	0.91	39733
4	0.71	0.79	0.75	40341
5	0.67	0.58	0.62	39864
accuracy		0.76		240000
macro avg	0.76	0.76	0.76	240000
weighted avg	0.76	0.76	0.76	240000



**TÉCNICA DE SELEÇÃO DE FEATURES:**

-NO SELECTION

**MODELO DE CLASSIFICAÇÃO:**

-REDE NEURAL

**TEMPO DE EXECUÇÃO:**

-62 MINUTOS

-----TRAINING DATA-----

Matriz de Confusão training\_data :

```

[[143623 9342 1549 1217 2882 1205]
 [ 14051 124383 3940 3277 10113 4226]
 [ 1706 5183 108802 3010 16682 24747]
 [ 912 3923 1638 146989 5773 1032]
 [ 3322 9170 9150 5897 127302 4818]
 [ 1771 6547 51542 1901 15078 83297]]

```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.87	0.90	0.88	159818
1	0.78	0.78	0.78	159990
2	0.62	0.68	0.65	160130
3	0.91	0.92	0.91	160267
4	0.72	0.80	0.75	159659
5	0.70	0.52	0.60	160136
accuracy			0.76	960000
macro avg	0.76	0.77	0.76	960000
weighted avg	0.76	0.76	0.76	960000

-----TESTING DATA-----

Matriz de Confusão testing\_data :

```
[[36077 2408 377 346 672 302]
 [ 3567 30781 1030 866 2655 1111]
 [ 441 1272 26992 785 4151 6229]
 [ 246 983 440 36355 1450 259]
 [ 807 2328 2333 1493 32103 1277]
 [ 402 1575 12929 477 3750 20731]]
```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.87	0.90	0.88	40182
1	0.78	0.77	0.78	40010
2	0.61	0.68	0.64	39870
3	0.90	0.91	0.91	39733
4	0.72	0.80	0.75	40341
5	0.69	0.52	0.59	39864
accuracy		0.76		240000
macro avg	0.76	0.76	0.76	240000
weighted avg	0.76	0.76	0.76	240000

**TÉCNICA DE SELEÇÃO DE FEATURES:****-SELECTKBEST****MODELO DE CLASSIFICAÇÃO:****-RANDOM FOREST****TEMPO DE EXECUÇÃO:****-79 MINUTOS**

-----TRAINING DATA-----

Matriz de Confusão training\_data :

```

[[139082 12110 1673 1282 3993 1678]
 [ 10703 130464 3013 2556 9426 3828]
 [ 2862 6642 115760 2780 14595 17491]
 [ 1148 5206 1398 145384 5914 1217]
 [ 4734 10823 5729 5328 127901 5144]
 [ 2761 7503 22364 1976 14325 111207]]

```

Reports (Precision - Recall - F1 Score)

```

precision recall f1-score support

0    0.86    0.87    0.87   159818
1    0.76    0.82    0.78   159990
2    0.77    0.72    0.75   160130
3    0.91    0.91    0.91   160267
4    0.73    0.80    0.76   159659
5    0.79    0.69    0.74   160136

accuracy                0.80  960000
macro avg              0.80  0.80  0.80  960000
weighted avg           0.80  0.80  0.80  960000

```

-----TESTING DATA-----

Matriz de Confusão testing\_data :

```
[[34205 3491 466 383 1149 488]
 [ 3145 31380 837 793 2749 1106]
 [ 769 1785 23677 823 3948 8868]
 [ 320 1470 450 35410 1722 361]
 [ 1397 2979 1762 1512 31035 1656]
 [ 751 1978 10683 529 3874 22049]]
```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.84	0.85	0.85	40182
1	0.73	0.78	0.76	40010
2	0.63	0.59	0.61	39870
3	0.90	0.89	0.89	39733
4	0.70	0.77	0.73	40341
5	0.64	0.55	0.59	39864
accuracy		0.74		240000
macro avg	0.74	0.74	0.74	240000
weighted avg	0.74	0.74	0.74	240000

**TÉCNICA DE SELEÇÃO DE FEATURES:****-SELECTKBEST****MODELO DE CLASSIFICAÇÃO:****-GRADIENT BOOST CLASSIFIER****TEMPO DE EXECUÇÃO:****-78 MINUTOS**

-----TRAINING DATA-----

Matriz de Confusão training\_data :

```

[[139239 11984 1722 1223 3877 1773]
 [ 11137 129240 3383 2512 9497 4221]
 [ 2639 6791 102746 2489 14624 30841]
 [ 893 4267 1399 147061 5432 1215]
 [ 4364 10821 7122 5283 125724 6345]
 [ 2421 7971 38406 1827 14263 95248]]

```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.87	0.87	0.87	159818
1	0.76	0.81	0.78	159990
2	0.66	0.64	0.65	160130
3	0.92	0.92	0.92	160267
4	0.72	0.79	0.75	159659
5	0.68	0.59	0.64	160136
accuracy			0.77	960000
macro avg	0.77	0.77	0.77	960000
weighted avg	0.77	0.77	0.77	960000

-----TESTING DATA-----

Matriz de Confusão testing\_data :

```
[[34754 3207 412 350 1020 439]
 [ 2967 31772 833 711 2578 1149]
 [ 650 1712 24536 675 3637 8660]
 [ 275 1118 396 36168 1460 316]
 [ 1145 2794 1896 1393 31459 1654]
 [ 646 1966 10258 451 3544 22999]]
```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.86	0.86	0.86	40182
1	0.75	0.79	0.77	40010
2	0.64	0.62	0.63	39870
3	0.91	0.91	0.91	39733
4	0.72	0.78	0.75	40341
5	0.65	0.58	0.61	39864
accuracy			0.76	240000
macro avg	0.75	0.76	0.76	240000
weighted avg	0.75	0.76	0.76	240000

**TÉCNICA DE SELEÇÃO DE FEATURES:****-SELECTKBEST****MODELO DE CLASSIFICAÇÃO:****-XGBOOST****TEMPO DE EXECUÇÃO:****-64 MINUTOS**

-----TRAINING DATA-----

Matriz de Confusão training\_data :

```

[[137543 13075 1853 1302 4349 1696]
 [ 11708 128432 3334 2464 9850 4202]
 [ 2816 7368 99823 2749 16071 31303]
 [ 924 5406 1453 145388 6044 1052]
 [ 4582 12183 6913 5448 124884 5649]
 [ 2647 8494 42433 1995 16202 88365]]

```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.86	0.86	0.86	159818
1	0.73	0.80	0.77	159990
2	0.64	0.62	0.63	160130
3	0.91	0.91	0.91	160267
4	0.70	0.78	0.74	159659
5	0.67	0.55	0.60	160136
accuracy			0.75	960000
macro avg	0.75	0.75	0.75	960000
weighted avg	0.75	0.75	0.75	960000

-----TESTING DATA-----

Matriz de Confusão testing\_data :

```
[[34603 3305 435 363 1069 407]
 [ 2979 31907 839 673 2519 1093]
 [ 687 1851 24553 745 3978 8056]
 [ 258 1385 391 35884 1540 275]
 [ 1187 3006 1815 1401 31496 1436]
 [ 688 2056 10682 465 4010 21963]]
```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.86	0.86	0.86	40182
1	0.73	0.80	0.76	40010
2	0.63	0.62	0.62	39870
3	0.91	0.90	0.91	39733
4	0.71	0.78	0.74	40341
5	0.66	0.55	0.60	39864
accuracy		0.75		240000
macro avg	0.75	0.75	0.75	240000
weighted avg	0.75	0.75	0.75	240000



**TÉCNICA DE SELEÇÃO DE FEATURES:****-SELECTKBEST****MODELO DE CLASSIFICAÇÃO:****-REDE NEURAL****TEMPO DE EXECUÇÃO:****-83 MINUTOS**

-----TRAINING DATA-----

Matriz de Confusão training\_data :

```

[[139319 11321 1685 1462 4408 1623]
 [ 13784 125659 2719 3252 10095 4481]
 [ 2879 6187 90793 2958 17319 39994]
 [ 1008 5040 1584 145243 6110 1282]
 [ 5316 10607 6058 6100 125107 6471]
 [ 2908 7607 35128 2096 15950 96447]]

```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.84	0.87	0.86	159818
1	0.76	0.79	0.77	159990
2	0.66	0.57	0.61	160130
3	0.90	0.91	0.90	160267
4	0.70	0.78	0.74	159659
5	0.64	0.60	0.62	160136
accuracy			0.75	960000
macro avg	0.75	0.75	0.75	960000
weighted avg	0.75	0.75	0.75	960000

-----TESTING DATA-----

Matriz de Confusão testing\_data :

```
[[35042 2905 394 406 1040 395]
 [ 3414 31291 694 832 2629 1150]
 [ 706 1537 22394 811 4279 10143]
 [ 277 1283 418 35842 1565 348]
 [ 1339 2665 1553 1551 31528 1705]
 [ 717 1836 8843 513 3929 24026]]
```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.84	0.87	0.86	40182
1	0.75	0.78	0.77	40010
2	0.65	0.56	0.60	39870
3	0.90	0.90	0.90	39733
4	0.70	0.78	0.74	40341
5	0.64	0.60	0.62	39864
accuracy			0.75	240000
macro avg	0.75	0.75	0.75	240000
weighted avg	0.75	0.75	0.75	240000

**TÉCNICA DE SELEÇÃO DE FEATURES:**

-EXTRA TREE

**MODELO DE CLASSIFICAÇÃO:**

-RANDOM SEARCH

**TEMPO DE EXECUÇÃO:**

-55 MINUTOS

-----TRAINING DATA-----

Matriz de Confusão training\_data :

```

[[139046 12164 1637 1308 3948 1715]
 [ 10618 130580 3001 2528 9363 3900]
 [ 2843 6635 115785 2760 14466 17641]
 [ 1147 5173 1353 145430 5872 1292]
 [ 4786 10852 5755 5320 127750 5196]
 [ 2739 7490 22224 1959 14200 111524]]

```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.86	0.87	0.87	159818
1	0.76	0.82	0.78	159990
2	0.77	0.72	0.75	160130
3	0.91	0.91	0.91	160267
4	0.73	0.80	0.76	159659
5	0.79	0.70	0.74	160136
accuracy			0.80	960000
macro avg	0.80	0.80	0.80	960000
weighted avg	0.80	0.80	0.80	960000

-----TESTING DATA-----

Matriz de Confusão testing\_data :

```
[[34188 3543 473 370 1145 463]
 [ 3089 31448 831 759 2742 1141]
 [ 775 1797 23630 809 3930 8929]
 [ 320 1464 435 35431 1699 384]
 [ 1411 2969 1785 1504 30988 1684]
 [ 752 1967 10648 525 3872 22100]]
```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.84	0.85	0.85	40182
1	0.73	0.79	0.76	40010
2	0.63	0.59	0.61	39870
3	0.90	0.89	0.90	39733
4	0.70	0.77	0.73	40341
5	0.64	0.55	0.59	39864
accuracy		0.74		240000
macro avg	0.74	0.74	0.74	240000
weighted avg	0.74	0.74	0.74	240000

**TÉCNICA DE SELEÇÃO DE FEATURES:**

-EXTRA TREE

**MODELO DE CLASSIFICAÇÃO:**

-GRADIENT BOOST CLASSIFIER

**TEMPO DE EXECUÇÃO:**

-81 MINUTOS

-----TRAINING DATA-----

Matriz de Confusão training\_data :

```

[[139239 11984 1722 1223 3877 1773]
 [ 11137 129240 3383 2512 9497 4221]
 [ 2639 6791 102746 2489 14624 30841]
 [ 893 4267 1399 147061 5432 1215]
 [ 4364 10821 7122 5283 125724 6345]
 [ 2421 7971 38406 1827 14263 95248]]

```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.87	0.87	0.87	159818
1	0.76	0.81	0.78	159990
2	0.66	0.64	0.65	160130
3	0.92	0.92	0.92	160267
4	0.72	0.79	0.75	159659
5	0.68	0.59	0.64	160136
accuracy			0.77	960000
macro avg	0.77	0.77	0.77	960000
weighted avg	0.77	0.77	0.77	960000

-----TESTING DATA-----

Matriz de Confusão testing\_data :

```
[[34754 3207 412 350 1020 439]
 [ 2967 31772 833 711 2578 1149]
 [ 650 1712 24536 675 3637 8660]
 [ 275 1118 396 36168 1460 316]
 [ 1145 2794 1896 1393 31459 1654]
 [ 646 1966 10258 451 3544 22999]]
```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.86	0.86	0.86	40182
1	0.75	0.79	0.77	40010
2	0.64	0.62	0.63	39870
3	0.91	0.91	0.91	39733
4	0.72	0.78	0.75	40341
5	0.65	0.58	0.61	39864
accuracy		0.76		240000
macro avg	0.75	0.76	0.76	240000
weighted avg	0.75	0.76	0.76	240000

**TÉCNICA DE SELEÇÃO DE FEATURES:**

-EXTRA TREE

**MODELO DE CLASSIFICAÇÃO:**

-XGBOOST

**TEMPO DE EXECUÇÃO:**

-80 MINUTOS

-----TRAINING DATA-----

Matriz de Confusão training\_data :

```

[[137543 13075 1853 1302 4349 1696]
 [ 11708 128432 3334 2464 9850 4202]
 [ 2816 7368 99823 2749 16071 31303]
 [ 924 5406 1453 145388 6044 1052]
 [ 4582 12183 6913 5448 124884 5649]
 [ 2647 8494 42433 1995 16202 88365]]

```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.86	0.86	0.86	159818
1	0.73	0.80	0.77	159990
2	0.64	0.62	0.63	160130
3	0.91	0.91	0.91	160267
4	0.70	0.78	0.74	159659
5	0.67	0.55	0.60	160136
accuracy			0.75	960000
macro avg	0.75	0.75	0.75	960000
weighted avg	0.75	0.75	0.75	960000

-----TESTING DATA-----

Matriz de Confusão testing\_data :

```
[[34603 3305 435 363 1069 407]
 [ 2979 31907 839 673 2519 1093]
 [ 687 1851 24553 745 3978 8056]
 [ 258 1385 391 35884 1540 275]
 [ 1187 3006 1815 1401 31496 1436]
 [ 688 2056 10682 465 4010 21963]]
```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.86	0.86	0.86	40182
1	0.73	0.80	0.76	40010
2	0.63	0.62	0.62	39870
3	0.91	0.90	0.91	39733
4	0.71	0.78	0.74	40341
5	0.66	0.55	0.60	39864
accuracy			0.75	240000
macro avg	0.75	0.75	0.75	240000
weighted avg	0.75	0.75	0.75	240000



**TÉCNICA DE SELEÇÃO DE FEATURES:**

-EXTRA TREE

**MODELO DE CLASSIFICAÇÃO:**

-REDE NEURAL

**TEMPO DE EXECUÇÃO:**

-66 MINUTOS

-----TRAINING DATA-----

Matriz de Confusão training\_data :

```

[[141548 10639 1308 1504 2843 1976]
 [ 14097 124098 3100 3286 10587 4822]
 [ 1518 6101 95605 3161 15923 37822]
 [ 1035 5140 1449 144304 5537 2802]
 [ 3378 10791 7750 6816 122660 8264]
 [ 1553 7097 40550 2353 15139 93444]]

```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.87	0.89	0.88	159818
1	0.76	0.78	0.77	159990
2	0.64	0.60	0.62	160130
3	0.89	0.90	0.90	160267
4	0.71	0.77	0.74	159659
5	0.63	0.58	0.60	160136
accuracy			0.75	960000
macro avg	0.75	0.75	0.75	960000
weighted avg	0.75	0.75	0.75	960000

-----TESTING DATA-----

Matriz de Confusão testing\_data :

```
[[35631 2703 334 358 683 473]
 [ 3520 30806 788 824 2785 1287]
 [ 362 1518 23627 851 3945 9567]
 [ 286 1291 364 35631 1412 749]
 [ 819 2690 2050 1697 30897 2188]
 [ 357 1728 10176 599 3702 23302]]
```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.87	0.89	0.88	40182
1	0.76	0.77	0.76	40010
2	0.63	0.59	0.61	39870
3	0.89	0.90	0.89	39733
4	0.71	0.77	0.74	40341
5	0.62	0.58	0.60	39864
accuracy			0.75	240000
macro avg	0.75	0.75	0.75	240000
weighted avg	0.75	0.75	0.75	240000

**TÉCNICA DE SELEÇÃO DE FEATURES:**

-CORRELATION MATRIX

**MODELO DE CLASSIFICAÇÃO:**

-RANDOM SEARCH

**TEMPO DE EXECUÇÃO:**

-68 MINUTOS

-----TRAINING DATA-----

Matriz de Confusão training\_data :

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.79	0.82	0.80	159818
1	0.75	0.82	0.78	159990
2	0.71	0.65	0.68	160130
3	0.90	0.90	0.90	160267
4	0.71	0.76	0.74	159659
5	0.74	0.65	0.69	160136
accuracy		0.77	0.77	960000
macro avg	0.77	0.77	0.76	960000
weighted avg	0.77	0.77	0.76	960000

-----TESTING DATA-----

Matriz de Confusão testing\_data :

```
[[31522 4015 838 696 2113 998]
 [ 3233 31570 749 786 2565 1107]
 [ 1907 1898 18010 965 4429 12661]
 [ 726 1508 459 35211 1436 393]]
```

[ 2626 2949 2746 1527 29223 1270]  
[ 1868 2046 13988 648 3560 17754]]

#### Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.75	0.78	0.77	40182
1	0.72	0.79	0.75	40010
2	0.49	0.45	0.47	39870
3	0.88	0.89	0.89	39733
4	0.67	0.72	0.70	40341
5	0.52	0.45	0.48	39864
accuracy			0.68	240000
macro avg	0.67	0.68	0.68	240000
weighted avg	0.67	0.68	0.68	240000

**TÉCNICA DE SELEÇÃO DE FEATURES:**

-CORRELATION MATRIX

**MODELO DE CLASSIFICAÇÃO:**

-XGBOOST

**TEMPO DE EXECUÇÃO:**

-71 MINUTOS

-----TRAINING DATA-----

Matriz de Confusão training\_data :

```

[[129718 13531 3447 2117 7429 3576]
 [ 11845 129406 3105 2492 9047 4095]
 [ 6140 7092 82777 2925 16759 44437]
 [ 2497 4372 1649 145734 4848 1167]
 [ 8646 10806 9906 5150 119616 5535]
 [ 5887 8020 50670 2156 13771 79632]]

```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.79	0.81	0.80	159818
1	0.75	0.81	0.78	159990
2	0.55	0.52	0.53	160130
3	0.91	0.91	0.91	160267
4	0.70	0.75	0.72	159659
5	0.58	0.50	0.53	160136
accuracy			0.72	960000
macro avg	0.71	0.72	0.71	960000
weighted avg	0.71	0.72	0.71	960000

-----TESTING DATA-----

Matriz de Confusão testing\_data :

```
[[32297 3658 839 599 1837 952]
 [ 3173 31790 779 709 2452 1107]
 [ 1580 1795 19346 805 4218 12126]
 [ 672 1160 469 35808 1290 334]
 [ 2253 2773 2617 1352 29922 1424]
 [ 1540 2000 13670 532 3344 18778]]
```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.78	0.80	0.79	40182
1	0.74	0.79	0.76	40010
2	0.51	0.49	0.50	39870
3	0.90	0.90	0.90	39733
4	0.69	0.74	0.72	40341
5	0.54	0.47	0.50	39864
accuracy			0.70	240000
macro avg	0.69	0.70	0.70	240000
weighted avg	0.69	0.70	0.70	240000

**TÉCNICA DE SELEÇÃO DE FEATURES:**

-CORRELATION MATRIX

**MODELO DE CLASSIFICAÇÃO:**

-GRADIENT BOOST CLASSIFIER

**TEMPO DE EXECUÇÃO:**

-76 MINUTOS

-----TRAINING DATA-----

Matriz de Confusão training\_data :

```

[[128160 14765 3642 2349 7552 3350]
 [ 13098 128497 2897 2306 9160 4032]
 [ 6753 7740 76825 3271 17591 47950]
 [ 2910 5496 1659 144206 4877 1119]
 [ 10147 11907 10822 5171 117189 4423]
 [ 6795 8897 52569 2408 14694 74773]]

```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.76	0.80	0.78	159818
1	0.72	0.80	0.76	159990
2	0.52	0.48	0.50	160130
3	0.90	0.90	0.90	160267
4	0.69	0.73	0.71	159659
5	0.55	0.47	0.51	160136
accuracy			0.70	960000
macro avg	0.69	0.70	0.69	960000
weighted avg	0.69	0.70	0.69	960000

-----TESTING DATA-----

Matriz de Confusão testing\_data :

```
[[32150 3803 884 633 1848 864]
 [ 3313 31854 741 626 2391 1085]
 [ 1696 1983 18873 891 4389 12038]
 [ 755 1401 450 35549 1261 317]
 [ 2609 2982 2842 1330 29487 1091]
 [ 1763 2186 13318 565 3502 18530]]
```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.76	0.80	0.78	40182
1	0.72	0.80	0.76	40010
2	0.51	0.47	0.49	39870
3	0.90	0.89	0.90	39733
4	0.69	0.73	0.71	40341
5	0.55	0.46	0.50	39864
accuracy		0.69		240000
macro avg	0.69	0.69	0.69	240000
weighted avg	0.69	0.69	0.69	240000



**TÉCNICA DE SELEÇÃO DE FEATURES:**

-CORRELATION MATRIX

**MODELO DE CLASSIFICAÇÃO:**

-GRADIENT BOOST CLASSIFIER

**TEMPO DE EXECUÇÃO:**

-49 MINUTOS

-----TRAINING DATA-----

Matriz de Confusão training\_data :

```

[[129766 12290 3678 2327 8457 3300]
 [ 14743 125923 2725 3001 9445 4153]
 [ 7738 6231 77778 3487 18114 46782]
 [ 3017 4569 1745 145206 4619 1111]
 [ 10732 9896 10312 5790 118713 4216]
 [ 7376 7424 54796 2421 14881 73238]]

```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.75	0.81	0.78	159818
1	0.76	0.79	0.77	159990
2	0.51	0.49	0.50	160130
3	0.90	0.91	0.90	160267
4	0.68	0.74	0.71	159659
5	0.55	0.46	0.50	160136
accuracy			0.70	960000
macro avg	0.69	0.70	0.69	960000
weighted avg	0.69	0.70	0.69	960000

-----TESTING DATA-----

Matriz de Confusão testing\_data :

```
[[32509 3170 935 613 2088 867]
 [ 3713 31205 730 823 2469 1070]
 [ 1924 1579 18976 941 4618 11832]
 [ 791 1205 492 35806 1147 292]
 [ 2693 2490 2674 1489 29914 1081]
 [ 1927 1801 13856 599 3618 18063]]
```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.75	0.81	0.78	40182
1	0.75	0.78	0.77	40010
2	0.50	0.48	0.49	39870
3	0.89	0.90	0.90	39733
4	0.68	0.74	0.71	40341
5	0.54	0.45	0.49	39864
accuracy			0.69	240000
macro avg	0.69	0.69	0.69	240000
weighted avg	0.69	0.69	0.69	240000

**TÉCNICA DE SELEÇÃO DE FEATURES:**

-LIGHTGBM

**MODELO DE CLASSIFICAÇÃO:**

-RANDOM FOREST

**TEMPO DE EXECUÇÃO:**

-62 MINUTOS

-----TRAINING DATA-----

Matriz de Confusão training\_data :

[Parallel(n\_jobs=1)]: Done 34 out of 34 | elapsed: 8.4s finished

```

[[141850 11114 1163 1499 2903 1289]
 [ 10601 130367 3003 2804 9609 3606]
 [ 2017 6700 116236 2972 15647 16558]
 [ 1066 8104 2225 138075 8753 2044]
 [ 3287 10843 5727 5917 128801 5084]
 [ 2069 7417 21895 2779 15227 110749]]

```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.88	0.89	0.88	159818
1	0.75	0.81	0.78	159990
2	0.77	0.73	0.75	160130
3	0.90	0.86	0.88	160267
4	0.71	0.81	0.76	159659
5	0.79	0.69	0.74	160136

accuracy			0.80	960000
macro avg	0.80	0.80	0.80	960000

weighted avg 0.80 0.80 0.80 960000

-----TESTING DATA-----

Matriz de Confusão testing\_data :

```
[[34968 3346 337 395 807 329]
 [ 3166 31235 871 918 2774 1046]
 [ 518 1818 23643 834 4285 8772]
 [ 308 2246 657 33459 2469 594]
 [ 914 3015 1827 1718 31202 1665]
 [ 531 1962 10665 746 4176 21784]]
```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.87	0.87	0.87	40182
1	0.72	0.78	0.75	40010
2	0.62	0.59	0.61	39870
3	0.88	0.84	0.86	39733
4	0.68	0.77	0.73	40341
5	0.64	0.55	0.59	39864
accuracy			0.73	240000
macro avg	0.73	0.73	0.73	240000
weighted avg	0.73	0.73	0.73	240000

**TÉCNICA DE SELEÇÃO DE FEATURES:**

-LIGHTGBM

**MODELO DE CLASSIFICAÇÃO:**

-GRADIENT BOOST CLASSIFIER

**TEMPO DE EXECUÇÃO:**

-79 MINUTOS

-----TRAINING DATA-----

Matriz de Confusão training\_data :

```

[[141577 11401 1221 1485 2757 1377]
 [ 11214 127585 3509 3475 10167 4040]
 [ 1737 6645 100608 3205 15650 32285]
 [ 982 6802 2160 139634 8472 2217]
 [ 3083 11076 7187 6542 125335 6436]
 [ 1726 7652 36485 2841 15115 96317]]

```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.88	0.89	0.88	159818
1	0.75	0.80	0.77	159990
2	0.67	0.63	0.65	160130
3	0.89	0.87	0.88	160267
4	0.71	0.79	0.74	159659
5	0.68	0.60	0.64	160136
accuracy			0.76	960000
macro avg	0.76	0.76	0.76	960000
weighted avg	0.76	0.76	0.76	960000

-----TESTING DATA-----

Matriz de Confusão testing\_data :

```
[[35273 3198 308 379 707 317]
 [ 3129 31225 893 954 2718 1091]
 [ 430 1689 23960 859 3933 8999]
 [ 257 1844 539 34272 2213 608]
 [ 780 2868 2011 1734 31234 1714]
 [ 407 1907 9835 708 3762 23245]]
```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.88	0.88	0.88	40182
1	0.73	0.78	0.75	40010
2	0.64	0.60	0.62	39870
3	0.88	0.86	0.87	39733
4	0.70	0.77	0.74	40341
5	0.65	0.58	0.61	39864
accuracy			0.75	240000
macro avg	0.75	0.75	0.75	240000
weighted avg	0.75	0.75	0.75	240000

**TÉCNICA DE SELEÇÃO DE FEATURES:****-LIGHTGBM****MODELO DE CLASSIFICAÇÃO:****-XGBOOST****TEMPO DE EXECUÇÃO:****-73 MINUTOS**

-----TRAINING DATA-----

Matriz de Confusão training\_data :

```

[[140170 12664 1293 1515 2848 1328]
 [ 12054 126639 3546 3345 10351 4055]
 [ 1934 7199 98473 3390 16871 32263]
 [ 1070 8152 2244 137005 9678 2118]
 [ 3537 12182 7094 6502 124296 6048]
 [ 1951 8077 40800 2972 16732 89604]]

```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.87	0.88	0.87	159818
1	0.72	0.79	0.76	159990
2	0.64	0.61	0.63	160130
3	0.89	0.85	0.87	160267
4	0.69	0.78	0.73	159659
5	0.66	0.56	0.61	160136
accuracy			0.75	960000
macro avg	0.75	0.75	0.74	960000
weighted avg	0.75	0.75	0.74	960000

-----TESTING DATA-----

Matriz de Confusão testing\_data :

```
[[35169 3291 304 358 732 328]
 [ 3136 31400 883 853 2686 1052]
 [ 485 1825 24185 884 4168 8323]
 [ 278 2091 586 33790 2423 565]
 [ 864 3022 1904 1663 31347 1541]
 [ 437 1996 10213 718 4061 22439]]
```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.87	0.88	0.87	40182
1	0.72	0.78	0.75	40010
2	0.64	0.61	0.62	39870
3	0.88	0.85	0.87	39733
4	0.69	0.78	0.73	40341
5	0.66	0.56	0.61	39864
accuracy			0.74	240000
macro avg	0.74	0.74	0.74	240000
weighted avg	0.74	0.74	0.74	240000



**TÉCNICA DE SELEÇÃO DE FEATURES:****-LIGHTGBM****MODELO DE CLASSIFICAÇÃO:****-REDE NEURAL****TEMPO DE EXECUÇÃO:****-59 MINUTOS**

-----TRAINING DATA-----

Matriz de Confusão training\_data :

```

[[142574  9190  1375  1346  3945  1388]
 [ 17275 118174  4156  4021 11645  4719]
 [  1705  5976 98163  3286 16277 34723]
 [  1819  8011  2838 132912 11998  2689]
 [  3452 10458  8991  6850 122180  7728]
 [  1766  6899 45414  2994 15920 87143]]

```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.85	0.89	0.87	159818
1	0.74	0.74	0.74	159990
2	0.61	0.61	0.61	160130
3	0.88	0.83	0.85	160267
4	0.67	0.77	0.72	159659
5	0.63	0.54	0.58	160136
accuracy			0.73	960000
macro avg	0.73	0.73	0.73	960000

weighted avg 0.73 0.73 0.73 960000

-----TESTING DATA-----

Matriz de Confusão testing\_data :

```
[[35852 2367 343 300 981 339]
 [ 4323 29328 1041 1067 3080 1171]
 [ 402 1490 24299 852 4078 8749]
 [ 474 2037 720 32871 2925 706]
 [ 846 2580 2288 1667 30969 1991]
 [ 398 1673 11390 737 3874 21792]]
```

Reports (Precision - Recall - F1 Score)

	precision	recall	f1-score	support
0	0.85	0.89	0.87	40182
1	0.74	0.73	0.74	40010
2	0.61	0.61	0.61	39870
3	0.88	0.83	0.85	39733
4	0.67	0.77	0.72	40341
5	0.63	0.55	0.58	39864

accuracy		0.73		240000
macro avg	0.73	0.73	0.73	240000
weighted avg	0.73	0.73	0.73	240000