Comparative Study of Classification Techniques in BSM Higgs Boson Searches

Chiruvanuru Kumar Tapaswin

Abstract

This thesis presents a comparative study of two classification techniques using multivariate analysis: Maximum Likelihood Estimate and Boosted Decision Trees (BDT), in the context of searching for Beyond Standard Model (BSM) Higgs bosons. The study focuses on distinguishing signal events of BSM Higgs bosons from the dominant background events of pair-produced top quarks in the 1-lepton bbWW channel. Using data from the ATLAS experiment at CERN's Large Hadron Collider, this research aims to improve the sensitivity of BSM Higgs boson searches within the framework of two-Higgs-doublet models (2HDM).

Contents

1 Introduction **4**

2 Theoretical Framework **6**

2.1 The Standard Model 6

2.1.1 The Fundamental Forces 7

2.1.2 Leptons 7

2.1.3 Quarks 7

2.1.4 Gauge Bosons 8

2.1.5 The Higgs Boson 8

2.1.6 Hadrons 9

2.2 Beyond the Standard Model 9

2.2.1 Two-Higgs-Doublet Models (2HDM) 10

3 Experimental Setup **12**

3.1 The Large Hadron Collider 12

3.2 The ATLAS Experiment 13

3.3 Detector Coordinates 14

3.4 Particle matter interaction 16

3.4.1 Ionization and Excitation 16

3.4.2 Bremsstrahlung 17

3.4.3 Electromagnetic Interactions: Pair Production and Photon Interactions 18

3.4.4 Nuclear Interactions 20

3.4.5 Cherenkov Radiation 22

3.5 Application of Cherenkov Radiation in the LUCID Detector 23

3.5.1 Principle of Cherenkov Radiation in LUCID 24

4 Analysis **25**

4.1 Analysis of Signal and Background Processes 25

4.1.1 Background Process 25

4.1.2 Signal Process with Beyond Standard Model (BSM) Particles 26

4.1.3 Comparative Analysis 26

4.2 Monte Carlo Simulation Samples 27

4.3 Input variables and Correlations 28

4.3.1 Kinematic Variables 28

4.3.2 Correlation Analysis 31

4.4 Classifier Performance 33

4.4.1 BDT Analysis 33

4.4.2 LikelihoodMIX Analysis 34

4.5 Background Rejection vs Signal Efficiency 35

4.5.1 ROC curve for BDT 35

4.5.2 ROC curve for LikelihoodMIX 36

4.6 Analysis of Multivariate Discriminant Methods 37

4.6.1 Maximum Likelihood Estimation Analysis 37

4.6.2 Boosted Decision Tree Analysis 38

4.6.3 Comparative Performance 39

5 Decision Trees **40**

5.1 Introduction 40

5.2 Decision Tree Structure 40

5.3 Mathematical Foundations of Decision Trees 41

5.3.1 Entropy, Information Gain and Gini Impurity 41

5.4 Decision Rules 45

5.5 Adaptive Boosting 46

5.5.1 Initial Weighting of Instances 46

5.5.2 Calculating Weighted Error 46

5.5.3 Training the Weak Learners . . . 47
5.5.4 Assigning Learner Weight . . . 47
5.5.5 Updating Instance Weights . . . 48
5.5.6 Constructing the Strong Classifier . . . 48
5.6 Application in High-Energy Physics . . . 49

6 Maximum Likelihood Estimation **50**

6.1 Likelihood Function . . . 50
6.2 Log-Likelihood Function . . . 51
6.3 Theoretical Properties of MLE . . . 51
6.3.1 Consistency . . . 51
6.3.2 Efficiency . . . 52
6.3.3 Asymptotic Normality . . . 52
6.4 Finding the Maximum Likelihood Estimate . . . 52
6.5 Example: Gaussian Distribution . . . 53
6.6 Maximum Likelihood Estimation Using Kernel Density Estimation . . . 54
6.6.1 Kernel Density Estimation Definition . . . 54
6.7 Likelihood Function Using KDE . . . 55
6.8 Finding the Optimal Bandwidth . . . 55
6.9 Likelihood Estimation Using Splines . . . 56
6.9.1 Defining Splines . . . 56
6.9.2 Likelihood Function Using Splines . . . 57
6.10 Hybrid Approach: KDE and Splines for MLE . . . 57
6.11 Model Performance . . . 58
6.11.1 Correlation Matrix . . . 58
6.11.2 ROC Curve . . . 59

A Appendix A **60**

A.1 Pseudocode for Decision Tree in C++ . . . 60
A.2 Pseudocode for AdaBoost implementation . . . 64

Chapter 1

Introduction

Everything in the known universe can be described by the four fundamental forces - gravity, electromagnetic force, weak nuclear force and strong nuclear force. Of these, the last three make up for the theoretical framework of elementary particles called the Standard Model (SM). The Standard Model has been really successful in describing fundamental particles and its interactions. However, there are strong indications that the Standard Model is not the final theory to describe nature. The Standard Model fails to account for the some of the known mysteries that still baffle scientists. Among them is the fact that the SM fails to explain the observed matter anti-matter asymmetry. It also fails to give an explanation of the nature of dark matter, also added to this is the stability of Higgs Boson's mass. It would suffice to say that the SM is not considered a complete theory, but an approximation of a more fundamental theory.

To address these shortcomings, many new theoretical models have been introduced. Of these, the Beyond the Standard Model (BSM) theories are promising. Within the BSM theories, we are particularly interested in one of its simpler extensions, the 2 Higgs Doublet Model (2HDM). In this theory, there arise 5 physical Higgs bosons through spontaneous symmetry breaking: a pair of charged Higgs (H^-, H^+), the light Higgs (H) and the heavy (H2) scalar Higgs Bosons, and one pseudoscalar boson (A). One variant of the 2HDM achieves consistency with the 125-GeV Higgs boson through a Gildener-Weinberg scalon scheme, which not only stabilizes the Higgs boson's mass, but it is also naturally aligned. That is, of all the scalars, its couplings to gauge bosons and fermions are exactly those of the single Higgs boson of the SM. The BSM Higgs bosons in GW-2HDM are well within reach of the current LHC energy. But finding the BSM Higgs bosons requires improved sensitivity to their low masses.

This Senior Capstone Experience (SCE) is a comparative study of 2 classification techniques using

multivariate analysis, namely Maximum Likelihood Estimation (MLE) and Boosted Decision Trees (BDT), for distinguishing signal events of BSM Higgs Bosons from the dominant background events of pair produced top quarks, in the 1-lepton bbWW channel. This SCE will have a detailed analysis of the mathematical underpinnings of each method, as well as their practical applications in particle physics to the said search, which would be carried out using the data recorded by the ATLAS experiment at CERN's Large Hadron Collider in Geneva, Switzerland.

Chapter 2

Theoretical Framework

2.1 The Standard Model

The Standard Model of particle physics (SM) is one of the most successful and well-tested theories in modern science, which has been developed over the course of the 20th century.It provides a comprehensive framework for understanding the fundamental particles and forces that govern the universe - excluding gravity.

According to this theoretical framework the most fundamental particles are quarks and leptons. They interact with each other by virtue of bosons. There is the Higgs Boson, which is one of the more heavier elementary particle that bestows mass to other elementary particles. Additionally, there are anti-particles associated with these elementary particles, with the same mass but opposite charge.

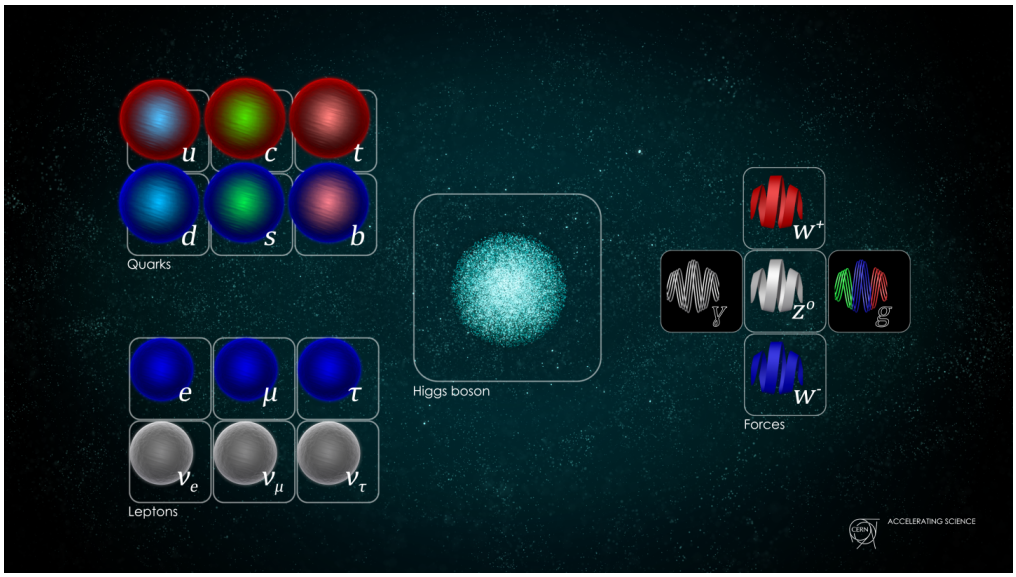


Figure 2.1: Particles and force carriers of the Standard Model [1]

2.1.1 The Fundamental Forces

We all are familiar with two of the fundamental forces: gravity and electromagnetic force. The force of gravity has still not been incorporated into the Standard Model, because a theory of quantum gravity hasn't yet been worked out, we just assume that gravity has negligible effects on the atomic scale, and it is mediated through the massless photon (represented as γ). The remaining two forces are the weak force and the strong force. The strong force plays its part in keeping the atom held together. The strong force overcomes the electromagnetic repulsion that the protons face.

The strong force is mediated through gluons, which are massless. The weak force is responsible for radioactive decay. The Glashow-Salam-Weinberg (GWS) theory unified electromagnetic force and the weak force. According to this theory, at sufficiently high energies, there is negligible difference between the two forces, and they act together as the electroweak force. The mediators of the weak force are the massive W^+,W^-, and the Z^0 bosons.

2.1.2 Leptons

Leptons are spin-1/2 fermions. Put simply, fermions are particles whose spin quantum number is an odd multiple of 1/2. They interact via the electromagnetic (if charged), gravity and weak forces. They are divided into three generations: electron (e^-), muon (μ), and tau (τ). Higher generations are heavier, and all three generations possess some charge. There also exist neutral leptons called neutrinos and are of three types: electron-neutrino (ν_e), muon-neutrino (ν_μ) and the tau-neutrino (ν_τ). Neutrinos do not carry any charge, and are not affected by the electromagnetic forces. As a result, they pass through matter undeterred. They are created through radioactive decay, and are even created by the nuclear reactions in the Sun. Fig 2.1 shows the leptons.

2.1.3 Quarks

Quarks are among the fundamental building blocks of matter. Protons and neutrons are made up of quarks. Their existence has been experimentally verified. Quarks are grouped into six flavors: up (u), down (d), charm (c), strange (s), top (t) and bottom (b). These flavors describe fundamental differences in properties such as charge and mass. The up, charm and top quarks carry a charge of $+\frac{2}{3}e$, where e is the elementary charge, while the down, strange, and bottom quarks have a charge of $-\frac{1}{3}e$.

Each quark flavor has its corresponding antiquark, which carries the opposite charge and other quantum numbers. In nature, quarks never appear alone but are always confined within larger particles called hadrons. Hadrons are classified into two main families: baryons and mesons. Baryons, such as protons and neutrons, are composed of three quarks, while mesons are made up of a quark-antiquark pair. Fig 2.1 shows that quarks are divided into three generations. Higher generations are less stable, the up and down quarks are the most stable.

Quarks are subject to the strong nuclear force. Quarks group together to form other particles. Proton is composed of two up quarks and a down quark (uud), whereas the neutron is composed of one up quark and two down quarks (udd). They also interact with other particles via the weak nuclear force. The weak nuclear force can cause quarks to turn into heavier generation by absorbing a W boson, or emitting a W boson to turn into a lighter generation.

An example to illustrate this is beta decay, where a neutron turns into a proton, an electron and a neutrino. In this process, physical properties such as charge and momentum are conserved. One of the down quarks in the neutron turns into a up quark, releasing a W^- boson. Now there are two down quarks and an up quark, making it a proton. The electron and neutrino are released by the decay of the W^- boson.

2.1.4 Gauge Bosons

Gauge bosons are the force-carrying particles. They mediate the fundamental interactions between elementary particles. The Standard Model describes three of the four known fundamental forces in nature: the electromagnetic force, the weak force, and the strong force. Each of these forces has its own corresponding gauge bosons, which arise from the symmetry principles governing the interactions. They are of three types: the photon (γ) carrying the electromagnetic force, the W boson and the Z boson carrying the weak force, and the gluons which carry the strong force. All of them have been experimentally verified. Fig 2.1 shows the gauge bosons.

2.1.5 The Higgs Boson

The Higgs boson is a fundamental particle associated with the Higgs field, a field that gives mass to elementary particles in the Standard Model of particle physics. Its discovery at the Large Hadron Collider (LHC) in 2012 by the ATLAS and CMS collaborations marked a significant milestone in modern physics,

confirming the last missing piece of the Standard Model.

The Higgs mechanism explains how particles acquire mass. In the Standard Model, all particles initially have no mass. However, as they interact with the Higgs field, which permeates all of space, they gain mass. The more strongly a particle interacts with this field, the heavier it becomes. The Higgs boson is an excitation, or quantum manifestation, of the Higgs field, much like a photon is an excitation of the electromagnetic field.

2.1.6 Hadrons

Hadrons are composite particles made up of quarks, bound together by the strong force, which "glues" quarks together inside hadrons. The two major classes of hadrons are baryons, which are made of three quarks, and mesons, which consist of one quark and one anti-quark. They are integral to the structure of atomicnuclei, as protons and neutrons, both baryons, are the building blocks of matter.

Protons are made up of two up quarks and a down quark, while neutrons are made up of one up quark and two down quarks. Examples of the mesons include the pion π^+ made up of $(u\bar{d})$ and the kiaon K^+ made up of $(u\bar{s})$.

2.2 Beyond the Standard Model

Even though the elements of the Standard Model have been experimentally verified, there are strong reasons to believe that this is not the ultimate theory of matter, rather it is a low energy approximation of a grander theory. There are several phenomena that remain unexplained by the Standard Model:

- **Neutrino Oscillations** Neutrinos (uncharged leptons) change flavor/generations as they travel. The most famous example of this phenomenon is the neutrino oscillations of the Sun, where an electron-neutrino which originally formed during the nuclear reaction of the Sun changes its flavor to muon-neutrino or a tau-neutrino as it reaches the Earth. These oscillations occur only when they have mass, but the Standard Model predicts them to be massless.

- **Matter Anti-Matter Asymmetry** The Standard Model predicts that there should be equal amounts of matter and anti-matter. But we know that the Universe is made up of only matter, the SM has not been able to explain this asymmetry.

- **Gravity** Gravity has not been included in the Standard Model. We just assume that gravity has negligible effects on the atomic scale, but if we think about it, gravity is weaker than the weak force, so it is baffling to know that we have yet to figure out how gravity works in this scale.

- **Dark Matter and Dark Energy** Observations over the years have verified that the known Universe is made up of 5% matter, the rest is dark matter and dark energy. The Standard Model can not explain this anomaly yet. We still do not have a good dark matter candidate.

These shortcomings of the Standard Model have led researchers to put their thinking caps on and look for theories beyond the Standard Model (BSM). This section is based on [2] Griffith's Introduction to Elementary Particles and [3] Martin and Shaw's Particle Physics.

2.2.1 Two-Higgs-Doublet Models (2HDM)

In the Standard Model (SM) of particle physics, there is only one Higgs doublet, which gives rise to a single Higgs boson. However, many Beyond the Standard Model (BSM) theories propose an extended Higgs sector to explain phenomena not covered by the SM, such as the hierarchy problem or the nature of dark matter. One of the simpler and more widely studied extensions is the Two-Higgs-Doublet Model (2HDM). This model assumes the existence of two Higgs doublets, Φ_1 and Φ_2, instead of just one. These two scalar fields contribute to a more complex Higgs sector, introducing multiple physical Higgs bosons.

In the 2HDM, spontaneous symmetry breaking leads to five distinct physical Higgs bosons: a pair of charged Higgs bosons (denoted as H^+ and H^-), one light neutral scalar Higgs boson (often referred to as h and identified with the observed 125-GeV SM-like Higgs boson), one heavy neutral scalar Higgs boson (denoted as H), and one neutral pseudoscalar Higgs boson (denoted as A). This enriched Higgs sector offers various new physics signatures and potential discoveries at high-energy experiments like those at the Large Hadron Collider (LHC).

A particularly interesting variant of the 2HDM achieves consistency with the experimentally observed 125-GeV Higgs boson through a mechanism known as the *Gildener-Weinberg (GW) scalon scheme*. This mechanism not only stabilizes the mass of the Higgs boson but also ensures that its couplings to gauge bosons and fermions are aligned with those of the single Higgs boson in the Standard Model. In this variant, the 125-GeV scalar behaves almost exactly like the SM Higgs, while the additional BSM Higgs bosons predicted by the model are within the reach of current LHC energy levels. However, detecting these

additional Higgs bosons requires improving experimental sensitivity to their potentially lower masses. This section is based on [4] Symmetry Breaking and Scalar Bosons by Gildener, Eldad and Weinberg, Steven.

In this thesis, one particular channel of interest for discovering BSM Higgs bosons in the 2HDM is the **1-lepton** $bbWW$ channel. In this channel, the background is dominated by pair-produced top quarks. This background can be fully reconstructed and suppressed due to the kinematic constraints imposed at all the decay vertices of the top-quark pairs. Similarly, the signal events from BSM Higgs bosons in this channel can also be fully reconstructed by considering the kinematic constraints at the decay vertices. This allows for a significant improvement in the ability to distinguish between the signal and the background, enhancing the overall sensitivity of the search for new physics.

Chapter 3

Experimental Setup

3.1 The Large Hadron Collider

The Large Hadron Collider (LHC), operated by by CERN, is the world's largest and most powerful particle accelerator. Situated underground at the Franco-Swiss border near Geneva, Switzerland, the LHC represents one of humanity's most ambitious scientific experiments. Spanning 27 kilometers in circumference, this massive experiment is designed to accelerate protons to near-light speeds and collide them with unprecedented energy levels. These high-energy collisions enable physicists to probe fundamental particles, providing insights into the particles and forces that compose the universe.

The reason for such a massive detector lies in the de Broglie laws we are all familiar with. The wavelength of a particle is inversely proportional to its momentum (a measure of energy). Therefore, to probe particles with really low wavelengths, we need really high energy. It must be noted that quantum mechanics is at play at this level, so these entities exhibit wave-particle duality.

We use a collider with two beams as their collision will have more energy than, say, a fixed target collider. We use a dipole magnetic field to accelerate these two beams, that are counter-circulating, to near-light speeds. Current collision energy at the LHC is 13.6TeV.

The LHC is designed to provide proton-proton collisions at a center of mass energy up to $\sqrt{s} = 14\text{TeV}$ for four experiments: ATLAS, CMS, LHCb, and ALICE. The accelerator complex, pictured in Fig 3.1 is a succession of machines with increasing energies. Each machine injects a beam to the next one, which further increases the beam energy. We will be focussing on the ATLAS experiment.

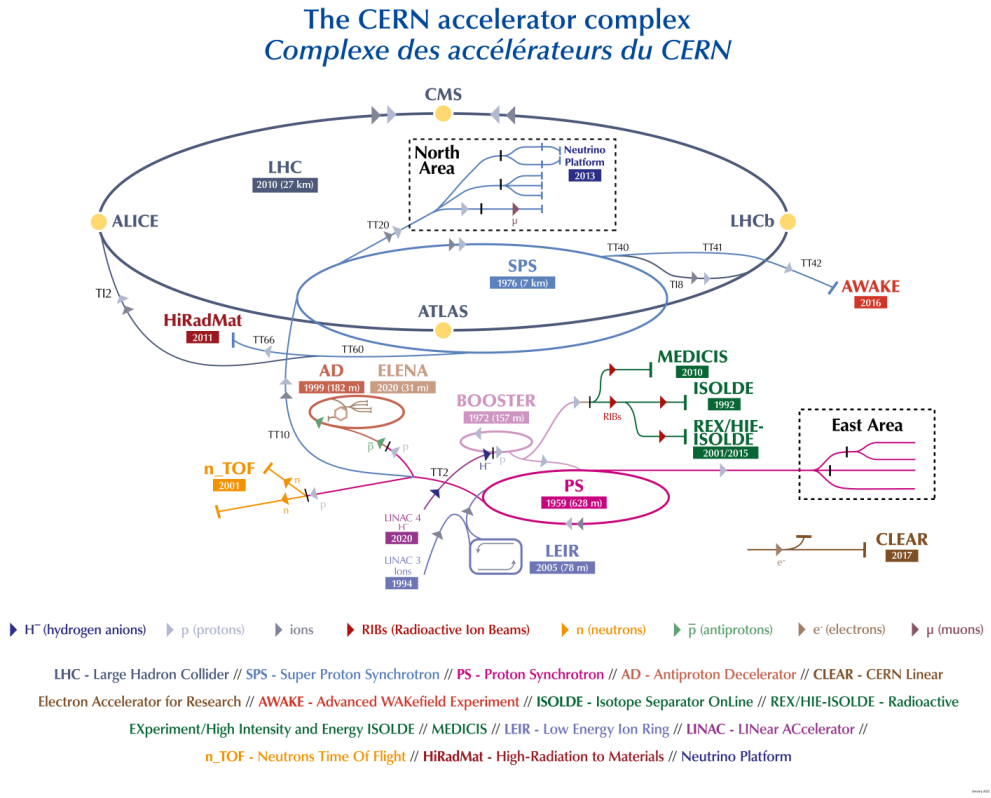


Figure 3.1: CERN accelerator complex [5]

3.2 The ATLAS Experiment

The ATLAS experiment is one of the four primary detectors at CERN's Large Hadron Collider (LHC), It is known formally as "A Toroidal LHC ApparatuS". It is a general-purpose detector designed to investigate a wide array of physics phenomena, from fundamental particles and forces to testing the boundaries of the Standard Model. The ATLAS experiment, along with the CMS experiment, discovered the Higgs Boson in 2012.

The detector, as monumental as the scientific goals it aims to achieve, is both massive and meticulously structured, standing 25 meters high and spanning 44 meters in length. ATLAS operates by layering different detection systems, each dedicated to capturing specific information about particles produced in collisions. At the heart of the detector lies the **Inner Detector**, the closest section to the collision points, which tracks the paths of charged particles as they pass through. This inner layer is crucial for measuring the momentum of particles with high precision, providing insights into particle interactions within moments of collision.

Surrounding the inner detector are the **calorimeters**, specialized systems that measure the energy of particles by absorbing them. These calorimeters are divided into two main types: the electromagnetic

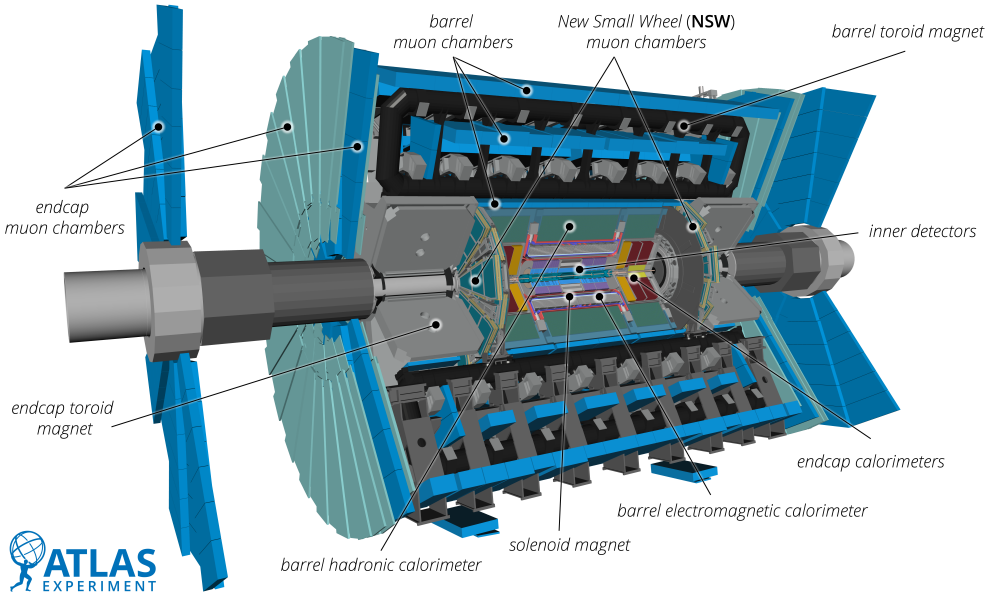


Figure 3.2: The ATLAS Detector [6]

calorimeter, which focuses on capturing data from photons and electrons, and the hadronic calorimeter, dedicated to recording information about hadrons, particles that contain quarks, such as protons and neutrons. By capturing the energy deposited by these particles, the calorimeters allow scientists to classify and analyze them with accuracy.

The **Muon Spectrometer**, positioned on the outer edges of the detector, is essential for detecting muons, a type of heavy particle that can pass through dense matter with minimal loss of energy. Muons are valuable in particle physics experiments because they provide a clean, distinct signal. The final layer, the **magnetic field system**, incorporates powerful magnets, enabling ATLAS to measure particles' momentum by bending their trajectories. This setup is especially effective for identifying the charge of particles and providing clues about their momentum.

3.3 Detector Coordinates

The ATLAS experiment at the Large Hadron Collider (LHC) uses a cylindrical coordinate system tailored to its detector geometry to analyze particle collision data. This system centers around the **z-axis**, which aligns with the LHC's beamline where proton collisions occur, with the origin located at the interaction point. The **radial distance** r measures the perpendicular distance from the beamline, tracking how far particles move outward from the collision center. The **azimuthal angle** ϕ, defined from 0 to 2π radians in the xy-plane, indicates rotational orientation around the z-axis.

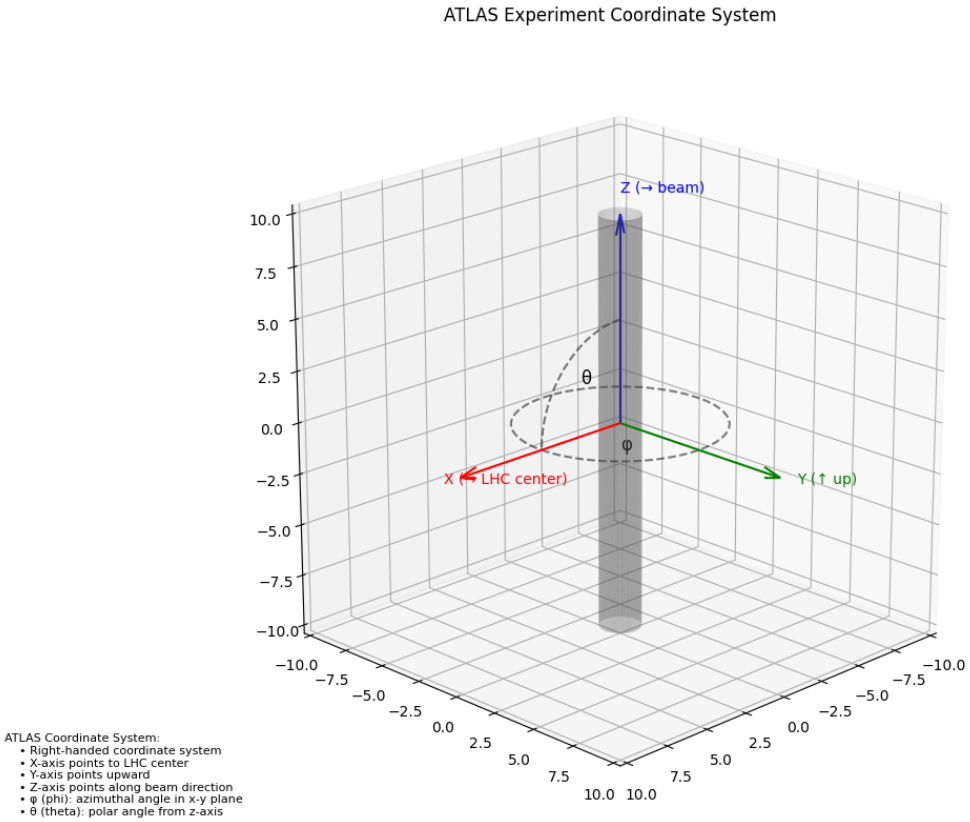


Figure 3.3: Coordinate System of the ATLAS Experiment

ATLAS also employs the **polar angle** θ, the angle a particle's trajectory makes with the beamline, and **pseudorapidity** η, a function of θ given by

$$\eta = -\ln\left(\tan\frac{\theta}{2}\right).$$

Pseudorapidity provides a finer resolution for describing particle directions, especially at high energies, making it particularly useful for particles traveling nearly parallel to the beamline. Defined in terms of the polar angle of a particle's trajectory, pseudorapidity remains nearly invariant under Lorentz boosts along the beam axis. This invariance is advantageous because particles in the ATLAS detector often move at velocities close to the speed of light, and their momenta are significant along the beamline. By using pseudorapidity, which is stable under these conditions, physicists can describe particle directions consistently across collision events, regardless of differences in reference frames moving along the beam axis.

Transverse momentum p_T, defined as the momentum perpendicular to the beamline, is a crucial measurement as it remains invariant under boosts along the z-axis. This property makes p_T reliable for interpreting collision dynamics, as any momentum in the transverse plane must originate from the collision

itself.

The ATLAS detector's components rely on this coordinate system to track particle paths and energy deposits. The Inner Detector uses r, ϕ, and z to trace charged particles, while the calorimeters measure energy deposits localized by η and ϕ. The Muon Spectrometer, using r, ϕ, and z, captures the trajectories and momenta of muons that penetrate to the detector's outer layers. Fig 3.3 represents the coordinates used at the ATLAS Experiment.

3.4 Particle matter interaction

The study of particle-matter interactions is fundamental in particle physics, as it underpins the principles by which detectors observe and identify particles produced in high-energy collisions. When particles traverse matter, they interact with atomic nuclei and electrons, resulting in energy loss, deflection, and, in some cases, the creation of secondary particles.

3.4.1 Ionization and Excitation

Charged particles, such as electrons, protons, and muons, lose energy primarily through ionization and excitation as they pass through matter. When a charged particle passes near an atom, it exerts an electric force on the atom's electrons. If this force is strong enough, it can remove an electron from the atom, causing *ionization*. Alternatively, it may excite the atom, raising an electron to a higher energy level without ejecting it. The energy lost by the particle in these processes is transferred to the material, leaving a trail of ionized atoms and excited states along its path. This trail is used in tracking detectors to measure particle trajectories and, indirectly, their energies.

The energy loss per unit distance, denoted by $\frac{dE}{dx}$, can be described by the Bethe-Bloch formula:

$$\frac{dE}{dx} = -\frac{4\pi N_A z^2 e^4}{m_e c^2 \beta^2} Z \left[\ln \left(\frac{2 m_e c^2 \beta^2 \gamma^2 T_{\max}}{I^2} \right) - \beta^2 \right],$$

where N_A is Avogadro's number, z is the charge of the particle, $\beta = \frac{v}{c}$ is the velocity of the particle relative to the speed of light, Z is the atomic number of the material, $T_{\max}$ is the maximum kinetic energy that can be transferred in a single collision, and I is the mean excitation potential of the material. The Bethe-Bloch formula is particularly accurate for heavy charged particles at moderate speeds. Fig 3.4 shows the graphical representation of the Bethe-Bloch formula.

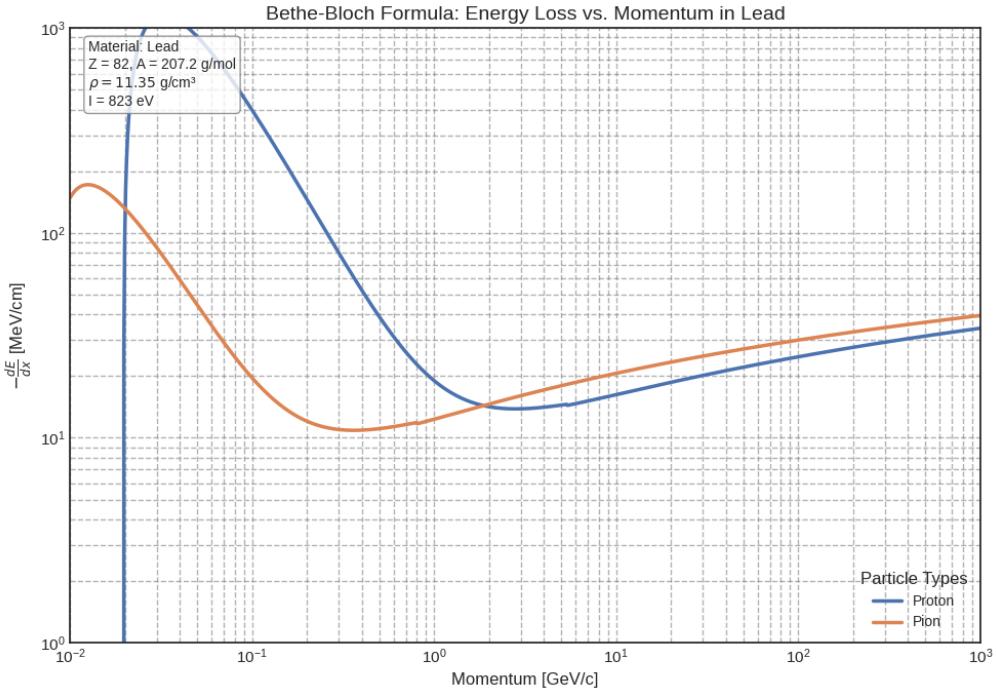


Figure 3.4: Bethe-Bloch formula for protons and pions in Lead

3.4.2 Bremsstrahlung

When a charged particle, particularly an electron, is accelerated in the electric field of an atomic nucleus, it can emit electromagnetic radiation known as *bremsstrahlung* (or "braking radiation"). This process is a significant mechanism of energy loss for electrons and positrons in dense materials, especially at high energies. The energy loss due to bremsstrahlung is proportional to the particle's energy and becomes dominant over ionization for electrons with energies above several MeV.

The energy loss per unit length due to bremsstrahlung can be approximated by:

$$-\frac{dE}{dx} \approx \frac{E}{L_R},$$

where E is the energy of the particle and L_R is the radiation length of the material, a characteristic distance over which the particle's energy is reduced by a factor of $1/e$. This property is critical in electromagnetic calorimeters, which rely on bremsstrahlung and pair production to measure the energy of electrons and photons. Fug 3.5 shows the graphical representation of this formula.

L_R is given as

$$\frac{1}{L_R} = 4\left(\frac{\hbar}{mc}\right) Z(Z+1)\alpha^2 n_a \log\left(\frac{183}{Z^{1/3}}\right)$$

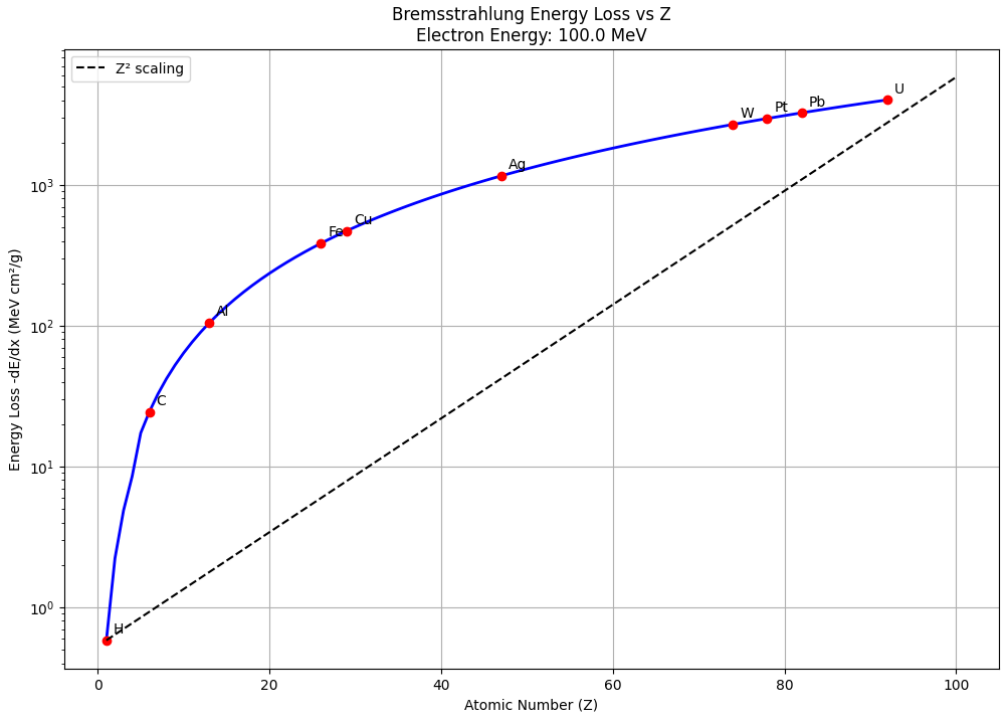


Figure 3.5: Rate of energy loss vs Atomic number

3.4.3 Electromagnetic Interactions: Pair Production and Photon Interactions

High-energy photons interact with matter primarily through three processes: the photoelectric effect, Compton scattering, and pair production. These interactions vary in significance depending on the photon's energy and the properties of the material, with each mechanism having distinct features and energy thresholds.

Photoelectric Effect The photoelectric effect is a process in which a photon transfers all of its energy to an atomic electron, ejecting it from the atom. This process is most likely to occur at low photon energies, typically less than a few hundred keV, and in materials with high atomic numbers Z due to the higher binding energies of inner-shell electrons. The energy of the ejected electron, or photoelectron, is given by:

$$E_e = h\nu - E_b,$$

where $h\nu$ is the energy of the incident photon, and E_b is the binding energy of the electron in its atomic shell. The photoelectric effect has a high cross-section for photons interacting with tightly bound inner electrons (e.g., K or L shells), making it the dominant interaction mechanism for low-energy photons in materials with high Z.

In particle detectors, the photoelectric effect is important in scintillation detectors and certain types of calorimeters, where the energy of the incident photon is transferred directly to the detector medium through ionization processes, enabling energy measurements.

Compton Scattering Compton scattering, shown in Fig 3.6 occurs when a photon interacts with a loosely bound or free electron, transferring a portion of its energy to the electron and being scattered in a different direction with reduced energy. The energy and angle of the scattered photon, E' and θ respectively, are related by the Compton formula:

$$E' = \frac{E}{1 + \frac{E}{m_e c^2}(1 - \cos\theta)},$$

where E is the energy of the incident photon, m_e is the electron rest mass, and c is the speed of light. Compton scattering is most significant for photon energies in the range of hundreds of keV to several MeV and plays a key role in photon interactions with matter in this energy range.

This process is particularly important in detectors designed for gamma rays, as Compton scattering provides information on the photon's original energy and direction through the analysis of the scattered photon and recoiling electron. Compton scattering is the dominant interaction mechanism for medium-energy photons in most materials.

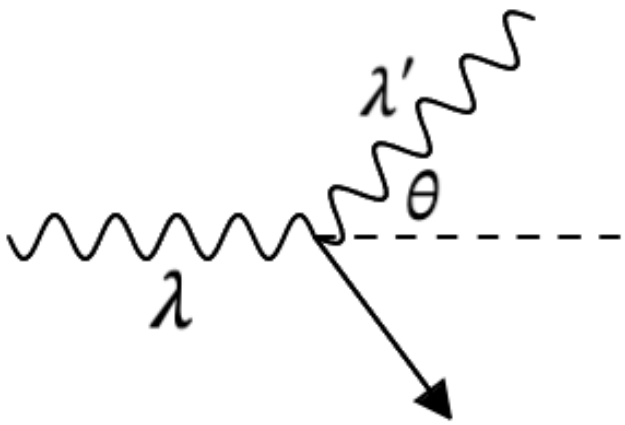


Figure 3.6: Compton Scattering

Pair Production Pair production, shown in Fig 3.7 is a process in which a photon with energy exceeding $2m_e c^2$ (the combined rest mass of an electron and a positron, or about 1.022 MeV) interacts with the electric field of a nucleus or an electron, creating an electron-positron pair. The excess energy of the photon beyond

the threshold of 1.022 MeV is transferred to the kinetic energy of the created particles. The pair production process is represented as:

$$\gamma \rightarrow e^- + e^+.$$

For pair production to occur, the photon must be in the vicinity of a nucleus or electron to conserve momentum, as a free photon cannot simultaneously conserve both energy and momentum in this process. The cross-section for pair production increases with the photon energy and becomes the dominant interaction mechanism for photons with energies in the GeV range, which is particularly relevant in high-energy physics experiments.

Pair production plays a critical role in electromagnetic calorimeters used in particle physics. In these detectors, high-energy photons entering the calorimeter initiate a cascade of secondary particles through successive pair production and bremsstrahlung interactions, creating an *electromagnetic shower*. By measuring the total energy deposited in the calorimeter, the energy of the original photon can be determined accurately.

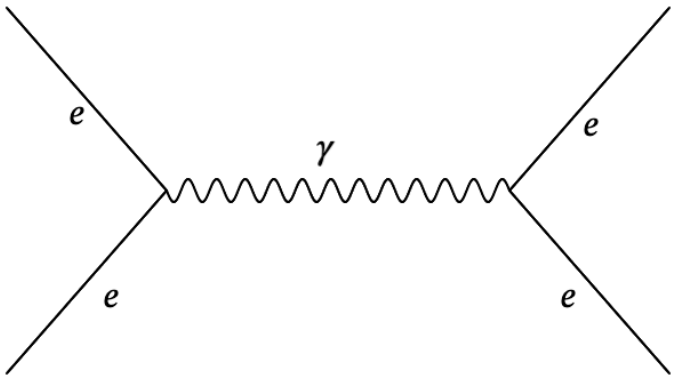


Figure 3.7: Pair Production

3.4.4 Nuclear Interactions

Nuclear interactions are a primary means by which neutral and charged hadrons interact with matter, as these particles do not engage significantly with atomic electrons through electromagnetic forces. Instead, neutral hadrons, such as neutrons and neutral kaons, interact with nuclei via the *strong nuclear force*. This interaction often results in *inelastic scattering* events, where the incoming hadron collides with a nucleus, leading to the production of secondary particles and nuclear fragments. Charged hadrons, such

as protons, charged pions, and kaons, also undergo nuclear interactions but can additionally lose energy through ionization as they traverse matter.

Inelastic Scattering and Hadronic Showers Inelastic scattering is a process in which a hadron strikes a nucleus, transferring some of its energy and inducing nuclear reactions. These interactions break up the target nucleus and produce a cascade of secondary particles, which can include pions, kaons, neutrons, protons, and even photons from nuclear de-excitations. The result is a *hadronic shower*—a complex sequence of interactions producing a multitude of particles, each carrying a portion of the initial energy. This cascading process is essential in hadronic calorimeters, which aim to measure the energy of incident hadrons by capturing and absorbing all secondary particles in the shower.

The development of a hadronic shower is highly stochastic, meaning that each individual interaction is governed by probabilistic rules. The energy of the primary hadron is progressively distributed among a large number of secondary particles, most of which eventually lose their energy through ionization or further nuclear interactions within the detector material. The depth and structure of the shower depend on the energy and type of the incident particle, as well as the material's nuclear properties.

Hadronic Calorimeters Hadronic calorimeters are designed specifically to measure the energy of hadrons by utilizing nuclear interactions to absorb their energy. These detectors are usually constructed with dense materials that have high atomic and mass numbers, which increase the probability of nuclear interactions and, consequently, the full development of hadronic showers. Common absorber materials in hadronic calorimeters include steel, iron, lead, and uranium, which have favorable properties for capturing secondary particles and localizing the shower.

Hadronic calorimeters can be classified as *sampling calorimeters* or *homogeneous calorimeters*. In a sampling calorimeter, alternating layers of absorber material and active material (such as scintillators or gaseous detectors) are used. The active layers detect the ionization signals from the charged particles in the hadronic shower, while the absorber layers encourage nuclear interactions and secondary particle production. Homogeneous calorimeters, by contrast, are made entirely of a single active material that both induces and detects the interactions, though this approach is more common in electromagnetic rather than hadronic calorimetry due to cost and efficiency factors.

Nuclear Interaction Length The nuclear interaction length, λ, is a key parameter in the design of hadronic calorimeters. It represents the mean distance a high-energy hadron travels in a material before undergoing a nuclear interaction. The interaction length depends on the material's nuclear cross-section and density, with shorter interaction lengths corresponding to a higher probability of nuclear interaction over a given distance. For effective energy measurement, the thickness of a hadronic calorimeter must be several interaction lengths, ensuring that most hadronic showers are fully contained within the detector. Typically, hadronic calorimeters at experiments like the ATLAS or CMS detectors are around 7λ to 11λ thick, allowing nearly complete absorption of hadronic showers, even from high-energy particles.

Differences Between Electromagnetic and Hadronic Showers Hadronic showers differ significantly from electromagnetic showers, which are created by electrons, positrons, and photons. Electromagnetic showers primarily involve processes like bremsstrahlung and pair production, which produce only electromagnetically interacting particles. Hadronic showers, however, involve a wide range of secondary particles, including neutrally and strongly interacting particles (e.g., neutrons and pions). Additionally, a portion of the energy in hadronic showers is lost to nuclear binding energy and other non-visible processes, which is often referred to as the *invisible energy*.

Invisible energy in hadronic showers can lead to fluctuations in the measured energy, as some of the energy from hadronic interactions is not deposited in the calorimeter's active regions. This property makes hadronic calorimeters less precise than electromagnetic calorimeters and often requires specific *calibration techniques* to correct for energy losses and improve measurement accuracy.

3.4.5 Cherenkov Radiation

Cherenkov radiation is emitted when a charged particle traverses a medium at a speed greater than the phase velocity of light in that medium. This condition leads to the emission of a cone of coherent light at a characteristic angle, known as the Cherenkov angle, θ_C, which depends on the particle's velocity and the refractive index of the medium. The Cherenkov angle is given by:

$$\cos\theta_C = \frac{c}{nv},$$

where c is the speed of light in a vacuum, n is the refractive index of the medium, and v is the velocity of the particle. Cherenkov radiation is analogous to a sonic boom in air, produced when an object moves

faster than the speed of sound. This radiation appears as a characteristic blue glow due to the emission of light predominantly in the blue and ultraviolet parts of the spectrum.

Conditions for Cherenkov Radiation For Cherenkov radiation to occur, two conditions must be met:

- The particle must be charged, as neutral particles do not interact with the medium's electromagnetic field in the same way.
- The particle's velocity v must exceed the phase velocity of light in the medium, $v_p = \frac{c}{n}$. Therefore, the threshold speed for Cherenkov radiation is $v > \frac{c}{n}$. As the refractive index n is greater than 1 for all transparent materials, this condition can be satisfied by particles moving at relativistic speeds.

The intensity of Cherenkov radiation depends on the speed of the particle and increases with the angle of emission, up to a maximum determined by the particle's speed and the medium's refractive index. The light is emitted as a coherent wavefront, forming a cone with its vertex along the particle's path.

Cherenkov Angle and Particle Velocity Measurement The Cherenkov angle θ_C is directly related to the particle's velocity and can therefore be used to measure the particle's speed when the refractive index n of the medium is known. For a given medium, θ_C is maximized for particles moving at or near the speed of light. By measuring θ_C precisely, one can determine the particle's velocity. If the particle's momentum is also known, its mass can be inferred, allowing for particle identification.

The Cherenkov angle is given by:

$$\theta_C = \arccos\left(\frac{c}{nv}\right).$$

For high-energy particles where $v \approx c$, this angle is close to a fixed maximum, determined primarily by the medium's refractive index. The sharp dependence of θ_C on v makes Cherenkov detectors effective at distinguishing between particles with small differences in velocity.

3.5 Application of Cherenkov Radiation in the LUCID Detector

The **LUCID (Luminosity Cherenkov Integrating Detector)** is a specialized subsystem within the ATLAS experiment at CERN's Large Hadron Collider (LHC). Its primary purpose is to measure the instantaneous and integrated luminosity of proton-proton collisions with high precision. This is essential

for understanding the collision rates and ensuring the accurate normalization of cross-section measurements in particle physics experiments.

3.5.1 Principle of Cherenkov Radiation in LUCID

Luminosity is a critical parameter in particle physics, quantifying the rate of potential particle interactions in a collider. It represents the number of possible particle collisions per unit area per unit time, effectively indicating the density of particles in the accelerator beam.

The LUCID detector measures luminosity by leveraging Cherenkov radiation. Positioned around the beam pipe at a carefully chosen distance, it captures particles emerging from proton-proton collisions at small angles. The detector consists of an array of quartz tubes that act as radiators, paired with photomultiplier tubes (PMTs) to detect and amplify Cherenkov light. When charged particles pass through the quartz radiators at high speeds, they produce Cherenkov radiation, which is collected and converted into electrical signals.

The amount of Cherenkov light detected is directly proportional to the number of charged particles passing through the detector, enabling an estimate of the collisions' luminosity. By tracking these measurements over time, LUCID calculates the total number of collisions occurring within a given period.

LUCID's luminosity measurements are essential for interpreting data from the ATLAS experiment. By determining collision rates, it allows researchers to accurately calculate cross-sections for key processes, such as Higgs boson production, rare particle decays, and searches for new physics beyond the Standard Model. Additionally, the precise and reliable nature of Cherenkov-based detection enhances the overall accuracy of ATLAS data analysis. This section is based on [3] Martin and Shaw's Particle Physics.

Chapter 4

Analysis

4.1 Analysis of Signal and Background Processes

The following section analyzes two processes: one representing a Standard Model (SM) background process, and the other illustrating a potential Beyond Standard Model (BSM) signal. Both diagrams depict interactions that could occur at the Large Hadron Collider (LHC).

4.1.1 Background Process

The first diagram (Figure 4.1) represents a SM background process involving gluon-gluon fusion, which leads to the production of a top quark (t) and an anti-top quark ($\bar{t}$) pair, followed by their decay.

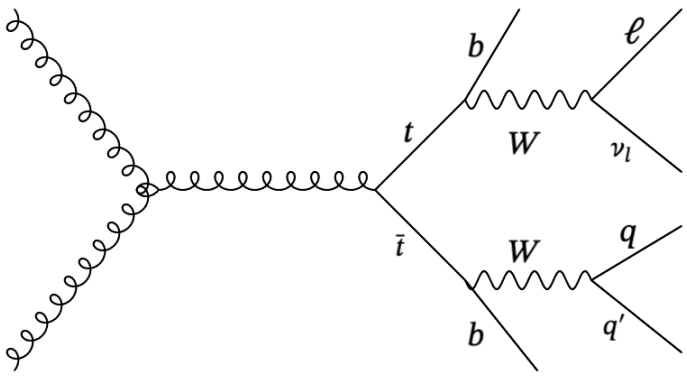


Figure 4.1: Standard Model background process: gluon-gluon fusion producing a $t\bar{t}$ pair.

In this process, two gluons (g) interact to produce a $t\bar{t}$ pair, a common occurrence in SM processes at the LHC. The top quark t subsequently decays into a W^+ boson and a b-quark, while the anti-top quark $\bar{t}$ decays into a W^- boson and another b-quark. The W^+ boson then decays into a lepton (ℓ) and a neutrino

(ν_ℓ), while the W^- boson decays into a pair of light quarks, denoted q and q'. As a result, the final state of this process includes a lepton, a neutrino, two b-quarks, and two light quarks. This background process is crucial in the study of top quark pair production, as it can mimic signals for new physics. Therefore, accurate modeling of this process is necessary to distinguish BSM signals from SM backgrounds.

4.1.2 Signal Process with Beyond Standard Model (BSM) Particles

The second diagram (Figure 4.2) depicts a BSM process involving hypothetical particles labeled A/H_2 and H^+. Such processes are predicted in models with extended Higgs sectors, such as the Two Higgs Doublet Model (2HDM).

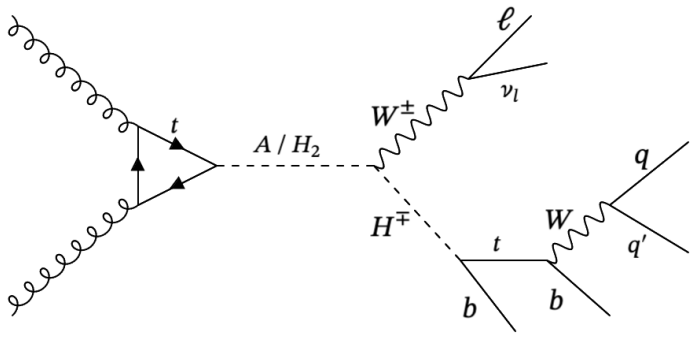


Figure 4.2: BSM signal process involving a neutral scalar or pseudoscalar A/H_2 and a charged Higgs boson H^+.

In this BSM scenario, two gluons initially fuse to produce a virtual top quark loop, which subsequently emits a neutral BSM particle, labeled as A or H_2. This particle decays into a $W^\pm$ boson and a charged Higgs boson ($H^\pm$), a signature that is characteristic of BSM theories. The $W^\pm$ boson decays into a lepton (ℓ) and a neutrino (ν_ℓ), while the $H^\pm$ decays into a top quark t and a b-quark. The produced top quark then decays into a W boson and another b-quark, with the W boson decaying into two light quarks, labeled q and q'. Consequently, the final state of this signal process also consists of a lepton, a neutrino, two b-quarks, and two light quarks, which is notably similar to the final state in the SM background process.

4.1.3 Comparative Analysis

The similarity in final states between the background process and the BSM signal process poses a significant challenge in experimental searches for new physics at the LHC. Since both processes result in the same observable particles, distinguishing the BSM signal from the SM background requires sophisticated

techniques. This challenge emphasizes the importance of accurate background modeling and the use of advanced methods, such as machine learning algorithms, to effectively separate BSM signals from SM backgrounds.

4.2 Monte Carlo Simulation Samples

To study the physics processes of interest, we needed to generate simulated collision events that could be compared with real data. These simulations were created using two main software packages: [7] MadGraph5_aMC@NLO , which generates the initial particle interactions, and [8]Pythia8, which simulates how these particles decay and interact with the detector.

Our simulation is based on a theoretical model called the Type-I Two-Higgs-Doublet Model (2HDM), which predicts the existence of additional Higgs bosons beyond the one discovered in 2012. We focused on two of these predicted particles: a neutral particle called the pseudoscalar A, and a charged particle called $H^{\pm}$. The model was implemented using specialized software that includes all the necessary physics calculations based on theortical predictions.

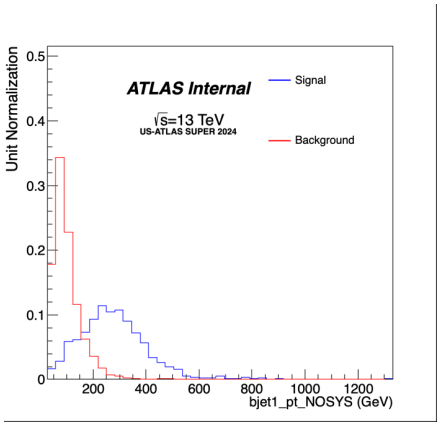
For each combination of particle masses we wanted to study, we generated 100,000 events. This number was chosen to match what we might expect to see in real data, based on theoretical predictions. We considered various mass combinations:

- The pseudoscalar A with masses ranging from 300 to 1000 GeV
- The charged Higgs $H^{\pm}$ with masses ranging from 200 to 800 GeV

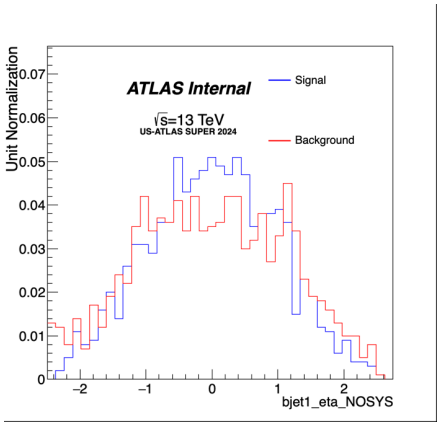
The simulation was performed using [9]AthGeneration 23.6.10 with AtlasFast3, utilizing the [7]MadGraph_NNPDF30NLO_Base_Fragment with the A14 tune and [10]NNPDF23LO PDF set. The events were generated at effective one-loop QCD in MadGraph at leading order (LO) using the four-flavor scheme.

4.3 Input variables and Correlations

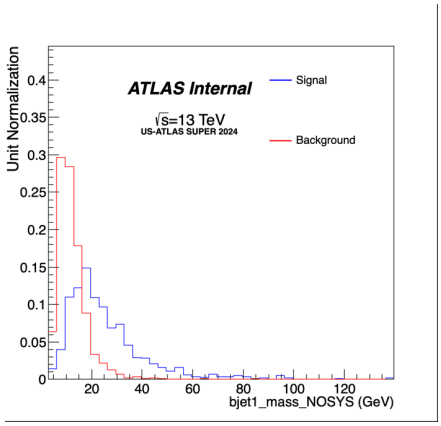
4.3.1 Kinematic Variables



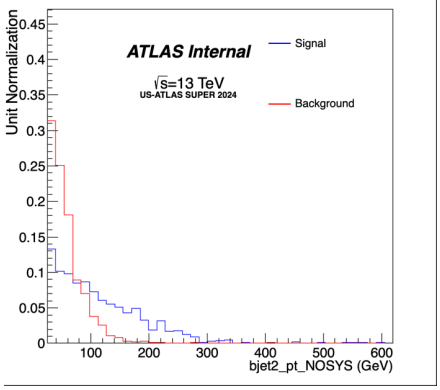
Transverse momentum for bjet1



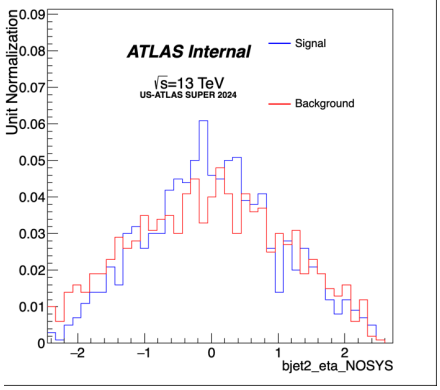
Eta measurement for bjet1



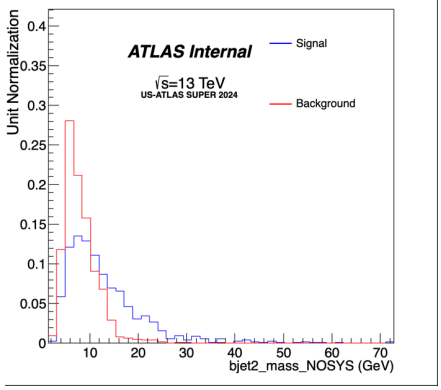
Mass measurement for bjet1



Transverse momentum for bjet2

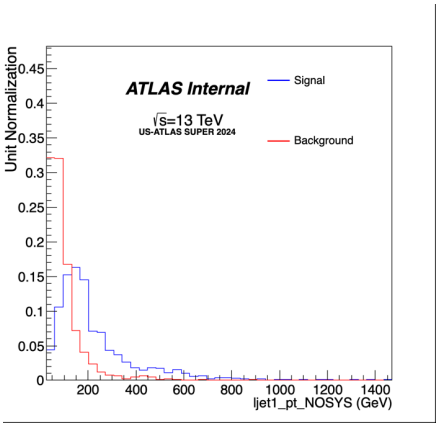


Eta measurement for bjet2

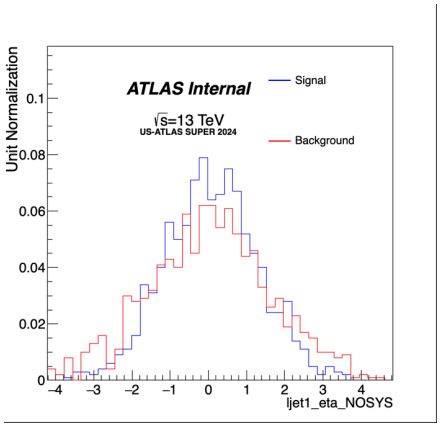


Mass measurement for bjet2

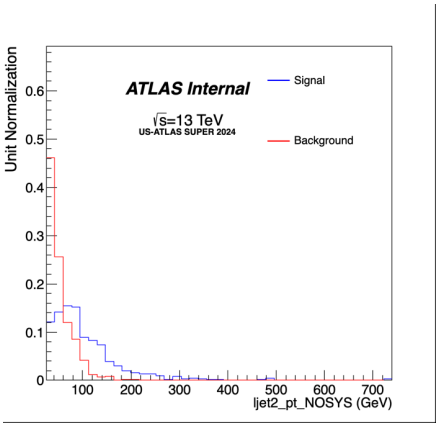
Figure 4.3: bjet input variables



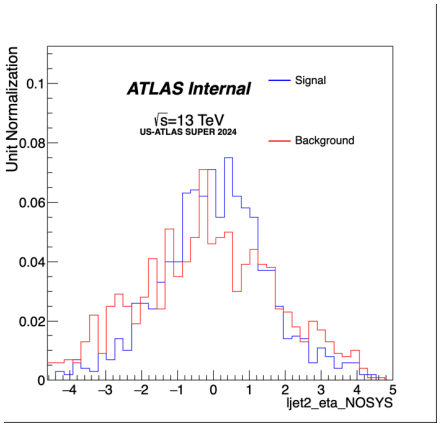
Transverse momentum for ljet1



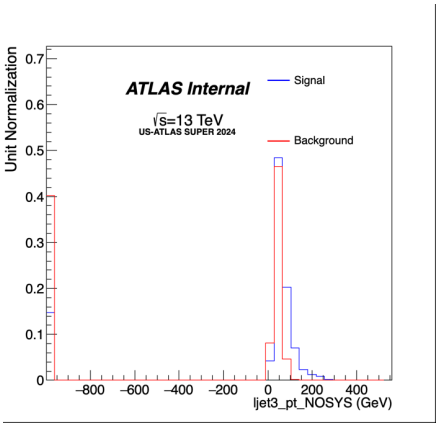
Eta measurement for ljet1



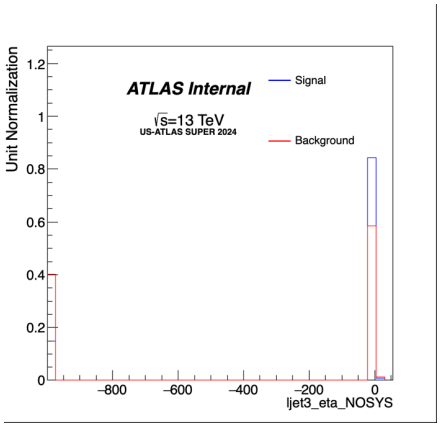
Transverse momemtum for ljet2



Eta measurement for ljet2

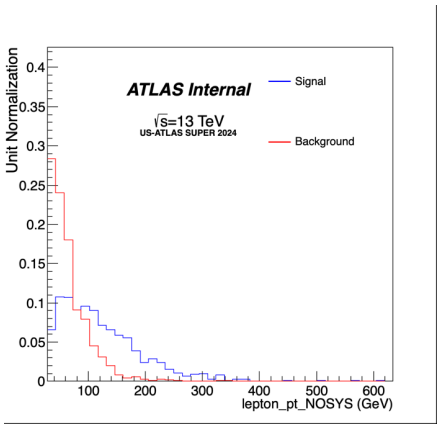


Transverse momentum for ljet3

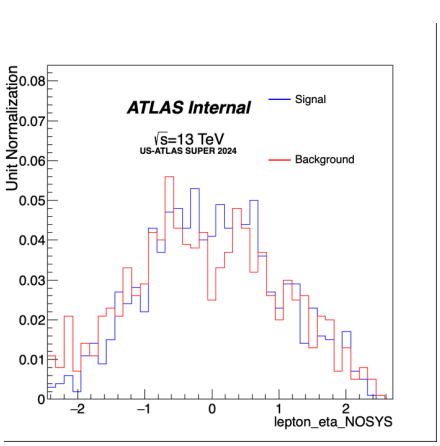


Eta measurement for ljet3

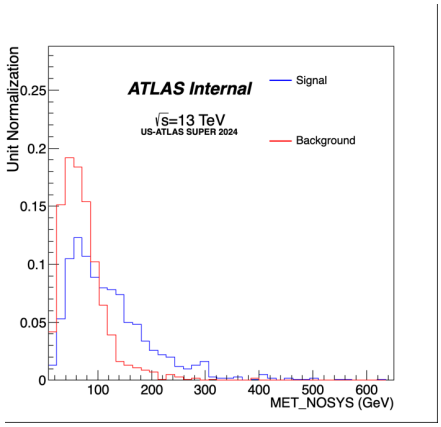
Figure 4.4: ljet input variables



Transverse momentum for lepton



Eta measurement for lepton



Missing Transverse Energy

Figure 4.5: Lepton and Missing Transverse Energy variables

Figures 4.3, 4.4, 4.5 show the input variables used in our analysis using Machine Learning techniques.

Transverse Momentum (p_T)

The transverse momentum distributions show distinct patterns for signal and background. The lepton p_T distribution reveals that the signal peaks at higher values compared to the background, suggesting that signal events tend to produce more energetic leptons. For the leading b-jet ($bjet1$) p_T, the signal distribution extends to higher values, indicating that signal events often contain more energetic b-jets. Similar trends are observed for light jets ($ljet1$, $ljet2$), with signal events showing harder p_T spectra.

Pseudorapidity (η)

The η distributions provide insights into the angular properties of particles. For lepton η, both signal and background show central distributions, but the signal appears slightly more central. The distributions for b-jets and light jets are generally symmetric around $\eta = 0$, with some variations between signal and background.

Mass Distributions

The mass distributions of b-jets offer additional discriminating power. The leading b-jet ($bjet1$) mass shows a broader distribution for the signal compared to the background, potentially indicating different decay processes. Similar trends are observed for the sub-leading b-jet ($bjet2$) mass, but with lower overall mass values.

Missing Transverse Energy (MET)

The MET_NOSYS distribution shows a clear separation between signal and background. Signal events tend to have higher MET values, suggesting the presence of undetected particles (e.g., neutrinos or dark matter candidates). Background events are more concentrated at lower MET values.

4.3.2 Correlation Analysis

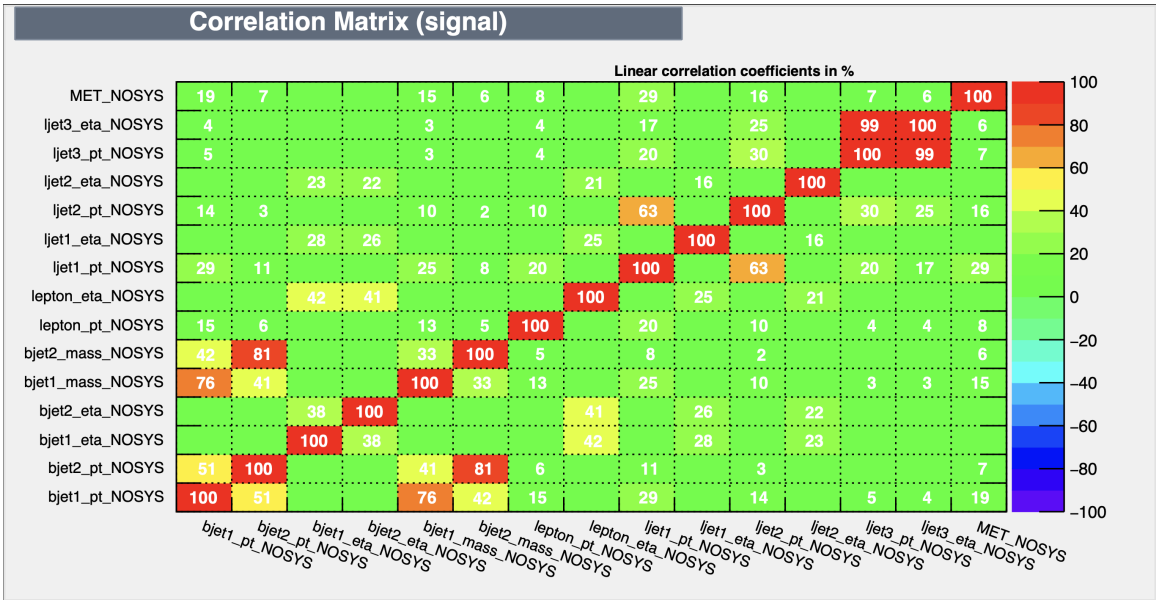


Figure 4.6: Correlation Matrix for Signal events

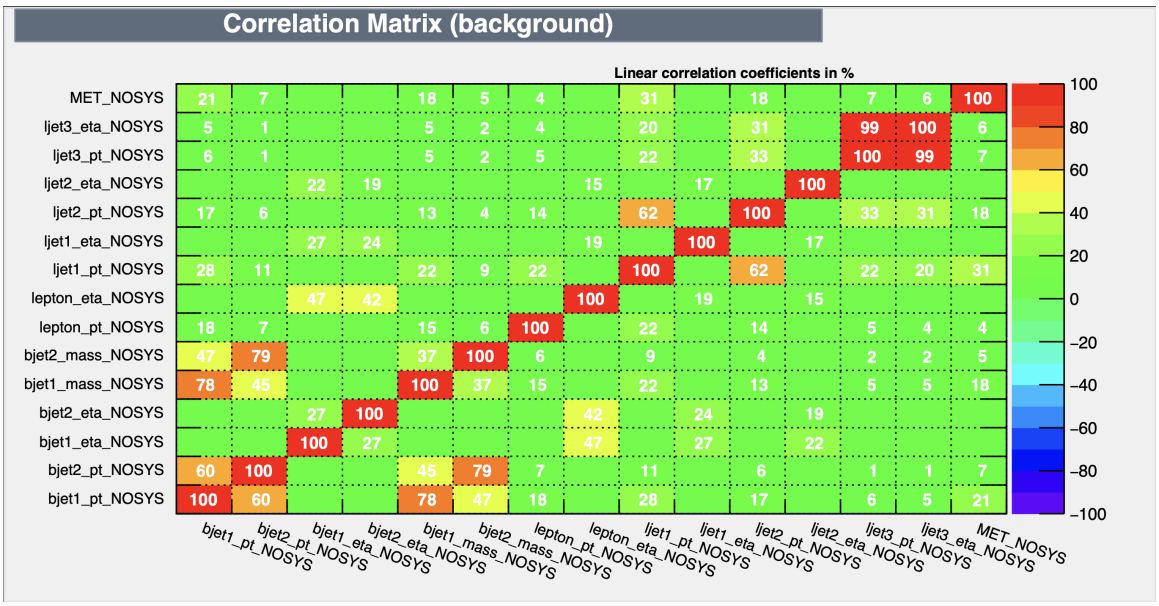


Figure 4.7: Correlation Matrix for Background events

The correlation matrices presented in Figures 4.6 and 4.7 reveal crucial patterns that inform our classification strategy. These correlations not only provide insight into the underlying physics but also guide the development of machine learning techniques for signal-background discrimination.

The correlation structure exhibits several notable characteristics that directly impact the development of classification algorithms. Most prominently, we observe a near-linear correlation between the third light jet's pseudorapidity and transverse momentum (ljet3_eta_NOSYS and ljet3_pt_NOSYS) in both signal and background distributions.

The b-jet sector presents particularly interesting features for classification. The correlation between the transverse momenta of the two b-jets (bjet1_pt_NOSYS and bjet2_pt_NOSYS) differs notably between background (60%) and signal (51%) events. This difference in correlation structure provides valuable discriminating power, suggesting that the relationship between b-jet kinematics, rather than individual variables alone, should be exploited in our classification approach.

The correlation analysis informs our feature engineering approach through several key observations. While traditional approaches might suggest removing highly correlated variables, the subtle differences between signal and background correlation patterns argue for more sophisticated dimensionality reduction techniques. Principal Component Analysis can be employed to create decorrelated features while preserving the discriminating information contained in the correlation differences.

The moderate correlations observed between different physics objects suggest the utility of engineered features that combine information from multiple objects. These composite features can capture higher-order correlations that are not immediately apparent in the linear correlation matrices. Angular separations and invariant masses constructed from multiple objects often provide powerful discriminating variables that complement the basic kinematic quantities.

4.4 Classifier Performance

4.4.1 BDT Analysis

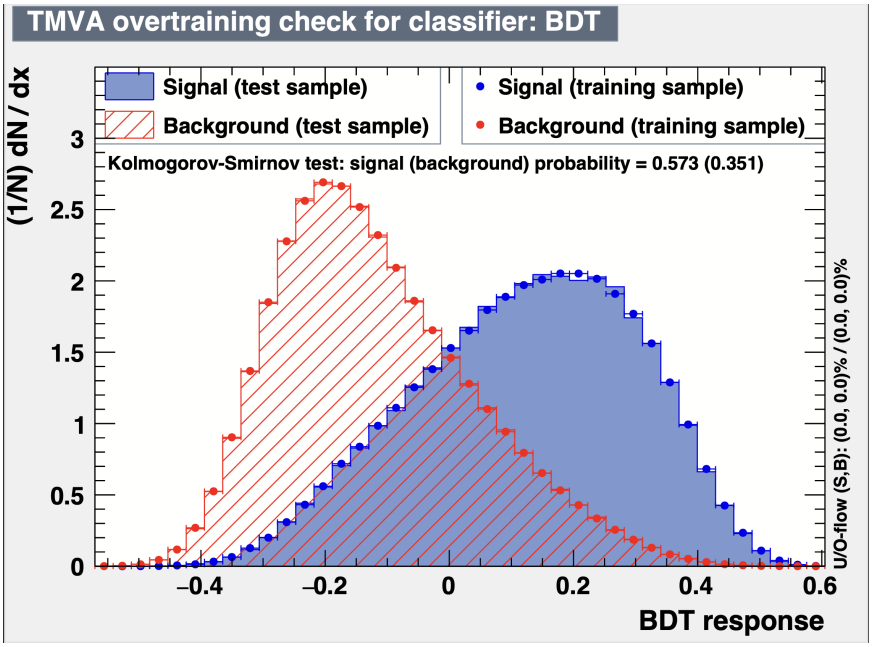


Figure 4.8: BDT Performance

This plot is used for assessing whether BDT classifier has learned generalizable patterns or whether it has become too tailored to the training data, which indicated over-training.

The x-axis of the plot represents the BDT response, which corresponds to the score produced by the classifier. Positive values of the BDT response suggests a higher likelihood that a sample is a signal event, while a negative value suggests a higher likelihood that a sample is a background event. The y-axis represents a normalized distribution of events, which allows for direct comparison between the test and training distributions by scaling the number of events in each bins.

Upon analyzing the plot, we see that there is a strong agreement between the training and test distributions for both signal and background samples. The training distributions closely follow the test distributions for each class, which suggests that the classifier is not overfitting. We also use the Kolmogorov-Smirnov (KS) test. It provides a probability value that measures the similarity between the training and test distributions. The KS values for for signal is 0.573 and for background is 0.351, well over the threshold of 0.05,suggesting no significant statistical difference between the training and test distributions for either class.

4.4.2 LikelihoodMIX Analysis

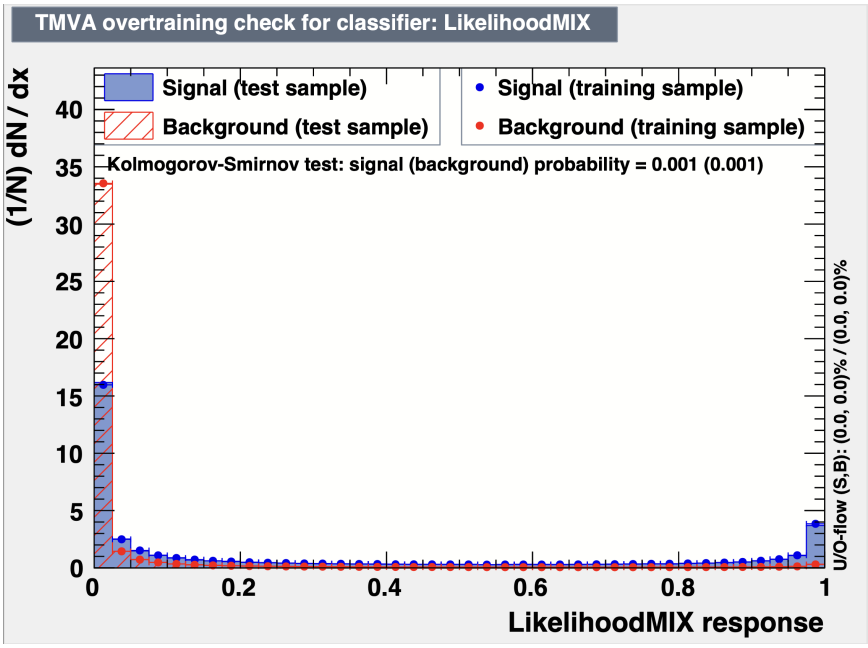


Figure 4.9: LikelihoodMIX Performance

The x-axis of the plot represents the response of the LikelihoodMIX classifier, where values near 0 indicate a background-like classification and values near 1 suggest a signal-like classification. The y-axis displays a normalized event distribution, allowing for an even comparison between the training and test datasets.

The distribution shapes in this plot are highly concentrated near 0 for both signal and background classes, with minimal distribution extending towards higher LikelihoodMIX response values. This concentration indicates that the classifier has not effectively separated signal from background samples, as both are clustered at the lower end of the response spectrum. Ideally, a well-performing classifier should show clear separation between signal and background distributions. The lack of such separation here suggests that the classifier may not be distinguishing between these classes effectively.

A critical indication of potential overtraining comes from the Kolmogorov-Smirnov (KS) test results displayed on the plot. The KS probabilities for both signal and background are extremely low (0.001), well below the standard significance threshold of 0.05. These low KS probabilities imply a significant difference between the training and test distributions for both classes, signaling a potential overfitting issue. Such a discrepancy suggests that the classifier may be performing well on the training data but struggles to generalize to new, unseen test data.

4.5 Background Rejection vs Signal Efficiency

4.5.1 ROC curve for BDT

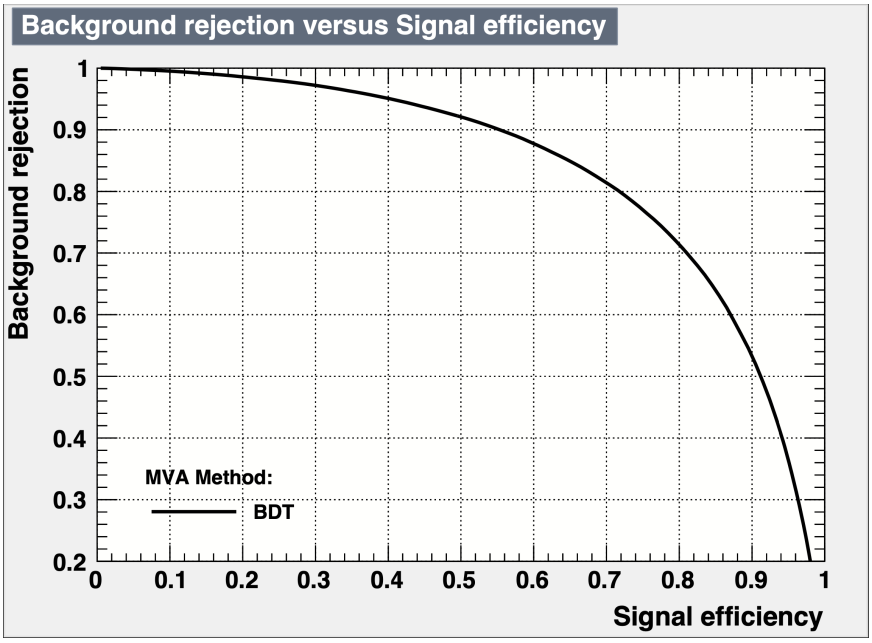


Figure 4.10: ROC Curve for BDT

This ROC curve highlights BDT's effectiveness in separating potential signal events from the dominant $t\bar{t}$ background. We see that BDT achieves excellent background rejection (above 0.9) for signal efficiencies up to approximately 0.5, a critical range for isolating BSM Higgs signatures from top quark pair production events. Beyond this point, a trade-off becomes evident, as increasing signal acceptance results in a more rapid decline in background rejection.

At a signal efficiency of 0.8 BDT still maintains a background rejection of about 0.7. This indicates the classifier's capability to significantly suppress the $t\bar{t}$ background while retaining a substantial portion of potential BSM Higgs events in the 1-lepton $bbWW$ final state. Such performance is crucial for maximizing the sensitivity of searches within the ATLAS experiment at the Large Hadron Collider, where robust background discrimination is key to identifying any potential BSM Higgs boson signals.

4.5.2 ROC curve for LikelihoodMIX

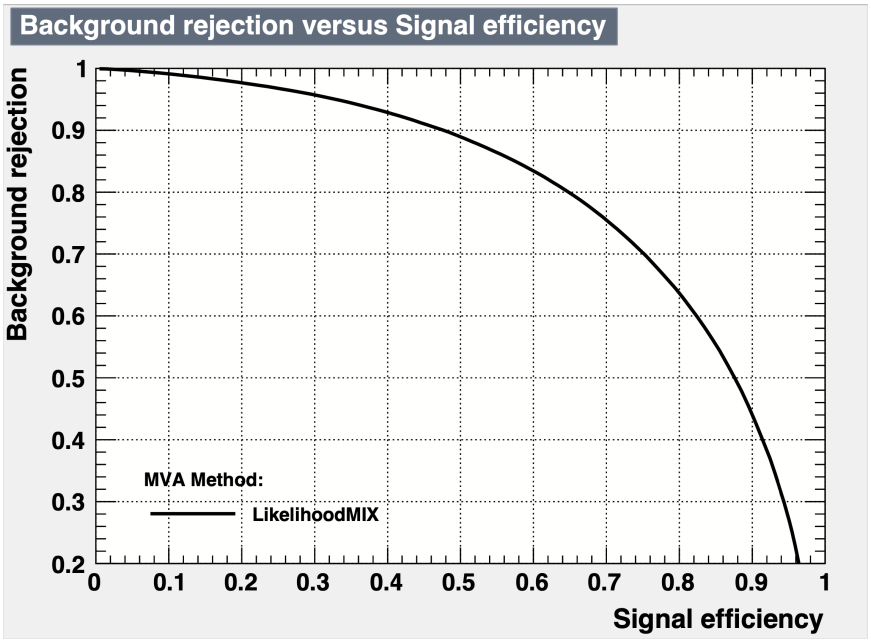


Figure 4.11: ROC Curve for LikelihoodMIX

This performance curve illustrates the LikelihoodMIX method's capability in discriminating signal events from $t\bar{t}$ background. The method demonstrates robust background rejection characteristics, maintaining a rejection rate above 0.95 for signal efficiencies up to 0.3. We a gradual decline in background rejection as signal efficiency increases, with a notable inflection point around 0.5 signal efficiency, where the background rejection begins to deteriorate more rapidly. At a signal efficiency of 0.8, the LikelihoodMIX method achieves approximately 0.7 background rejection, indicating its effectiveness in preserving potential BSM Higgs signals while maintaining substantial background suppression.

4.6 Analysis of Multivariate Discriminant Methods

4.6.1 Maximum Likelihood Estimation Analysis

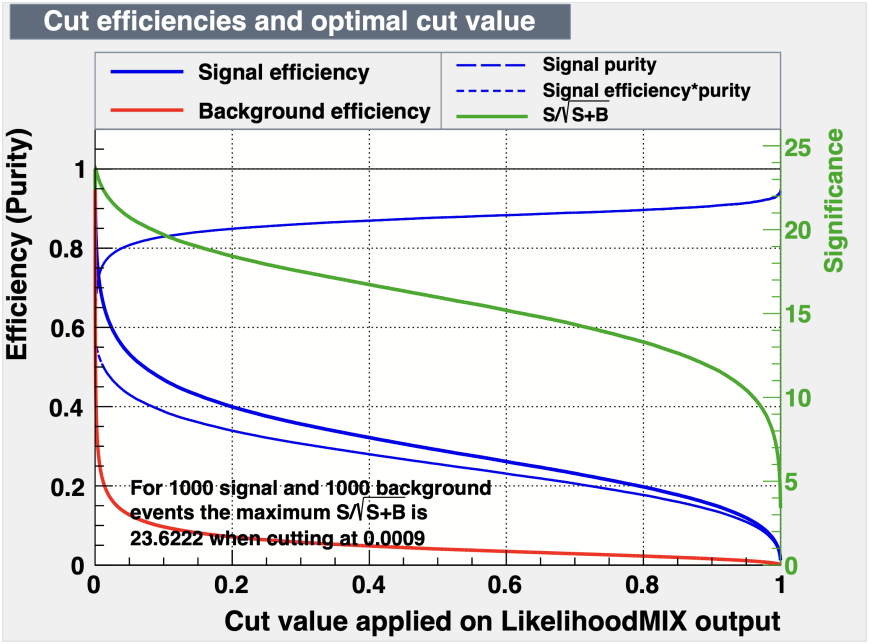


Figure 4.12: LikelihoodMIX Efficiency

The LikelihoodMIX discriminant demonstrates characteristic behavior that reflects the underlying probability density estimation approach. When analyzing the cut efficiency curves across the output range of [0,1], the signal efficiency shows a gradual but consistent improvement, starting at approximately 0.8 and climbing to 0.9 across the range. This behavior indicates the method's ability to maintain strong signal retention even as the selection criteria become more stringent.

The background rejection capabilities of the LikelihoodMIX method are particularly noteworthy. The background efficiency curve exhibits a sharp initial decline, dropping precipitously in the first 20% of the cut range. After this point, the background efficiency plateaus at approximately 0.05, indicating highly effective background suppression. This rapid background rejection while maintaining reasonable signal efficiency is a key strength of the likelihood-based approach.

The signal purity, represented in Figure 4.12, shows steady improvement as stricter cuts are applied, demonstrating the method's ability to progressively isolate the signal component. The optimal cut value for the LikelihoodMIX output was determined to be 0.0009, achieving a maximum significance ($S/\sqrt{S+B}$) of 23.6222. This optimization point represents the best compromise between signal retention and background suppression for the given sample of 1000 signal and 1000 background events.

4.6.2 Boosted Decision Tree Analysis

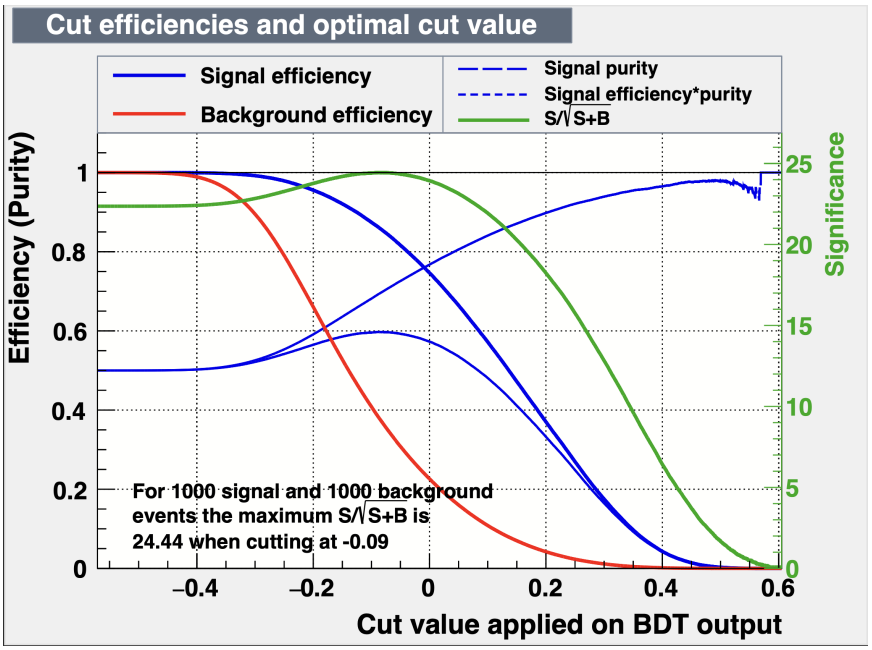


Figure 4.13: BDT Efficiency

The Boosted Decision Tree (BDT) approach reveals substantially different characteristics in its discrimination performance compared to the likelihood-based method. The BDT output, ranging from [-0.5,0.6], shows remarkable signal retention capabilities across a broad range of cut values. The signal efficiency maintains near-unity performance until approximately -0.2, indicating superior signal preservation in the initial cutting regime.

One of the most striking features of the BDT performance, shown in Figure 4.13, is the symmetrical nature of its efficiency curves around the optimal cut point. The background efficiency demonstrates a more gradual decline compared to the LikelihoodMIX method, suggesting a more nuanced separation of signal and background events. This behavior results in a more stable operating region around the optimal cut value, which could prove advantageous when considering systematic uncertainties in a final analysis.

The BDT achieves its maximum significance of 24.44 at a cut value of -0.09, surpassing the performance of the LikelihoodMIX method. This superior performance is achieved through a fundamentally different approach to event discrimination, where the recursive binary splits of the decision tree structure appear to better capture the multidimensional correlations in the input variables. The higher significance value, combined with better signal retention, suggests that the BDT provides more optimal separation between signal and background events for this particular dataset.

The efficiency curves for the BDT also exhibit a more controlled descent in both signal and back-

ground efficiencies after the optimal cut point, providing a wider range of viable cut values that maintain good discrimination power. This characteristic offers greater flexibility in final cut selection, allowing for adjustments based on specific analysis requirements without severe penalties in performance.

4.6.3 Comparative Performance

The distinct behaviors of these two multivariate methods illustrate their underlying mathematical approaches to event classification. The LikelihoodMIX method excels at rapid background rejection but sacrifices some signal efficiency to achieve this. In contrast, the BDT maintains high signal efficiency while achieving comparable or better background rejection through its hierarchical decision structure. The BDT's superior maximum significance value, coupled with its more robust performance across a range of cut values, suggests it may be the more suitable choice for this particular analysis.

The location of optimal cut values (-0.09 for BDT versus 0.0009 for LikelihoodMIX) reflects the fundamental differences in how these methods transform the input variable space. The BDT's negative optimal cut point indicates that its discrimination power benefits from retaining events that would be classified as slightly background-like by a simpler discriminant, likely due to its ability to identify complex patterns in the multidimensional feature space.

This can be quantitatively summarized in Table 4.1.

Table 4.1: Performance comparison of multivariate methods

Metric	LikelihoodMIX	BDT
Maximum Significance	23.6222	24.44
Optimal Cut Value	0.0009	-0.09
Signal Efficiency at Optimal Cut	$\sim$0.85	$\sim$0.95
Background Efficiency at Optimal Cut	$\sim$0.15	$\sim$0.20

Chapter 5

Decision Trees

5.1 Introduction

A decision tree is a hierarchical model that supports decision-making by visually representing decisions and their possible consequences. These consequences include chance event outcomes, resource costs, and utility. Decision trees are employed in operations research as a tool to help pinpoint the most effective strategy for achieving a goal. They are widely used in machine learning. One of their major applications is to visually represent algorithms that solely use conditional control statements.

5.2 Decision Tree Structure

Decision trees have a structure similar to flowcharts. In this structure **Internal nodes** represent "tests" on an attribute. Attributes are data elements used in a model. For instance, an internal node could represent a test on the outcome of a coin flip, determining whether it lands on heads or tails. Each branch stemming from an internal node symbolizes the result of the test represented by that node. Each **leaf node** represents a class label. A class here is a category that a model predicts for an input data. This label signifies that a decision has been reached after evaluating all attributes.

Classification rules are represented as the paths connecting the root node to the leaf node. These paths illustrate the decision-making logic encoded within the tree. All decision trees have three primary types of nodes. **Decision nodes** depicted as squares that represent points where a decision needs to be made. A **chance node** depicted as a circle that signify points where an uncertain event/chance occurrence influences outcomes, and **end nodes** depicted by rectangles that indicate the final outcome of a particular

decision path.

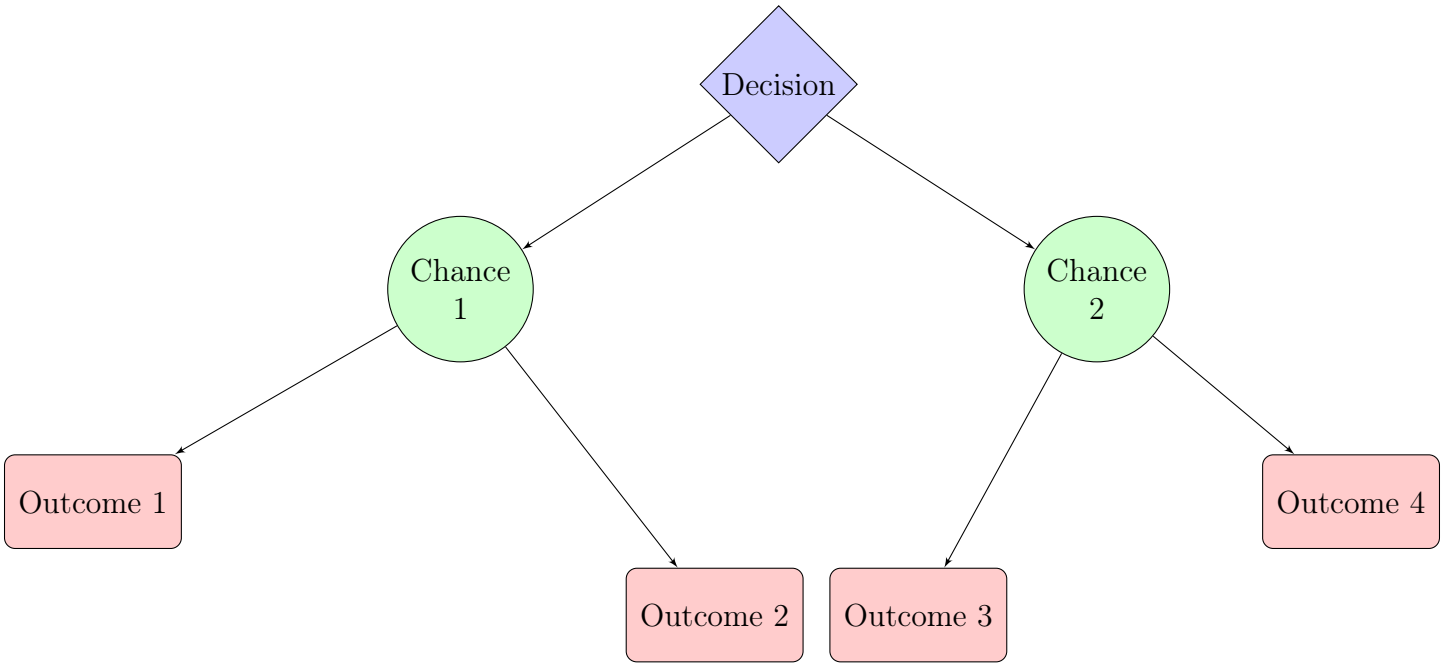


Figure 5.1: A Simple Decision Tree

Figure 5.1 illustrates a simple decision tree with one decision node, two chance nodes, and four possible outcomes.

5.3 Mathematical Foundations of Decision Trees

The foundations of decision trees are rooted in several core areas in mathematics, including but not limited to information theory, probability, statistics and optimization. Discussed below are some common metrics that stem from the theories mentioned above.

5.3.1 Entropy, Information Gain and Gini Impurity

The physics definition of entropy we are all familiar with that entropy is the measure of uncertainty or disorder of a system. In the context of decision tree classification, it assesses the impurity in the dataset. Impurity of a dataset is a measure of how mixed up or heterogeneous the data is. A higher entropy indicates more randomness/uncertainty about the class labels.

Before we introduce the mathematical definition of entropy, there are some prerequisite terms that need to be established.

Definition 1. *The **proportion of instances**, denoted p_i is defined as*

$$p_i = \frac{|S_i|}{|S|}.$$

This proportion quantifies how many of the total instances in a dataset S belong to a particular class i. Here $|S_i|$ is the ordinality/number of instances in the subset S_i that belong to class i, and $|S|$ is the total number of instances in the dataset S. The value of p_i ranges from 0to1. A higher value indicates a larger proportion of the dataset belongs to class i.

Example 1. For a dataset of 100 datasets, if 30 of them belong to class i, then $p_i = \frac{|S_i|}{|S|} = \frac{30}{100} = 0.3$ or 30%.

Definition 2. *The **number of distinct classes** present in a dataset S is represented as $c(s)$. Each class represents a unique category or label that the instance on the dataset can belong to.*

Now, equipped with the above definitions, we can talk about entropy in the mathematical sense.

Definition 3. ***Entropy** of a dataset S, denoted by $H(S)$ is defined as:*

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

where p_i denotes the proportion of instances in class i in the dataset S, and $c(s)$ refers to the number of distinct classes in the dataset.

The value of $p_i \log_2(p_i)$ measures the uncertainty associated with class i, and summing over all classes gives the total entropy. The proportion p_i is crucial for calculating entropy $H(S)$. In the entropy formula, it reflects the uncertainty associated with the distribution of classes in the dataset. If p_i is equal across all classes, the entropy will be higher, indicating greater uncertainty. Conversely, if one class dominates (e.g., p_i is close to 1 for one class), the entropy will be lower, indicating more certainty about class membership.

The value of c is important for entropy calculations because it determines the scope of the summation in the entropy formula $H(S)$. Each class i contributes to the overall entropy, so knowing how many classes there are helps in understanding the complexity of the classification problem. More distinct classes (higher c) typically lead to greater uncertainty in the dataset, as there are more possible categories that an instance could belong to. Conversely, a dataset with fewer classes (lower c) might have lower entropy, especially if the instances are unevenly distributed among the classes.

Definition 4. ***Splitting*** *a dataset on an attribute A is defined as the process of partitioning the dataset into distinct subsets based on the values of A. Mathematically,*

$$S = \bigcup_{i=1}^{n} A_i$$

where $A_i \cap A_k \neq 0$ for $i \neq k$.

This technique is fundamental in decision tree learning, where splits are chosen to maximize the separation or "purity" of classes within the subsets, with the goal of increasing homogeneity in terms of a target variable.

Definition 5. ***Conditional Entropy****, denoted as $H(S|A)$ represents the remaining uncertainty after knowing the value of A, where A is an attribute on which the dataset is split. It is defined as:*

$$H(S|A) = \sum_{i}^{c} p_i H(S_i)$$

This formula represents the weighted sum of the entropies of each subset S_i resulting from the split, where $|S_i|$ denotes the size (number of instances) of subset S_i, created by splitting S based on attribute A, and $|S|$ is the total number of instances in the dataset S. Each subset S_i contributes to the overall conditional entropy based on its size relative to the entire dataset. Larger subsets have more influence on the conditional entropy.

With these definitions set, we can talk about Information Gain.

Definition 6. ***Information Gain (IG)*** *is defined as the difference between the entropy of the original dataset S and the entropy after splitting the dataset by a given attribute A:*

$$IG(A) = H(S) - H(S|A)$$

where $IG(A)$ represents the information gain from splitting the dataset S based on attribute A. $H(S)$ denotes the entropy of the original dataset S, which quantifies the uncertainty in the dataset before any split. $H(S|A)$ represents the conditional entropy, or the weighted sum of the entropies of the subsets created by splitting S based on A.

This measures the uncertainty that remains after splitting on A. Information gain (IG) is a metric used in decision trees and other algorithms to determine how well a feature (or attribute) splits a dataset. It's

a measure of how much "information" is gained about the target variable after observing the feature. The idea is to reduce uncertainty (measured by entropy) by selecting attributes that provide the most useful information for classification.

Another important metric used in decision tree algorithms to measure impurity of a dataset is the Gini Impurity.

Definition 7. ***Gini Impurity*** *is defined as:*

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

where $Gini(S)$ represents the Gini impurity of the dataset S, p_i denotes the proportion of instances in class i within the dataset, and c refers to the total number of distinct classes present in the dataset.

It is a measure of how often a randomly chosen element from the dataset would be incorrectly classified if it were randomly labeled according to the distribution of class labels in a dataset. It quantifies the level of uncertainty/impurity in a group of data points, and is used in decision trees to decide where to split the data.

Example 2. To understand how Gini impurity works, consider a dataset with three classes: A, B, and C. If you calculate the Gini impurity for this dataset, you would first find the proportion of instances in each class. For example, if the dataset contains 50 instances, with 30 belonging to class A, 10 to class B, and 10 to class C, the proportions would be calculated as follows: $p_A = \frac{30}{50} = 0.6$, $p_B = \frac{10}{50} = 0.2$, and $p_C = \frac{10}{50} = 0.2$.

You can then compute the Gini impurity as follows:

$$Gini(S) = 1 - (p_A^2 + p_B^2 + p_C^2) = 1 - (0.6^2 + 0.2^2 + 0.2^2) = 1 - (0.36 + 0.04 + 0.04) = 1 - 0.44 = 0.56$$

In this case, a Gini impurity of 0.56 suggests that the dataset is relatively impure, meaning there is a mix of classes with no single class overwhelmingly dominant. In practice, this level of impurity may prompt a decision tree algorithm to further split the dataset to achieve clearer separation between classes and reduce impurity.

The goal when building decision trees is to minimize the Gini impurity at each split. When evaluating potential splits, the algorithm calculates the Gini impurity for each subset created by the split and takes a weighted average based on the number of instances in each subset. The split that results in the lowest weighted Gini impurity is chosen, as it indicates a more homogeneous grouping of instances.

5.4 Decision Rules

It is possible to express the information contained in a decision tree in a linearized format known as decision rules. These rules provide a condensed representation of the decision-making logic. In these rules, the outcome specified by a particular rule corresponds to the content of the leaf node it leads to. The conditions along the path from the root node to that leaf node are linked together using the conjunction "and" in the "if" clause of the rule. The general format of these decision rules is:

if condition1 and condition2 and condition3 then outcome

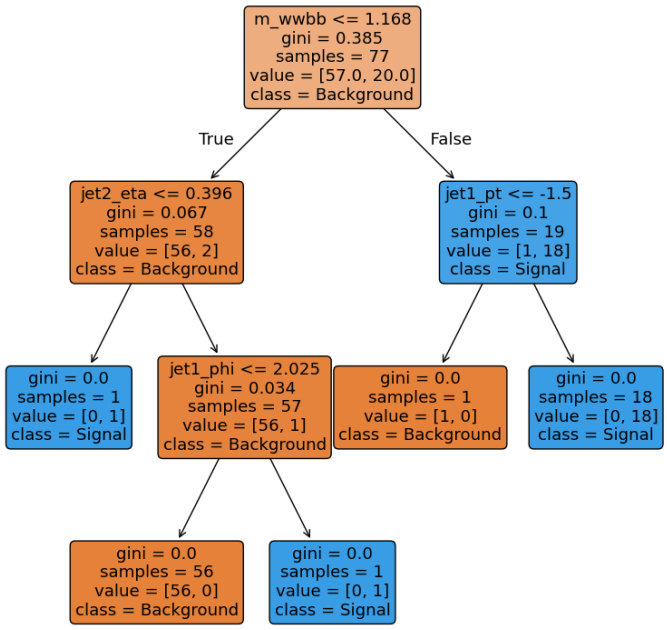


Figure 5.2: A hypothetical decision tree for Higgs Boson searches

In the context of Higgs boson searches, decision rules can be applied to classify events in a particle physics experiment as either a **Signal** (Higgs boson event) or **Background** event based on various features. The features used in the decision tree are `m_wwbb`, `jet2_eta`, `jet1_phi`, and `jet1_pt`.

The conditions derived from this hypothetical decision tree might be as follows: `m_wwbb` ≤ 1.168, `jet2_eta` ≤ 0.396, and `jet1_phi` ≤ 2.025. The corresponding decision rule for classifying an event as a **Signal** event could be stated as:

if `m_wwbb` ≤ 1.168 and `jet2_eta` ≤ 0.396 and jet1_phi ≤ 2.025 then classify as Signal

These decision rules can be generated by constructing association rules, with the target variable (Signal or Background) placed on the right side of the rule. They can also be utilized to represent relationships between features and classification outcomes in your decision tree.

5.5 Adaptive Boosting

Adaptive Boosting, often referred to as AdaBoost, was introduced by Freund and Schapire in 1997 as a powerful ensemble method that enhances the predictive capability of weak learners to create a robust classifier. By iteratively adjusting weights on misclassified instances, AdaBoost is particularly effective in classification tasks involving imbalanced or complex datasets, such as identifying signal events in particle physics experiments. The AdaBoost algorithm operates by sequentially training weak learners, adjusting instance weights to focus on previously misclassified examples. This section provides formal definitions for each step of the algorithm.

5.5.1 Initial Weighting of Instances

Let n represent the number of training instances, and let each instance (x_i, y_i) consist of a feature vector x_i and a label $y_i \in \{-1, 1\}$. At the outset, all instances are assigned an equal weight:

$$w_i^{(1)} = \frac{1}{n}, \quad \text{for } i = 1, 2, \ldots, n. \tag{5.1}$$

This uniform weighting assumes that each instance initially contributes equally to the model.

5.5.2 Calculating Weighted Error

Once a weak learner $h_t(x)$ is trained, its performance is evaluated by calculating the *weighted error rate* ϵ_t, defined as:

$$\epsilon_t = \frac{\sum_{i=1}^{n} w_i^{(t)} \cdot \mathbb{I}(h_t(x_i) \neq y_i)}{\sum_{i=1}^{n} w_i^{(t)}}, \tag{5.2}$$

where $\mathbb{I}(h_t(x_i) \neq y_i)$ is an indicator function that equals 1 if $h_t(x_i) \neq y_i$ (indicating misclassification) and 0 otherwise. Thus, ϵ_t reflects the proportion of incorrectly classified instances, weighted by their respective weights.

5.5.3 Training the Weak Learners

For each iteration $t = 1, 2, \ldots, T$, a weak learner $h_t(x)$ is trained on the weighted dataset. Here, a *weak learner* is defined as a model that performs only slightly better than random guessing.

Definition 8. *A **weak learner** is a classifier with an error rate $\epsilon_t < 0.5$ over the distribution of training data.*

Example 3. A decision stump, a decision tree with only a single split, can serve as a weak learner. For instance, in a dataset of particles classified by energy levels, a decision stump might classify particles based on whether their energy level exceeds a certain threshold.

5.5.4 Assigning Learner Weight

In AdaBoost, each weak learner's contribution to the final model is determined by a weight coefficient that reflects its accuracy.

Definition 9. *For a weak learner $h_t(x)$ with error rate ϵ_t, its weight α_t is defined as:*

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right).$$

This weight quantifies the learner's reliability: a lower error rate ϵ_t yields a higher weight α_t, giving accurate learners more influence in the ensemble's decisions.

Example 4. Let us examine two weak learners with different error rates. When h_1 has $\epsilon_1 = 0.2$, its weight becomes $\alpha_1 \approx 0.693$. In contrast, when h_2 has $\epsilon_2 = 0.4$, its weight is only $\alpha_2 \approx 0.201$. The more accurate learner h_1 receives more than triple the weight of h_2, demonstrating how AdaBoost favors more accurate weak learners.

5.5.5 Updating Instance Weights

After each iteration, AdaBoost adjusts the training instance weights to focus subsequent learners on the challenging cases.

Definition 10. *The weight $w_i^{(t)}$ of instance i at iteration t is updated according to:*

$$w_i^{(t+1)} = w_i^{(t)} \cdot e^{-\alpha_t y_i h_t(x_i)}$$

where y_i is the true label and $h_t(x_i)$ is the predicted label.

Example 5. Consider a weak learner with weight $\alpha_t = 0.5$. For a correctly classified instance where $y_i h_t(x_i) = 1$, the weight update becomes:

$$w_i^{(t+1)} = w_i^{(t)} \cdot e^{-0.5} \approx 0.607 w_i^{(t)}$$

In contrast, for a misclassified instance where $y_i h_t(x_i) = -1$, the update yields:

$$w_i^{(t+1)} = w_i^{(t)} \cdot e^{0.5} \approx 1.649 w_i^{(t)}$$

This calculation demonstrates how misclassified instances receive significantly higher weights for the next iteration.

The weights are then normalized to sum to one:

$$w_i^{(t+1)} = \frac{w_i^{(t+1)}}{\sum_{j=1}^{n} w_j^{(t+1)}}, \quad \text{for } i = 1, 2, \ldots, n$$

5.5.6 Constructing the Strong Classifier

The final step combines all weak learners into a robust ensemble classifier.

Definition 11. *After T iterations, the strong classifier $H(x)$ is defined as:*

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

where sign(z) returns 1 if $z > 0$ and -1 otherwise.

Example 6. Consider three weak learners with weights $\alpha_1 = 0.7$, $\alpha_2 = 0.3$, $\alpha_3 = 0.4$ making predictions $h_1(x) = 1$, $h_2(x) = -1$, $h_3(x) = 1$ for an instance x. The weighted sum of their predictions is:

$$\sum_{t=1}^{3} \alpha_t h_t(x) = 0.7(1) + 0.3(-1) + 0.4(1) = 0.8$$

Since $0.8 > 0$, the strong classifier outputs $H(x) = 1$.

5.6 Application in High-Energy Physics

In high-energy physics, such as data analysis at the Large Hadron Collider (LHC), AdaBoost is valuable for separating signal events from background noise. The `ROOT` framework, specifically the Toolkit for Multivariate Analysis (TMVA), includes an implementation of AdaBoost, making it accessible for analyzing particle collision events.

Example 7. In experiments seeking evidence for the Higgs boson, AdaBoost can classify collision events as signal (Higgs-like) or background (non-Higgs). By iteratively focusing on events misclassified in previous stages, AdaBoost enhances the model's sensitivity to subtle signals amidst noisy data.

Chapter 6

Maximum Likelihood Estimation

Definition 12. ***Maximum Likelihood Estimation (MLE)*** *is a statistical method used to estimate the parameters of a model by maximizing the likelihood function. Given a set of observations, MLE finds the parameter values that make the observed data most probable.*

To apply MLE, we first construct a likelihood function, which measures how likely the observed data is for various parameter values. The parameters that maximize this function are considered the best estimates.

6.1 Likelihood Function

Definition 13. *Given a set of observations $X = (x_1, x_2, \ldots, x_n)$ that are independently and identically distributed (i.i.d.) and a parameter θ to be estimated, the* ***likelihood function*** *$L(\theta|X)$ is defined as:*

$$L(\theta|X) = P(X|\theta) = \prod_{i=1}^{n} P(x_i|\theta),$$

where $P(X|\theta)$ represents the probability of observing the data X given θ. This product form arises because the observations are independent.

Example 8. Suppose we observe data points $X = (x_1, x_2, x_3)$ from a distribution parameterized by θ. If the observations are independent, the likelihood function would be the product:

$$L(\theta|X) = P(x_1|\theta) \cdot P(x_2|\theta) \cdot P(x_3|\theta).$$

This function gives a measure of how likely it is to observe X under different values of θ.

The goal of MLE is to find the parameter value θ that maximizes $L(\theta|X)$.

6.2 Log-Likelihood Function

Definition 14. *The **log-likelihood function** $\ell(\theta|X)$ is defined as the natural logarithm of the likelihood function:*

$$\ell(\theta|X) = \log L(\theta|X) = \sum_{i=1}^{n} \log P(x_i|\theta).$$

Maximizing the log-likelihood function is equivalent to maximizing the likelihood function, due to the monotonicity of the logarithm function.

Example 9. Continuing from the previous example, if we have three observations and the likelihood function is:

$$L(\theta|X) = P(x_1|\theta) \cdot P(x_2|\theta) \cdot P(x_3|\theta),$$

the log-likelihood function becomes:

$$\ell(\theta|X) = \log P(x_1|\theta) + \log P(x_2|\theta) + \log P(x_3|\theta).$$

Working with $\ell(\theta|X)$ simplifies the optimization since products become sums.

6.3 Theoretical Properties of MLE

MLE has several key properties: consistency, efficiency, and asymptotic normality.

6.3.1 Consistency

Definition 15. *An **estimator** $\hat{\theta}$ is **consistent** if it converges to the true parameter θ_0 as the sample size n approaches infinity. In the context of MLE, this means that as more data is observed, the estimate $\hat{\theta}$ approaches the actual value of θ_0.*

Example 10. Suppose the true parameter $\theta_0 = 2$ and we use MLE on increasingly large datasets to estimate it. As the sample size n grows, a consistent estimator would yield $\hat{\theta}$ values that converge to 2.

6.3.2 Efficiency

Definition 16. *An estimator is **efficient** if it achieves the lowest possible variance among all unbiased estimators. MLE is asymptotically efficient, meaning it has the minimum possible variance for large sample sizes.*

This efficiency is related to the **Cramér-Rao lower bound (CRLB)**, which provides a lower limit on the variance of unbiased estimators. MLE asymptotically attains the CRLB, making it a desirable estimator in large samples.

6.3.3 Asymptotic Normality

Definition 17. *MLE is said to be **asymptotically normally distributed** if, as n becomes large, the distribution of the estimator $\hat{\theta}$ approaches a normal distribution with mean θ_0 and variance equal to the inverse of the Fisher Information $I(\theta_0)$.*

$$\sqrt{n}(\hat{\theta} - \theta_0) \xrightarrow{d} N(0, I(\theta_0)^{-1}). \tag{6.1}$$

Example 11. If $\theta_0 = 5$ and $I(\theta_0) = 4$, then for large n, the MLE $\hat{\theta}$ would approximately follow a normal distribution with mean 5 and variance $\frac{1}{4} = 0.25$.

6.4 Finding the Maximum Likelihood Estimate

To find $\hat{\theta}$, we take the derivative of the log-likelihood function $\ell(\theta|X)$ with respect to θ and set it equal to zero:

$$\frac{d\ell(\theta|X)}{d\theta} = 0. \tag{6.2}$$

Definition 18. *A critical point is a value of θ where $\frac{d\ell(\theta|X)}{d\theta} = 0$. A critical point is a **maximum** if the second derivative is negative at this point.*

Example 12. For an observed dataset $X = (2, 4, 5)$ from a model with parameter θ, suppose we obtain a critical point $\theta = 3$. By checking the second derivative, we can confirm if it corresponds to a maximum.

6.5 Example: Gaussian Distribution

Example 13. Consider $X = (x_1, x_2, \ldots, x_n)$ following a Gaussian distribution with unknown mean μ and variance σ^2. The probability density function is:

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

The likelihood function for X is:

$$L(\mu, \sigma^2|X) = \prod_{i=1}^{n} f(x_i|\mu, \sigma^2).$$

Taking the logarithm yields:

$$\ell(\mu, \sigma^2|X) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log\big(\sigma^2\big) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2.$$

To find $\hat{\mu}$ and $\hat{\sigma}^2$, we take partial derivatives with respect to each parameter and solve.

Estimating the Mean

$$\frac{\partial \ell}{\partial \mu} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i - \mu) = 0 \Rightarrow \hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i.$$

Estimating the Variance

$$\frac{\partial \ell}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{n}(x_i - \mu)^2 = 0 \Rightarrow \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})^2.$$

Thus, the MLE for the mean is the sample mean, and the MLE for variance is the sample variance.

6.6 Maximum Likelihood Estimation Using Kernel Density Estimation

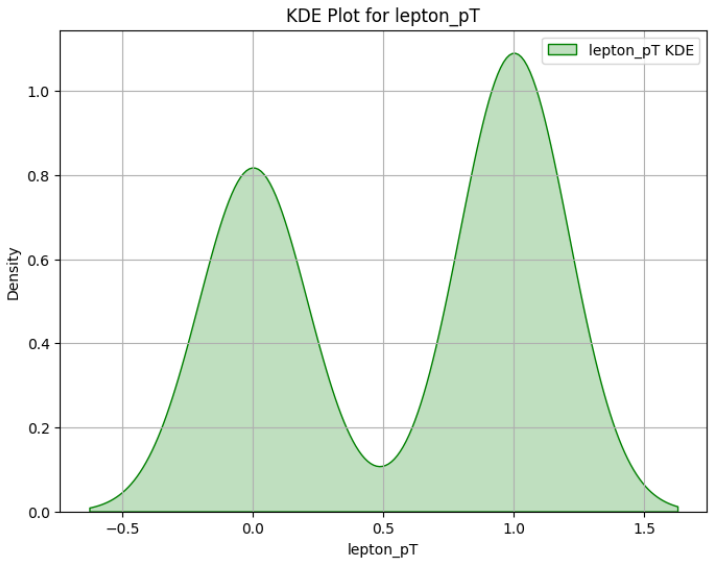


Figure 6.1: A hypothetical MLE using KDE for Higgs Boson searches

Kernel Density Estimation (KDE) is a flexible, non-parametric method for estimating the probability density function (PDF) of a random variable. Unlike parametric approaches, KDE does not assume any specific distributional form, allowing for a more accurate and adaptable representation of the data distribution.

Definition 19. ***Kernel Density Estimation (KDE)*** *is a statistical method for estimating the PDF of a dataset by placing a kernel function K over each observed data point. The* ***bandwidth*** *h controls the smoothness of the resulting density estimate. A small h may lead to overfitting, capturing noise in the data, while a large h may oversmooth, hiding essential features.*

6.6.1 Kernel Density Estimation Definition

Given a set of n i.i.d. observations $X = (x_1, x_2, \ldots, x_n)$, the KDE for the density function $f(x)$ is defined as:

$$f(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right), \tag{6.3}$$

where $K(\cdot)$ is a kernel function, such as the Gaussian or Epanechnikov kernel, and h is the bandwidth parameter.

Example 14. Suppose we have observations $X = (2, 3, 5, 8)$ and use a Gaussian kernel $K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}$. The density estimate $f(x)$ at a point x with a bandwidth $h = 1$ would be:

$$f(x) = \frac{1}{4 \times 1} \sum_{i=1}^{4} \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-x_i)^2}{2}}.$$

This density estimate provides a smooth approximation to the underlying distribution.

6.7 Likelihood Function Using KDE

Definition 20. *The **likelihood function** $L(X|h)$ for observed data $X = (x_1, x_2, \ldots, x_n)$ using KDE is given by:*

$$L(X|h) = \prod_{i=1}^{n} f(x_i) = \prod_{i=1}^{n} \left(\frac{1}{nh} \sum_{j=1}^{n} K\left(\frac{x_i - x_j}{h} \right) \right),$$

where $f(x_i)$ is the density estimate at each observation x_i.

Example 15. For observations $X = (2, 3)$ and Gaussian kernel $K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}$, the likelihood function with $h = 1$ is:

$$L(X|1) = \left(\frac{1}{2} \sum_{j=1}^{2} \frac{1}{\sqrt{2\pi}} e^{-\frac{(2-x_j)^2}{2}} \right) \times \left(\frac{1}{2} \sum_{j=1}^{2} \frac{1}{\sqrt{2\pi}} e^{-\frac{(3-x_j)^2}{2}} \right).$$

The log-likelihood function is typically used for maximization:

$$\ell(h|X) = \log L(X|h) = -n \log(nh) + \sum_{i=1}^{n} \log \left(\sum_{j=1}^{n} K\left(\frac{x_i - x_j}{h} \right) \right). \tag{6.4}$$

6.8 Finding the Optimal Bandwidth

To find the bandwidth h that maximizes the likelihood function, we differentiate the log-likelihood $\ell(h|X)$ with respect to h and set it to zero:

$$\frac{\partial \ell}{\partial h} = -\frac{n}{h} + \sum_{i=1}^{n} \frac{1}{\sum_{j=1}^{n} K\left(\frac{x_i - x_j}{h} \right)} \left(-\frac{1}{h^2} \sum_{j=1}^{n} K'\left(\frac{x_i - x_j}{h} \right) \cdot (x_i - x_j) \right). \tag{6.5}$$

Solving for h yields the optimal bandwidth.

6.9 Likelihood Estimation Using Splines

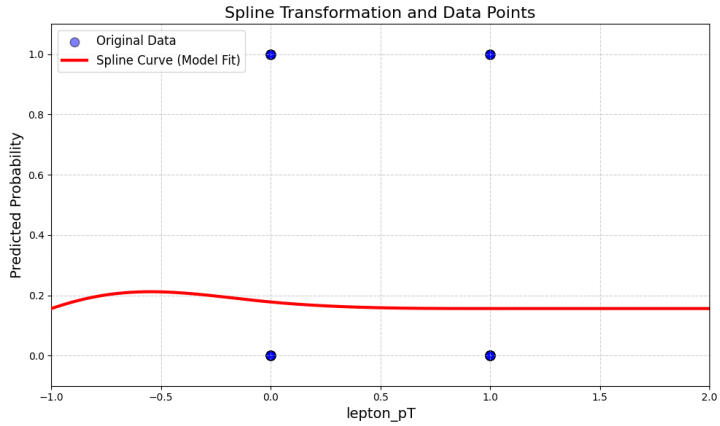


Figure 6.2: A hypothetical MLE using Splines for Higgs Boson searches

Splines are flexible piecewise polynomial functions that can model non-linear relationships in data. They are especially useful in situations where data behavior varies across its range.

Definition 21. *A **spline** is a piecewise polynomial function that is continuous and differentiable at specified points, called knots. A **cubic spline**, for example, is a function constructed from third-degree polynomials joined smoothly at these knots.*

6.9.1 Defining Splines

A spline function $S(x)$ can be written as a sum of basis functions:

$$S(x) = \sum_{i=1}^{k} a_i B_i(x), \tag{6.6}$$

where $B_i(x)$ are the basis functions for the intervals defined by the knots, and a_i are coefficients to be estimated.

Example 16. For three knots at points $x = 1, 2, 3$, a cubic spline function could be:

$$S(x) = a_1 B_1(x) + a_2 B_2(x) + a_3 B_3(x),$$

where B_1, B_2, and B_3 are cubic polynomials defined on the intervals created by these knots.

6.9.2 Likelihood Function Using Splines

Given observations $Y = (y_1, y_2, \ldots, y_n)$ corresponding to inputs $X = (x_1, x_2, \ldots, x_n)$, the likelihood function can be formulated as follows.

Assuming Gaussian noise, the likelihood for spline-based estimation is:

$$L(S|X,Y) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - S(x_i))^2}{2\sigma^2}\right). \tag{6.7}$$

Taking the log-likelihood:

$$\ell(S|X,Y) = -\frac{n}{2}\log\big(2\pi\sigma^2\big) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - S(x_i))^2. \tag{6.8}$$

To estimate the spline coefficients a_i, we maximize this log-likelihood.

6.10 Hybrid Approach: KDE and Splines for MLE

Combining KDE and splines can enhance model accuracy and adaptability by capturing complex structures in data.

Definition 22. *A **hybrid KDE-spline model** uses KDE to estimate an initial density function and refines it using splines, which offer local flexibility.*

Example 17. Given a dataset $X = (x_1, x_2, \ldots, x_n)$, we first compute the KDE:

$$f(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right),$$

and model this estimate as a spline:

$$S(x) = \sum_{i=1}^{k} a_i B_i(x),$$

allowing for an adaptable approximation to the data distribution.

The likelihood function for this model is:

$$L(S|X) = \prod_{i=1}^{n} f(x_i|S(x_i)) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_i - S(x_i))^2}{2\sigma^2}\right). \tag{6.9}$$

Maximizing the log-likelihood with respect to a_i and h provides a robust model that captures both global and local data features.

6.11 Model Performance

6.11.1 Correlation Matrix

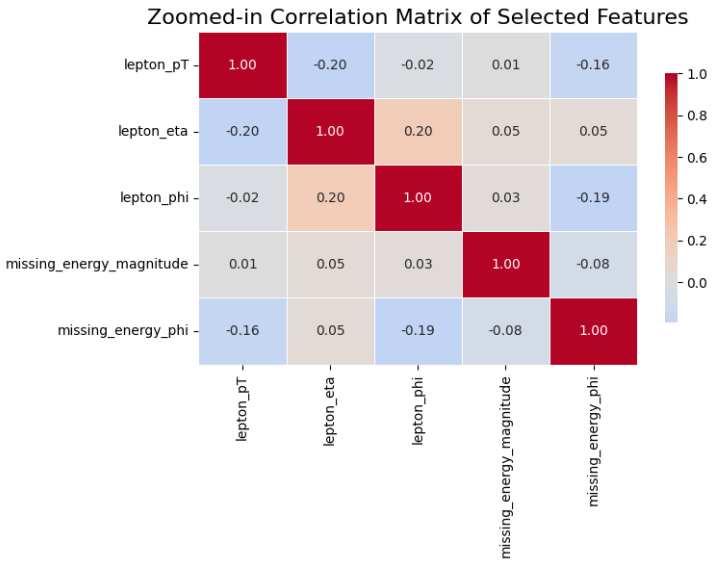
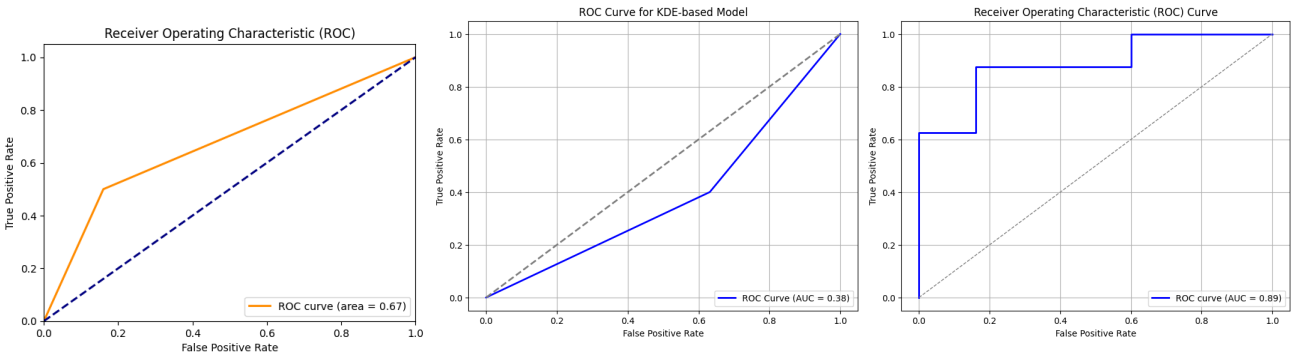


Figure 6.3: A zoomed in Correlation Matrix for Higgs Boson searches

A correlation matrix is a table that displays the correlation coefficients between multiple variables. Each cell in the table shows the correlation between two variables, ranging from -1 to 1. A value of 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 indicates no correlation. This matrix is crucial for understanding relationships between features and can help identify multicollinearity, which may affect model performance.

6.11.2 ROC Curve



(a) ROC curve for Decision Tree (b) ROC Curve for MLE using KDE (c) ROC Curve for MLE using Splines

Figure 6.4: ROC Curves for hpyothetical Machine Learning Algorithms for Higgs Boson searches

The Receiver Operating Characteristic (ROC) curve is a graphical representation used to assess the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The curve plots the true positive rate (sensitivity) against the false positive rate (1 - specificity). The area under the ROC curve (AUC) provides a single measure of overall accuracy that is independent of any specific threshold. An AUC of 0.5 suggests no discriminative ability, while an AUC of 1.0 indicates perfect discrimination.

Appendix A

Appendix A

A.1 Pseudocode for Decision Tree in C++

```
class TreeNode {
    string feature              // Feature used for splitting
    double threshold            // Threshold value for the split
    TreeNode left               // Pointer to the left child
    TreeNode right              // Pointer to the right child
    string classLabel           // Class label for leaf nodes
    bool isLeaf                 // Indicates if the node is a leaf
}

// Main class for Decision Tree
class DecisionTree {
    TreeNode root               // Root node of the tree

    // Function to build the decision tree
    function buildTree(data, labels) {
        if all instances in data belong to the same class then
            return createLeafNode(classLabel)
```

```
    if no features left to split then
        return createLeafNode(most common class)

    // Find the best feature and threshold for splitting
    bestFeature, bestThreshold = findBestSplit(data, labels)
    TreeNode node = new TreeNode(bestFeature, bestThreshold)

    // Split the data based on the best feature and threshold
    leftData, leftLabels, rightData, rightLabels = splitData(data, labels, bestFeature, be

    // Recursively build the left and right subtrees
    node.left = buildTree(leftData, leftLabels)
    node.right = buildTree(rightData, rightLabels)

    return node
}

// Function to predict the class label for a new instance
function predict(node, instance) {
    if node is a leaf node then
        return node.classLabel

    if instance[node.feature] <= node.threshold then
        return predict(node.left, instance)
    else
        return predict(node.right, instance)
}

// Function to find the best feature and threshold for splitting
function findBestSplit(data, labels) {
```

```
    double bestGain = -infinity
    string bestFeature
    double bestThreshold

    for each feature in data do
        for each unique threshold in feature do
            // Calculate the gain from this split
            gain = calculateInformationGain(data, labels, feature, threshold)
            if gain > bestGain then
                bestGain = gain
                bestFeature = feature
                bestThreshold = threshold

    return (bestFeature, bestThreshold)
}

// Function to calculate Information Gain
function calculateInformationGain(data, labels, feature, threshold) {
    leftData, leftLabels, rightData, rightLabels = splitData(data, labels, feature, thresh
    entropyBefore = calculateEntropy(labels)
    entropyAfter = calculateConditionalEntropy(leftLabels, rightLabels)
    return entropyBefore - entropyAfter
}

// Function to calculate Entropy
function calculateEntropy(labels) {
    map<string, int> classCounts
    for each label in labels do
        classCounts[label] += 1
```

```
    double entropy = 0.0
    total = labels.size()
    for each class in classCounts do
        probability = classCounts[class] / total
        if probability > 0 then
            entropy -= probability * log2(probability)

    return entropy
}

// Function to calculate Conditional Entropy
function calculateConditionalEntropy(leftLabels, rightLabels) {
    total = leftLabels.size() + rightLabels.size()
    pLeft = leftLabels.size() / total
    pRight = rightLabels.size() / total
    return pLeft * calculateEntropy(leftLabels) + pRight * calculateEntropy(rightLabels)
}

// Function to create a leaf node
function createLeafNode(classLabel) {
    TreeNode leaf = new TreeNode()
    leaf.classLabel = classLabel
    leaf.isLeaf = true
    return leaf
}

// Function to split the dataset based on a feature and threshold
function splitData(data, labels, feature, threshold) {
    leftData = []
    leftLabels = []
```

```
        rightData = []
        rightLabels = []

        for each instance in data do
            if instance[feature] <= threshold then
                leftData.append(instance)
                leftLabels.append(labels[instance])
            else
                rightData.append(instance)
                rightLabels.append(labels[instance])

        return (leftData, leftLabels, rightData, rightLabels)
    }
}
```

A.2 Pseudocode for AdaBoost implementation

```
class AdaptiveBoost {
    list<DecisionTree> weakLearners   // List of weak decision trees
    list<double> alpha                 // Weights for each weak learner

    // Function to train the Adaptive Boost model
    void train(data, labels, numIterations) {
        int numInstances = size(data)
        list<double> weights(numInstances, 1.0 / numInstances) // Initialize weights

        for (int i = 0; i < numIterations; i++) {
            // Train a weak learner on the weighted dataset
            DecisionTree weakLearner = new DecisionTree()
            weakLearner.train(data, labels, weights)
```

```
        // Make predictions on the training data
        list<int> predictions = weakLearner.predict(data)

        // Calculate error rate
        double error = calculateError(predictions, labels, weights)

        // Compute the weight of the weak learner
        double alpha = 0.5 * log((1 - error) / max(error, 1e-10))
        weakLearners.push_back(weakLearner)
        alpha.push_back(alpha)

        // Update weights
        for (int j = 0; j < numInstances; j++) {
            if (predictions[j] != labels[j] then
                weights[j] *= exp(alpha) // Increase weight for misclassified instances
            else
                weights[j] *= exp(-alpha) // Decrease weight for correctly classified inst
        }

        // Normalize weights
        normalize(weights)
    }
}

// Function to calculate the weighted error of predictions
double calculateError(predictions, labels, weights) {
    double error = 0.0
    int numInstances = size(predictions)
```

```
        for (int j = 0; j < numInstances; j++) {
            if (predictions[j] != labels[j] then
                error += weights[j] // Accumulate weights of misclassified instances
        }
        return error
    }

    // Function to make predictions using the ensemble of weak learners
    int predict(instance) {
        double finalPrediction = 0.0

        for (int i = 0; i < weakLearners.size(); i++) {
            // Aggregate predictions weighted by alpha
            finalPrediction += alpha[i] * weakLearners[i].predict(instance)
        }

        // Return final class label based on the aggregated prediction
        return (finalPrediction >= 0.5) ? 1 : 0
    }

    // Function to normalize weights
    void normalize(weights) {
        double sum = 0.0
        for each weight in weights do
            sum += weight
        for (int j = 0; j < size(weights); j++) {
            weights[j] /= sum // Normalize each weight
        }
    }
}
```

Bibliography

[1] CERN. Open-pho-chart-2015-001-1. https://cds.cern.ch/images/OPEN-PHO-CHART-2015-001-1/file?size=large, 2015.

[2] David J Griffiths. *Introduction to elementary particles; 2nd rev. version*. Physics textbook. Wiley, New York, NY, 2008.

[3] B. R. Martin and Graham Shaw. *Particle Physics*. Manchester Physics Series. Wiley, 3rd edition, 1992.

[4] Eldad Gildener and Steven Weinberg. Symmetry Breaking and Scalar Bosons. *Phys. Rev. D*, 13:3333, 1976.

[5] CERN. Ccc-v2022. https://cds.cern.ch/record/2800984/files/CCC-v2022.png?subformat=icon-640, 2022.

[6] ATLAS Collaboration. Atlas 2022 schematic. https://atlas.cern/sites/default/files/resources/colouringbook/ATLAS-2022-schematic.png, 2022.

[7] MadGraph5_aMC@NLO. https://launchpad.net/mg5amcnlo.

[8] Pythia8. http://home.thep.lu.se/Pythia/.

[9] Athgeneration. https://gitlab.cern.ch/atlas/athena/tree/21.6/Projects/AthGeneration.

[10] NNPDF30NLO. https://nnpdf.mi.infn.it/.

[11] Kenneth Lane and Eric Pilon. Phenomenology of the new light higgs bosons in gildener-weinberg models. *Physical Review D*, 101(5), March 2020.

[12] Estia J. Eichten and Kenneth Lane. Gildener-weinberg two-higgs-doublet model at two loops. *Physical Review D*, 107(7), April 2023.

[13] Estia Eichten and Kenneth Lane. Higgs alignment and the top quark. *Physical Review D*, 103, 06 2021.

[14] Kenneth Lane and William Shepherd. Natural stabilization of the higgs boson's mass and alignment. *Physical Review D*, 99(5), March 2019.

[15] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[16] Eri Asakawa, Daisuke Harada, Shinya Kanemura, Yasuhiro Okada, and Koji Tsumura. Discovery potential for T-odd Higgs Bosons at the LHC. *arXiv:1107.3391*, 2011.

[17] C.P. Robert and G. Casella. *Monte Carlo statistical methods*. Springer Verlag, 2004.