Predicting Physics Observables by Analysing LHC Data

Nissim Sahoo
Roll No.: 24B1818

Mentor: Deependra Sharma

WiDS, Analytics Club, IIT Bombay

January 7, 2026

Abstract

This document serves as a structured log and technical report for the project *Predicting Physics Observables by Analysing LHC Data*. The report documents the step-by-step exploration of ROOT files, understanding of data structures, and preliminary analysis techniques used in high-energy physics workflows.

Contents

1 Introduction

High-energy physics experiments such as those conducted at the Large Hadron Collider (LHC) generate extremely large and structured datasets. These datasets are commonly stored using the ROOT framework, which provides efficient data storage, access, and visualization capabilities.

The objective of this project is to develop a systematic understanding of ROOT files and their internal structure as a first step toward predicting physics observables from LHC data.

2 Understanding ROOT

Exploration of AnalysisResults67.root

This section documents the systematic exploration of the ROOT file `AnalysisResults67.root`. The objective was to understand the internal structure of the file, identify the types of stored objects, and verify whether the data is stored in event-wise tabular form or as aggregated analysis outputs.

Overview of the ROOT File Structure

The file was opened using the ROOT framework and its top-level contents were listed. The file does not contain any objects directly at the top level; instead, it is organized into multiple task-based directories:

- track-propagation
- tof-signal
- event-selection-task
- bc-selection-task
- ft0-corrected-table
- centrality-table
- hf-candidate-creator-dstar
- hf-candidate-creator-dstar-expressions
- hf-candidate-selector-dstar-to-d0-pi
- hf-task-dstar-to-d0-pi

This indicates that the file corresponds to an *analysis output* rather than raw event-level data.

Nested Directory Exploration

Each directory was explored individually. For example, the `track-propagation` directory contains:

- hDCAxyVsPtRec (TH2F)
- hDCAxyVsPtMC (TH2F)
- hDCAzVsPtRec (TH2F)

- hDCAzVsPtMC (TH2F)
- trackTunedTracks (TH1D)

No TTree objects were found.

Python-Based Inspection Using `uproot`

The `uproot` library can be used to inspect the data programmatically:

```
import uproot
file = uproot.open("AnalysisResults67.root")
file.keys()
```

This confirms that the file contains only histogram objects, suitable for aggregated analysis, not per-event exploration.

Exploration of Prompt_DstarToD0Pi.root

This file contains a single `TTree` called `treeMLDstar`, storing event-wise data of reconstructed $D^* \to D^0\pi$ candidates. The structure is flat and columnar:

- Kinematic variables: fPt, fEta, fPhi
- Topological variables: fDecayLengthD0, fImpactParameter
- PID variables: fNSigTpc, fNSigTof

Python can read this TTree into a pandas DataFrame:

```
tree = file["treeMLDstar"]
df = tree.arrays(library="pd")
df.shape
```

The DataFrame contains 797,119 rows and 51 columns.

3 Fundamental Particle Physics Background

3.1 Relativistic Kinematics

- Spatial resolution: $\lambda = \frac{h}{p}$
- Lorentz invariant interval: $ds^2 = c^2dt^2 - dx^2 - dy^2 - dz^2$
- Energy-momentum relation: $E^2 = p^2c^2 + m^2c^4$
- Natural units: $c = \hbar = 1 \implies E^2 = p^2 + m^2$
- Rapidity: $y = \frac{1}{2}\ln\frac{E+p_z}{E-p_z}$
- Transverse momentum: $p_T = \sqrt{p_x^2 + p_y^2}$
- Pseudorapidity: $\eta = -\ln\tan\frac{\theta}{2}$

3.2 Elementary and Composite Particles

- Fermions: quarks and leptons
- Bosons: photon, gluon, W, Z, Higgs
- Hadrons: mesons (quark-antiquark), baryons (three quarks)

3.3 Heavy Quarks and Heavy-Flavor Mesons

- Charm (c) and bottom (b) quarks
- D^0: $c\bar{u}$, B mesons: $b\bar{q}$
- Prompt vs Non-Prompt D^* production

3.4 Strong Interaction

- Quantum Chromodynamics (QCD)
- Perturbative QCD at high energies
- Quark-Gluon Plasma at extreme temperature

4 Experimental Reconstruction of D Mesons

- $D^{*+} \to D^0\pi^+_{\text{soft}}, D^0 \to K^-\pi^+$
- Invariant mass reconstruction: $m^2 = (E_1 + E_2)^2 - (\vec{p}_1 + \vec{p}_2)^2$
- Transverse momentum: $p_T = \sqrt{p_x^2 + p_y^2}$

5 Decay Topology and Machine Learning

- Decay geometry: decay length, impact parameter, pointing angle
- Pointing angle: $\cos\theta_p = \frac{\vec{p}_D \cdot \vec{L}}{|\vec{p}_D||\vec{L}|}$
- ML classifier separates prompt, non-prompt, background
- Input features exclude invariant mass and transverse momentum

6 Analysis Workflow

Proton–Proton Collision

$\downarrow$

Track Reconstruction

$\downarrow$

D Meson Candidate Formation

$\downarrow$

Invariant Mass Selection

$\downarrow$

Topological Variable Calculation

$\downarrow$

Machine Learning Classification

$\downarrow$

Prompt / Non-Prompt Separation

$\downarrow$

Physics Observable Extraction

7 Dataset Structure and Variable Inspection

- Three ROOT files: Prompt, Non-Prompt, Background
- Each has a single TTree: treeMLDstar
- Branches: topological, kinematic, PID variables

8 Transverse Momentum Binning

- $p_T = \sqrt{p_x^2 + p_y^2}$
- pT intervals: $[1, 1.5, 2, 3, 4, 6, 8, 10, 12, 16, 24]$ GeV/c
- Example: $7 < p_T < 10$ GeV/c used for detailed analysis

9 Exploratory Analysis of Topological Variables

- Candidate-level distributions, normalized
- Pointing angle cosine: $\cos\theta = \frac{\vec{L}\cdot\vec{p}}{|\vec{L}||\vec{p}|}$
- Transverse pointing angle: $fCpaXYD0$
- Transverse decay length: $L_{xy} = |\vec{r}_{SV} - \vec{r}_{PV}|_{xy}$
- Impact parameters of decay tracks

10 Summary

This report documents a structured understanding of ROOT file structures, event-wise candidate data, topological observables, and the machine-learning workflow used to separate prompt, non-prompt, and background D^* candidates. All analyses can be performed using Python (`uproot` + `pandas`) without requiring images or ROOT GUI visualization.