technische universität
dortmund

Advanced Laboratory Course

Particle Physics

Multivariant analysis for the selection of LHCb data

Koen Denekamp & Riana Shaba

June 2025

1 Introduction

1.1 The LHCb detector

The LHCb detector was conceived specifically to perform experiments in flavour physics. According to the design specifications [6], it was created to study CP-violation and other phenomena related to the b quark. CP-violation concerns the problem that there should have been an equal amount of matter and antimatter created at the Big Bang. Matter and antimatter annihilate when they get close together, so there should be no matter in the universe. This is not true, as everything around us is made out of matter.
The LHCb detector, shown in Figure 1, has several components. In this diagram, the protons collide at the left, where $z = 0$, and in the middle, where $y = 0$. The particles created in the collision will thus be detected if they move from left to right. It should be noted that particles are created in every direction, but particles containing b and c quarks mostly move in the positive z or negative z direction. This is the order that will be followed in explaining the different components as well.

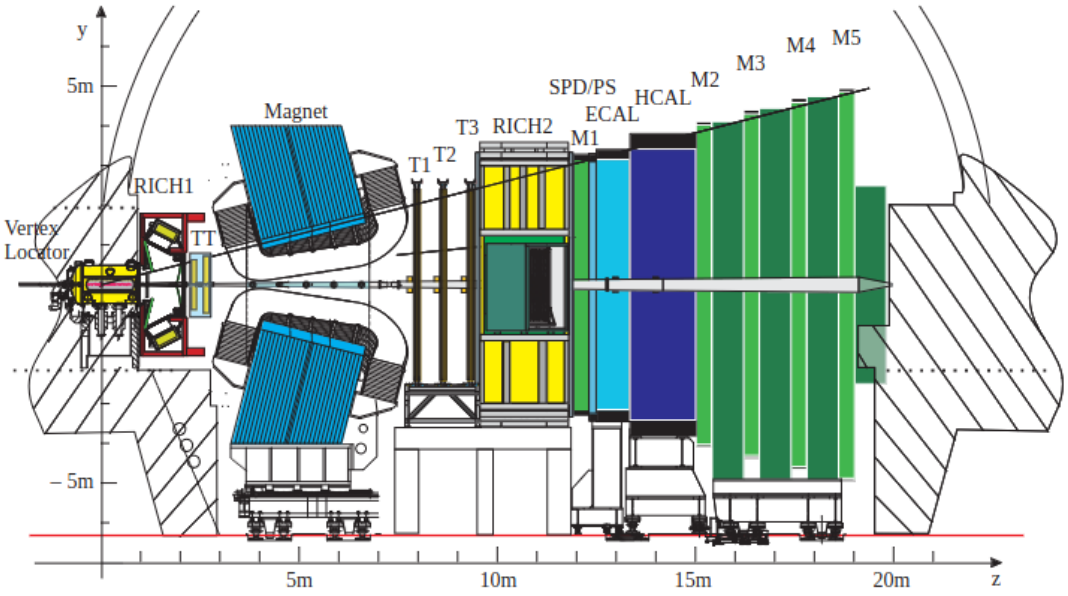


Figure 1: Diagram of the LHCb detector with the separate components. From Ref. [6].

The decaying particles first encounter the Vertex Locator (VELO). This segment is part of the tracking system, as it both reconstructs primary and secondary vertices of particles,

their decays, and the trajectories [4].
During Run 2, the time on which the Monte Carlo simulations that generated the data were based, the particles then entered the Tracker Turicensis (TT).
Then the particles enter the first Ring-Imaging Cherenkov detector (RICH1). This part of the detector is used for particle identification [1]. It works due to the fact that different mediums have a different speed of light: when a particle travels faster than the speed of light in that medium, the particle disrupts the molecules of the medium in such a way that they release photons.
Based on the Cherenkov angle between the track of the particle and the line one can draw across the side of the rings, the velocity of the particle can be determined [2]. This means the angle is related to the velocity.
Since the momentum of the particle is measured using the tracking stations, and the magnet for charged particles [2], it is possible to combine the momentum and velocity measures to find an estimate of the mass of a particle.
Then there is a magnet used to measure whether a particle is positively, negatively, or neutrally charged, as these groups of particles will bend in different directions, or not at all for neutrally charged particles. It is also used to calculate the momentum of charged particles since high-momentum particles bend less than low-momentum particles.
T1, T2, and T3 are tracking stations consisting of two parts. The inner part, known as the Inner Tracker (IT), is close to the beam pipe. The outer part, known as the Outer Tracker (OT) in placed around the IT. The IT and the TT together are also known as the Silicon Tracker [8].
Then another RICH detector is encountered, before entering the SPD/PS layers. SPD stands for Scintillator Pad Detector, while PS stands for PreShower. The goal of the SPD is to identify charged particles and to separate electrons from protons. The PS identifies electromagnetic particles. These steps are taken, since the calorimeters that follow these layers require good background rejection and reasonable efficiency [7].
There are two calorimeters. One specifically for electrons and photons, known as the ECAL and one for hadrons, known as the HCAL. These detectors measure the energy of the incoming particles.
Lastly there are the five muon detector layers, which are also part of the tracking system. Muons pass through the subsystems of the detector, as they have a very low interaction probability.
Due to the massive amount of data collected, it is not feasible to store everything. This is why, in Run 2, there were hardware and software triggers that only recorded data if it meets certain criteria, such as a high enough kinetic energy, and it thus deemed interesting for analysis. The hardware triggers (known as L0) were built into the detector, while the software triggers (which are applied at two levels known as HLT1 and HLT2)

happen when the data has been transferred to computers. These triggers reduce the frequency of events that are recorded from 40 MHz to 12.5 kHz [5].

1.2 Physics

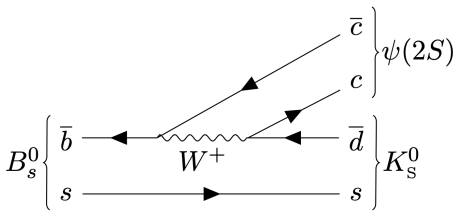
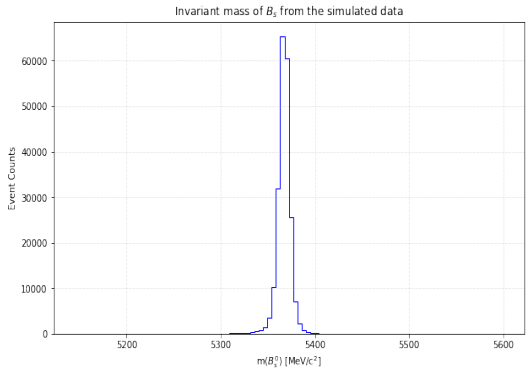


Figure 2: Feynman diagram at the leading order of $\mathrm{B_s^0} \to \psi(2\mathrm{S})\mathrm{K_s^0}$. From Ref. [3].

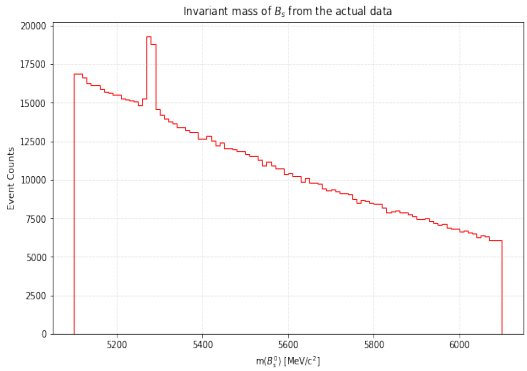
In this laboratory exercise we are interested in the decay $\mathrm{B_s^0} \to \psi(2\mathrm{S})\mathrm{K_s^0}$, whose Feynman diagram in the leading order is shown in Figure 2. However, this signal is hidden in the dataset because of the enormous amount of background events that are present. Therefore, the peak we see in Fig. 3b is not directly the signal we are looking for, but it corresponds to the very kinematically-similar decay of $\mathrm{B^0} \to \psi(2\mathrm{S})\mathrm{K_s^0}$. In this study we assume that the dataset contains only the combinatorial background, which is the dominating background, the visible $\mathrm{B^0}$ peak and possibly $\mathrm{B_s^0}$ events. The combinatorial background is the background that is generated from mismatching final state particles that are detected. This generally creates a random mass distribution, that is exponentially decaying in abundance when the mass increases.
In order to obtain a clean signal with the right signal to background ratio, we will use a machine learning algorithm that will be able to classify the signal and background.

2 Data Analysis

This data analysis for studying the $\mathrm{B_s^0} \to \psi(2\mathrm{S})\mathrm{K_s}$ decay is based on three samples: the real dataset, and two Monte Carlo simulations for the B_s^0 decay and the kinematically similar B^0 decay. First we start by exploring the first two samples by plotting the invariant mass of B_s^0 in the simulation data and in the actual data, which are shown in Figure 3.



(a) Histogram showing the invariant mass distribution of B_s^0 in the simulation data, showing a peak at $\mathrm{m(B_s^0)} = 5365.37$ MeV.



(b) Histogram showing the invariant mass distribution of B_s^0 in the real data. The peak does not correspond to B_s^0.

Figure 3: Histograms showing the distributions of the invariant mass of B_s^0 in the simulation sample and real dataset. While in the simulation we have a peak corresponding to B_s^0 close to nominal mass, in the dataset the B_s^0 signal is hidden in the vast amount of events and background.

In Figure 3a we have the distribution of the invariant mass from the MC simulation sample. We can see a peak corresponding to $m(B_s^0) = 5365.37$ MeV, which is near the nominal mass of $m(B_s^0) = 5366.88$ MeV. While in Figure 3b we have the real data distribution of the invariant mass. The peak showing here does not correspond to the signal we are after. Even though the data used has been preprocessed, there still remains an enormous amount of uninteresting events and backgrounds, especially in this case it is dominated by the combinatorial background, that hides the signal.

In the following we define a window where we expect signal to appear by choosing the shortest interval that contains 99% of the MC mass distribution. We find the narrowest signal window to be [5333.42, 5394.60] MeV. The next step is to select the background sample, because the classifier should not use the signal window to train on, otherwise it will result to be a biased one. Therefore, suitable signal and background training samples to work on are needed. In this case, the signal training sample is the signal simulation, and the background training sample is the upper sideband (UBS) of the B_s^0 peak that is hidden in the data. The reason for this is because the USB contains reconstructed B_s^0 masses larger than the nominal one, and it contains combinatorial background which we want to remove, so it is suitable for the training. A full visualization of the data mass distribution, the signal window defined centered on the nominal B_s^0 mass, and the back-

ground training sample is shown in Figure 4.

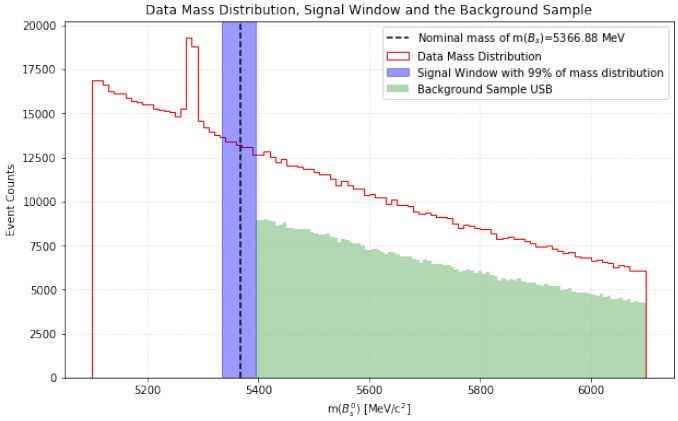


Figure 4: An overview of the data mass distribution, the signal window centered on the nominal B_s^0 mass, and the background training sample.

Simulation algorithms remain imperfect, mostly because theoretical models themselves come with their own uncertainties, so there is also some amount of error in the modelling of the detector to simulate responses. To correct to some extent these errors, we use weights. Some aspects of the decay kinematics that are imperfect in simulation are corrected using `kinematic_weights`. Other properties of the decay can be mismodelled by simulation, so we need our classifier to not only be able to distinguish between simulation and data, but between signal and background too. Hence, we evaluate how similar simulation and data are in each variable without having B_s^0 data to compare the simulation with. For this purpose we use $\mathrm{B}^0 \to \psi(2\mathrm{S})\mathrm{K_s}$ decay because it is very abundant in the dataset and moreover, it is kinematically similar to B_s^0 decay, so it will be important in the removal of combinatorial background. The sWeights, `sweights_sig`, have been computed and give pure B^0 events from the real dataset. These weights are used to evaluate the difference between the data and simultated control sample. Figure 5 shows the reconstructed mass of the B^0 meson with and without sWeights.

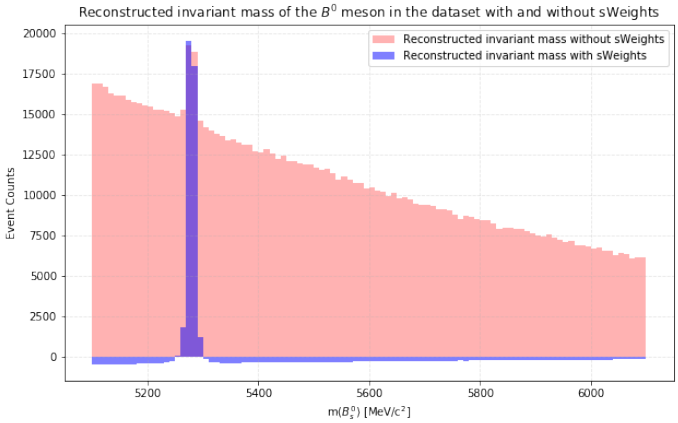


Figure 5: Histogram showing the reconstructed invariant mass of the B^0 meson in the dataset with and without sWeights.

Next we create several plots to show the distribution of each variable for the B^0 in the dataset by using `sweights_sig`, and the B^0 from the specific simulated sample of this decay using the `kinematic_weights`. Some of the relevant variables in this study are the distribution of energy, pseudorapidity (η), azimuthal angle (ϕ) and the transverse momentum (p_T).

The next step is to perform a feature selection. Since there are over 800 variables in the dataset, this is very important. First we remove any unrelated variables, such as errors, trigger information, etc. This leaves nearly 400 variables. Here we compare the B^0 simulation and B^0 data using a Kolmogorov–Smirnov test, where we calculate the Kolmogorov-Smirnov statistics according using the Empirical Cumulative Distribution Function

$$\mathrm{F_W(x)} = \frac{\sum_{i|x_i \in D, x_i \leq x} w_i}{\sum_{j|x_j \in D} w_j}, \tag{1}$$

so that

$$\mathrm{KS} = \sup_x |F_{W_1}(x) - F_{W_2}(x)|. \tag{2}$$

The reason for this is to only consider variables in our training that are adequately simulated to a degree that we can expect the same behavior with the simulated data as with real data later. This is important, since using variables that are simulated in a different way from reality invalidates the use of those variables, as it would train the model to distinguish simulated and real data, rather than the mesons we are interested in distinguishing. Looking at the histogram shown in Figure 6, we apply a cut-off of $\mathrm{KS} < 0.05$,

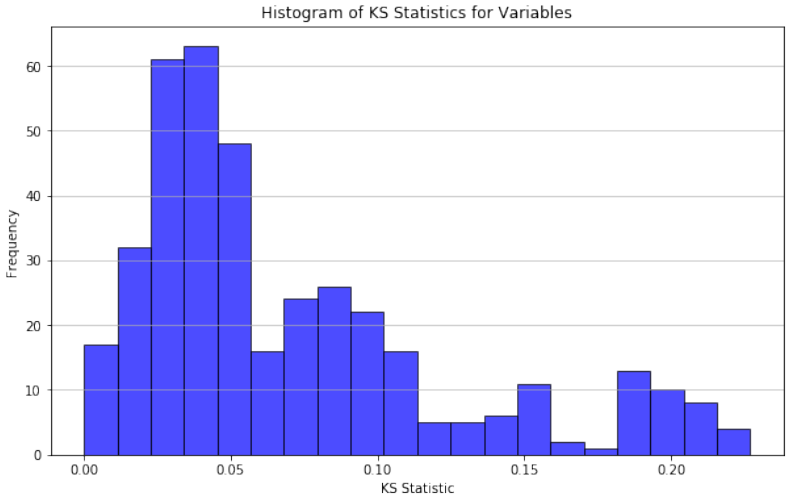


Figure 6: Histogram showing the Kolmogorov–Smirnov test value.

because this value keeps a lot of highly similar data, while still removing many variables. This means we keep variables that show a value of 0.05 or less.

There remain exactly 200 variables. We calculate the test values again, but now to compare signal and background. Here we keep only variables with a large value, larger than 0.20 and this results in 93 variables.

Then we remove variables that are highly correlated between themselves, as this will reduce the effectiveness of the individual variables. We only keep one variable of each pair, or more, that has a value larger than 0.6.

In the end, there are 26 variables that we will use in training the classifier, shown in Table 1.

Table 1: List of Variables

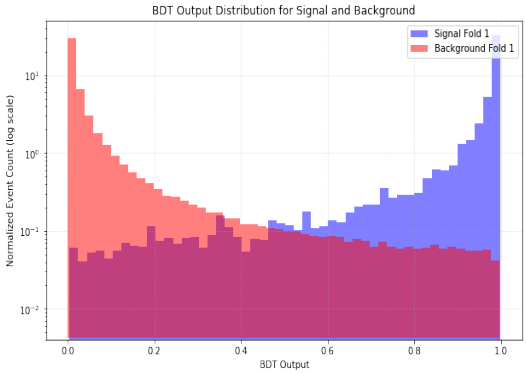
Variable Name
B_LOKI_ETA
B_Vtx_Chi2NDOF
B_MINIP
B_OWNPV_CHI2
B_FDCHI2_OWNPV
B_PT
B_FitDaughtersConst_KS0_M_flat

Continued on next page

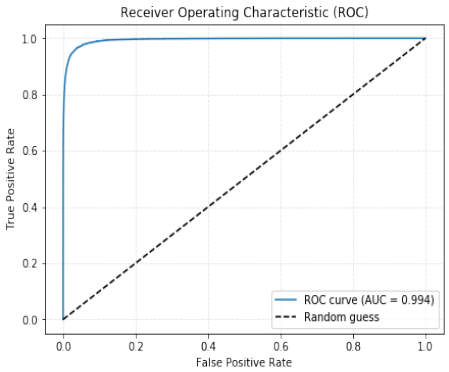
Table 1 – continued from previous page

Variable Name
B_FitDaughtersConst_psi_2S_M_flat
B_FitDaughtersPVConst_KS0_M_flat
B_FitDaughtersPVConst_psi_2S_M_flat
B_FitPVConst_psi_2S_M_flat
psi_2S_Vtx_Chi2NDOF
psi_2S_FD_ORIVX
muplus_MINIP
muplus_IP_ORIVX
muplus_PIDp
muplus_ProbNNp
muplus_ProbNNghost
muminus_MINIP
muminus_IP_ORIVX
muminus_PIDp
muminus_ProbNNp
muminus_ProbNNpi
muminus_ProbNNghost
KS0_IP_ORIVX
KS0_DIRA_ORIVX

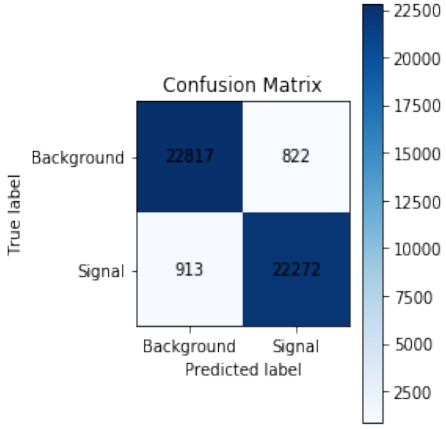
In this exercise, we consider a Boosted Decision Tree (BDT), from the `xgboost` package, which is a common choice in High Energy Physics. Before we train the model we must define some weights for the background, since there is a large imbalance between classes in the training set. This weight is defined as the sum of the signal weights, i.e. the amount of signal, divided by the number of background. This will create a weighted 50/50 split.
We train the model using 5-fold cross validation, to reduce the variance in the test set, thus improving the generalization of the model. We keeps these fold separate until we optimize the FOM, as described below. Then we calculate the required variables for each split individually, and take the sum. Metrics are calculated, and the plots are fitted. The plots for the first fold are shown in Figure 7 as an example.



(a) BDT output histograms



(b) ROC curve



(c) Confusion Matrix

Figure 7: Figures showing the metrics considered for the first split.

Now that we have a model that gives us an output, we need to optimize the selection. In order to do this, we use the Punzi Figure of Merit (FOM). This is defined as

$$s = \frac{\epsilon_{\text{sig}}}{\frac{5}{2} + \sqrt{n_{\text{bkg}}}}, \tag{3}$$

where ϵ_{sig} is the signal efficiency and $n_{\mathbf{bkg}}$ is the number of background events in the same region. We calculate different values of the FOM between 0 and 1 for every split, and then take the mean value for each position. Calculating the cut value belonging to the maximum FOM, we choose our cut value to be 0.990. This means that we only keep events that have a BDT output of 0.990 or larger.

Then we apply the classifier to the full dataset and plot the mass distribution as shown in Figure 8. The two peaks correspond to the decays B^0 and B_s^0, while throughout the distribution there is also the combinatorial background present.

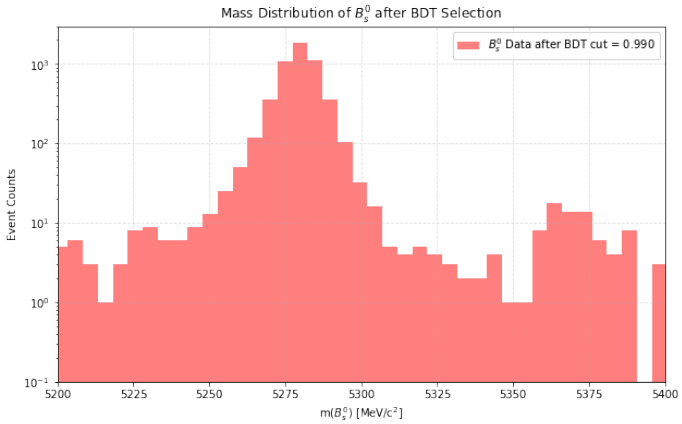


Figure 8: Histogram showing the mass distribution of B_s^0 in the full dataset. With the BDT cut of 0.990 applied, the two peaks, B^0 and B_s^0, can be seen.

To evaluate the efficiency versus background retention, we model the entire mass distribution. We define the full model as a combination of Double Gaussian fits to the two peaking structures that correspond to the BDT selected B^0 simulation and B_s^0 signal simulation, and a decreasing exponential function fit to the combinatorial background. Prior to fitting, both signal simulations are classified and the BDT selection is applied to ensure that the peak shapes in simulation are consistent with those in data. The shape parameters obtained from the fits are then fixed in the fit to the data.
Before performing the fit we plot the mass distribution after applying the BDT cut in order to estimate by eye the ranges needed for the fit parameters. The results of the selected B_s^0 signal simulation are given in Table 2 and Figure 9.

Parameter	Value
μ	5367.126 ± 0.016
σ_1	5.041 ± 0.016
σ_2	18.979 ± 0.240
fraction	0.927 ± 0.002

Table 2: Fit parameters for the B_s^0 signal simulation.

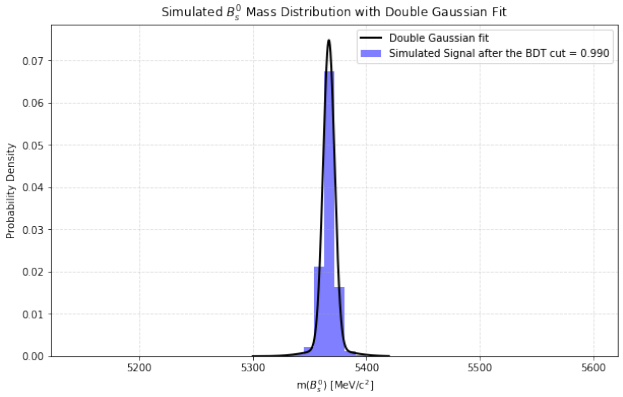


Figure 9: Histogram showing the mass distribution of the simulated B_s^0 with a Double Gaussian Fit.

We do the same for the selected B^0 simulation. These results are shown in Table 3 and Figure 10.

Parameter	Value
μ	5279.884 ± 0.018
σ_1	4.786 ± 0.016
σ_2	18.979 ± 0.018
fraction	0.928 ± 0.002

Table 3: Fit parameters for the B^0 signal simulation.

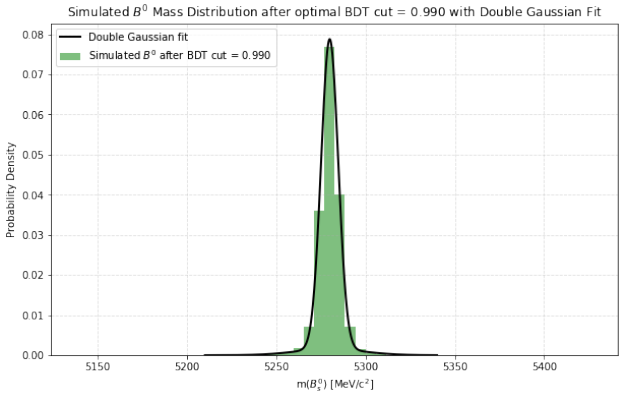


Figure 10: Histogram showing the mass distribution of the simulated B^0 with a Double Gaussian Fit.

Although the classifier performs well, it does not capture all combinatorial background that is present in the data sample. Therefore, it is important to include an explicit background component in the models, for which we use a decreasing exponential function. The results of this fit with the three submodels are given in Table 4, and a plot with

all the submodels and the selected data is shown in Figure 11.

Parameter	Value
B^0 fraction	0.960 ± 0.004
B_s^0 fraction	0.008 ± 0.002
Background	-0.002 ± 0.001

Table 4: Fit parameters for the Full model.

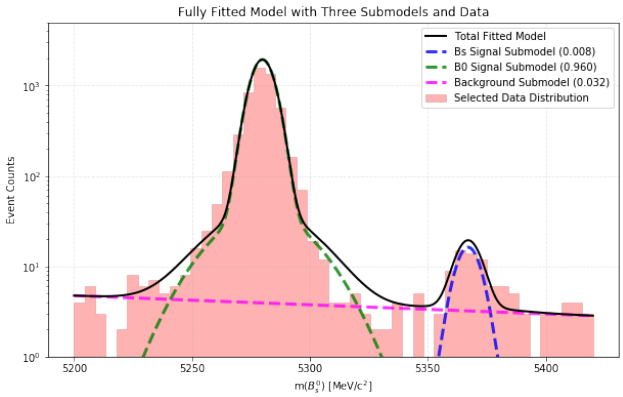


Figure 11: Histogram showing the mass distribution of the fully fitted model with all the three submodels and data.

From the values in the Table 4 and the plot in Figure 11, we can see that the signal we are interested in, B_s^0, is now visible but extremely small compared to the B^0 events that dominate the data. The rest is a small fraction of background.
We compute the statistical significance of our observation using the following formula:

$$m = \frac{n_{sig}}{\sqrt{n_{sig} + n_{bkg}}} \tag{4}$$

where n_{sig} = 48.49 is the number of signal events, and n_{bkg} = 182.32 is the number of background events in the signal region.
We find the significance to be m = 3.19, which means that the signal is above a 3σ threshold, representing an evidence of the $B_s^0 \to J/\psi\, K_s$ decay mode.

3 Conclusions

In this laboratory exercise we completed a full study of extracting signal from a dataset that was heavily dominated by combinatorial background.

We started by exploring and comparing the simulated data with the actual data. Then we defined the narrowest window containing 99% of the Monte Carlo mass distribution where we expected signal to appear.

In the following we selected a background training sample to be the upper sideband of the mass distribution since it contained combinatorial background needed to be removed, and a signal training sample to be the signal simulation. Then, we evaluated how similar simulation and data are, because the classifier needs to be able to distinguish between signal and background, not only simulation and data. We found a good match between the distributions of energy, pseudorapidity, azimuthal angle, and the transverse momentum for both the simulated sample and the real dataset.

We then performed feature selection. We reduced the number of features from over 800 to just 26. We use these variables to train five BDTs, each on a different subset of the data. These are evaluated using Confusion Matrices and ROC curves. We then find the optimum cut value to be at 0.990. We applied the classifier and the BDT cut to the signal and B^0 simulations, then we model the full mass distribution which includes both peaks. We also did a fit to the background using a decreasing exponential function, then we plotted the mass distribution of the fully fitted model with all the three submodels and data. We found an extremely small, but visible peak of the B_s^0 mass with a signal yield of 48.49, and a significance of m = 3.19 which is statistically a satisfying result.

References

[1] N H Brook et al. *LHCb RICH 1 Engineering Design Review Report*. en. Aug. 2004. URL: https://cds.cern.ch/record/897981/files/lhcb-2004-121.pdf.

[2] LHCb collaboration et al. "LHCb Detector Performance". en. In: *International Journal of Modern Physics A* 30.07 (Mar. 2015). arXiv:1412.6352 [hep-ex], p. 1530022. ISSN: 0217-751X, 1793-656X. DOI: 10.1142/S0217751X15300227. URL: http://arxiv.org/abs/1412.6352 (visited on 04/16/2024).

[3] LHCb bei E5. "Selection of $B_s^0 \to \psi(2S)K_s^0$". In: (June 24, 2023).

[4] P. Kopciewicz, S. Maccolini, and T. Szumlak. "The LHCb vertex locator upgrade — the detector calibration overview". en. In: *Journal of Instrumentation* 17.01 (Jan. 2022), p. C01046. ISSN: 1748-0221. DOI: 10.1088/1748-0221/17/01/C01046. URL: https://iopscience.iop.org/article/10.1088/1748-0221/17/01/C01046 (visited on 11/20/2023).

[5] LHCb Collaboration. "Design and performance of the LHCb trigger and full real-time reconstruction in Run 2 of the LHC. Performance of the LHCb trigger and full real-time reconstruction in Run 2 of the LHC". In: *JINST* 14.04 (2019). _eprint: 1812.10790, P04013. DOI: 10.1088/1748-0221/14/04/P04013. URL: https://cds.cern.ch/record/2652801 (visited on 05/08/2024).

[6] LHCb Collaboration. "LHCb reoptimized detector design and performance : Technical Design Report". In: *CERN-LHCC-2003-030*. Technical Design Report (July 2003). http://cds.cern.ch/record/630827.

[7] Eduardo Picatoste Olloqui. "LHCb Preshower(PS) and Scintillating Pad Detector (SPD): commissioning, calibration, and monitoring". In: *J. Phys.: Conf. Ser.* 160 (2009), p. 012046. DOI: 10.1088/1742-6596/160/1/012046. URL: https://cds.cern.ch/record/1293075 (visited on 11/20/2023).

[8] J. van Tilburg. "Tracking performance in LHCb". en. In: *The European Physical Journal C - Particles and Fields* 34.1 (July 2004), s397–s401. ISSN: 1434-6052. DOI: 10.1140/epjcd/s2004-04-041-7. URL: https://doi.org/10.1140/epjcd/s2004-04-041-7 (visited on 04/09/2024).