ECS784 Coursework 1 - Using SVC and K-Neighbors Algorithms to Distinguish Between Two Z^0 Boson Decay Modes (75.8% accuracy)

Ben Malpas

March 2023

1 Introduction and background information

1.1 Summary

This report details the use of two machine learning algorithms (SVC and K-Neighbors) to attempt to distinguish between two decay modes of the Z^0 boson (a neutral elementary particle) using labelled data originally from CERN. The results achieved via the two different approaches will be assessed based on accuracy on unseen test data.

1.2 Background information

The Z^0 boson is a neutral exchange particle which mediates the weak interaction. The Z^0 boson has three general groups of possible decays. It can decay into charged lepton-antilepton pairs, neutrino-antineutrino pairs, or quark-antiquark pairs. This investigation is concerned with a subset of the first group of decays. There are three possible charged lepton-antilepton pairs: electron-positron, muon-antimuon and tau-antitau. These are the only charged lepton decays which satisfy the relevant conservation laws as required [1].

Note that the positron is the antiparticle of an electron. The charged leptons (electron, muon and tau) all have negative charge, and their respective antiparticles have positive charge. This means that when the electrically neutral Z^0 boson decays into a lepton and an antilepton, the system remains neutrally charged and so electrical charge conservation is not violated.

The basic coordinate system generally employed in collider experiments is detailed in Figure 1.

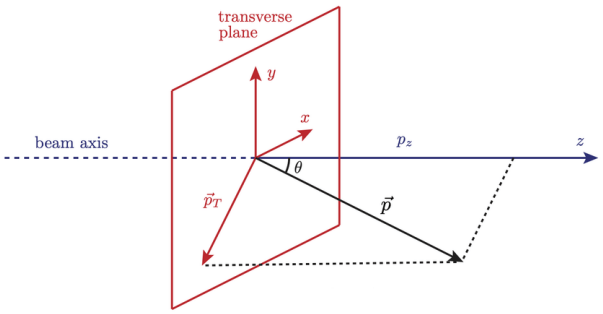


Figure 1: Standard geometry of a collider. The beam is oriented along the z axis and the pseudorapidity (η) is a measure of the θ angle. [2] The ϕ angle (phi) is not shown here but it is the angle of the transverse momentum (p_T) to the x axis.

Formulas for the cartesian components of the momentum are given in Equation 1. The first two relations follow from trigonometry and the definition of the phi angle.

$$p_x = p_T \cos(\phi),\ p_y = p_T \sin(\phi),\ p_z = p_T \sinh(\eta) \text{ [3]} \quad (1)$$

1.3 Problem statement and hypothesis

The goal of this project is to distinguish between two of the charged lepton-antilepton decay modes of the Z^0 boson using two machine learning techniques. The two decays in question are the electron-positron and the muon-antimuon decay modes. All other possible decays are not present in the data set, which contains largely kinematic data collected by CERN as well as labels specifying the type of decay for each data-point.

The electron has a mass of 0.511Mev, the muon has a mass of 105.66Mev and the Z^0 boson has a mass of 91,200Mev [4]. As the mass of the Z^0 boson is far greater than that of the decay products in both cases, the disparity in kinetic energies of the resulting decay particles for the electron-positron and muon-antimuon case will be small compared to the total kinetic energy of the decay products. This means that the kinematics (what we will be largely using to differentiate the decay modes) will likely be similar in each case. Our data set also does not specify the total energy of the products that lead to the Z^0 boson or even whether or not it was constant for each collision. The original source of the data set also directly specifies that only a subset of the full event information is given, and so the data is not suitable for full physical analysis. Taking these factors and any unforeseen variation in the data into account leads to the hypothesis that a high accuracy classifier model is an unlikely outcome.

1.4 Context within literature

For several decades, machine learning techniques have been used in high energy particle physics, often under the name of multivariate analysis. Such analysis employs machine learning techniques such as the use of neural networks, decision trees and support vector machines. These techniques experienced performance limitations when applied to data sets with many features. The development of deep learning around 2012 has helped to overcome this performance bottleneck, as the very large neural networks were able to tackle more complex and higher dimensional problems with better performance [5]. Machine learning methods will become even more prominent within high energy physics with the advent of the high luminosity LHC, which is set to deliver around 100 times the luminosity (directly proportional to the amount of detections) as the LHC. Such large data sets will provide significant performance problems, with machine learning promising some solutions [6].

This project uses traditional machine learning techniques to tackle a toy problem within the field of high energy physics, motivated by the need for machine learning approaches in this domain.

Omid Baghcheh Saraei undertook a similar project using the same data set and was able to achieve accuracies of 0.828, 0.872 and 0.909 using K-Neighbors, SVC and TensorFlow respectively [7].

2 Data processing

2.1 Data source

The data used in this project was downloaded from Kaggle and can be accessed here[1]. The data set is a subset of a data set provided by CERN Open Data Portal which in turn is a small portion of event data collected by CERN.

2.2 Description of data

The raw downloaded data consists of 20,000 data-points with eleven features and one label for prediction. The label gives the class of decay, either Zee or Zmumu, which refers to one of the decays shown in Equation 2 and the features are summarised in the table in Figure 2. Approximately half of the data-points belonged

[1] https://www.kaggle.com/datasets/omidbaghchehsaraei/identification-of-two-modes-of-z-boson?datasetId=2903822

to each category of decay, which meets physical expectation.

$$\text{Zee: } Z^0 \rightarrow e^- + e^+, \text{ Zmumu: } Z^0 \rightarrow \mu^- + \mu^+ \quad (2)$$

Variable	Description
Unnamed: 0	An index labelling each event
Run	The run number of the event
Event	The event number
pt1	The transverse momentum of the first lepton (in units of GeV)
eta1	The pseudorapidity (η) of the first lepton
phi1	The phi (ϕ) angle (in radians) of the first lepton
Q1	The charge of the first lepton
pt2	The transverse momentum of the second lepton (in units of GeV)
eta2	The pseudorapidity (η) of the second lepton
phi2	The phi (ϕ) angle (in radians) of the second lepton
Q2	The charge of the second lepton

Figure 2: A table summarising each feature. Many variable descriptions have been adapted from the Kaggle data set.

2.3 Cleaning the data

Upon inspection, it was realised that the data contained 2745 events where two like charged particles were detected (either two Q=1 or two Q=-1). This contradicts charge conservation as the Z^0 boson is neutral as previously discussed. To deal with these paradoxical data-points, a copy of the original data, df2, was made and the contradictory events were removed from the copy leaving the original DataFrame, df, unchanged. Throughout the rest of the project, these two data sets were operated on simultaneously allowing for an eventual comparison to be made.

At this initial stage it was also safe to remove some of the non physical features, such as the "Unnamed: 0" index, the run and the event numbers, as it is not feasible that these arbitrary numbers affect the physics of the decay.

2.4 Feature manipulation

Physically it would make sense that the muon-antimuon decay products have less total momentum than the electron-positron decay products, as muons have a larger rest mass, which "takes up" some of the energy released by the decaying Z^0 boson. Motivated by this intuition, the phi (ϕ) angle, the pseudorapidity (η) and the transverse momentum were combined to find the individual particles' total momentum as well as the total momentum of the two-particle system. In order to find the correct values, the equations in figure 2 were used to find the cartesian components, and these were combined to find the appropriate magnitudes.

It was also observed that the momentum related variables were not normally distributed, and were distributed over a much larger range of values than the phi (ϕ) angle and pseudorapidity (η) variables. Taking the log of the values helped both of these problems, particularly the latter.

2.5 Feature Selection

Using a pandas heatmap (Figure 3), correlations between features were inspected. As expected, the various momentum features and their log counterparts shared similar correlations to all other variables and to each other. For reasons previously mentioned and for the reason that the logged variables appeared to have stronger correlations with the class number, the unlogged features were removed from the data set.

It was also noticed that each of the ϕ angles had little correlation to any features other than the other particle's ϕ angle making them ineffective in determining the class of the decay. Physically this makes sense due to the rotational symmetry of the beam axis. The same argument

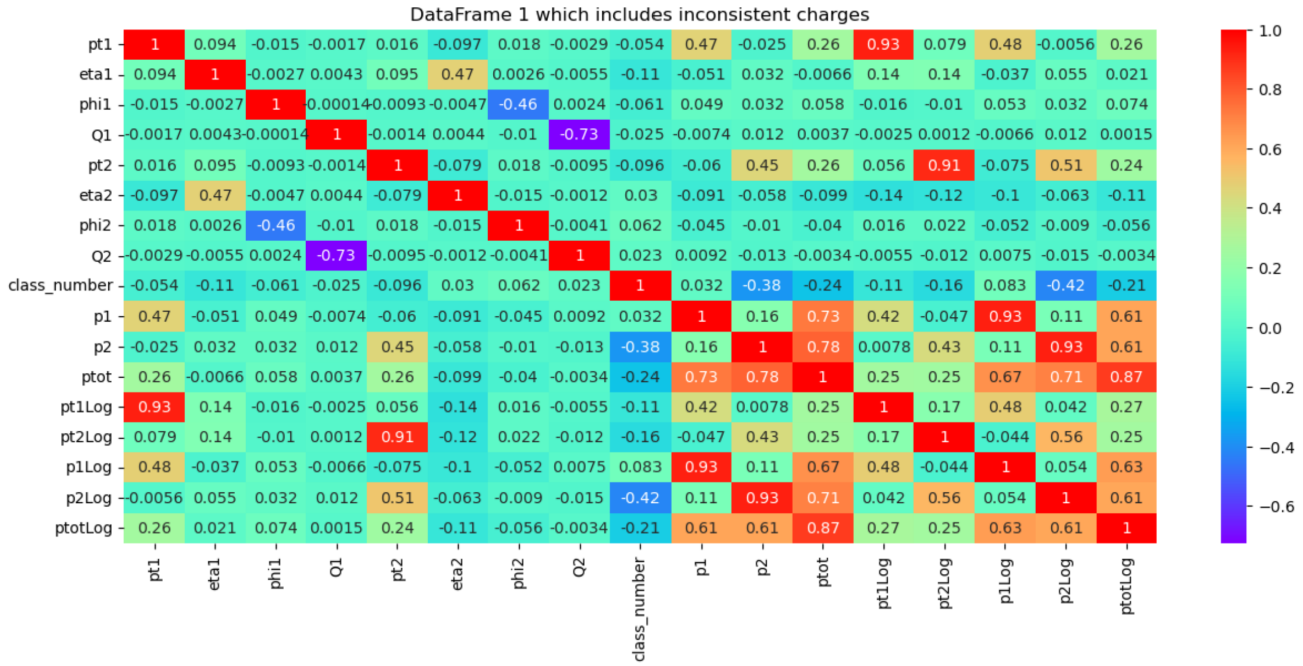


Figure 3: Pandas heatmap showing correlations between variables.

can be made for the two charges (Q1 and Q2), where this time the symmetry is a more abstract symmetry related to the decay. These features were therefore removed from the data set.

After the feature selection process, our DataFrame was left with seven features and one label.

3 Learning methods

3.1 Description of learning methods

The two machine learning methods used in this project were a support vector classifier (SVC) and the K-Neighbors classifier.

SVC classifies data-points based on their separation in a number of features and requires classification labels. SVC separates data-points of different classes by creating a hyperplane with margin boundaries. Any data-point on or within the margin boundaries is called a support vector. A pair of margin boundary hyperplanes are decided such as to minimise the number of data-points within the margin boundaries, and on the wrong side of the margin boundaries, as well as maximise the distance separating the margin boundaries. The classification boundary is qualitatively the hyperplane which resides in the middle of the two margin boundaries [8].

A K-Neighbors classifier essentially classifies any particular data-point based on the nearest k data-points surrounding it. The data-points may also be weighted based on their distance to the point to be predicted using some distance metric.

3.2 Justification of learning methods

SVC and K-Neighbors are both natural choices for labelled classification problems. SVC is versatile, with different kernel functions [8] and is typically better than logistic regression due to its use of support vectors and margin boundaries. K-Neighbors, whilst very simple, can make an effective classifier. K-Neighbors does not train a model so it is very easy to implement and "training time" is negligible. K-Nearest Neighbors is also a non-linear classifier [10], making it a good option in scenarios where data of different classes cannot be separated by a linear hyperplane.

3.3 Model assessment

k-fold cross-validation is a method by which data is split into k groups, and each group is used as a validation data set with the other k-1 groups acting in combination as a training data set [10]. The average of the accuracies on these k validation groups gives a good indication of the ability of the model to predict new data without it being exposed to the test data.

Five-fold cross-validation was used in order to assess the accuracy of the model when selecting the best hyperparameters. The final assessment of the models was mainly influenced by its accuracy on the previously unseen test data. The accuracy on training data was also considered to provide an insight into whether overfitting was present.

3.4 Hyperparameter tuning

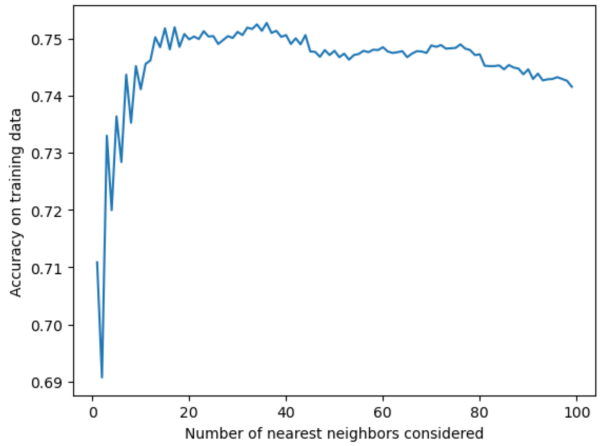


Figure 4: Plot showing how cross validation accuracy changes with n_neighbors for the K-Neighbors classifier. Similar plots were produced for the SVC case.

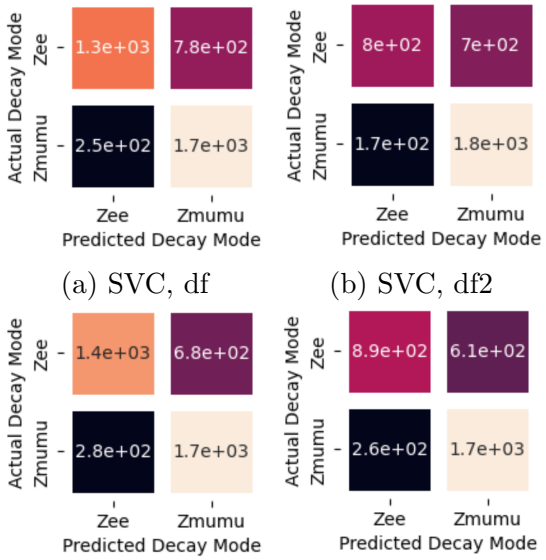
One hyperparameter was altered for each model. In order to choose the appropriate value in each case, a range of values were tested systematically and a plot was produced (plots shown in Figure 4). In the SVC case, the regularisation parameter, C, was adjusted, whilst in the K-Neighbors case the number of neighbours considered, n_neighbors, was adjusted. When finding the ideal value of the regularisation parameter in the SVC case, the order of magnitude was determined before the final value used.

The final hyperparameter values chosen for the regularisation parameter and the number of neighbours considered respectively were C=100 and n_neighbors=17.

Although regularisation parameter values beyond 100 increased the accuracy of the model on the training data, C=100 was chosen as large values of the regularisation parameter can lead to overfitting.

4 Results

Figures 5 and 6 show the final results for the two algorithms on the two different data sets. When inspecting the results, keep in mind that df was the DataFrame with the charge paradoxical points kept in, whilst df2 is the DataFrame with the charge paradoxical points removed.



(a) SVC, df (b) SVC, df2

(c) K-Neighbors, df (d) K-Neighbors, df2

Figure 5: Confusion matrices showing the numbers of events correctly and incorrectly predicted for each catagory.

4.1 Analysis

Although the accuracy is not ideal, it appears that overfitting is not a big problem based on the

	df	df2
SVC	Training data accuracy: 0.746 Test data accuracy: 0.743	Training data accuracy: 0.754 Test data accuracy: 0.746
K-Neighbors	Training data accuracy: 0.780 Test data accuracy: 0.758	Training data accuracy: 0.780 Test data accuracy: 0.748

Figure 6: A table summarising each feature. Many variable descriptions have been adapted from the Kaggle data set.

fairly minor deviation between accuracies on test and training data in each case (maximum deviation of 3.2%). Removing the paradoxical points seemed to help a negligible amount (0.3%) in the SVC case, and hinder by a similarly small amount (1%) in the K-Neighbors case.

Both techniques yielded a very similar accuracy on the test data with K-Neighbors being 1.5% and 0.2% more accurate for df (charge paradoxical points left in) and df2 (charge paradoxical points removed) respectively.

5 Conclusion

The overall accuracy (74.3% - 75.8%) of the models created/ algorithms employed on the data was certainly not high enough to be useful in any meaningful way. The accuracy is also significantly lower than the project of Saraei O.B. who was able to use machine learning techniques on the same data set to achieve an accuracy of 90%.

The disparity could be partially due to time limitations with regards to hyperparameter tuning, as only one hyperparameter was adjusted for each algorithm. One overall limitation of this data set comes from the fact that it only presents a subset of collision information. It may be the case that for this data set it is impossible to use the provided features to separate points of different classes with high accuracy if it is true that significant decays of different classes could have identical features in certain ranges. In this scenario, more features, such as the energy or momentum of the products, would have to be included in the data set in order to achieve high accuracy. If this is not the case, then maybe other methods, such as deep learning, could be used to better distinguish the decay events.

6 References

1. atlas.physicsmasterclasses.org. (n.d.). International Physics Masterclasses. [online] Available at: https://atlas.physicsmasterclasses.org/en/zpath_lhcphysics2.htm#:~:text=Since%20Z%20is%20neutral%20the [Accessed 9 Mar. 2023]

2. Franceschini, Roberto & Kim, Doojin & Kong, Kyoungchul & Matchev, Konstantin & Park, Myeonghun & Shyamsundar, Prasanth. Kinematic Variables and Feature Engineering for Particle Phenomenology. 10.48550/arXiv.2206.13431. (2022)

3. Wong, C.Y. Introduction to High-Energy Heavy-Ion Collisions. World Scientific. (1994)

4. Workman, R.L. et al. (Particle Data Group), Prog. Theor. Exp. Phys. 2022, 083C01 (2022)

5. Advances in Multi-Variate Analysis Methods for New Physics Searches at the Large Hadron Collider, Reviews in Physics, Volume 7: Stakia A., Dorigo T., Banelli G., Bortoletto D., Casa A.,

Castro P., Delaere C., Donini J., Finos L., Gallinaro M., Giammanco A., Held A., Morales F.J., Kotkowski G, Liew S.P., Maltoni F., Menardi G.,... Weiler A. (2021)

6. Machine Learning in High Energy Physics Community White Paper: Kim Albertsson et al J. Phys.: Conf. Ser. 1085 022008 (2018)

7. Identifying two modes of Z boson(ACC 90%) Saraei O.B. Available at: kaggle.com/code/omidbaghchehsaraei/identifying-two-modes-of-z-boson-acc-90#KNeighborsClassifier (2023)

8. Scikit-learn (2019). scikit-learn: machine learning in Python — scikit-learn 0.20.3 documentation. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/index.html.

9. Logunova, I. (2022). K-Nearest Neighbors (KNN) Algorithm for Machine Learning. [online] Available at: https://serokell.io/blog/knn-algorithm-in-ml.

10. Brownlee, J. (2018). A Gentle Introduction to k-fold Cross-Validation. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/k-fold-cross-validation/.