An Investigation into Heavy Higgs decay using machine learning techniques

Jacob Gordon

201416406

A Thesis submitted in partial fulfilment of the requirements for the degree of

Bachelor of Science

Under the supervision of Nikolaos Rompotis

At the

Department of Physics

May 2022

Table of Contents

Declaration of Authorship

I, Jacob Gordon confirm that this paper “An Investigation into Heavy Higgs decay using machine learning techniques” is completely my own. I also confirm that:

- Any sources used to support my claims in this paper are stated in the Bibliography.
- I have not received external help on this project other than that of my supervisor.
- All code written for this project was my own or adapted from code sent by my supervisor.

Signed: Jacob Gordon

Date: 13/05/2022

Abstract:

Heavy Higgs is a theoretical particle beyond the standard model of physics, more specifically the Two-Higgs doublet model. Heavy Higgs was born out of the discovery of the Higgs boson in 2012. Since then, other Higgs particles similar to Heavy Higgs are seeking to be proven. One of the main methods of proving their existence is machine learning, which is a method of using computer algorithms to take input data, look for pattens and try and predict output data. The machine learning algorithms used within this project are support vector machine, Neural Network, Random Forest classifier and TPOT classifier. These methods lead to varying accuracy scores of 89%, 72%, 50% and 91% proving that some algorithms are much better than others. Despite some of these low accuracy scores the high accuracy scores of 89% and 91% are enough to prove the existence of Heavy Higgs within the dataset.

Introduction

The purpose of this project is to improve the search for the Heavy Higgs boson using a variety of machine learning techniques. This includes Support Vector Machine, Neural Networks and Random Forest classifier (more detail on this further down in the report). The specific decay scheme we will be focusing on in this report is as follows:

$$A \rightarrow ZH \rightarrow t\bar{t}$$

Decay scheme 1 – Heavy Higgs decay scheme

In the above decay scheme, A represents Heavy Higgs, Z another boson and H another Heavy Higgs. The $t\bar{t}$ signifies a top quark, anti-top quark pair, this quark pair is part of the background events from the decay scheme. All data analysed in this report is simulated data from the ATLAS experiment.

The data set used for this undertaking was extensive. There were 77 different sample numbers, each of which contained 56 different variables. Each one of these variables influenced the Heavy Higgs in one way or another. One of the big initial challenges in this project was deciding which variables we would select to analyse using machine learning. As some variables would positively contribute to our models while others would negatively contribute. Subsequently we would then take these variables to build the different models that aim to detect the Heavy Higgs and compare them. As Heavy Higgs is technically a theoretical particle, the best model will be a good indication as to wherever or not the particle does truly exist.

History and context:

The search for Heavy Higgs was born out of the discovery of the Higgs boson in 2012, which was detected in the large hadron collider via the ATLAS experiment [1]. Before this, the Higgs boson had been theorised by Peter Higgs in 1964 [2], although it would be decades until he was proven right and the particle was attributed to him. The large hadron collider is the world's most powerful particle accelerator. It consists of 27-kilometre ring of superconducting magnets to accelerate the particles [3]. Although many different particle accelerators have been proposed over the years, the large hadron collider was only built in 2008 and has only been in operation since 2010. It was built on the Franco-Swiss border at the European Organisation for Nuclear Research outside of Geneva.

A Toroidal LHC Apparatus or ATLAS for short is one of the general-purpose particle detectors at CERN and is one of the two particle detectors responsible for the discovery of the Higgs Boson [4]. ATLAS is a global effort, relying on scientists, universities and laboratories worldwide as well as at CERN.

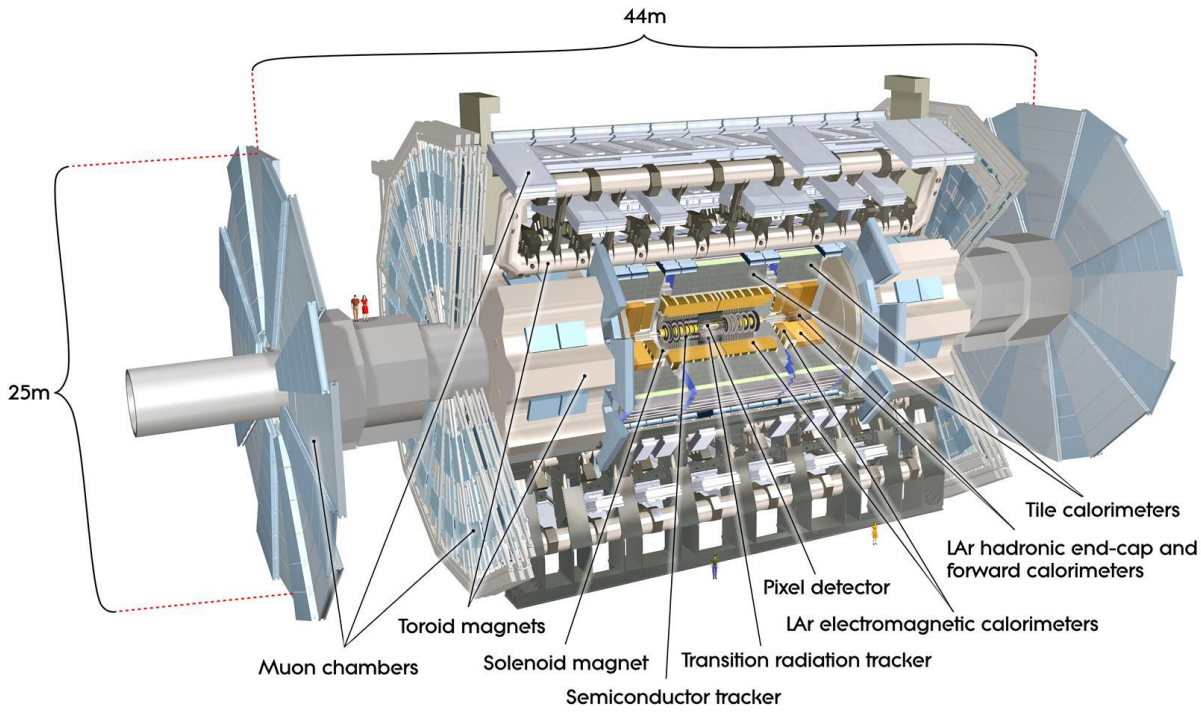

Figure 1 – A diagram of the ATLAS detector.

Figure 1 shows the full scale of the ATLAS detector. At 25m tall and 44m wide ATLAS is the largest particle detector ever constructed in terms of volume. A huge magnet system bends the path of the charged particles so the momenta can be measured. An enormous amount of data is generated from the detector [3].

The ATLAS experiment found the mass of the Higgs boson to be 125Gev. The detection of the Higgs Boson led the standard model to be challenged. The Higgs boson does exist within the parameters of the standard model, however alternative models have been proposed which include more or different versions of the Higgs. For example, Heavy Higgs and its decay scheme investigated in this project are featured in the Two-Higgs doublet model, which introduces a total of 5 Higgs bosons. 2 of which are present in decay scheme 1, as the A and the H are both Heavy Higgs.

Project motivation:

The motivation behind the project is to discover the best algorithm for detecting the Heavy Higgs by training our algorithms off previously generated data. Different types of machine learning techniques and pipelines will be used to evaluate the data and from that the best model will be chosen to give the best chance of detecting Heavy Higgs in the future. In the past machine learning has been used to try and improve the search for Higgs bosons [5]. This project uses a variety of different machine learning techniques (some which may not have been attempted previously) to find the best model.

Theory

Search for Heavy Higgs:

Heavy Higgs particles fall outside the standard model, the standard model only accounts for one Higgs boson after electroweak symmetry breaking. This does not comply with the $A \rightarrow ZH \rightarrow t\bar{t}$ decay scheme as there is two Higgs particles present. As a result, we must look beyond the standard model, specifically at the Two-Higgs doublet model (2HDM) mentioned earlier in this report. This model includes both the H and A Higgs present in our decay scheme. Both particles are technically theoretical being beyond the standard model so any evidence of their existence would advance our understanding of particle physics further.

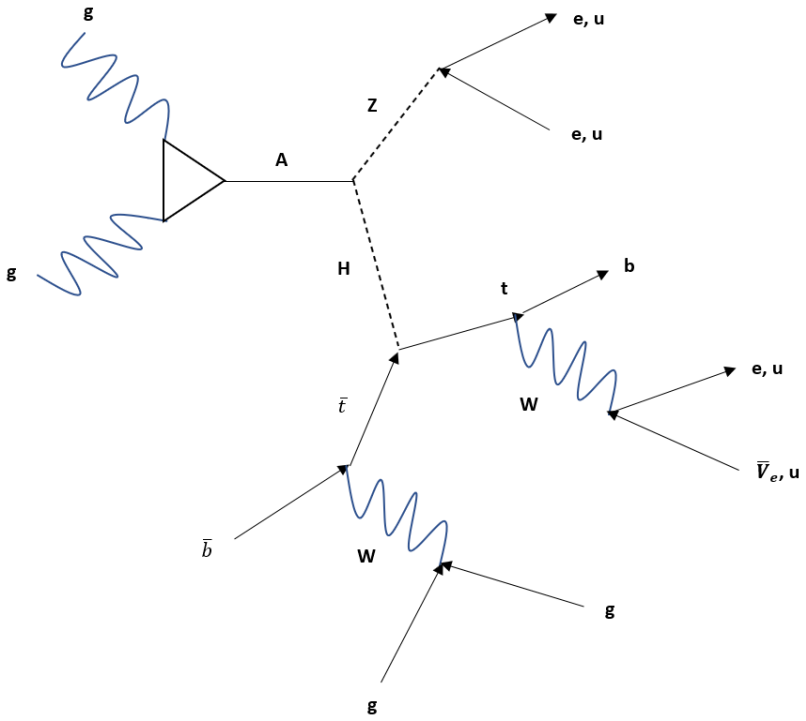

Figure 2 – A Feynman diagram of the $A \rightarrow ZH \rightarrow t\bar{t}$ decay scheme.

Figure 2 is a Feynman diagram showing the production of the $A \rightarrow ZH \rightarrow t\bar{t}$ decay scheme via gluon – gluon fusion. Gluon – gluon fusion produces 2 b jets which decay from H. A jet is a narrow cone of particles produced by the hadronization of the gluons. Gluon – gluon fusion is one of two production methods for our decay scheme, the other is b – associated production.[6] This is where a b quark and anti b quark annihilate to form A. This method is not included in this report as the data made available was for gluon – gluon fusion only.

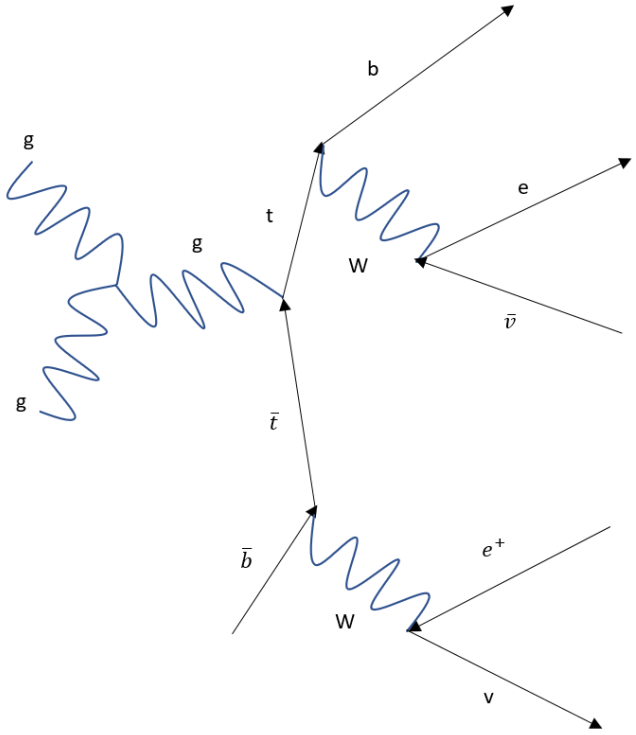

Figure 3 – A Feynman diagram to show the $t\bar{t}$ background process.

Figure 3 shows the final state in the decay scheme, $t\bar{t}$. The $t\bar{t}$ state represents the background process for the $A \rightarrow ZH$ decay scheme and takes up most events in the experiment. $t\bar{t}$ is one of the final states that can arise from $A \rightarrow ZH$ decay, the other being $V\bar{V}$ which represents the neutrino, anti-neutrino final state. This state however does not need to be considered for this project as all data sourced was for the $t\bar{t}$ state.

How machine learning works:

As explained briefly in the introduction, machine learning is the application of computer algorithms to analyse and find patterns in data without being given explicit instructions. This analytical technique is useful to analyse our dataset, this is because a massive dataset with 100,000's of entries cannot be accurately modelled by more traditional methods. The goal of a machine learning algorithm is to predict data. As the task set in this project is to predict the presence of Heavy Higgs, machine learning is an obvious choice.

There are two major types of machine learning. The first type of machine learning is the classifier. A machine learning classifier is a technique where the code is trying to identify a discrete event. In reference to this project, we are trying to identify the presence of Heavy Higgs, as the dataset being used is simulated ATLAS data, we already know Heavy Higgs is present. This makes our algorithm a classifier, more specify, as there can only be Heavy Higgs or not, this makes it a binary classifier. The other type is regression, this is where the algorithm tries to predict a set of continuous data, for example this might be temperature or some other constantly changing factor. However, as our project concerns only classifiers, it is not as important to this undertaking. It is important to state that all the algorithms used in this project follow a supervised approach, this means we already know the inputs and outputs in our data and the algorithm is trying to identify the output (I.e., Heavy Higgs).

Machine learning works by training and testing data, this means the data is split is taken into two parts at a 70% and 30% split respectively. The training set is where the algorithm learns and recognises the patterns in the dataset, it then applies this to the testing data in order to see how accurate the model was. This process is true over all machine learning algorithms.

Support Vector Machine:

Support Vector Machine is an algorithm that creates a line or "hyperplane" which separates data into classes. This clearly makes it a strong classifier, meaning it is perfect to analyse the dataset.

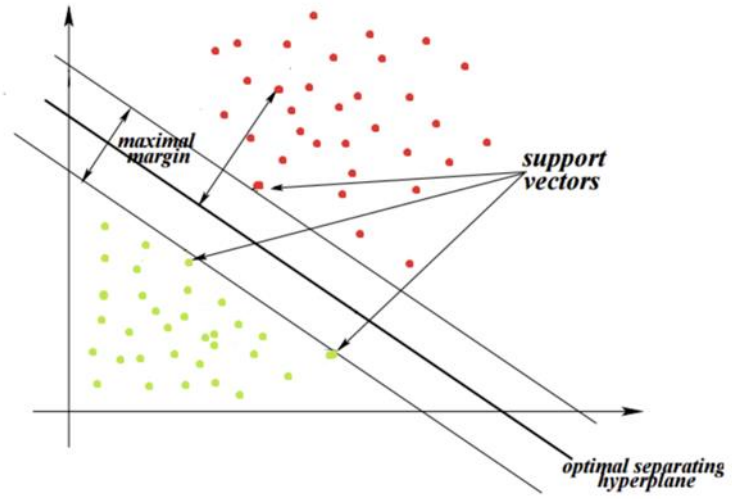

Figure 4 – A diagram of an ideal support vector machine.

In figure 4 you can see the hyperplane which separates the two variables. In our case support vector machine is a binary classifier, therefore it separates the data into two distinct classes either Heavy Higgs is present or not present. In figure 4 you can also see the datapoints closest to the hyperplane labelled support vectors, support vectors are called as such because they influence the position of the hyperplane, if they shift, so does the hyperplane [7]. Support Vector Machine is often one of the easier machine learning algorithms to implement and it outputs a good final accuracy. This made it a strong first choice to model our data as it would give a good indication to the presence of Heavy Higgs.

Neural Network:

A Neural Network is a machine learning algorithm inspired by the human brain to map a series of inputs to outputs. Neural Networks typically fall under the category of deep learning, which is a subset of machine learning.

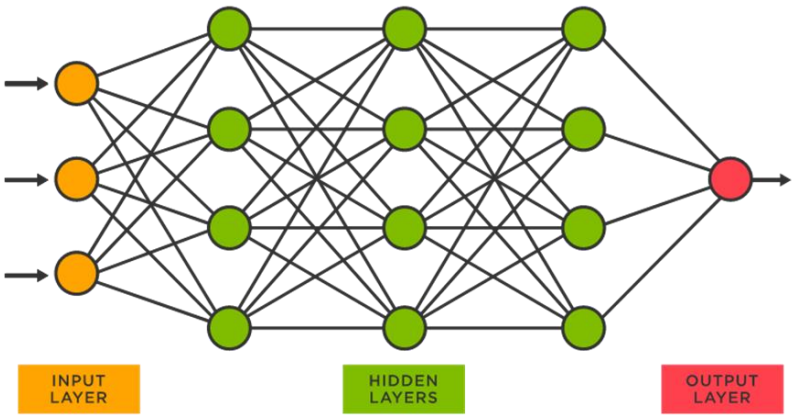

Figure 5 – A simple diagram showing the process of a Neural Network.

Deep learning algorithms such as Neural Networks contain an input layer, multiple hidden layers and an output layer, which can be seen in figure 5. Each individual node acts as its own linear regression module, containing an input, output, weights and a bias (or threshold) [8]. Weights determine the importance of a given variable and bias which allows us to shift the activation function. The activation function is the point at which the algorithm decides that a certain variable is significant or not.

Neural Networks are one of the more advanced machine learning algorithms. As a result, neural networks have the ability to model complex, non – linear data (such as the one in our dataset) to produce a model more accurate to reality. As our dataset is quite large and complicated a Neural Network would be ideal for accurately identifying Heavy Higgs particles.

Random Forest Classifier:

The final algorithm chosen was the Random Forest classifier. This is a technique where the algorithm creates many different decision trees and takes the final output as an average of all the trees. As the name suggests, each decision tree has an element of randomness involved when it is created.

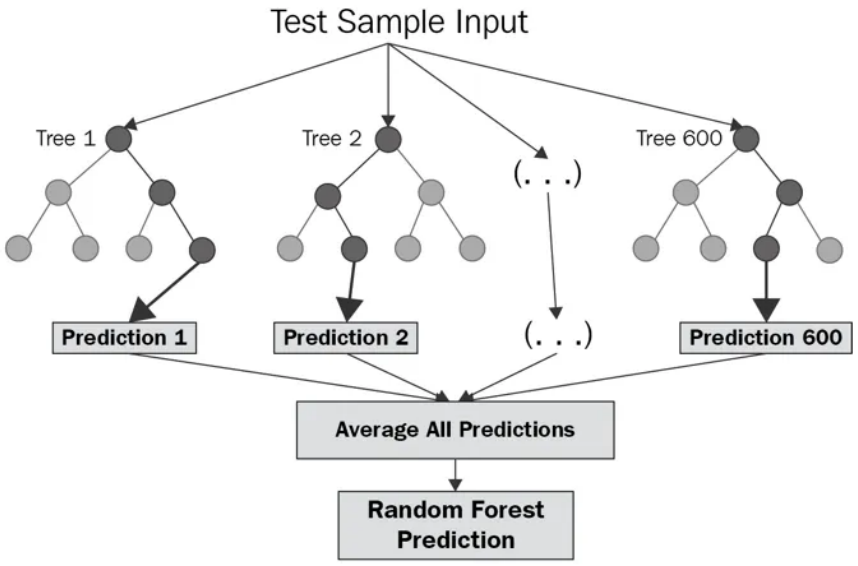

Figure 6 – A diagram of a Random Forest classifier.

Figure 6 shows the process of the Random Forest classifier. As our problem is binary, each individual decision tree will make a prediction as to whoever or not Heavy Higgs is present and assign it a value based on its estimation. It will them sum up all the decision trees and assign a final accuracy to the model.

TPOT classifier:

TPOT (Tree based Pipeline Optimization Tool) is an automatic machine learning process. This means that instead of manually choosing and coding a machine learning model for our data, this tool will automatically run 1000's of different machine learning techniques and conclusively choose the best model for our data.

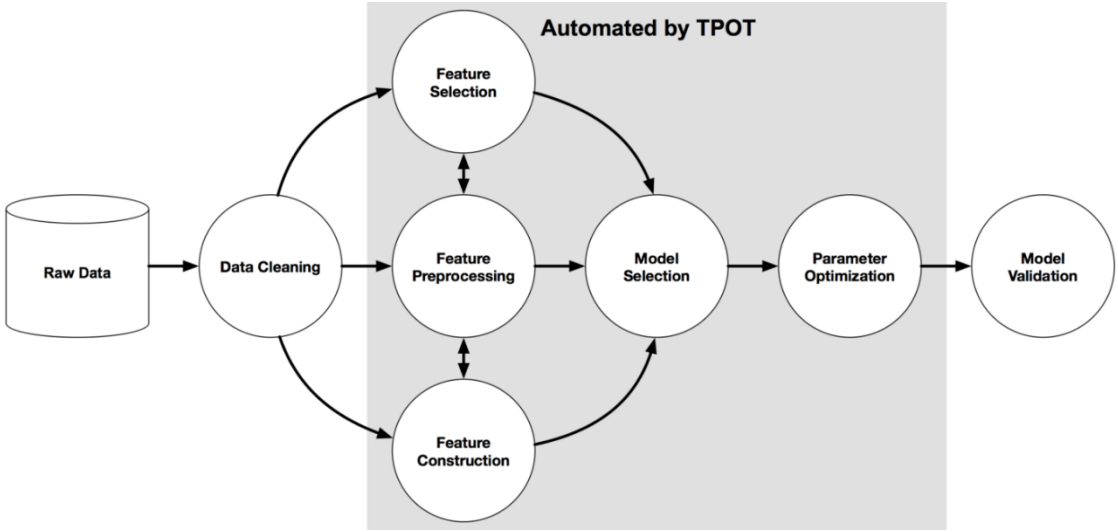

Figure 7 – The TPOT machine learning process.

In figure 7 you can see the machine learning process; this is the process that all the machine learning techniques have had to follow in order to produce an accurate model. The advantage of TPOT is that the difficult undertaking of choosing a machine learning algorithm and tuning the hyperparameters to fit the data is done automatically. Hyperparameters are variables used to control the learning process for specific algorithms [9]. For example, this could include the test – train split ratio, the activation function, or the number of generations in an algorithm.

Results and analysis

Significance

Before any models could be made, we had to decide which variables were the most significant. This was important because the greater the significance of a certain variable the more it positively impacted the machine learning algorithms. It is important to reduce the number of variables used down because variables with a low significance would not contribute to our models and some may even negatively contribute. As a result, the top ten most significant variables from the dataset were chosen to be analysed. All machine learning algorithms were run using these variables. To calculate significance the following equation was used:

$$Z = \pm\sqrt{nln\left(\frac{n}{b}\right) - (n-b)}$$

Equation 1 – The significance equation

Equation 1 [10] was used to calculate the significance of each variable where n was the signal number plus background and b was just the background. This equation is derived from the gaussian approximation without uncertainty (full derivation in Appendix A), this is because the more significant variables follow a gaussian distribution. Equation 1 was used to get a linear significance which is a good approximation but to improve the significance of the result was taken in quadrature as well:

$$\Delta Z = \pm\sqrt{(\frac{dZ}{dn})^2 + (\frac{dZ}{db})^2}$$

Equation 2 – The significance in quadrature

Equation 2 shows the significance in quadrature. To achieve this the partial diffraction of Z was taken with respect to both n and b (full derivation in Appendix A). It should be noted that for both of these equations the more significant a given variable the closer it lied to zero. Typically, when calculating significance, the more significant variables lie closer to one, however in this case the opposite is true. This is proven throughout the results section as you can see how the more significant variables line up with its corresponding plot being more gaussian like, it is also proven in the actual machine learning models as accurate models were produced using the most significant variables selected.

Variable name	Significance (linear)	Significance (quadrature)
mVH	0.013555	0.004698
mH	0.014306	0.004958
topHad_m	0.015438	0.005351
mWHad	0.016488	0.005714
mWlep_nu	0.016900	0.005857
dm_VH_H	0.018144	0.006288
pTJ3	0.020063	0.006953
pTWHad	0.020170	0.006990
pTVH	0.020390	0.007067
pTH	0.020578	0.007132

Table 1 – The top ten most significant variables

Table 1 shows the top ten most significant variables. These were the variables used in all machine learning techniques throughout this project. To further reinforce that these variables are the most significant we can compare them to their corresponding ATLAS graphs which plots the number of events against a given variable. It should be noted that all initial graphs presented in this results section are for signal number "AZH_lltt_mA1200_mH600". This signal number was chosen because it contained a large amount of signal data. The presence of signal data means we are more likely to detect Heavy Higgs making this an ideal first choice. It would also give us a good baseline to compare other sample numbers to after all initial analysis has been done.

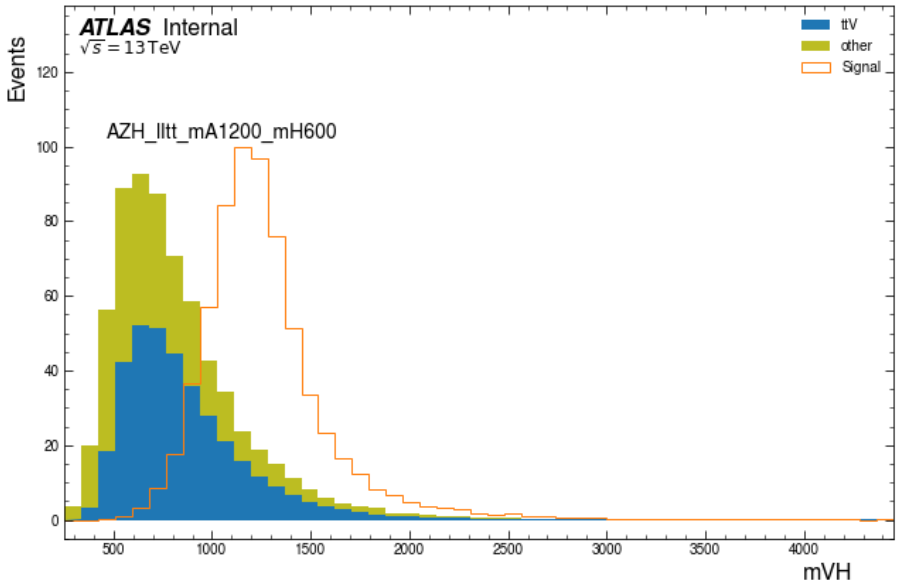

Figure 8 – A graph to show the significance of mVH.

Figure 8 shows a plot of the most significant variable mVH. From this plot alone, you can see that it is very significant as both the signal and background follow a gaussian distribution. While this does make the variable more significant, what makes this variable particularly more significant than the others are the fact that the signal is removed from the background. This can be seen on the graph as the signal distribution appears to be shifted to the right of the background. This shows us that for this particular variable the signal (Heavy Higgs) does not blend with the background making it easier to detect and therefore more significant. All other variables in the top ten follow a similar distribution to mVH.

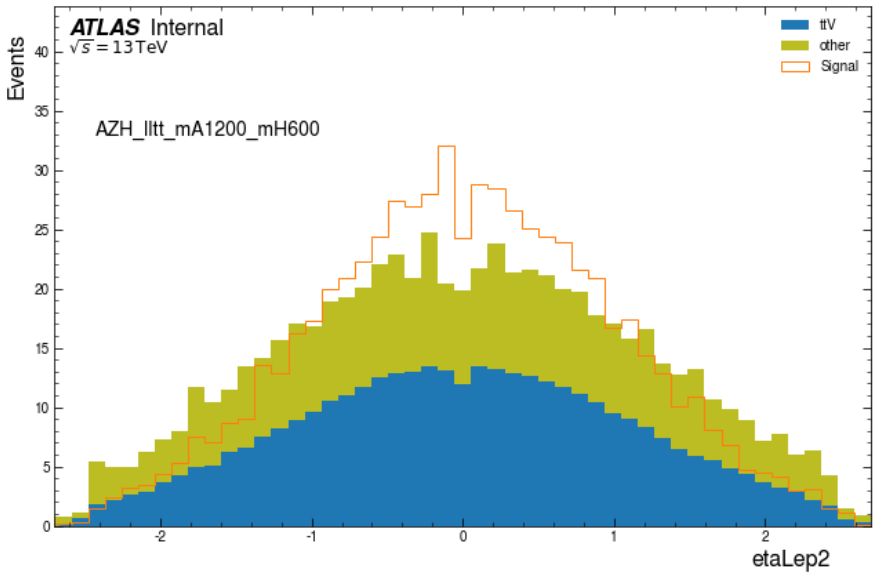

Figure 9 – A graph to show the significance of etaLep2.

Figure 9 shows the significance of the variable etaLep2. This is a much less significant variable with a linear significance of 0.378329 and quadrature significance of 0.132558. The purpose of this figure is to demonstrate the difference in significance. When you compare figure 9 to figure 8 it is immediately apparent that figure 8 is much more significant without even comparing the significance values. In figure 9 you can see that the gaussian distribution is much wider than that in figure 8, you can also see that the signal distribution is not removed from that of the background meaning it would be difficult to identify Heavy Higgs using this variable.

This figure also sees to justify the significance equations on page 14. The most significant variables seen in table 1 correspond to the most gaussian graphs, variables not on this table such as the one in figure 9 show a poor gaussian distribution, hence why the significance is so low. For further evidence of this, all graphs showing the significance of all variables are in the Appendix. It is important to only take the top ten most significant variables as taking anymore may harm the accuracy of the machine learning models.

<u>Correlation heat map</u>

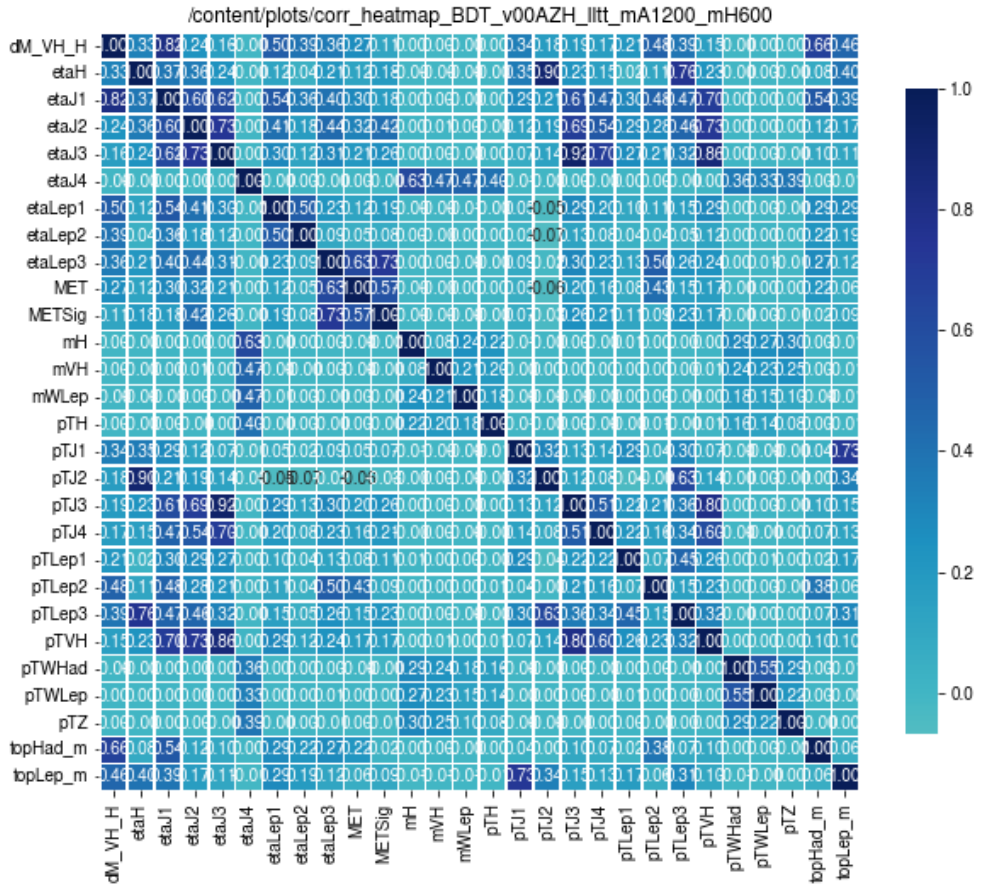

Figure 10 – Correlation heat map of all variables.

Figure 10 shows a correlation heatmap of every variable in the dataset. A correlation heatmap shows the significance of one variable compared to another. The dark black line that goes diagonally through the heatmap is the significance of one variable when compared to itself, this obviously nets a significance of one. You can see on the scale to the right of figure 10 that the darker the colours the more significant the variable. For example, when you compare METSig to etaLep3 it has a significance rating of 0.73 meaning that these two variables are significantly related. Some squares on the heatmap are a more turquoise colour, these are variables that are negatively correlated, for example etaLep1 and pTJ2. This is important because any variables that negatively correlate will have a negative influence on the machine learning models.

The correlation heatmap is a visual representation of why it is important to reduce down the number of variables used. Many variables in this dataset have no correlation with each other, so if they were included in our models, it would not improve them and in some cases the variables are negatively correlated, meaning the models would be worse. Based on this information and the significance calculated from equations 1 and 2 table 1 was concluded to be the best set of variables to use.

Data preparation

As already mentioned, the simulated ATLAS dataset used in this experiment was massive and without any data manipulation was disarranged. To correct this the signal data and background data had to be extracted and rearranged into a matrix that would be easier to analyse. This was done by slicing the dataset to extract only the signal numbers and event numbers, the event numbers correspond to the collision of the particles. Once this was done the specific sample number used was extracted, in our case "AZH_lltt_mA1200_mH600", this was done using a for loop. This signal data was then concatenated with the background data to create one big array that could be analysed. It is important to mention that this sample was chosen as any tag with "AZH" contained signal whereas any that did not was background only.

This array was 100,000's of entries long. One of the problems identified during this process is that the machine learning code took a considerable number of hours to run, to the point where it would be unrealistic to analyse the whole array. To counteract this only 100,000 entries were analysed, specifically between 300,000 and 400,000 as this section contained a large amount of signal data.

Support vector machine

The first machine learning technique used was support vector machine. To graphically show how well the model performed a ROC was created.

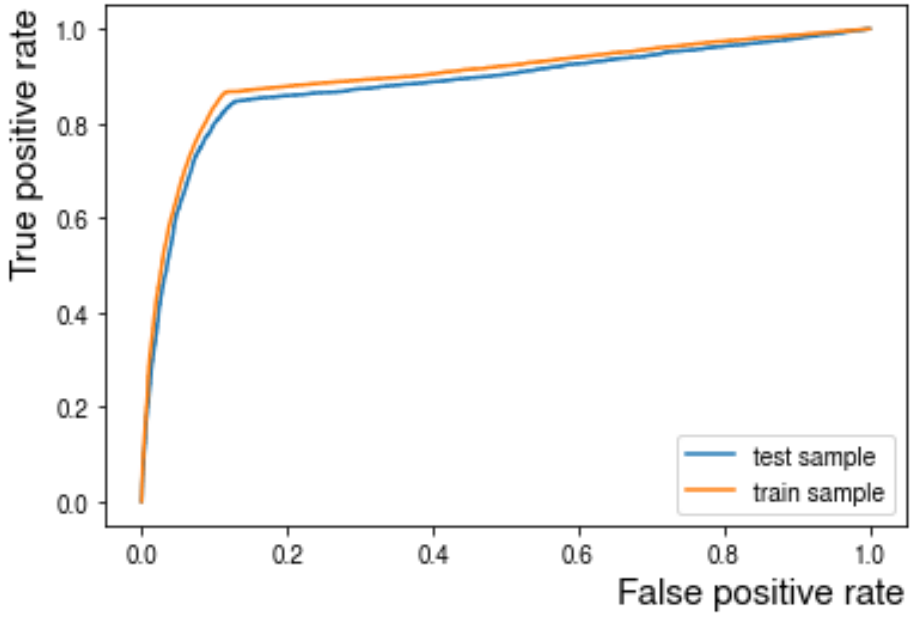

Figure 11 – A ROC curve to show the accuracy of the support vector machine model.

A Receiver Operating Curve or ROC for short shows the diagnostic ability of a binary classifier, in this case the support vector machine algorithm. The ROC has true positive rate on the y axis and false positive rate on the x axis. For the model to be accurate both the test and train sample have to curve closer to the true positive rate. The orange line represents the train sample and the blue the test sample. Typically, the train sample always lies closer to a true positive rate as it has had more data to analyse (70%, 30% split).

Figure 11 shows a very accurate ROC for support vector machine; both the true and test samples fall very close to the true positive rate. The train sample had an accuracy of 89.8% and the test sample had an accuracy of 88%. When support vector machine was applied to the whole dataset this gave us an accuracy of 89%, this means that the model is accurate 9 out of 10 times. In the greater context of our project that means that Heavy Higgs is detected with an 89% accuracy using this model.

During the development of this algorithm a plot similar to that of figure 4 was to be created. This was to show how the algorithm splits the data into Heavy Higgs and not Heavy Higgs and visually show how the algorithm works. This however is not as simple for our dataset as the example in figure 4, this is because figure 4 only has two variables whereas we are using 10 in

our analysis. This makes it difficult to easily split the data as the hyperplane can only split data in 2 dimensions whereas we had 10. To try and combat this a dimensional reduction analysis was performed [11]. This is a method of reducing down the number of variables in the training data to 2. This is done by a technique called feature selection that selects the most significant variables and reduces the training data down to them.

The final result of this however was quite poor and did not resemble the graph in figure 4, instead of there being a clear distinction between the detection of Heavy Higgs and not Heavy Higgs, both sets of data were mixed together. In this case data is lost in dimensional analysis reduction, which would explain the poor result.

Neural Network

The second machine learning technique was the Neural Network. For this algorithm a model loss graph was made.

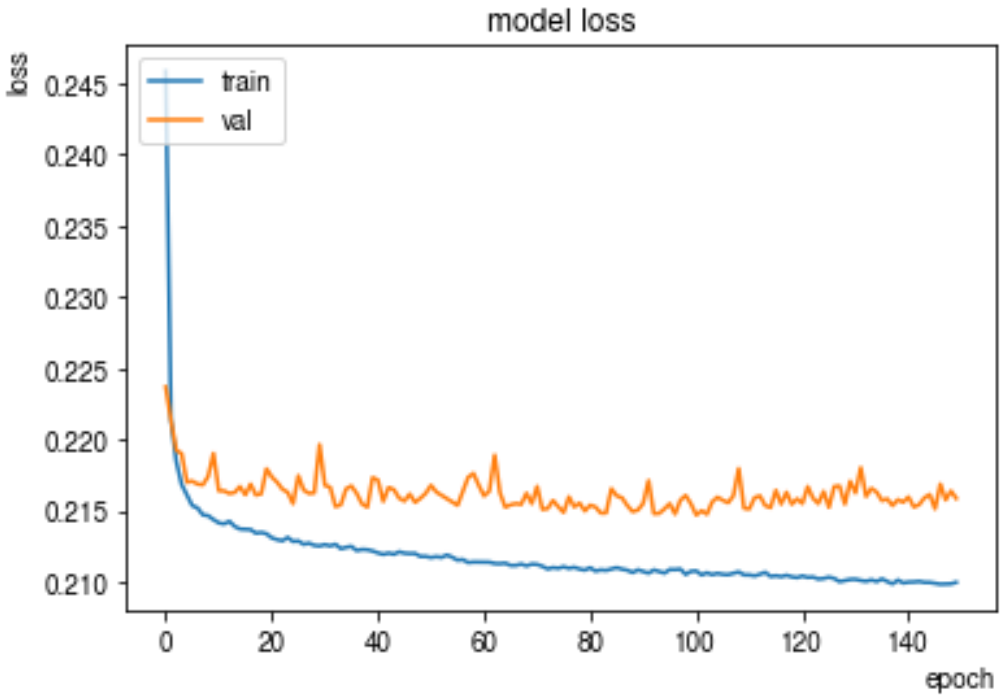

Figure 12 – A graph to show the model loss of the Neural Network.

A model loss graph shows how accurate the algorithm has been per epoch. The y-axis shows the loss of the model, loss shows how poorly the model performed when applied to a single example. The x-axis shows the epoch. One epoch is one loop of the Neural Network, this means the Neural Network trains off the training data to completion once, then does it again for a set number of epoch in our plot that was 150. The training data in figure 12 represented by the blue line shoots down rapidly and slowly begins to level off around a loss of 0.210, this shows

that the model has a good learning rate, meaning the training data does not take long to learn the patterns in the dataset. The orange line represents the accuracy of the model at each epoch, as you can see from figure 12 this does not necessarily get better with each consecutive loop as the highest peak is around 30 epochs.

The total accuracy of this model was 72%, this means the model is can only accurately predict the presence of Heavy Higgs 7 out of 10 times. This value is not high enough to give an accurate predication for Heavy Higgs as most scientific disciplines require an accuracy of 90% - 99% to be remarkable. The value of 72% is therefore much lower than the support vector machine value of 89% making support vector machine a substantially more accurate model.

Algorithms such as Neural Networks can suffer from overtraining. Overtraining is where the final model my not be representative of the whole dataset. A classifier such as a Neural Network may appear good when applied to the test data, but this is not always the case when applied to the whole dataset.

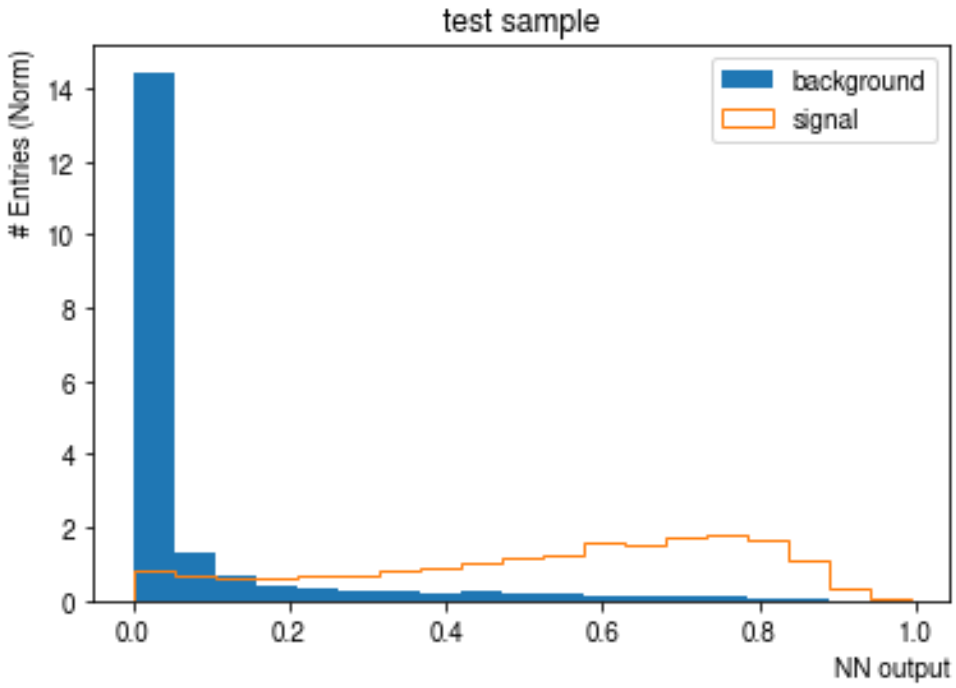

Figure 13 – Neural Network output when applied to the test data.

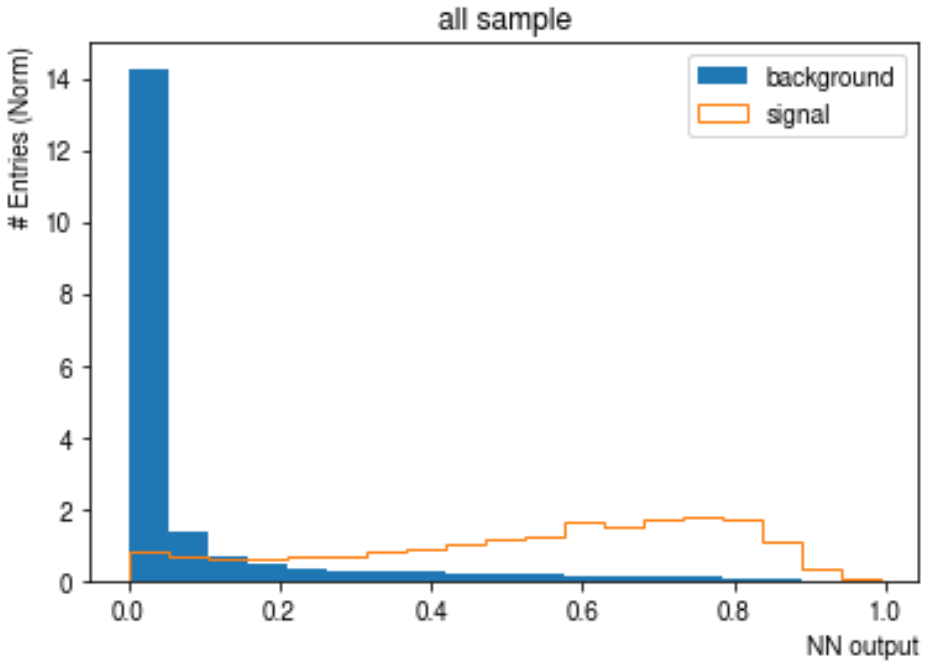

Figure 14 – Neural Network output when applied to the whole dataset.

Figures 13 and 14 show the output of the Neural Network when applied to both the test data and the dataset as a whole. You can see that both figures are identical. This is good because it confirms that the Neural Network has the same output when applied to both datasets, meaning it does not suffer from overtraining.

Random Forest classifier

The third machine learning technique was the Random Forest classifier, to show the accuracy of this model another ROC was produced.

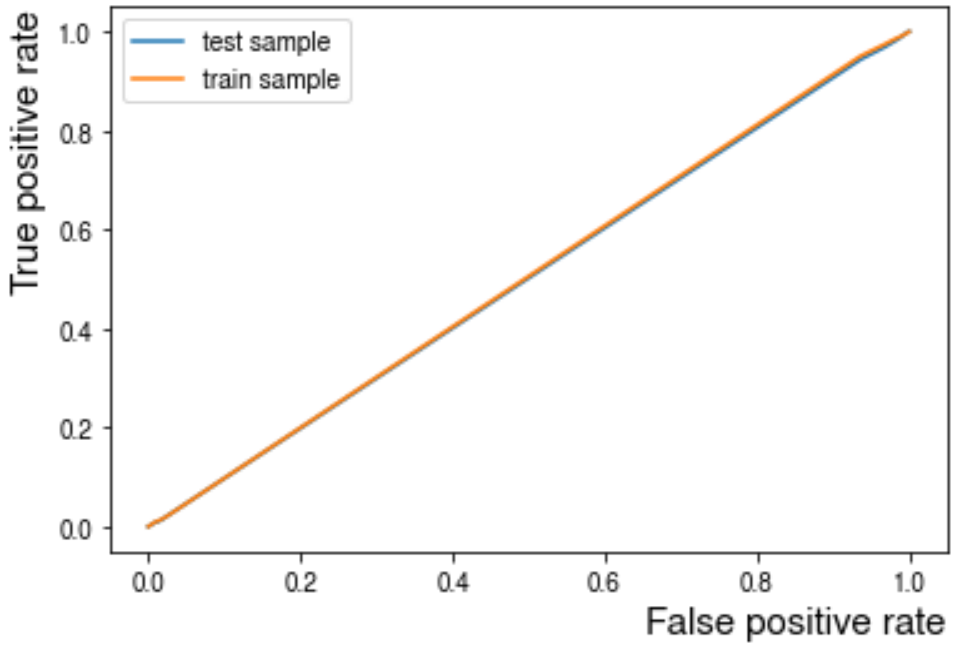

Figure 15 – An ROC to show the accuracy of the Random Forest classifier model.

Figure 15 shows the results of the ROC for Random Forest. The plot shows that both the test and train samples follow an almost linear model as they are diagonal across the graph. The test sample had an accuracy of 50.1% and the train sample 50.5%, this netted a final accuracy of 50% when applied to the whole dataset. This accuracy is obviously very poor, ROCs with an accuracy on or around 50% did not show a leniency to true positive or false positive rates. This means that they are completely random in their prediction, in other words this Random Forest model is no more accurate in predicting the presence of a Heavy Higgs than guessing at random in the dataset.

The reason for the incredibly poor performance of this model is likely due to how the algorithm works. As was mentioned in the theory section, a Random Forest classifier works by creating multiple decision trees and making a prediction on wherever or not Heavy Higgs is present, it then takes the average of this to give a final result. This process is random, as the name suggests, the dataset used was skewed as all the signal data was clustered in a specific region of the array making this model unfit to make a successful prediction.

TPOT classifier

The TPOT classifier produced the most unique and surprising results out of any of the algorithms, as previously mentioned TPOT runs 1000's of algorithm combinations to try and find the best one possible for the dataset. It does this in successive generations improving the accuracy of its search with each generation, for this project three generations were run. TPOT found a modified version of the Random Forest classifier to be the best fit for the dataset, this is rather unexpected as the previous attempt at Random Forest produced a very poor result. TPOT achieved this by varying the Random Forest hyperparameters and incorporating a Min-Max scalar algorithm [12].

$$X_\sigma = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Equation 3 – The Min-Max scalar equation

Equation 3 is the Min-Max scalar equation. This equation is a transformation feature that normalises the scaling, this means all the data will be normalised around the average mean in the dataset. This solves the problem of the data being skewed mentioned in the Random Forest

classifier section, this equation especially reformats the dataset to be suitable for Random Forest classification.

The hyperparameters that were changed to improve the accuracy of the Random Forest were as follows: The bootstrap was set to true; this means some samples may be used multiple times in construction of decision trees, the number of "leaf's" or depth of the trees was set to 8 and the criterion was set to "gini", this ranks the importance of each tree.

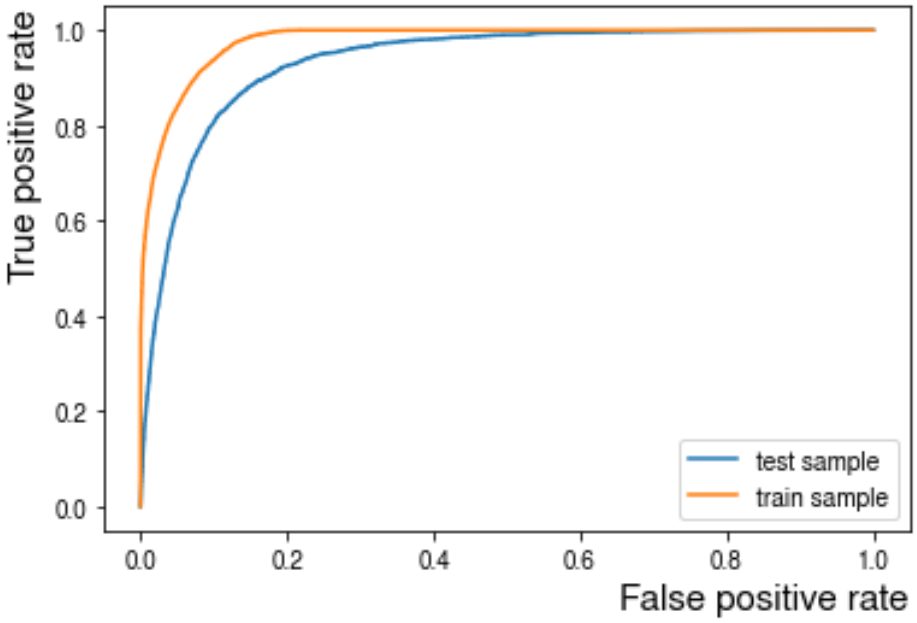

Figure 16 – An ROC for an improved Random Forest classifier.

Figure 16 shows an ROC for an improved Random Forest and the difference in accuracy is immediately apparent. The accuracy of the train sample was 97% and the test sample 93%, when applied to the whole dataset the final accuracy was 91%. This means that this Random Forest model can predict Heavy Higgs precisely more than 9 out of 10 times, this makes it our most accurate model and is a substantial improvement over the previous Random Forest model.

Changing the sample number

All analysis so far has been done for sample number "AZH_lltt_mA1200_mH600". For the last part of this study the sample number will be changed and the results compared to the results already obtained. The new signal number used was "AZH_lltt_mA600_mH400". This was chosen as similarly to the first sample number it contained a large amount of signal data. It was considered that a non-signal number sample would be compared such as "ttbarWW" to show the substantial difference between the two. However, as none of these sample numbers would contain the presence of Heavy Higgs this idea was discontinued.

Variable name	Significance (linear)	Significance (quadrature)
mVH	0.013850	0.004800
mH	0.016473	0.005709
topHad_m	0.019618	0.006799
dm_VH_H	0.019626	0.006820
mWHad	0.020827	0.007218
pTWHad	0.022208	0.007697
pTJ3	0.022528	0.007818
pTVH	0.022687	0.007863
pTH	0.022774	0.007893
pTZ	0.024938	0.008643

Table 2 – The top ten significant variables for signal number "AZH_lltt_mA600_mH400"

Table 2 shows the top 10 most significant variables for signal number "AZH_lltt_mA600_mH400". As you can see from the tables all of the variables (with the exception of pTZ) are the same as in table 1, some just appear to be more significant than others for this particular sample number. When the significance numbers are compared it is clear that they tend to lie very close to one another, this suggests that the top significant variables tend to remain the same over all sample numbers.

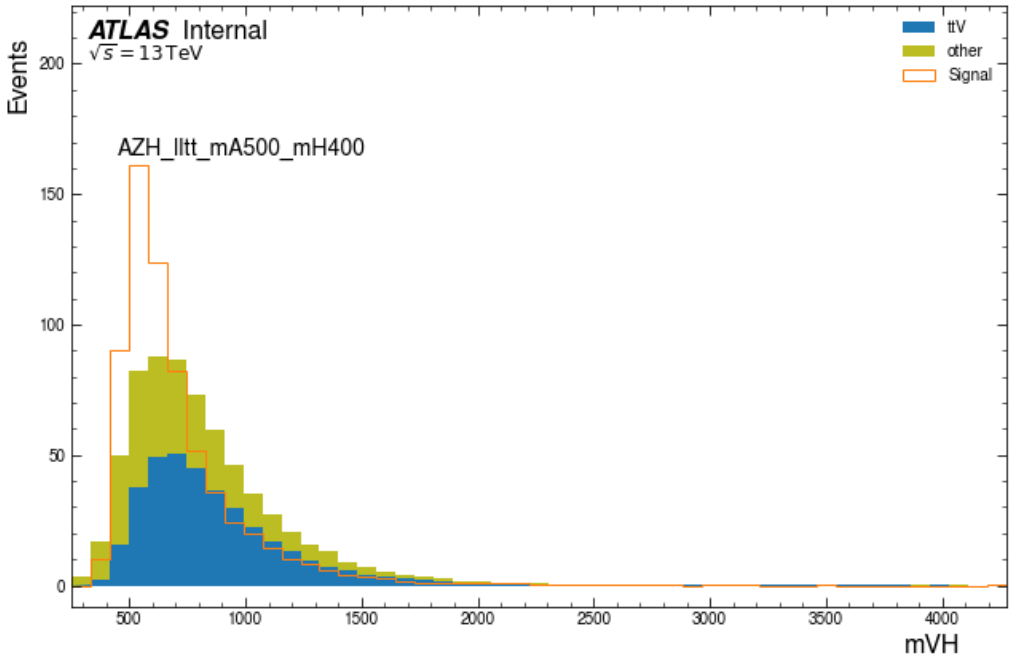

Figure 17 – A graph to show the significance of mVH for sample number "AZH_lltt_mA600_mH400".

Figure 17 shows the significance of mVH for sample number "AZH_lltt_mA600_mH400". When this is compared to the previous sample numbers equivalent in figure 8 it can be seen that there are many similarities. For example, both graphs then follow a strong gaussian distribution for both the background and the signal, this is what leads to its good significance rating of 0.004800, slightly worse than that of figure 8's 0.004698.

The reason for this difference is likely due to where the signal data falls on the plot. In figure 8 the signal data is removed from the background appearing to be "shifted" to the right, as a result it makes it easier to detect signal data. This is not true for figure 17 as the signal data appears to centre directly over the background data. From these preliminary results it can be concluded that "AZH_lltt_mA1200_mH600" is better for analysing signal data than "AZH_lltt_mA600_mH400".

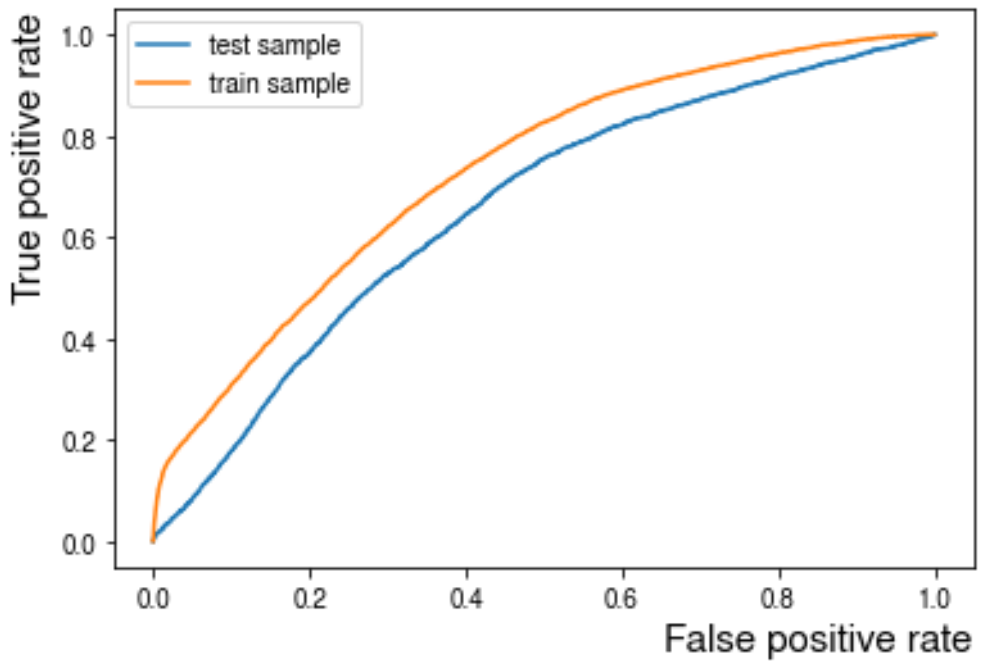

Figure 18 – Support vector machine analysis for "AZH_lltt_mA600_mH400".

Figure 18 shows the support vector analysis results for sample number "AZH_lltt_mA600_mH400". From the graph alone you can see that the accuracy is much worse than the previous sample number as both the test and train samples tend much further towards the false positive rate than in figure 11. The accuracy of the train sample was 73% and the test sample 65%. This means that the model is only accurate roughly 6.5 times out of 10. This is much worse than the 89% accuracy of the previous support vector machine algorithm and further supports the conclusion that this sample number is much worse for machine learning analysis.

The dataset contains a large amount of sample numbers, one sample number takes hours to run just 1 machine learning algorithm. As a result, it would be impossible to analyse them all to the extent of the first sample number. Other sample numbers are not dissimilar to the ones already analysed and the first sample number appears to yield excellent results, therefore further examination would be unproductive given the limitations of this project.

Discussion of results.

The results in greater context

The results of this project yield some excellent models but also some surprising conclusions, for example, the relatively low score of the Neural Network. The Neural Network only returned an accuracy of 72%. This figure is not necessarily low but when applied to the context of discovering a new particle it is nowhere near accurate enough. In the field of machine learning it is generally thought that Neural Networks tend to produce the best models due to their ability to analyse complex datasets, however, in this case they were outperformed by both the support vector machine and revised Random Forest classifier having scored 89% and 91% respectively. The reason for this low score could be due to the nature of the dataset, all the signal data which provides a positive recognition of Heavy Higgs was skewed to one end of the array, it could be for this reason that the algorithm had trouble identify complex patterns in the data. Another problem could be that the data was not correctly normalized before it was run through the Neural Network, this could have damaged the accuracy of the model.

Another surprising result is the improvement of the Random Forest Classifier from an incredibly poor accuracy of 50% to an accuracy of 91%. This staggering jump in accuracy demonstrates the importance of tuning the hyperparameters of the algorithm for specific machine learning problems. The Min-Max scalar equation mentioned in the results essentially "fixes" the problems encountered by the algorithm by first normalizing the data to allow the algorithm to function effectively. TPOT was instrumental in improving the algorithm. Before TPOT was ran it was not expected that Random Forest would take precedence given its poor performance earlier on in the project. However, with a few simple hyperparameter changes it proved to be the best algorithm to use, changing the hyperparameters for the Neural Network could also improve its accuracy.

At the beginning of this project, it was expected that some algorithms would perform much more poorly than others. For example, it was thought that the Random Forest classifier would be the worst functioning model. While this prediction did come true initially, it ended up being the best performing algorithm overall. The two algorithms that achieved the best results were the support vector machine (89%) and the improved Random Forest (91%), these accuracy scores are sufficient to suggest the presence of Heavy Higgs within the dataset, or at least

warrant further study. Because of this the project has been an overall success as two good models to identify Heavy Higgs have been found.

Problems encountered

One of the biggest problems faced during this project was timing, specifically the time taken to perform the machine learning. As mentioned already, the dataset had to be cutdown to only 100,000 entries in order to perform any analysis. This was because the code took an extensive amount of time to run, even for this reduced dataset. Before the dataset was reduced, support vector machine was run on the whole array. This ended up taking over 4 hours without any result. It was not feasible to wait this long to produce 1 model when there was many more that needed making. Even with the reduced dataset it still took upwards of 1 hour to produce the plots seen in figures 11 – 16. Before TPOT produced the revised Random Forest model it took upwards of 6 hours to complete, this is when it was applied to 100,000 entries, so if it was applied to the whole array of over 400,000 entries it could have taken up to 24 hours to run one algorithm.

The reason for this incredibly long waiting time is due to technological limitations. Machine learning algorithms require immense amounts of processing power to run and with large datasets (such as the one utilised in this project) require multiple computer cores and GPU's to run in a timely manner, this was not available during the project. To try and combat this Google Colab was used for all coding purposes as this makes use of the cloud to power its algorithms. However, Google Colab only allows the use of 2 cores, as a result this did not help to solve the problem by any great margin.

Another problem faced was the shear size of the dataset. As mentioned in the introduction, the dataset contained 77 sample numbers each of which contained over 50 variables, this means that there are millions of entries within this dataset. This problem feeds into the problem of long run times as the large amount of data results in the long run times. The size of the dataset also limits the analysis possible. Due to the total time taken to analyse one sample number to its extent, it would be unreasonable to analyse all of the sample numbers within the time frame of this project.

Another problem faced was the class imbalance within the dataset. The vast majority of the data was background, as all the algorithms are binary classifiers, they were setting no Heavy Higgs (background) as 0. As there is a tiny amount of signal data within the dataset when

compared to background, this causes the algorithms to train to find background much more than that of signal. This can cause the algorithms to become “biased” towards background and reduce the accuracy of the models. In an ideal scenario there would be a better ratio of signal to background data.

Further study

If this project were to be repeated it could be improved by using higher power hardware and software. This would not only allow the machine learning algorithms to be completed at a faster rate but would also allow a complete analysis of all sample numbers, giving a more accurate representation of the presence of Heavy Higgs. TPOT classifier should have also been the first algorithm employed, this would have allowed us to determine that Random Forest was the best algorithm to use straight away. This way more models could have been built of Random Forest instead of different models being tried at random.

The project could have been improved if more time was allocated. Working as an individual 12 weeks is not enough time to analyse the dataset to its full extent. Expanding the time allowed could also give the opportunity to go beyond the scope of the project, i.e., analysing and comparing other Heavy Higgs decay schemes to this one. One way to fix the problem of class imbalance mentioned in the problems section would be to utilise the use of SMOTE. Synthetic Minority Oversampling Technique [13] or SMOTE for short is an algorithm that resamples data. This means that instead of running the whole dataset through the algorithm it selects a more even amount of background and signal data. This helps prevent the algorithm from becoming biased.

Conclusion

To conclude, various machine learning algorithms were developed to evaluate the presence of Heavy Higgs particles within a given simulated ATLAS dataset. These machine learning models consisted of support vector machine, Neural Network, Random Forest classifier and an improved Random Forest classifier found via TPOT. These machine learning models scored an accuracy of 89%, 72%. 50% and 91% respectively. While not all these models are accurate, the ones that are, are very accurate and prove the existence of Heavy Higgs within this dataset. This makes the project an overall success. If this project were to be repeated more powerful software and hardware would have to be exploited to analyse more of the data in a quicker amount of time. This would lead to a higher accuracy for all algorithms used and gives the chance of even more algorithms being used for analysis.

Self-reflection

This project was chosen due to a growing interest in machine learning. Throughout my studies at university, I have gained more of an interest in python and the computing-based modules. This started out with simple graphing using matplotlib which was used in lab work and grew into more advanced analysis in my 2nd year computing project. Before this project I had no experience with machine learning, but I knew it was a skill that I wanted to develop as it is an increasingly important ability in the field of data science and data engineering. Due to this lack of expertise, it was initially difficult for me to grasp a lot of the key concepts behind machine learning. As a result, the first few weeks of the project were spent familiarising myself with machine learning. The same can be said about the physics behind the project. At the time it had been a while since I had studied particle physics, so a lot of the theories and terminology were unfamiliar to me, however meetings with my supervisor helped me to understand these quickly. Ultimately I am grateful to have had the opportunity to work on this project despite the challenges it presented.

Bibliography

[1] - ATLAS Experiment at CERN. (2012). *The Higgs boson: a landmark discovery*. [online] Available at: https://atlas.cern/Discover/Physics/Higgs#:~:text=On%204%20July%202012%2C%20the [Accessed 8 May 2022].

[2] - Aron, J. (n.d.). *A brief history of a boson: Timeline of Higgs*. [online] New Scientist. Available at: https://www.newscientist.com/article/dn22008-a-brief-history-of-a-boson-timeline-of-higgs/#:~:text=1964. [Accessed 8 May 2022]

[3] - CERN (2019). *The Large Hadron Collider | CERN*. [online] Cern. Available at: https://home.cern/science/accelerators/large-hadron-collider. [Accessed 8 May 2022]

[4] - ATLAS Experiment at CERN. (n.d.). *About*. [online] Available at: https://atlas.cern/about. [Accessed 8 May 2022]

[5] - Mott, A., Job, J., Vlimant, JR. *et al.* Solving a Higgs optimization problem with quantum annealing for machine learning. *Nature* **550,** 375–379 (2017). https://doi.org/10.1038/nature24047 [Accessed 8 May 2022]

[6] - Krauss, F., Napoletano, D. and Schumann, S. (2017). Simulating b -associated production of Z and Higgs bosons with the Sherpa event generator. *Physical Review D*, 95(3). doi:10.1103/physrevd.95.036012. [Accessed 8 May 2022]

[7] - Kumar, N. (2021). *Introduction to Support Vector Machines (SVMs)*. [online] MarkTechPost. Available at: https://www.marktechpost.com/2021/03/25/introduction-to-support-vector-machines-svms/. [Accessed 8 May 2022]

[8] - www.ibm.com. (n.d.). *What are Neural Networks?* [online] Available at: https://www.ibm.com/cloud/learn/neural-networks#:~:text=Neural%20networks%2C%20also%20known%20as. [Accessed 8 May 2022]

[9] - Nyuytiymbiy, K. (2021). *Parameters and Hyperparameters in Machine Learning and Deep Learning*. [online] Medium. Available at: https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac. [Accessed 8 May 2022]

[10] - ATLAS PUB Note Formulae for Estimating Significance. (2020). [online] Available at: http://cds.cern.ch/record/2736148/files/ATL-PHYS-PUB-2020-025.pdf?version=1 [Accessed 8 May 2022].

[11] - Brownlee, J. (2020). *Introduction to Dimensionality Reduction for Machine Learning*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/dimensionality-reduction-for-machine-learning/#:~:text=Dimensionality%20reduction%20refers%20to%20techniques%20for%20reducing%20the%20number%20of. [Accessed 8 May 2022].

[12] - Scikit-learn.org. (2019). *sklearn.preprocessing.MinMaxScaler — scikit-learn 0.22.1 documentation*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html [Accessed 8 May 2022].

[13] - Brownlee, J. (2020). *SMOTE for Imbalanced Classification with Python*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/ [Accessed 8 May 2022].

Appendix A - Derivations

The following derivation is for equations 1:

Gaussian approximation with uncertainty:

$$Z = \frac{n - b}{\sqrt{b + \sigma^2}}$$

Poisson model without uncertainty:

$$P(n, b) = \frac{b^n}{n!} e^{-b}$$

Uncapped test statistic:

$$q_0 = \pm -2ln(\lambda)$$

For large n, wilks theorem can be used:

$$Z = \pm \sqrt{-2\ln \lambda(0)}$$

$$Z = \frac{n - b}{\sqrt{b}} [1 + O(\frac{n - b}{b})]$$

Poisson-Poisson model with asymptotic formula:

$$L(s) = P(n, s + b)P(n, \tau b)$$

$$\lambda(0) \equiv \frac{L(0, \hat{b})}{L(\hat{s}, \hat{b})} = \left(\frac{n + m}{1 + \tau}\right)^{n+m} \frac{\tau^m}{n^n m^m}$$

Set $m = \tau b$ and combine with Z:

$$Z = \pm \sqrt{2(nln\left[\frac{n(b + \sigma^2)}{b^2 + n\sigma^2}\right] - \frac{b^2}{\sigma^2}\ln\left[1 + \frac{\sigma^2(n - b)}{b(b + \sigma^2)}\right])}$$

Discount error σ:

$$Z = \pm \sqrt{nln\left(\frac{n}{b}\right) - (n - b)}$$

For more detail see reference [10].

The following derivation is more equation 2:

Start with significance equation:

$$Z = \pm\sqrt{nln\left(\frac{n}{b}\right) - (n-b)}$$

$$\frac{dZ}{dn} = \frac{\ln\left(\frac{n}{b}\right)}{\sqrt{2}b\sqrt{nln\left(\frac{n}{b}\right)} - (n+b)}$$

$$\frac{dZ}{db} = \frac{-(n+b)}{\sqrt{2}b\sqrt{nln(\frac{n}{b})} - (n+b)}$$

$$\Delta Z = \pm\sqrt{(\frac{dZ}{dn})^2 + (\frac{dZ}{db})^2}$$

Appendix B – Further tables and graphs

Significance table

Variable name	Significance (linear)	Significance (quadrature)
mVH	0.013555	0.004698
mH	0.014306	0.004958
topHad_m	0.015438	0.005351
mWHad	0.016488	0.005714
mWlep_nu	0.016900	0.005857
dm_VH_H	0.018144	0.006288
pTJ3	0.020063	0.006953
pTWHad	0.020170	0.006990
pTVH	0.020390	0.007067
pTH	0.020578	0.007132
topLep_m	0.020927	0.007253
pTZ	0.021302	0.007383
PTZ_nu	0.021302	0.007383
mWLep	0.022093	0.007657
PTLep3	0.022405	0.007765
pTWLep	0.023513	0.008149
MET	0.023836	0.008261
pTJ4	0.025422	0.008811
pTLep1	0.025707	0.008909
pTJ1	0.025713	0.008911
pTLep2	0.036275	0.012572
pTJ2	0.036290	0.012577
mJ3	0.057704	0.019999
mJ4	0.061288	0.021241
mJ1	0.061881	0.021446
mJ2	0.088082	0.030527
METSig	0.146809	0.050880
mZ	0.196550	0.068123
etaVH	0.204543	0.070895
etaH	0.234312	0.081226
NSigJets	0.253737	0.087978
phiJ4	0.350659	0.122257
phiLep3	0.350659	0.122257

phiMET	0.350659	0.122257
phiJ2	0.350659	0.122257
PhiLep1	0.350659	0.122257
phiJ3	0.350659	0.122257
phiLep2	0.350660	0.122258
phiH	0.350660	0.122258
phiVH	0.350660	0.122258
phiJ1	0.350661	0.122258
etaLep3	0.378283	0.132540
etaLep2	0.378329	0.132558
etaLep1	0.378429	0.132596
NForwardJets	0.393087	0.138260
etaJ3	0.393088	0.138260
etaJ4	0.393089	0.138261
etaJ1	0.393089	0.138261
etaJ2	0.393090	0.138261
typeLep2	0.878970	1.353190
typeLep1	0.878970	1.353190
typeLep3	0.878970	1.353190
NTags	NaN	NaN
OSSF	NaN	NaN

Table 3 – All significance values for sample number "AZH_lltt_mA1200_mH600"

Most significant graphs not included in main report

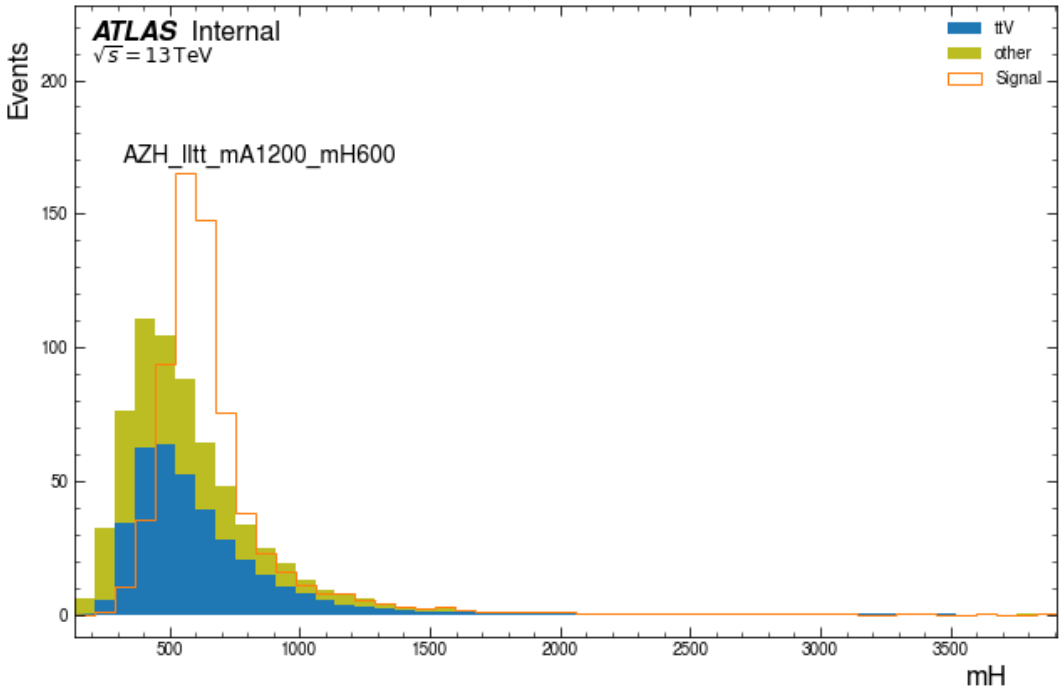

Figure 19 – A plot to show the significance of mH

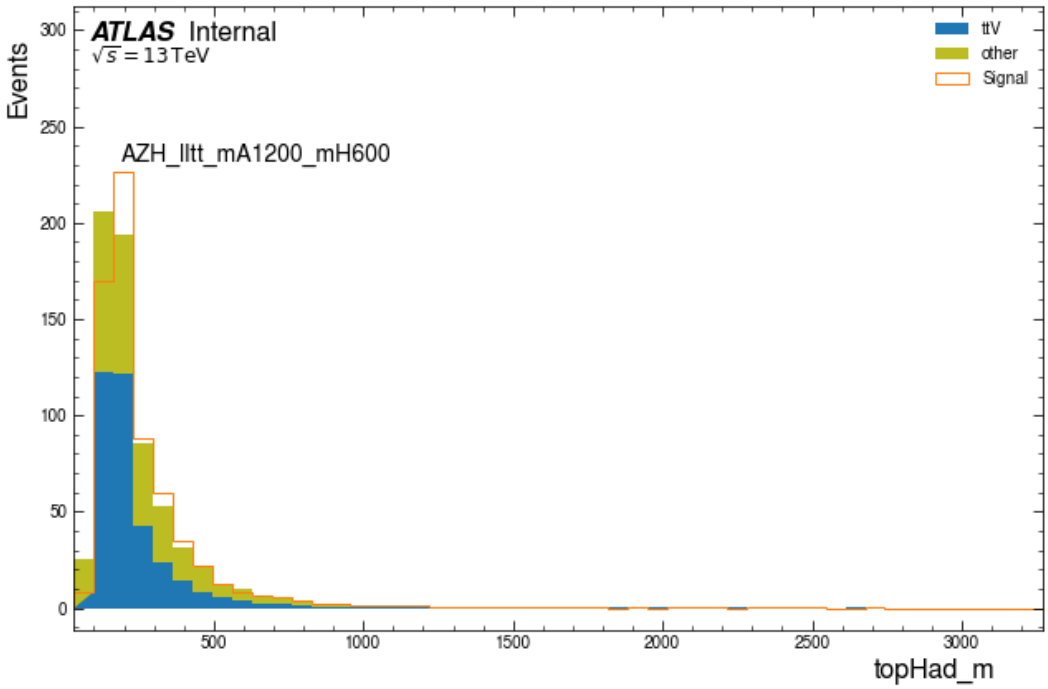

Figure 20 – A plot to show the significance of topHad_m

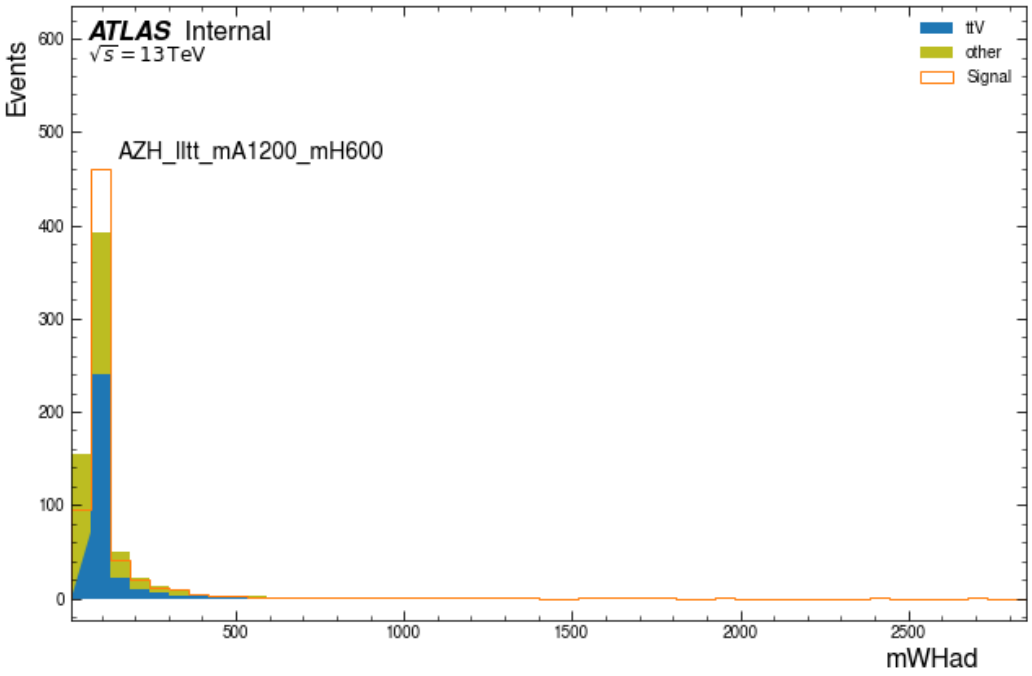

Figure 21 – A plot to show the significance of mWHad

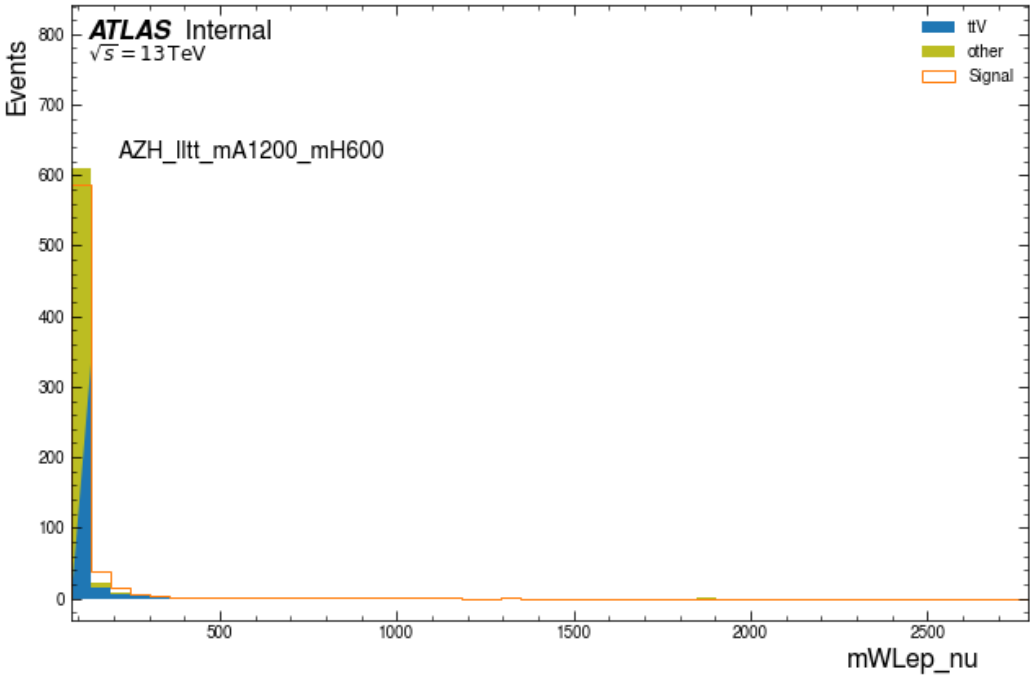

Figure 22 – A plot to show the significance of mWLep_nu

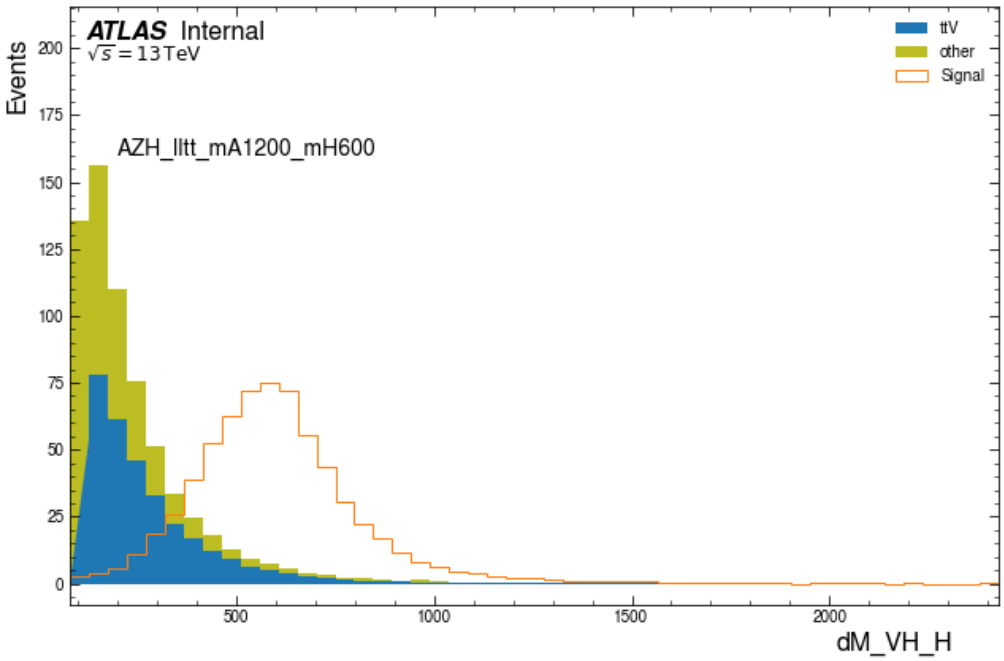

Figure 23 – A plot to show the significance of dm_VH_H

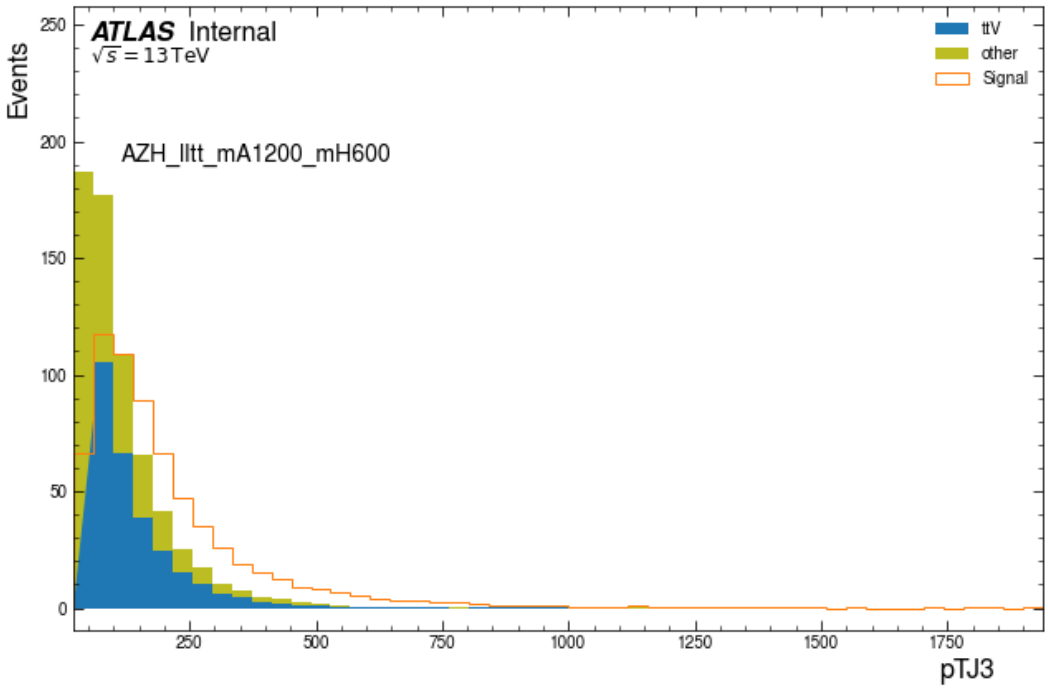

Figure 24 – A plot to show the significance of Ptj3

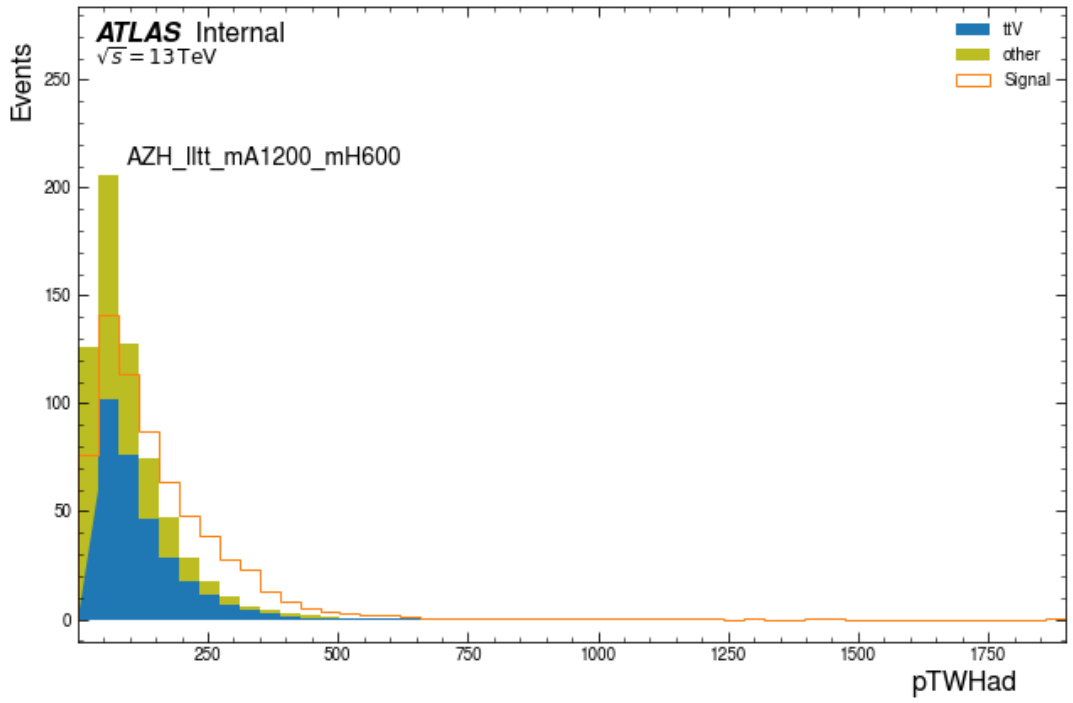

Figure 25 – A plot to show the significance of pTWHad

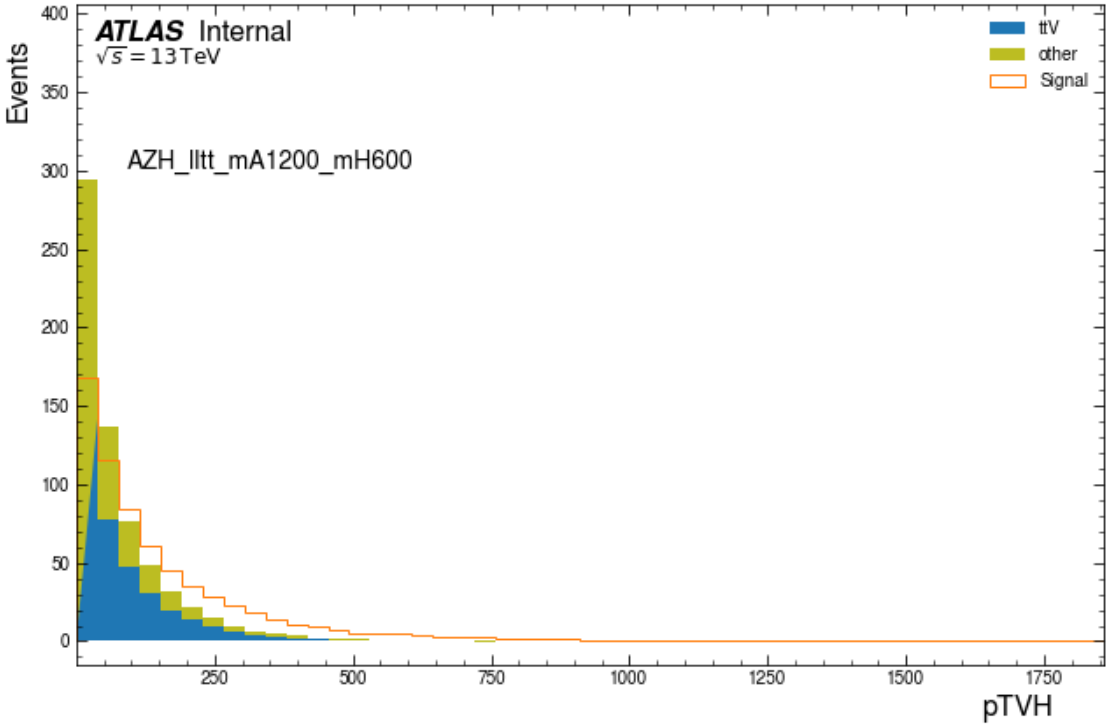

Figure 26 – A plot to show the significance of pTVH

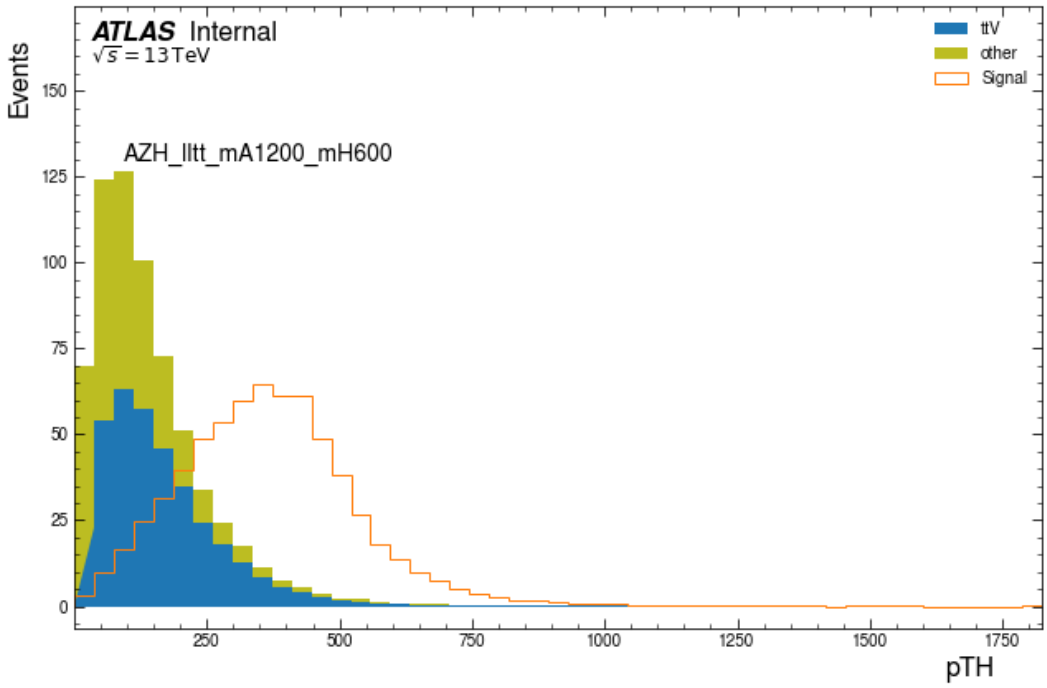

Figure 27 – A plot to show the significance of pTH

Appendix C – Project repository

A total of 3435 lines of code were written for this project. The whole project was written in Google Colab and was used to produce all significance plots, ROCs, and other models. The code was quite challenging to write, parts that were particularly challenging have been commented to explain it more clearly. This code can be accessed at:
https://github.com/JacobGordon1800/ATLAS-Heavy-Higgs/blob/main/ATLAS_Data_anaysis.ipynb

Appendix D - Presentation questions

Question 1: In your Feynman diagram you have the particle labelled A (Heavy Higgs) coupling directly to gluons. I throughout the particle coupled to mass, and gluons are massless, so is this diagram perhaps missing an intermediate step?

This is correct the Feynman diagram present in the presentation slides is incorrect. The correct Feynman diagram that should have been present is figure 2 in this report. The difference between the 2 diagrams is that figure 2 has a loop connecting the gluons to A. A loop bridges the gap between the gluons and Heavy Higgs by consisting of a particle which both particles can interact with. This particle could be for example $t\bar{t}$, the loop is represented by the triangle shape seen in the top right of figure 2.

Question 2: Your correlation matrix indicates how strongly all of your variables are correlated to one another, so by picking the variables which have the highest number from there are you not selecting the most correlated variables? Perhaps you selected the variables which correlate at most to a particular target variable?

The correlation heatmap shows how significant a variable is when compared to one another, not how significant each variable is in the whole dataset. The most significant variables shown in the presentation slides and shown throughout this report are the most significant variables for there given sample number. The correlation heat map seeks to support the results made from the significance equations (equations 1 and 2).

Picking the variables with the highest number from the correlation heatmap will not yield any significant results, firstly because the highest correlated variable is any variable with itself as they would all have a significance of 1. This means that using this method every variable would be the most significant variable, furthermore some variables that are highly significant to one another are not necessarily significant in the whole dataset, for example in figure 10 METSig has a high correlation with MET (phiMET) of 0.73. However, when all the variables present in table 3 are examined it can be seen that neither variable is very significant, meaning that if it was included in the machine leering algorithms it would negatively affect the models.

Appendix E – Project Proposal

An investigation into the Heavy Higgs decay using neural networks and machine learning techniques.

Jacob Gordon

Student number: 201416406

Project supervisor: Nikolaos Rompotis

This project is about the decay of a Heavy Higgs boson called A into a Z boson and another Heavy Higgs particle called H. The decay chain A → ZH appears in several Higgs models. This project will use the A →ZH →ttbar final state. Throughout this project various types of machine learning techniques will be used to analysis a large data set from the ATLAS project. These techniques include support vector machine, kerns neural networks and random forest.

Project plan:

Week	Tasks to completed in that week
Week 1	1. First meeting with project supervisor; discussion of project in detail and familiarisation with Heavy Higgs. 2. Completion of the risk assessment. 3. Completion of the project plan.
Week 2	1. Begin the analysis of the large data set. The large data set contains 71 samples. The plan is to plot one all the variables for the signal, ttbar and background to produce the number of plots that corresponds to number of variables in that sample. 2. The above task will then be repeated for each sample number in the large data set.
Week 3	1. Once all plots have been made apply a cut-based (or simple cut based) analysis. This means to apply a certain selection criterion to our plots in order to remove events that

	are of no use to us. For example, in this project it could be the transverse energy or momentum.
Week 4	1. In week 4 the machine learning code can commence. The first machine learning code will be support vector machine.
Week 5	1. In week 5 a machine learning neural network will be set up to analyse and classify the dataset. 2. This machine learning technique will be compared to the support vector machine.
Week 6	1. The last machine learning technique will be random forest. 2. All 3 different techniques will then be compared to one another
Week 7	1. At this point in the project most of the code should be done. A plan of the presentation should be formulated.
Week 8	1. Give presentation and answer any questions asked. 2. Work on a plan for the final report.
Week 9	1. Begin writing final report.
Week 10	1. Continue work on final report.
Week 11	1. Finish final report.
Week 12	1. Proofread final report and submit.