PROJECT REPORT

Course: CMPE 256

Project: Higgs Boson Particle Analysis

Team: Olympians

Dhruvil Patel
(013729625)
dhruviljiteshbhai.patel@sjsu.edu

Saumil Shah
(013761293)
saumil.j.shah@sjsu.edu

Saurabh Mane
(012548094)
saurabh.mane@sjsu.edu

Under Guidance of
Prof. Magdalini Eirinaki

Table of contents

1. Introduction

1.1. Motivation

The Higgs Boson particle was first discovered by the ATLAS experiment at the Large Hadron Collider, CERN in 2012. [1] The characteristic of a particle is how often it decays into other particles. The ATLAS experiment observed a signal of the Higgs boson decaying into two tau particles, but this decay is a small signal buried in background noise.

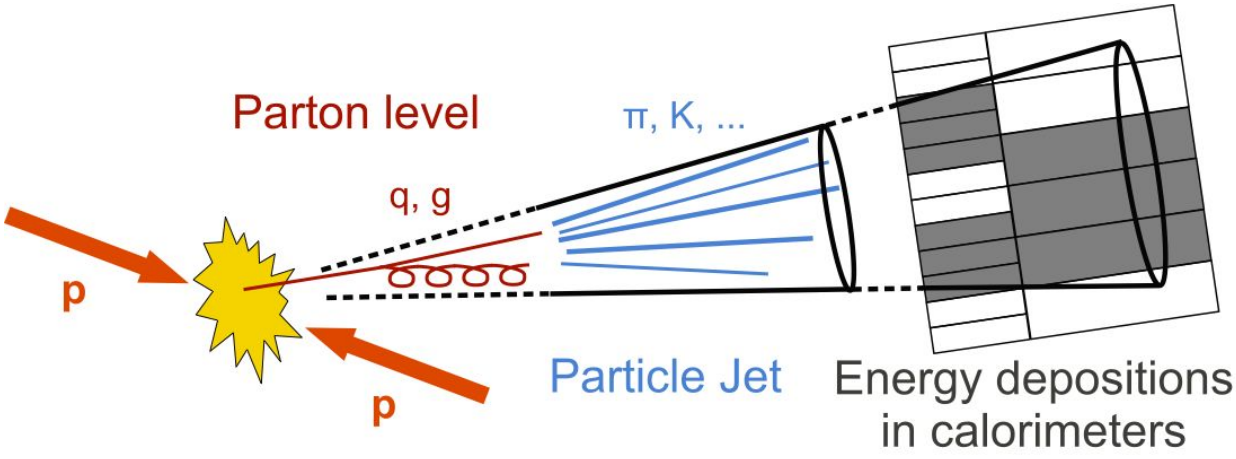


Fig. Large Hadron Collider particle collision
https://www.datasciencecentral.com/profiles/blogs/predicting-the-higgs-boson-signal

The Higgs Boson decays into two tau particles giving rise to a small signal buried in background noise. The goal of the project is to improve the classification of the events into Higgs Boson decay signal and background noise.

1.2. Objective

The objective of the project is to use the simulated dataset to classify the characterizing events detected by ATLAS into “tau tau decay of a Higgs boson particle” versus “background”. [2]

2. SYSTEM DESIGN AND IMPLEMENTATION DETAILS

2.1 Algorithms Overview:

There are three main tasks in the project: classifying events into Higgs Boson signal and background noise, predicting the weight value for each event and dimensionality reduction to visualize the data in a lower dimension.
The algorithms considered for the classification model are: Logistic Regression, Random Forests Classifier and XGBoost Classifier.
For regression model, algorithms that were tested are: Linear Regression and XGBoost Regressor.
For dimensionality reduction, Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE) and UMAP were used.

- **Algorithms for classification**

    - **Logistic Regression:** Logistic Regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. Logistic Regression model computes a weighted sum of the input features (plus a bias term), but instead of outputting the result directly like the Linear Regression model does, it outputs the logistic of the result. The logistic is a sigmoid function that outputs a number between 0 and 1. The sigmoid function helps to classify inputs into classes.

    - **Random Forest classifier:** Random Forest is an ensemble of Decision Trees, generally trained via the bagging method (or sometimes pasting), typically with max_samples set to the size of the training set. It creates a set of decision trees from randomly selected subset of training set and then aggregates the votes from different decision trees. The Random Forest algorithm introduces extra randomness when growing trees; instead of searching for the very best feature when splitting a node.

    - **XGBoost Classifier:** Gradient Boosting Machines fit into a category of Machine Learning called Ensemble Learning, that train and predict with many models at once to produce a single superior output. It is proved to be faster than other implementations of gradient boosting. Some of the major benefits of XGBoost are that its highly scalable/parallelizable, quick to execute, and typically outperforms other algorithms.

- **Algorithms for regression**
    - **Linear Regression:** Linear Regression is an approach for predicting a quantitative response Y on the basis of a single predictor variable X. It

assumes that there is approximately a linear relationship between X and Y.

- **XGBoost Regressor:** Gradient Boosting works by sequentially adding predictors to an ensemble, each one correcting its predecessor. However, instead of tweaking the instance weights at every iteration like AdaBoost does, this method tries to fit the new predictor to the residual errors made by the previous predictor.

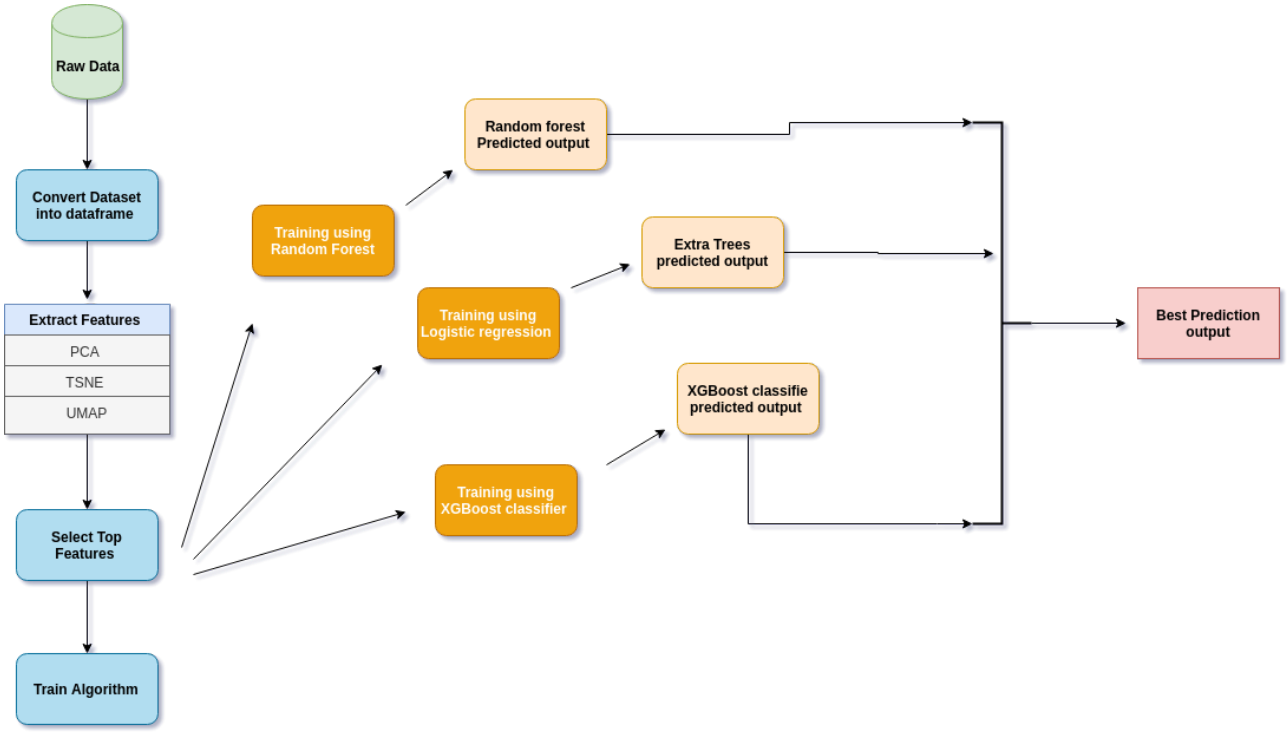
- **Algorithms for dimensionality reduction**
  - **Principal Component Analysis:** PCA identifies the axis that accounts for the largest amount of variance in the training set. It also finds a second axis, orthogonal to the first one, that accounts for the largest amount of remaining variance. If it were a higher-dimensional dataset, PCA would also find a third axis, orthogonal to both previous axes, and a fourth, a fifth, and so on—as many axes as the number of dimensions in the dataset.

  - **t-Distributed Stochastic Neighbour Embedding:** t-Distributed Stochastic Neighbor Embedding (t-SNE) is an unsupervised, non-linear technique primarily used for data exploration and visualizing high-dimensional data. In simpler terms, t-SNE gives you a feel or intuition of how the data is arranged in a high-dimensional space. The t-SNE algorithm calculates a similarity measure between pairs of instances in the high dimensional space and in the low dimensional space.

2.2 Technologies & Tools used:

- Language: Python
- Frameworks: pandas, numpy, sklearn, scipy, xgboost, matplotlib, seaborn
- Tools: Jupyter Notebook

2.3 System Design:



2.4 Visualization:

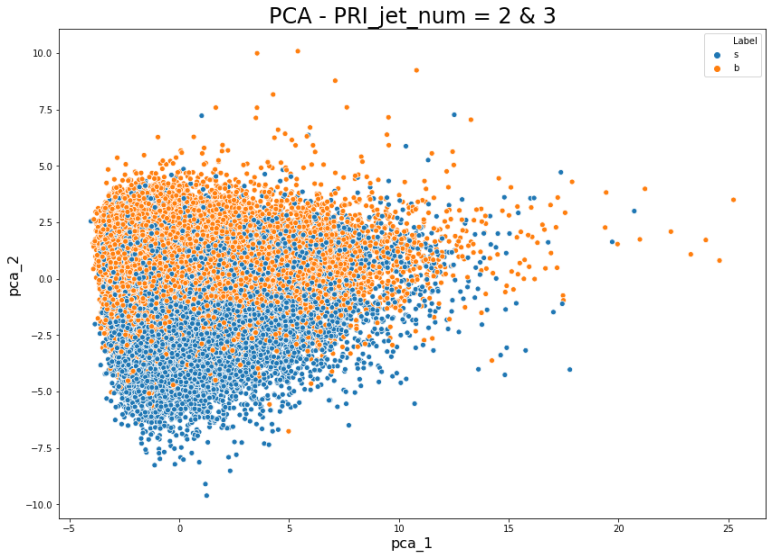


Fig. Scatter plot of data obtained using PCA on data with PRI_jet_num > 1

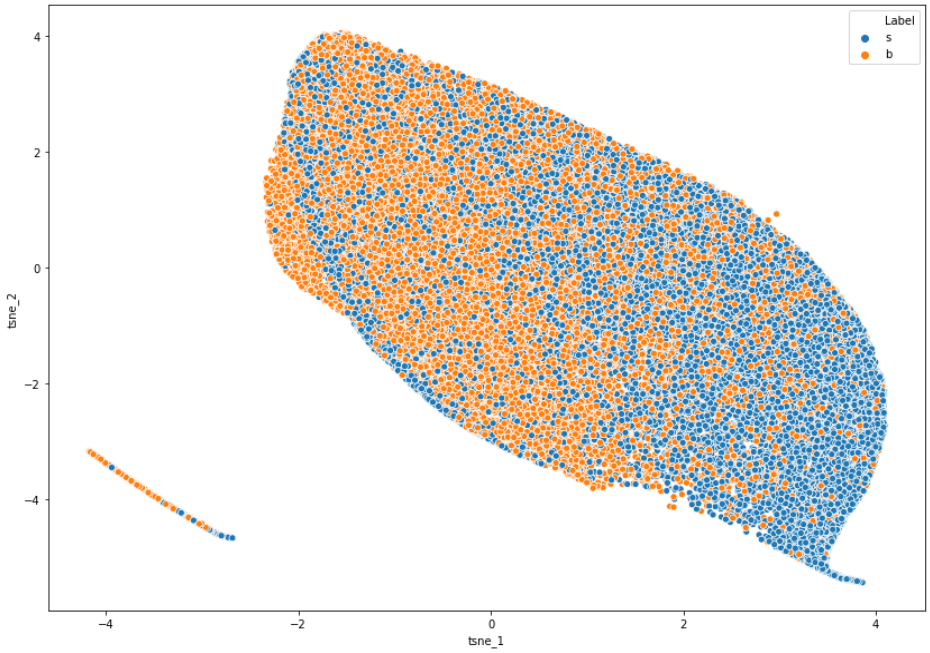


Fig. Scatter plot of clusters calculated by t-sne algorithm on data with PRI_jet_num > 2

3. Experiments / Proof of Concept Evaluation

3.1 Dataset Used

Source: ATLAS collaboration (2014). Dataset from the ATLAS Higgs Boson Machine Learning Challenge 2014. CERN Open Data Portal.

Data Size: 818238 instances, 35 features.

The project is based on a dataset from the ATLAS Higgs Boson Machine Learning Challenge 2014. The dataset is built from official ATLAS full-detector simulation, with "Higgs to tautau" events mixed with different backgrounds. The dataset is a combination of categorical and numerical values. The target class is a categorical value (s or b).

3.2 Data Preprocessing decisions

The dataset has a feature called PRI_jet_num it with 4 values from 0 - 3. Events with PRI_jet_num = 0 and 1 have null values in the data for multiple columns since the values cannot be calculated. The first step was to group the data into three subgroups. The next step is to remove null value columns from the subgroups. The dataset still has several outliers in the data. To manage the outliers, Robust Scaler was used as it performs best with outliers. The values of ‘s’ and ‘b’ in the target variable were changes to numerical values to enable running numerical methods on the data.

PCA was used to for feature extraction to reduce the dimensionality of the original dataset. As a result, the size of the data was reduced and it enabled visualization of the data in a lower dimensional space.

3.3 Methodology followed

For regression and classification models, we have used a sample of 70% of the data for testing and evaluating different models/algorithms. The remaining 30% was then used to test the accuracy of different prediction algorithms.

3.4 Graphs showing different parameters / algorithms evaluated

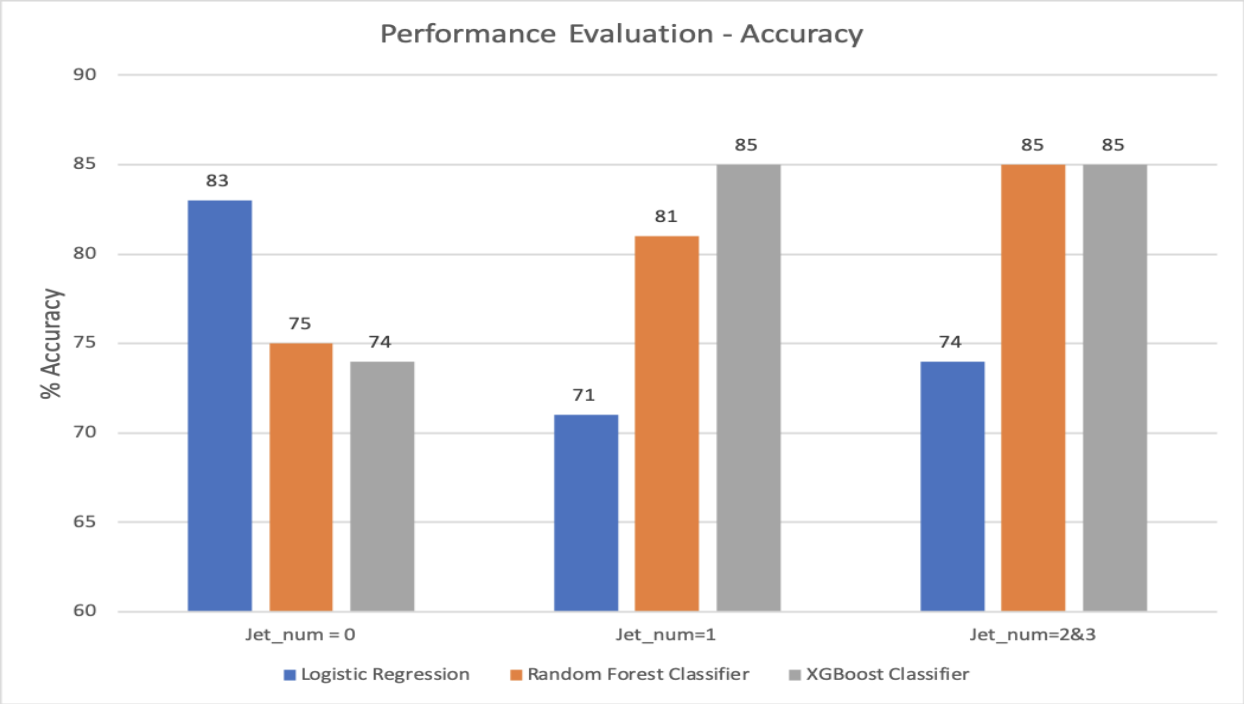


Fig. Comparison between the performance(accuracy) of the different models

3.5 Analysis of results

Based on these comparisons, XGBoost Classifier with 0.1 learning rate was the most accurate model giving the highest accuracy. This is followed by Random Forest Classifier and then Logistic Regression. Increasing the estimators resulted in dropping of test accuracy, due to overfitting. The dataset is divided into 3 parts and the graph shows the accuracy of each algorithm for all the 3 parts.

4. Discussion & Conclusions

4.1 Decisions made

- We had to decide how to change our dataset to make it relevant. Our main discussion points were on choosing methods for preprocessing the dataset for dimensionality reduction and which algorithms to choose for optimal prediction.
- We then decided to apply PCA, t-SNE and UMAP for dimensionality reduction. We also decided to choose Logistic Regression, XGBoost Classifier, Random Forest Classifier to make predictions for the dataset.

4.2 Difficulties faced

- The most challenging part was to choose an optimal machine learning algorithm for our prediction because the data had many null values and it was difficult to find a pattern to manage the null values.
- Dimensionality reduction methods like t-SNE and UMAP are very expensive processes and are very slow for big datasets.

4.3 Things that worked

- We divided the dataset into 3 parts, depending on the number of 'PRI_jet_num'. This gave us good results when we implemented our algorithms.
- Applying PCA as a preprocessing step for t-SNE saved a lot of time to visualize the data in a lower dimensional space as it reduced the number of features.
- Applying ensemble methods gave decent results after some preprocessing was applied.

4.4 Things that didn't work well

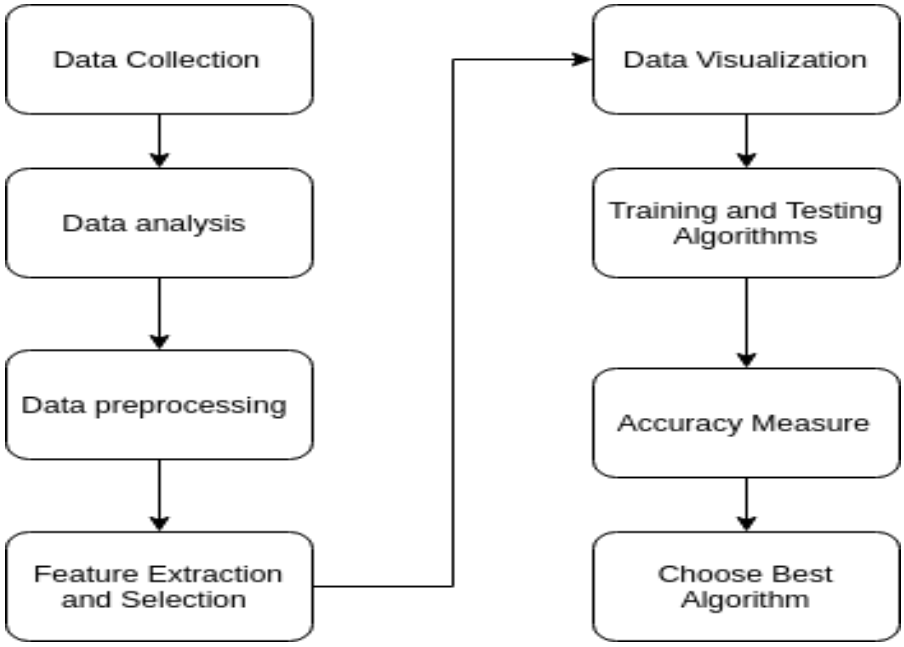
- Applying XGBoost Classifier without optimally preprocessing the dataset led to overfitting.
- Applying SVM on this dataset was very difficult. Training the model on 10% of the data took a very long time and we decided not to implement it on the entire dataset.
- UMAP and t-SNE did not give satisfactory results in clustering

4.5 Conclusion

- We understood the importance of data preprocessing and how it is the most important step when applying any algorithm.
- Random Forest CLassifier tends to overfit with an increase in estimators.
- The XGBoost Classifier gave us the best prediction result.

5. Project Plan / Task Distribution

5.1 Project Plan



5.2 Task Distribution

Saurabh Mane:

- Primary contributor in data cleaning and data preprocessing
- Removed outliers from the input data by calculating percentile data distribution.
- Initially performed classification using ExtraTreesClassifier.

Dhruvil Patel:

- Primary contributor in data visualization, training and testing.
- Applied dimensionality reduction using PCA, t-SNE and UMAP.
- Built several classification models to classify events into signal and background noise.

Saumil Shah:

- Initially performed classification using SVM.
- Implemented XGBClassifier to classify events into signal and background noise.
- Implemented regression models to predict weights for all events.

References:

[1] https://www.datasciencecentral.com/profiles/blogs/predicting-the-higgs-boson-signal
[2] http://opendata.cern.ch/record/328