1. Key EDA Points

- Shape: (568630, 31)
- We only have two datatypes, float and integer, here.
- There are no missing values and only one duplicate entry.
- From 1 distribution of legitimate (0) and fraudulent (1) transactions is equal, hence there is no imbalance.
- From 2, Amount varies from 50.01 to 24039.93 and is uniformly distributed.
- Values for V1 to V28 and Cost/Amount are on a different scale, there is a need for standard scaling (graphs for each feature are on Fraud.ipynb notebook).
- Drawing insights from Fig 3, we dropped 'ids' column, as it is nothing more than an index.
- Anonymous features were skewed. So, we removed outliers in every feature using IQR range values specific to each feature.

Figure 1: Target Variable Distribution

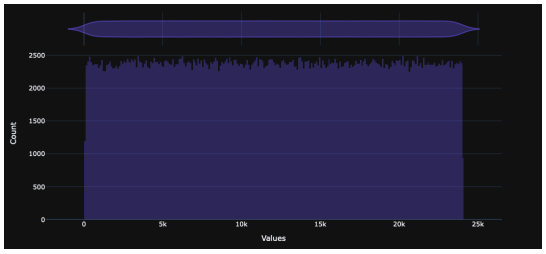

Figure 2: Amount

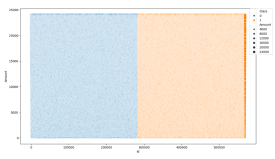

Figure 3: Amount vs Ids color coded by our target variable

2. Comparision with other other projects and our adopted methodology with results.

   - Referring the review document, we decided to explore the relationship between collinear features and its impact on our target variable by creating two datasets, one without highly correlated variables (threshold=0.6) and one with all the features. In the case of the former, for a set of highly correlated features, we dropped the feature that had a lower correlation with our target variable.
   - In 4, correlation metrics observed by us were different as we removed outliers before calculation itself.

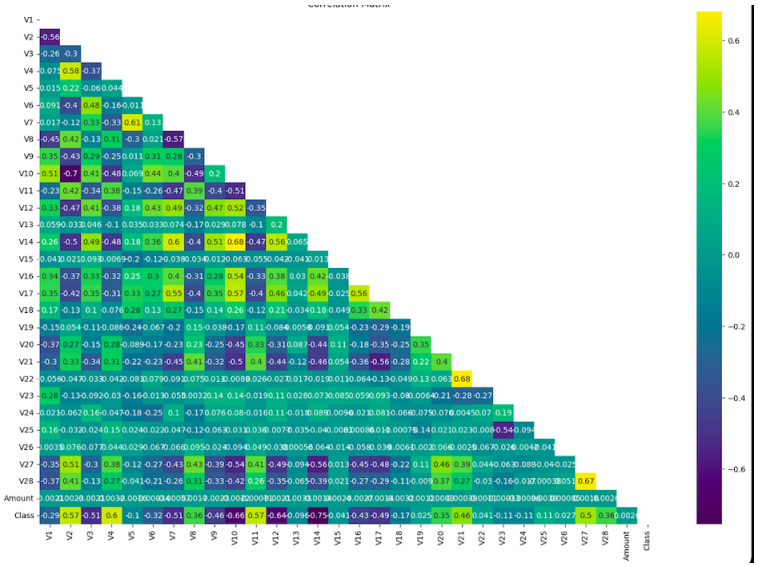

Figure 4: Correlation Matrix

   - For this part of our project, Srikar worked on a Random Forest classifier and a Logistic Regression for his classification tasks whereas Tharun implemented Naive Bayes and Linear SVM. They compared their models over various metrics including Accuracy Score, F1, Recall and Sensitivity, creating confusion matrices and ROC curves for every model.

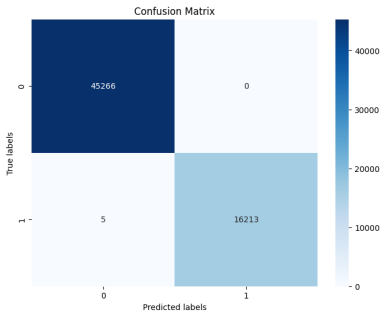

Figure 5: Confusion Matrix for Random Forest Regressor

     – The best metrics observed by Srikar are obtained from the Model_noThr RF with no tuning has the best performance
     F1: 0.9998766954377312
     AS: 0.9999349424240452

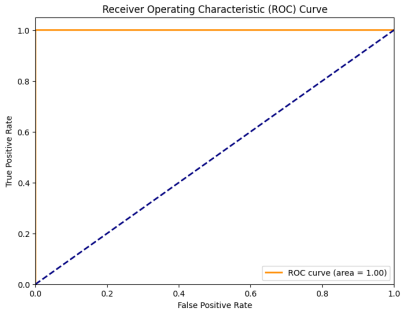

Figure 6: ROC curve for Random Forrest Regressor

P: 0.9997534212797435
Recall: 1.0
Specificity: 0.9999116334555738
In detail observations noted by Srikar can be found here.

- The best metrics observed by Tharun are obtained from the SVM with Thres = 0.6 model has the best performance.
  F1: 0.9945046557777438
  AS: 0.9970723971781161
  P: 1.0
  Recall: 0.989069379080006
  Sensitivity: 0.989069379080006
  Tharun's observations are here.

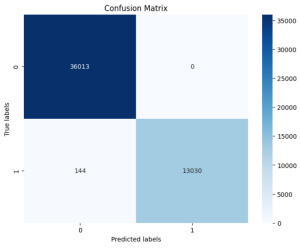

Figure 7: Confusion Matrix for Linear SVM

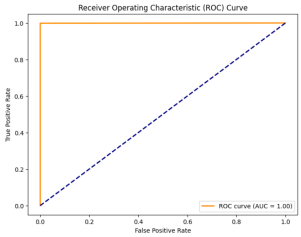

Figure 8: ROC curve for Linear SVM

- Drawing insights from the time taken to train the models by Srikar and Tharun, we decided to extract features from the original dataset using PCA. Considering 9, Abhijeet decided to explore the metrics for a Decision Tree Classifier using only one component, as this was clearly the

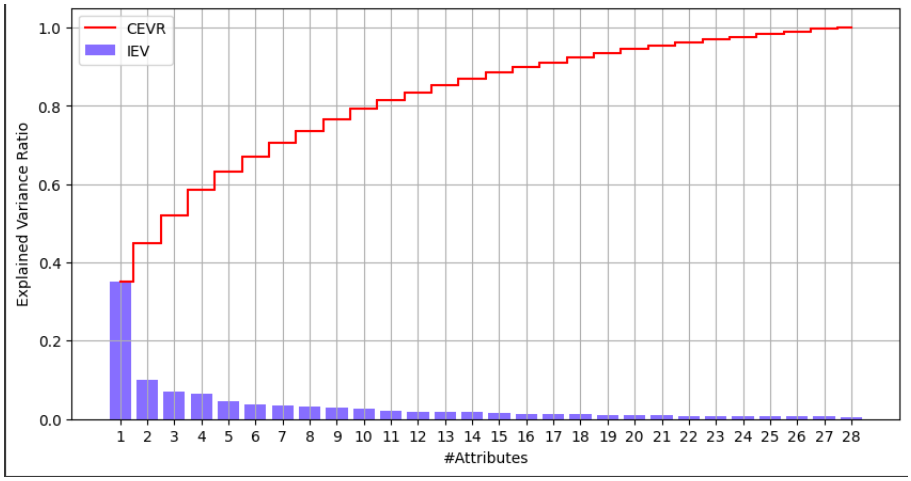

Figure 9: Explained Variance by #Attributes extracted using PCA

elbow point and ten components as here, the cumulative explained variance touched 0.8, to represent all the anonymous features (without Amount).

- In this workflow, Amount/Cost was divided into 3 bins to create new categories. Other anonymous features were standard scaled and then, extraced using PCA.
- Since Scikit Learn's Decision Tree classifier cannot handle categorical data that is non-numeric in nature, we decided to create a custom Decision Tree classifer using BaseEstimator and ClassifierMixin.
- The customDT class initialises a pipeline that label encodes categorical data and passes other features, which ends with a Decision Tree Classifier.
- Here, the best accuracy score observed was *0.9961803064196176*, for a tree without any limit on its maximum depth with 10 components extracted from PCA. The mean of absolute SHAPEly values is in 10.

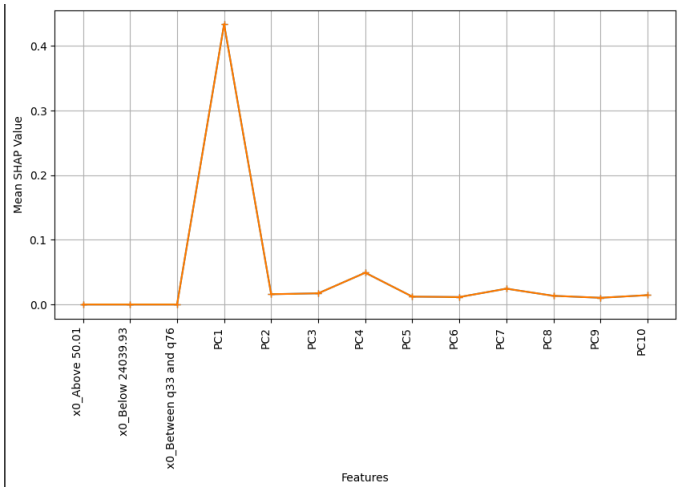

Figure 10: Shap Values for 10 attributes extracted

- Since a tree with so many features and unspecified depth turned out to unreadable, the customDT's functionality was extended to handle Grid Search Cross Value with only one feature

extracted from PCA and our categorical Amount's column. The best parameters were : *'criterion': 'gini', 'max_depth': 2, 'min_samples_leaf': 1, 'min_samples_split': 2*. The decision tree plotted for this model is in Fig11.

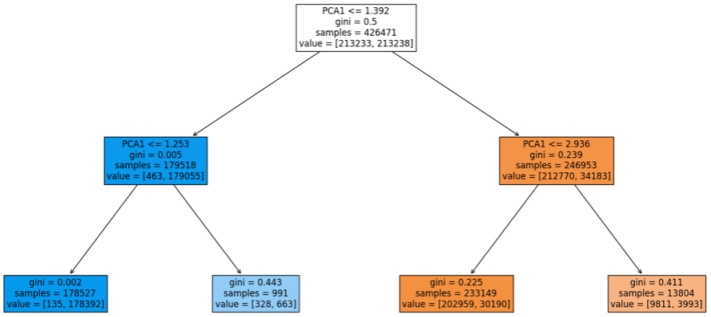

Figure 11: Decision Tree Plot for #attributes=1 from PCA

- The tree in Fig11 had an accuracy score of *0.918126310* and its other metrics were: *Precision: 0.997189366, Recall/Sensitivity: 0.838609376, Specificity: 0.997636532, F1 Score: 0.911050142*

3. **Key observation from PCA and SHAPE values graphs**: The features that explain a higher amount of variance in the dataset contribute more towards the outcome of classification task.

4. Future Scope of Work:
   Build a decision tree that can handle categorical features.
   Extend the current decision tree classifier to work with dataframes too.

5. Our individual notebooks with our presentation can be found on our shared Google Drive.