Credit Card Fraud Detection: a 4 Stage Process

Name:	**Vatsala Nema**
Registration No./Roll No.:	19332
Institute/University Name:	IISER Bhopal
Program/Stream:	EECS
Problem Release date:	February 02, 2022
Date of Submission:	24th April 2022

1 Introduction

Through this project on Credit Card Fraud detection, we hope to detect fraudulent activities with credit card transactions in real-time to enable banking systems to deny such fraudulent transactions and thus in turn prevent such malpractices from happening. This is a binary classification problem, and on exploring the data, it was found that there were 492 frauds out of 284,807 transactions performed. Since class labels are already known to us (0-Routine transaction 1- Fraud), the problem is solved through supervised learning algorithms.

Credit card fraud detection (CCFD) is like looking for needles in a haystack. It requires finding, out of millions of daily transactions, which ones are fraudulent. Multiple Supervised and Semi-Supervised machine learning techniques are used for fraud detection. Yet, there are three main challenges to overcome with card frauds related data sets i.e., strong class imbalance, the inclusion of labeled and unlabelled samples, and to increase the ability to process a large number of transactions

2 Methods

Github repo link: Github - CCFD

2.1 Pre-processing and Feature Engineering

Since all our data points are numerical, no label encoding is required. To balance the data set we can use downsample the data. For feature selection, use feature correlation matrix and Mutual Information Gain.

(A) On understanding the features of the data we saw that all the V features are scaled from 0-1 but the time and amount is not. To scale the data set completely and determine the independent features we will have to scale Time and Amount. We use Robust Scaler, a method that scales features using statistics that are robust to outliers. This Scaler removes the median and scales the data according to the quantile range

(B) Feature Selection: We select the most significant features using the metric, Mutual information between two random variables is a non-negative value, which measures the dependency between the variables. Here, we select the best 15 features to build our classifier.

Different Supervised machine learning algorithms like Decision Trees, Logistic Regression and Random Forests are used to detect fraudulent transactions in real-time data sets.
Total number of features in the original data set given for credit card fraud detection model are 30. The significance of the features are hidden due to privacy reasons, so the features not interpretable by a human. Initially, the training data set is split into training and validation set(at a ratio of 70:30 respectively) using the StratifiedShuffleSplit module from sklearn library. In this case of binary classification, this module splits the data set into two parts keeping the ratio of number of points from one class to that of another, the same in both the training and validation set.

2.2 Logistic Regression

Logistic regression model is a regression technique in which the dependent variable i.e., the target is categorical. It is a classification model, which is very easy to realize and achieves very good performance with linearly

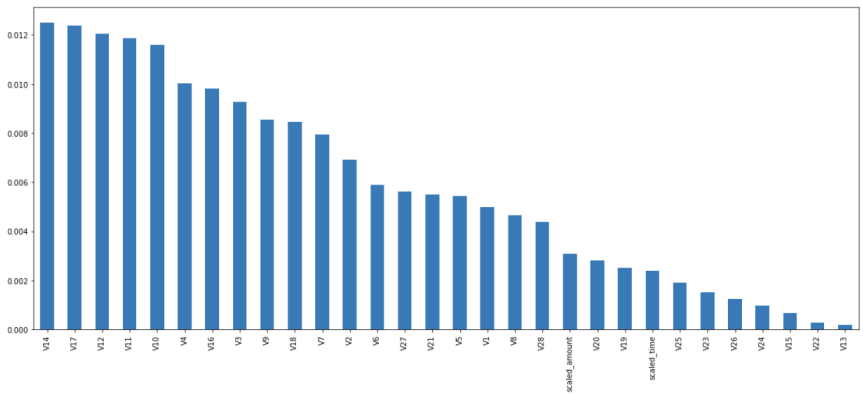

Figure 1: P
lot of the feature importance based on Mutual Information.

separable classes.LR is a transformation of a linear regression using the sigmoid function. The vertical axis stands for the probability for a given classification and the horizontal axis is the value of x. It assumes that the distribution of $y|x$ is Bernoulli distribution. The formula for LR is $F = 1/(1 + e^{-(\beta_0+\beta_1 x)})$ Here $\beta_0 + \beta_1 x$ is similar to the linear model y = ax + b. The logistic function applies a sigmoid function to restrict the y value from a large scale to within the range 0–1.

2.3 Decision Tree Classifier

Decision tree builds classification models in the form of a tree structure. A decision tree classifier is expressed as a recursive partition of the instance space. The decision tree consists of nodes that form a rooted tree. In a decision tree each terminal node splits the given feature space into two or more sub-spaces according to some splitting criteria like information gain, gini index, gain ratio, etc. In a decision tree each terminal node splits the given feature space into two or more sub-spaces according to a certain discrete function of the feature values of the given data set. In the case of numeric feature values i.e., for continuous features the condition refers to a range. Each leaf node is assigned to one class representing the most appropriate target value. Here, decision tree is used for classification purpose on continuous data as interpretation of decision trees are intuitive due to the graphical and tree like representation.

Also, too many features in the data set reduces the predictive power of the classifier dues to the curse of dimensionality. So the best subset of features need to be selected from all the features to improve the predictive power of the decision tree algorithm and also to reject redundant or irrelevant features. To select the best 10 features out of the given 30 features, here we have used Mutual Information. Mutual information (MI) between two random variables is always a non-negative value and it measures the inter-dependency of the variables. It is equal to zero if and only if two random variables are independent, and higher values mean higher dependency.

After selecting the 10 best features from the original training data set, the decision tree model was trained with default parameters on the new training data set, and then the AUC-ROC score was checked on the training and validation set, it showed a sign of over-fitting as the scores differed by a measure of around 10%. To overcome this over-fitting hyper-parameters were tuned using a grid search in which scoring parameter was AUC-ROC and 10-fold cross validation was also incorporated.

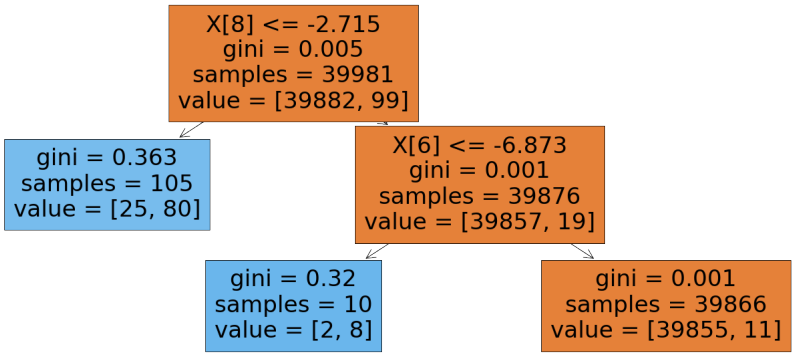

Figure 2: D
ecision tree after hyper-parameter tuning.

The algorithm for the classifier can be depicted as:

Algorithm 1 Classifier using Decision Tree

1. Split the data set into training and validation set using StratifiedShuffleSplit.
2. Select best k features using the Mutual Information in Classifier in the training data.
3. Train a Decision Tree model with default parameters.
4. Begin Hyper-parameter tuning.
    - Put the desired values for hyper-parameters in grid search parameters, for e.g. criterion, max_depth, max_features, etc.
    - Generate a list of effective values of alpha for pruning the tree.
    - Select the alpha value for which the bias and variance is lowest.
5. Train the decision tree extracted out of the grid search on the derived best alpha value.
6. Check the AUC-ROC score of the model on the validation set to determine if over-fitting exists. If it exists return to the step 4.

2.4 Random Forest Classifier

Random forests are simple to implement, fast in operation, and have proven to be extremely successful in a variety of domains. The key principle underlying the random forest approach comprises the construction of many "simple" decision trees in the training stage and the majority vote (mode) across them in the classification stage. Among other benefits, this voting strategy has the effect of correcting for the undesirable property of decision trees to overfit training data.

3 Experimental Analysis

The evaluation of the classifiers in this project have been done using three criteria, one is micro-averaged precision, recall and f-measure, AUC-ROC score and the precision-recall curve. The recall, specificity, precision, and F1 score metrics, are threshold-based metrics,and have well-known limitations. This is due to their dependence on a decision threshold which is difficult to determine in practice, and strongly depends on the case-specific constraints.

The AUC ROC is currently the de-facto metric for assessing fraud detection accuracies. The AUC ROC and AP metrics aim at assessing the performance for all possible decision thresholds, and are referred to as threshold-free metrics. Recent research has however recommended using the Precision-Recall curve and Average Precision metric instead. The ROC curve is obtained by computing the TPR and FPR for all possible fraud probability values returned by a classifier.

For the decision tree, to overcome the issue of over-fitting, grid search and cost complexity pruning were used. In the grid search, for max_depth, a range of values from 1 to 20 were used, while criterion were either "gini" or "entropy", also max_features was assigned a list with "auto", "sqrt" and "log2". After extracting the best estimator from the grid search, cost complexity pruning was applied to get a decision tree with lowest bias and variance. Here, a list of effective values of alphas were extracted with the cost_complexity_pruning_path function and a graph was plotted between alpha values and AUC-ROC score of both training and test data and then the best alpha value was chosen for which both bias and variance were lowest. For the tuned decision tree, the AUC-ROC score for the validation set was 91.83% and the values for macro-averaged precision, recall and f-measure were 0.867, 0.918, and 0.891 respectively.

As the Random-Forest Classifier chooses random decision trees on a subset of features and records, the random forest is a good fit for data sets in which the features are not intuitive or manually interpretable like the data set provided here. Random-forest classifiers helps boost the accuracy of the prediction as the decision trees get over-fitted on smaller subsets of data, their accuracy on the same increases considerably on the same subset. No hyper-parameter tuning was performed on the random-forest classifier. The AUC-ROC score for the random forest classifier was 94.037% and the macro-averaged precision, recall and f-measure were 0.939, 0.918, and 0.928 respectively.

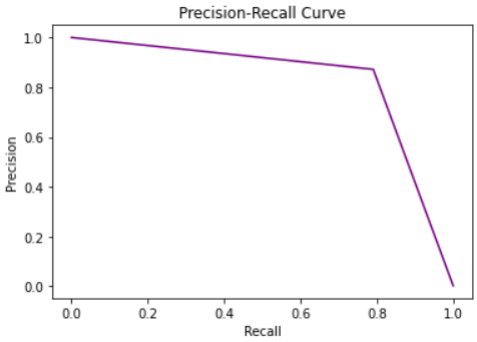

Figure 3: Precision-Recall Curve for the Decision Tree Classifier

4 Discussions:

Credit card fraud detection (CCFD) is a challenging problem, which requires analyzing large volumes of transaction data to identify fraud patterns. The large volumes of data, along with the ever-evolving techniques of fraudsters, make it impossible for human investigators to efficiently address this problem.

While we were quite successful with supervised learning models that were used, the hardware limitations on our side limits our scope to other ensemble methods with parameters being tuned. Ensemble based classifiers are quite good for imbalanced data sets like one provided here due to the combination of many weak classifiers within them which master on smaller subsets of data and then by majority voting, ensemble based methods are able to avert over-fitting to some extent, producing good results on new data.Hence, we plan on using Ensemble learning techniques like XGBoost with good resources. We plan to use it in the near future and develop a more robust ensemble learning framework for Fraud detection.

5 Contribution:

- Data Exploration, Decision Tree, Random Forest Classifier, Hyperparameter Tuning and Iterative Evaluation - Ajay
- Feature Processing, Logistic Regression Model and Iterative Model evaluation, Downsampling and Robust Scaling- Vatsala