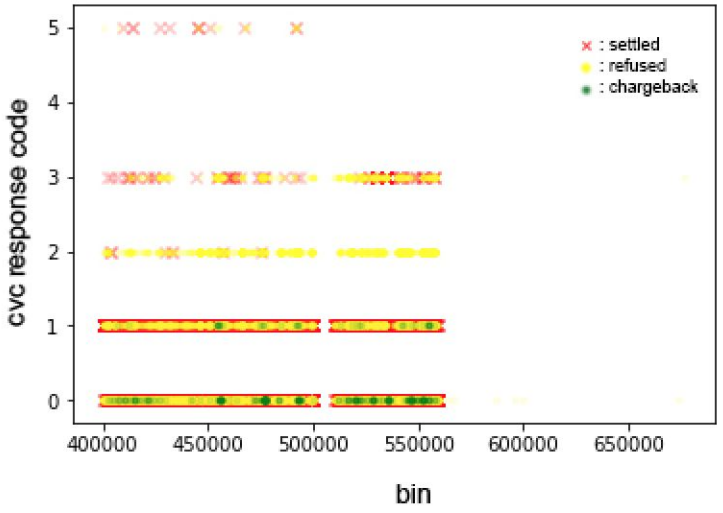
Cyber Data Analytics Assignment 1

Chitra Balasubramanian(4742907)
Gouri Viravalli (4738888)

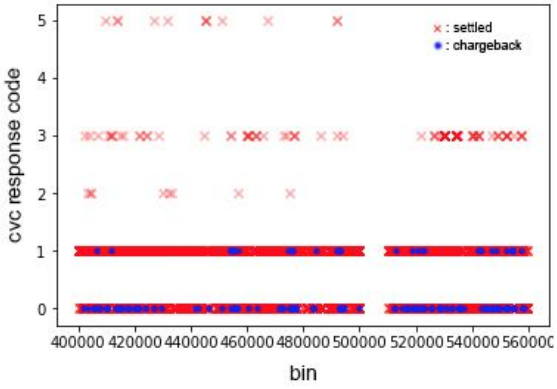
GitHub link - https://github.com/strawbeerry/CDA (for code and dataset)

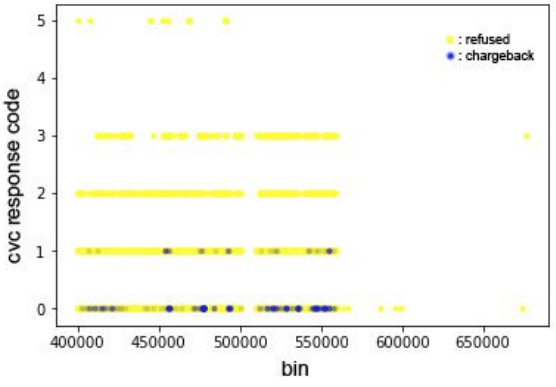
1a. **Visualization**

The scatterplot below contains all data points showing the relationship between the different card issuers (bin) and the status of the CVC/CVV response code, while highlighting in different colours the status of the transaction ie., 'simple journal'. The red colours correspond to authorized transactions, and out of the 6 options in the response code (y axis), only '1' means that the CVV code has actually matched. Hence, any red mark in '1' can be ignored. But the green (ie., fraudulent transactions) corresponds to cases where the attacker is aware of the cvc code. The yellow corresponds to refused transactions and although it could mean that the transaction was refused due to low balance, it could also mean that the transaction was found to be fraudulent. On the other hand, the transactions that were successful when the cvv code did not match (ie, when the response code is '2') corresponds to false negatives. It is seen from the visualization that there are also transactions that were accepted although the cvc response was unknown (ie, response code = 0) and that the maximum number of chargebacks occurred when the CVC was unknown. Hence proper verification of CVC code could be one control to reduce the fraudulency in the data.



The two graphs below are for a better understanding of the case - it represents the CVC response code and the card issuer for settled and chargebacks, and for refused and chargebacks.





It is notable that although most of the transactions are benign (red), there are very many refused transactions (yellow) that have no been classified as fraudulent but could be.

1b. **EXPERIMENTATION WITH SMOTED AND UNSMOTED DATA**

SMOTE is a process of oversampling the dataset to prevent the bias on an unbalanced data. Here there are very few fraud cases compared to the non-fraud cases. In other to equalize the effect we use the method of SMOTEing on our dataset. Usually unbalanced data can raise the number of false positives and false negatives in the dataset.

Dataset

There are many features that are provided in the fraud dataset. For the classification task we considered Bin, amount, cardVerificationCodeSupplied, cvcResponseCode, label, cardId, txVarientCode and shopper interaction as the variables. The dataset also consists of two derived variables, CountryCodeDifference(if there is a difference in the country of purchase of the credit card and country in which it is being used gets a value of 1 else it gets a value of 0) and normalisedAmount(this parameter is cut down to a scale of 1-10 as it might contribute as a highly important feature in the classification problem, which may not be the case in real world criteria). Some of the features are converted into their numerical codes so that it would be more meaningful for the classifier). The "label" represents the labels either 1 or 0 for fraud and non fraud data. This is obtained from simple_journal column of the original dataset, where chargeback is labelled 1 and Settled and Refused is labelled 0(as they are benign). This is the preprocessing step done before SMOTE.

The dataset is now smoted and the oversampled dataset is now 232018x2, i.e the size of the largest labelled dataset.

Classification

The next step to identifying the fraudulent cases is classification. The dataset is now divided into test and train sets in a ratio of 20 to 80. This means that 20% of the data is used for testing purposes and the remaining 80% of the data are used for training. The classifiers are first trained on the training dataset and then validated using the testing dataset. There are 4 main classifiers on which this dataset was tested. They are, RANDOM FOREST CLASSIFIER, K- NEAREST NEIGHBOR CLASSIFIER, LINEAR DISCRIMINANT CLASSIFIER and a simple NAIVE BAYES CLASSIFIER.

Random Forest Classifier

This is one of the most simplest classifiers, as the name suggests it builds a number of decision trees and merges them all together to get a more accurate and stable prediction.

K-Nearest-Neighbor Classifier

This classifier forms hyperplanes/decision boundaries over it's k-nearest neighbors, in our case we consider k=6. It is based on feature similarity, it groups data points with similar features into one group and most likely that group contains a common label.

Linear Discriminant Classifier

In this classifier a linear combination of features group together to separate two or more classes or groups. It is mainly used in dimensionality reduction.

Naive Bayes Classifier

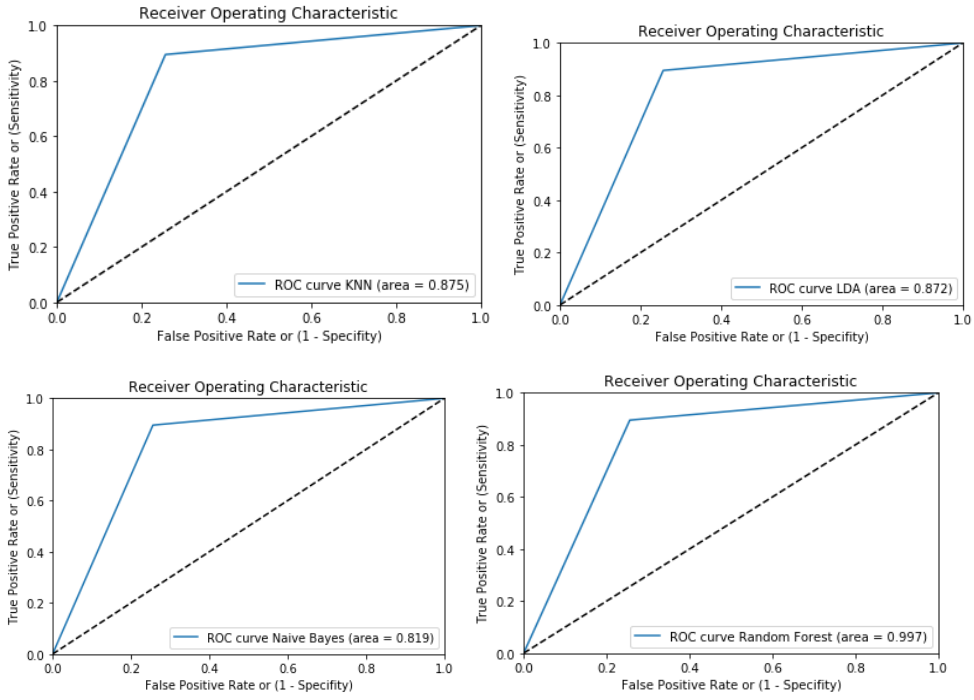
Having seen good results in the LDA classifier we decided to pick the simple Naive Bayes to simply test for fraudulent cases in our dataset.

We trained and tested the classifiers on the preprocessed dataset for both SMOTED and UNSMOTED data with the following results.

	SMOTED				UNSMOTED			
	Accuracy	Precision	Recall	F-Score	Accuracy	Precision	Recall	F-Score
R.F	99.74	99.69	99.69	99.69	99.69	57.07	50.85	51.44
KNN	87.48	87.48	87.48	87.48	99.88	55.95	50.85	51.44
LDA	87.24	88.85	87.2	87.1	99.23	50.33	52.24	50.47
NB	81.36	82.13	81.39	81.26	99.86	49.93	49.99	49.99

From the above table of the 4 classifiers we can easily tell that we cannot rely completely on accuracy. In the case of Unsomted data we can see that in all the cases there is over 99% of accuracy. This is completely misleading as there is a high chance of assuming all transactions are fraudulent. On the other hand precision and recall tell us the % of true fraud out of all fraud and %of identified true fraud out of supposed to be true fraud cases respectively. Always a high precision and a high recall is ideal in the case of credit card fraudulency as the credit card company will not go over a loss and all frauds can be identified. But the tradeoff is, there would be a discrepancy when it comes to assuming regular customers as fraudulent on even minor misleading transactions. Based on what the agency wants, no frauds or customer satisfaction or a balance between the two,they can pick a suitable classifier.

ROC Curves and AUC for the above mentioned 4 classifiers



ROC curve is another measure to predict the accurateness of an algorithm. It is a plot of sensitivity vs (1-specificity) which is nothing but recall and precision. The AUC shows how much of true positives fall under this curve.

Confusion matrix
LDA - [19966 6239] [790 25213]]
KNN - [[22283 3922] [2203 23800]]
RF - [[26086   119] [41 25962]]
NB - [[19301 6904] [2826 23177]]
The confusion matrix can tell us that there is at least a 100+ cases out of the 1000+ cases correctly classified as fraudulent, by making note of the True Positives.

2A4 BLACK BOX and WHITE BOX classifiers

The Black Box classifier considered here is the Random Forest classifier. The data considered here is the same pre-processed data in the previous question. But when we use a 10 fold cross validation we need to make sure that the test data is not SMOTED as it is data coming from the real world and it is quite impractical to add the new upcoming dynamic data to the training sample and SMOTEing it. Hence we need to make sure that one part is UNSMOTED data and the remaining is SMOTED.
In random forests every tree is built from a sample drawn with replacement from the training set. A commonly used class of ensemble methods is Random forests. Ensemble methods exploit the basic exploitation of base learners since the error can be reduced dramatically by averaging. This algorithm goes one step further and even makes the threshold splitting in trees random.

Cross validation accuracy of every run: [0.65408588 0.9955577 0.99862254 0.99879472 0.99879472 0.99882912 0.99882912 0.99882908 0.99882908 0.99879464]
The average accuracy of Random Forest :**0.9639**

The white box algorithm considered here is Linear Discriminant Analysis, where we know every step in the classification process and there is no randomization or any guess work in every run. There is a fixed set of rules which the algorithm follows. Here features are grouped to form a linear decision boundary to differentiate between fraud and non fraud.

Cross validation accuracy of every run: [0.52942594 0.97251971 0.99190743 0.99862254 0.99879472 0.99882912 0.99872581 0.99882908 0.99507525 0.9853635]
The average accuracy of Random Forest: **0.9468**

Here we can see that in a high performing classifier and a simple Linear Classifier the accuracy is a minor difference. But, there is a condition in both the classifiers wherein the accuracy is really low to 65% and 52% in the above classifiers. These correspond to the case where the testing data has a high population of fraud cases, which in reality is not the case.