# Final Term Project: Credit Card Fraud Detection using Machine Learning

## Saurabh Biswas

## DSC 550 T302

## Bellevue University

## **Abstract:**

The world has witnessed a tremendous growth of card-based transaction volume in recent years. But credit card fraud remains one of the biggest concerns since starting. Credit Card Fraud is one of the major sources of revenue loss for Payment Industry. Financial institutions use rule based traditional fraud detection system to combat fraud. But there are some drawbacks like human dependency, high operating cost etc. with traditional fraud detection system. The aim of this project is to use machine learning and build classification models to use it as an alternative to traditional fraud detection system. This project will build fraud detection system prototypes using Random Forest, Logistic Regression and Artificial Neural Network models. It will evaluate and compare the efficiency of each model and select the best model to use a fraud detection tool.

**Introduction:**

Credit card fraud is unauthorized access of someone’s credit card. By federal law, individuals are not liable for unauthorized use of their credit cards. This loss is consumed by payment card industry.

The world is leaning more towards card-based payments. According to an article, here is how US transaction market share will look like by 2022:

A picture containing clock, device

Description automatically generated

With growing market share of card-based payments, the financial loss due to fraudulent transactions is also increasing. In 2018, $24.26 Billion was lost due to payment card fraud worldwide. The United States leads as the most credit fraud prone country with 38.6% of reported card fraud losses in 2018. Payment Industry introduced various measures like introducing EMV (chip) card, biometric mobile payments such as apple pay, google pay etc. to reduce transactional frauds. As a result, the trend has shifted from in-store fraud into ecommerce fraud. But it remains one of the biggest concerns of financial institutions.

Currently financial institutions use traditional fraud detection system to stop fraud. But traditional fraud detection systems have may drawbacks such as minting a team of domain experts to analyze fraud data, high operating cost, human dependency etc. Financial institution is looking for alternatives to replace decades old traditional fraud detection system. Machine Learning techniques can be a great fit to build an automated fraud detection system that requires minimal human intervention.

Machine Learning offers a wide verity of classification algorithms that can be used to build fraud system prototypes. Machine Learning techniques also addresses all the drawbacks from traditional model. It requires very little human intervention once the model is trained. The operating cost is also very low, and it doesn’t require domain expertise to build and maintain this.

**Proposal:**

The aim of this project is to use machine learning techniques to combat fraud. Machine learning offers various classification models. This project will use Random Forest, Logistic Regression and Artificial Neural Network to build models. These models will be trained offline using the historical data. These trained models will be cross validated, and results will be compared to select the best model. It will be tested against an unseen test data. Once satisfied test result is obtained, it will be trained against the entire dataset and the prototype will be ready for use.

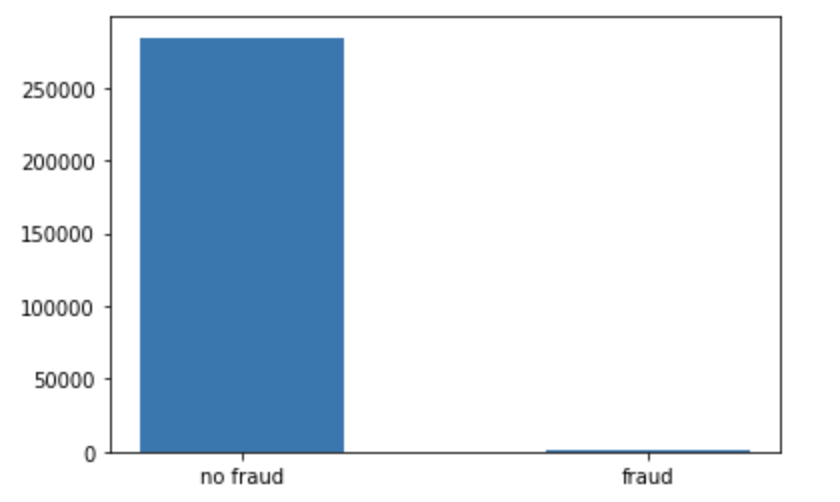
**Process Overview:**

Historical credit card transaction data from Kaggle will be used for this project. Basic analysis will be done on this data to understand the basic structure. Then the dataset will be split into train and test dataset in 75%/25% ratio. After that Exploratory Data Analysis will be performed to understand the underlaying relationship between different variables. Then outliers will be removed. Pipeline will be created with data transformation logic. The model parameters will be tuned with the help of cross validation. Models’ performance will be evaluated and compared. Pipeline will be set up for best model and will be trained on the entire dataset.

**Data:**

creditcard.csv is used from Kaggle. It contains two days credit card transactions from 2013 September from Europe. It contains 284,807 transactions. To maintain the data confidentiality, most of the variables were PCA transformed. Only ‘Amount’ and ‘Time’ variables are in their original form. Variables V1 through V28 contain only numeric values as a result of PCA transformation. ‘Class’ variable contains the outcome of the transactions – 1 signifies fraudulent and 0 – signifies non-fraudulent transactions.

To understand how the output ‘Class’ variable is distributed between fraudulent and non-fraudulent, following bar diagram is plotted.



It shows the dataset is highly imbalanced. Out of 284807 total transactions only 492 transactions are fraudulent i.e. it represents only 0.17% of the entire dataset.

To understand how amount is distributed for each Class, the amount distribution is plotted for fraud and non-fraud class.

A screenshot of a social media post

Description automatically generated

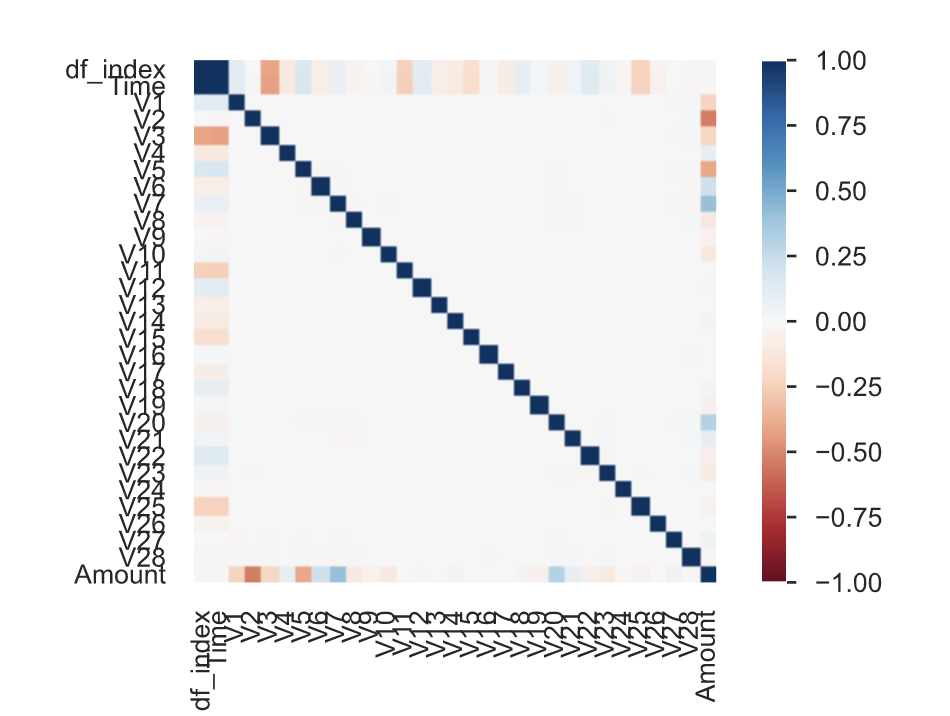
Based on this plot, it can be observed that most of the fraudulent transactions are of low amount. Whereas there are some outliers or very high amount for non-fraudulent transactions.

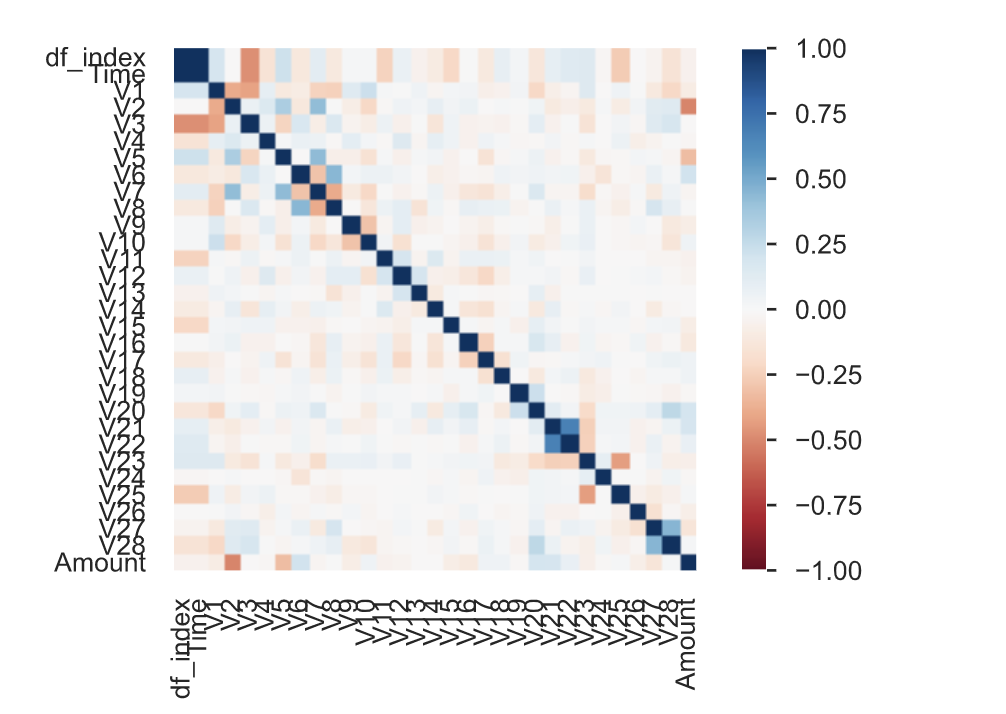
After getting basic idea about the dataset, training and test dataset is created with the help of sklearn.model\_selection train\_test\_split. A random seed value is used for reproducibility. stratify parameter is used to retain the original fraud- non-fraud distribution is preserved in training and test dataset.

**Exploratory Data Analysis and Data Cleaning:**

Exploratory Data Analysis is one of the most important steps in data science methodology. It reveals important relationship between variables that is not visible from outside. It also identifies the most important variables on the dataset that have most impact on the output class.

Exploratory Data Analysis is done on training dataset to reveal important. pandas\_profiling ProfileReport is used to get distribution of all variables as well as to get the univariate correlation between all variables.





Above Pearson and spearman plot shows the univariate linear and non-linear correlation between each variable with other variables. It also shows that none of the two variables are highly correlated.

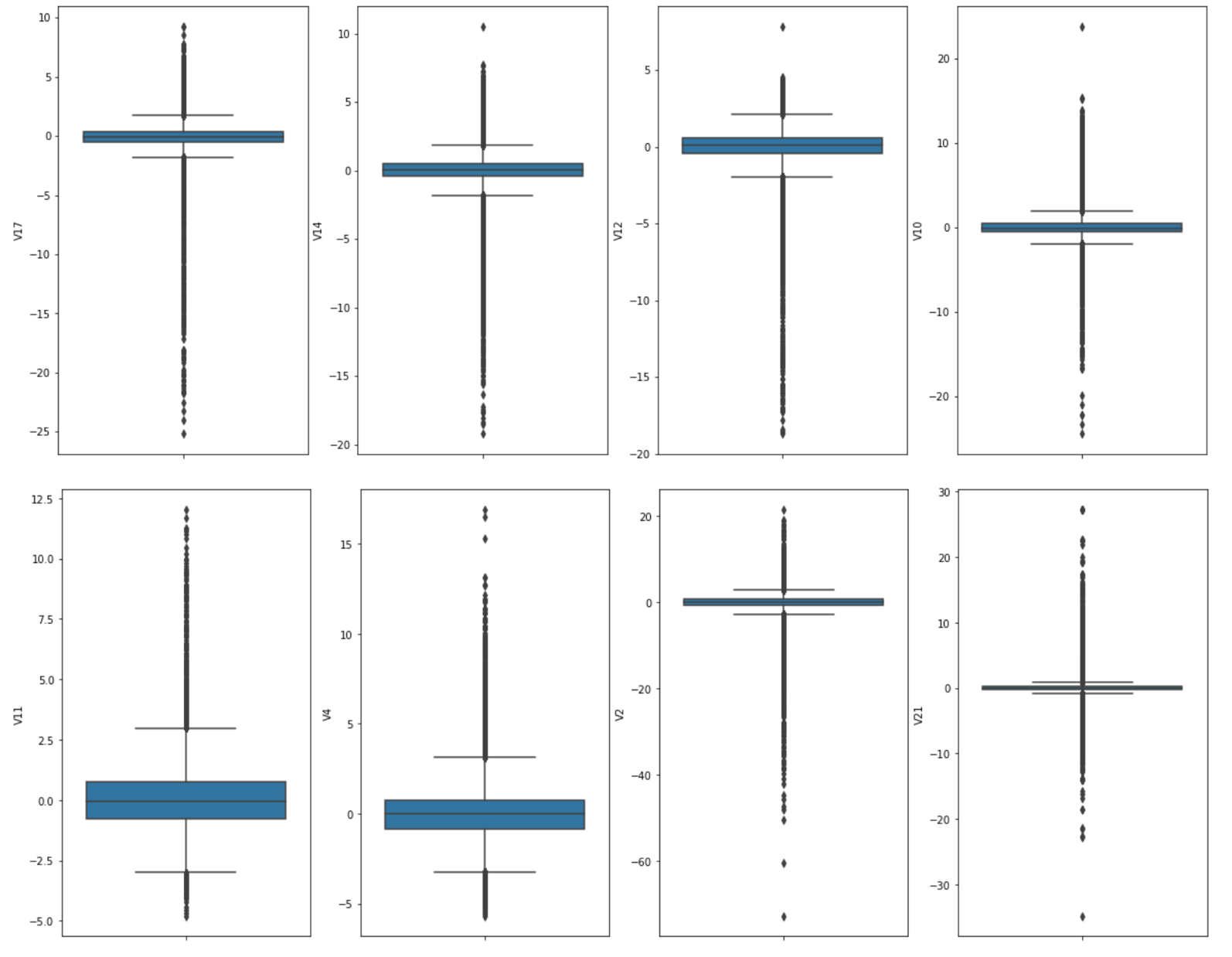
Pearson correlation is calculated between all independent variables with output ‘Class’ variable to identify important variables.

A close up of a logo

Description automatically generated

From the above plot, it can be observed that V17, V14, V12, V10, V11, V4, V2, V21 has higher correlation with output class variable. Any outliers or extreme values in these variables can introduce bias and affect model outcome. So, it is necessary to check for outliers for these variables and remove them if there are any.

Boxplots are drawn for V17, V14, V12, V10, V11, V4, V2, V21. This will show the outliers if there are any.



Amount, V14, V12, V10, V2 & V21 have outliers. And these outliers need to be removed prior to model training. Outliers need to be removed from both fraud and non-fraud Class separately. zscore is used to detect outliers. To remove outliers first the training dataset is split based on fraud and non-fraud class. zscore is calculated for each of these variables and all observations with zscore >= 3 is removed. Once the outliers are removed both classes are merged into one training dataset. As a result of this outlier removal process, it removed 27 fraudulent transactions and 18534 non-fraudulent transactions from training dataset.

Because of PCA transformation, there is no null values in any of the variables. Also, V1 through V28 variables are scaled because these are PCA transformed.

Amount and Time variables are not scaled. In order to normalize the data within a range, it is required to scale these two variables. Scaling will be done in pipelines.

**Pipeline Set-up:**

ColumnTransformer is used to transform the columns. RobustScaler is used for scaling the ‘Amount’ and ‘Time’ columns. All remaining columns are passthrough, no transformation will occur for those columns. Pipeline will have transformation built into it so that it can transform test dataset before feeding into model.

**Evaluation Matrix:**

There are various criteria that can measure the performance of a classification model.

**Accuracy:** It measures how many observations were correctly classified. But in this case if a model predicts all transactions as non-fraudulent transactions, then also it will achieve 99.83% accuracy. So, it won’t be suitable for comparing model’s performance.

**F1 Score:** It’s a harmonic mean between precision and recall. One major problem is, F1 score, it uses a predicated class, not a score.

**ROC Area Under the Curve:** It’s a plot between False positive rate v/s True positive rate. It uses a score from the classification model rather than predicated class. But this is not suitable for evaluating a highly imbalanced dataset.

**Average Precision:** It uses predicated score like ROC AUC does. But it put more importance on the positive class. It looks at positive predictive valuePPVand true positive rateTPR.As we care more about how model is preforming to detect positive classes e.g. detect fraud, Average precision is the best metrics to evaluate the model performance.

As our dataset is highly imbalanced, average precision score will be used as evaluation criteria.

**Models:**

**Random Forest:** It creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It doesn’t have overfitting problem like a regular decision tree. Some of the important parameters of random forest model are:

**n\_estimators – number of trees in the forest.**

**max\_depth** – maximum number of depths in the tree

**max\_features** - The number of features to consider when looking for the best split.

**class\_weight** – weights associated with classes. The “balanced” mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data.

**Logistic Regression:** Logistic regression is a predictive analysis. It is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

Some of the important parameters are:

C: Inverse of regularization strength; must be a positive float.

**class\_weight** – weights associated with classes. The “balanced” mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data.

**Artificial Neural Network:** It works like a human brain. It accepts data through input layer, passes through the one or more hidden layer to the output layer. Each node is connected through weight. It adjusts these weights to minimize the error between predicated and actual output.

Some of the important parameters are:

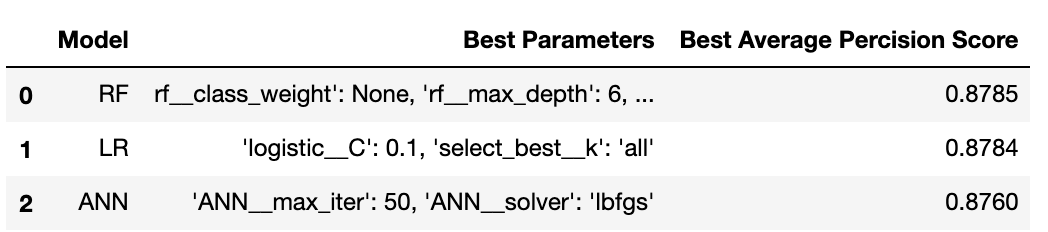
**solver:** The solver for weight optimization. ‘lbfgs’ is an optimizer in the family of quasi-Newton methods. ‘sgd’ refers to stochastic gradient descent.

**max\_iter:** maximum number of iteration.

**learning\_rate\_init**: the initial learning rate used.

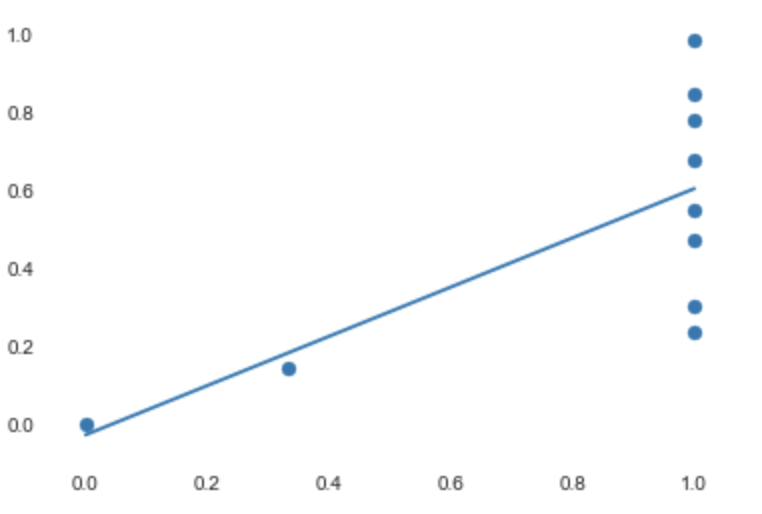
**Model Parameter Tuning:**

GridSearchCV is used to fine tune the parameter. It returns the best fit parameter depending upon average precision score.



**Best Model Selection:**

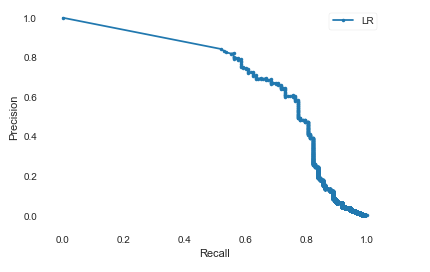
Random Forest model is not calibrated. It returns a vote rather than a probability score. The calibration\_curve shows that the random forest model requires further calibration.



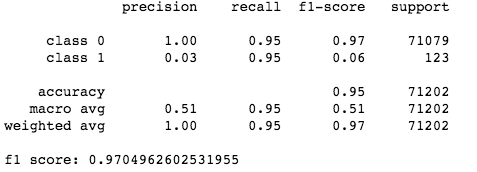
Logistic Regression is selected based on the average precision score. This is less complex than artificial neural network model.

**Model Evaluation:**

Logistic Regression model is tested against the test dataset and an average precision score of 0.66 is obtained.



Higher area under the average precision recall curve signifies, our selected model performed good on test dataset.





**Conclusion:**

* All the Models performed Pretty Well.
* Average Precision Scores are Similar.
* Random Forest can be Improved with Calibration.
* Selected Logistic Regression model based on the Cross-Validation Score and simplicity of this model.
* Trained Logistic Regression model on Entire Dataset using best Parameter Pipeline.

Ref:

<https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html>

<https://shiftprocessing.com/credit-card-fraud-statistics/>