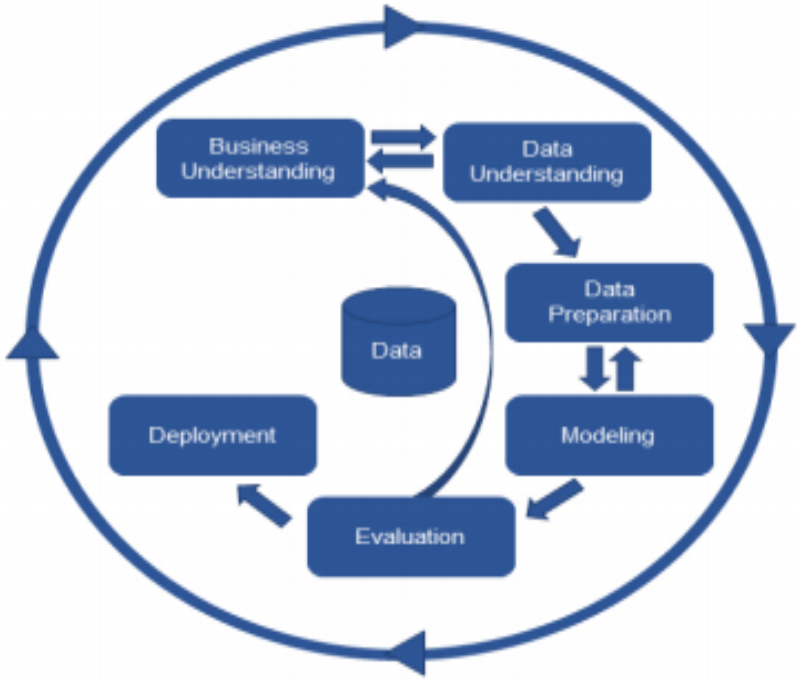
DETECTING ANOMALIES IN CREDIT CARD TRANSACTIONS

FALL 2018 - PREDICTIVE ANALYTICS

PROJECT DESCRIPTION

- **Credit cards** are used by many customers of various **financial institutions** to perform various online, ATM transactions.
- These transactions can sometimes be fraudulent done by people who aim to gain monetary **benefit without authorization**.
- This leads to **financial losses** for the banks and creates a **sense of mistrust** between the bank and customer and could be a major source for banks losing their customers and trust.
- Hence, it becomes necessary for financial institutions to identify and hold such **transactions accountable for security purposes.**

PROJECT WORKFLOW – CRISP DM



- **Business Understanding:** Build a model using the credit card transactional data that can help a financial institution predict fraudulent transactions.
- **Data Understanding:** 492 fake credit card transactions out 284,807 transactions so need to perform Data balancing.
- **Data Preparation:** Data is cleaned, processed and prepared using ETL and EDA.
- **Modelling:** Feature selection and iterating over models to select machine learning algorithms that are a good fit to give a predictive power.
- **Model evaluation:** Select which model suits the business requirement the best using various mathematical measures like Accuracy, AUC etc.
- **Model deployment** : Deployed in the Anaconda Ipython Notebook

https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining

PROJECT WORKFLOW – CRISP DM

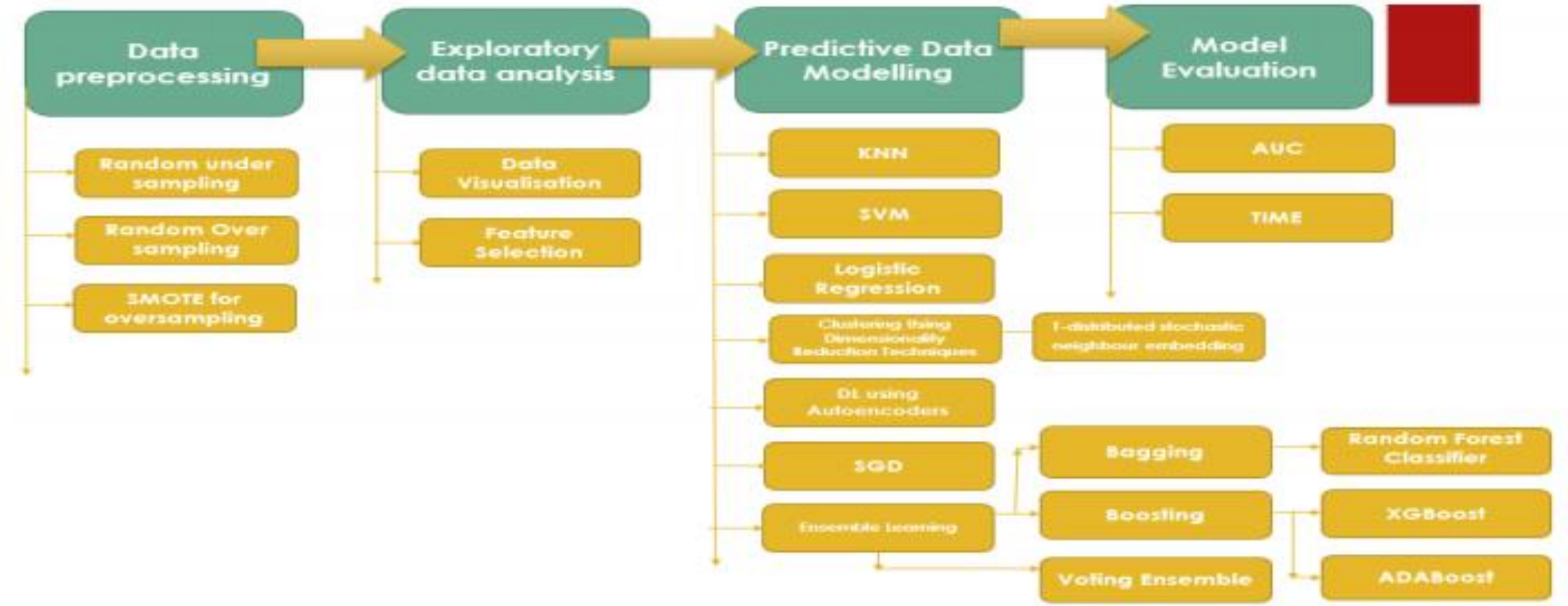


Figure 3: CRISP-DM MODEL

1.BUSINESS UNDERSTANDING

- Analyzing the **credit card transaction data** can help banks find **patterns** that do not **conform to normal transactional patterns.**
- **Predictive modelling** helps us build **a statistical model** made up of number of **predictive variables** that can help us predict **whether the future transactions are fraud or not**.
- **PROBLEM STATEMENT:** This project thus proposes to **build such a model** using **the credit card transactional data** that can help a **financial institution**
- identify **100% of fraudulent transactions** or **anomalous transactions** using various **data mining and machine learning techniques**
- minimise the incorrect number of fraud classifications.

2. DATA UNDERSTANDING

- **Dataset: https://www.kaggle.com/mlg-ulb/creditcardfraud**

COLUMNS

- **Time** - Number of seconds elapsed between this transaction and the first transaction in the dataset.
- **Amount** - Transaction amount. (Not Transformed Data)
- V1, V2, V3…V28 – **PRINCIPAL COMPONENTS** obtained through PCA (Renamed for security).
- **Class1 for fraudulent transactions, 0 otherwise**
- **It's a classification model.**

2. DATA UNDERSTANDING

A. DATA IMBALANCE

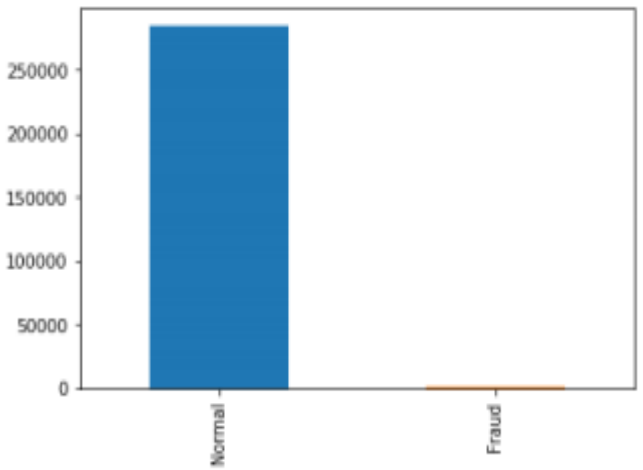


Figure 4: Normal vs Fraud transactions count

- We can see that the dataset contains 31 columns and 284,407 rows out of which 492 transactions are only fraudulent which accounts for 0.172% and hence data imbalance.

2. DATA UNDERSTANDING

B. FEATURE SCALING

- As the other columns are transformed into a standard normal form using PCA we need to transform the columns Time and Amount too into standard normal form.

- Given the distribution of the data, each value in the dataset will have the sample mean value subtracted, and then divided by the standard deviation of the whole dataset so that all features are on the same scale.

2. DATA UNDERSTANDING

C. EXPLORATORY DATA ANALYSIS

- There are no null values in the dataset. So, there is no need to handle such missing values.
- There are 284315 normal transactions and 492 fraudulent transactions.
- Time is not a significant factor in distinguishing fraudulent from non-fraudulent (not significant at 1st glance).

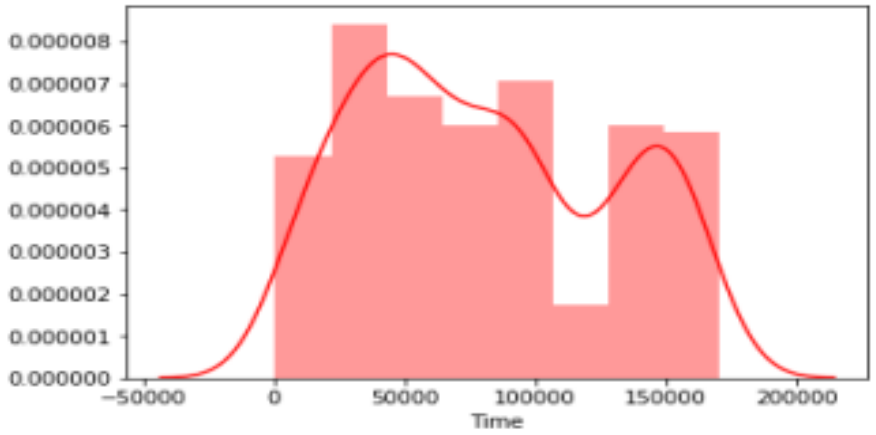


Figure 5: Fraudulent Transactions Vs Time

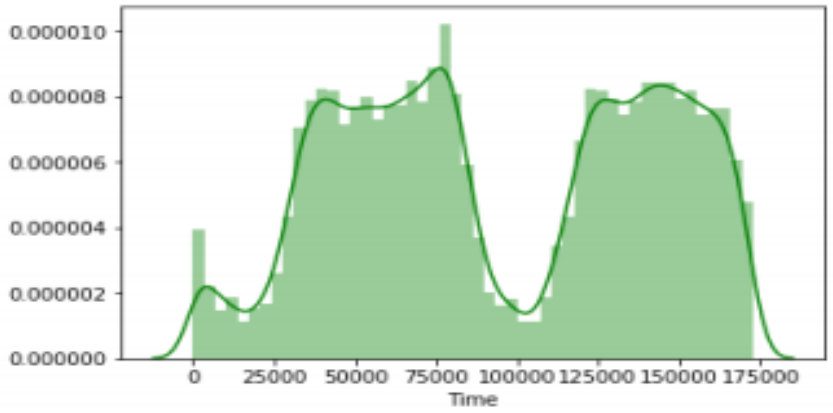


Figure 6: Non-Fraudulent Transactions Vs Time

2. DATA UNDERSTANDING

C. EXPLORATORY DATA ANALYSIS

- **Amount is also not a significant factor in distinguishing fraudulent from non-fraudulent.**

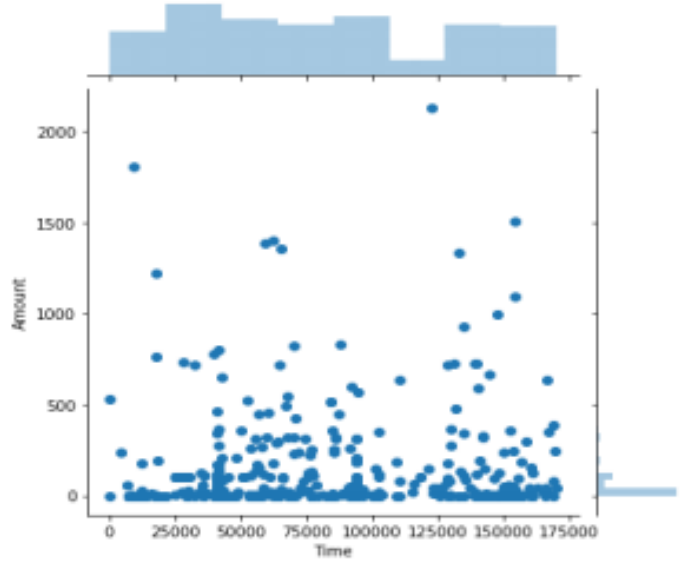


Figure 7: Fraudulent Transactions Amount Vs Time

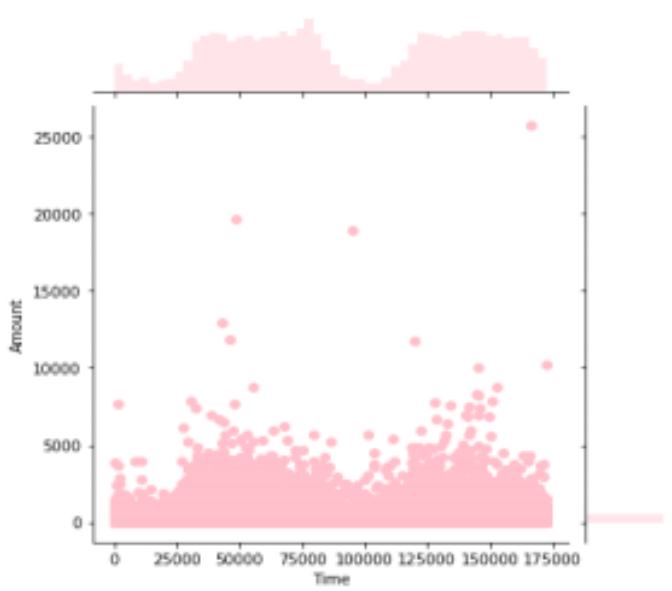


Figure 8: Nonfraudulent Transactions Amount Vs Time

2. DATA UNDERSTANDING

Correlation Analysis

- Heatmap that shows the correlation between the features in the dataset.
- Feature wise correlation to find features distinguishing between fraudulent and non-fraudulent classes. V1-V7, V9, V10, V11, V12, V14, V16- V19 V21 are distinguishing.

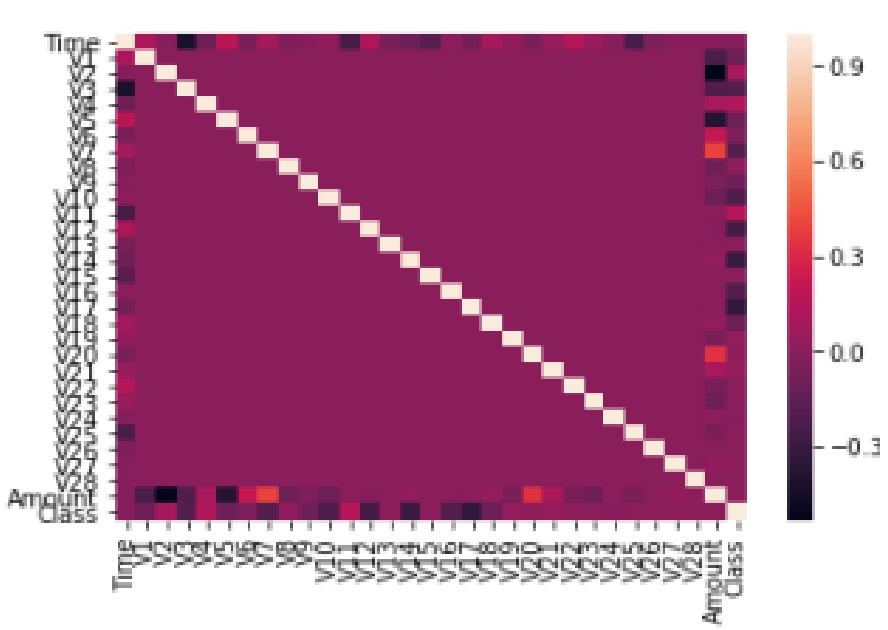
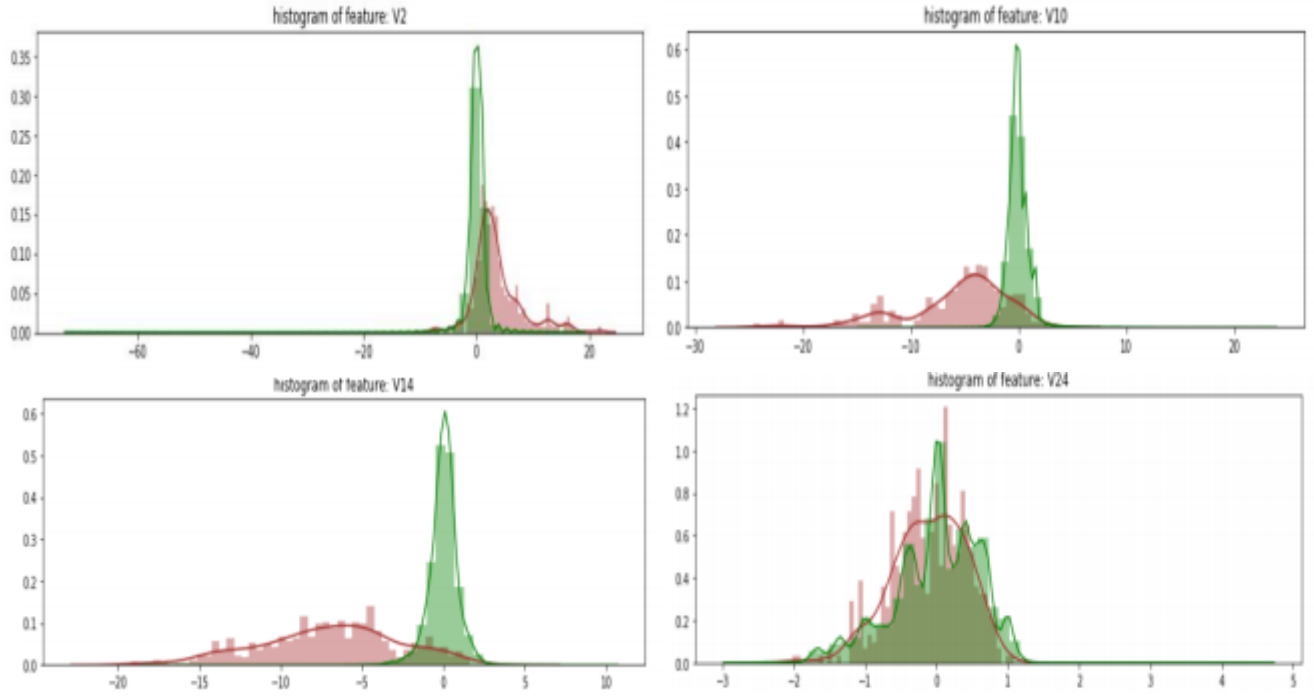


Figure 9: Correlation between features in the dataset



3. DATA PREPARATION

DATA BALANCING STRATEGIES

IMBALANCED DATASET

- Imbalanced data needs to balanced as:

1. It causes overfitting of the data and assume the major class as the output for the testing set.

2. We can fail to understand the correlations between the features due to these anomalies as they are in insignificant amount compared to the major class.

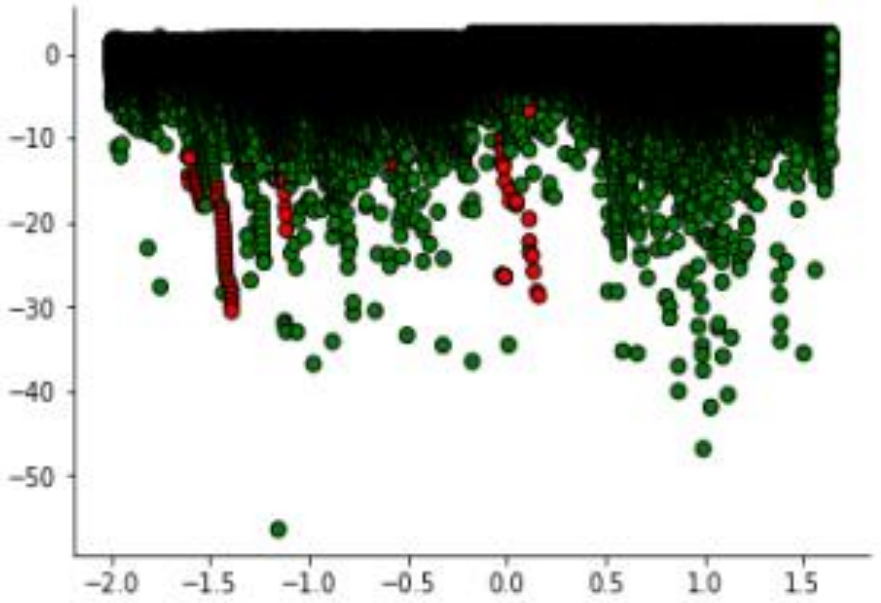


Figure 11: Scatterplot of Imbalanced Dataset

3. DATA PREPARATION

DATA BALANCING STRATEGIES

RANDOM UNDERSAMPLING

- Sampling shifts the data around to increase the "numerical stability" of the resulting models.
- Removes samples from the majority class to equal the minority class.

ADVANTAGES: Reduces Storage Space and Runtime.

DISADVANTAGES: Loss of Important Data.

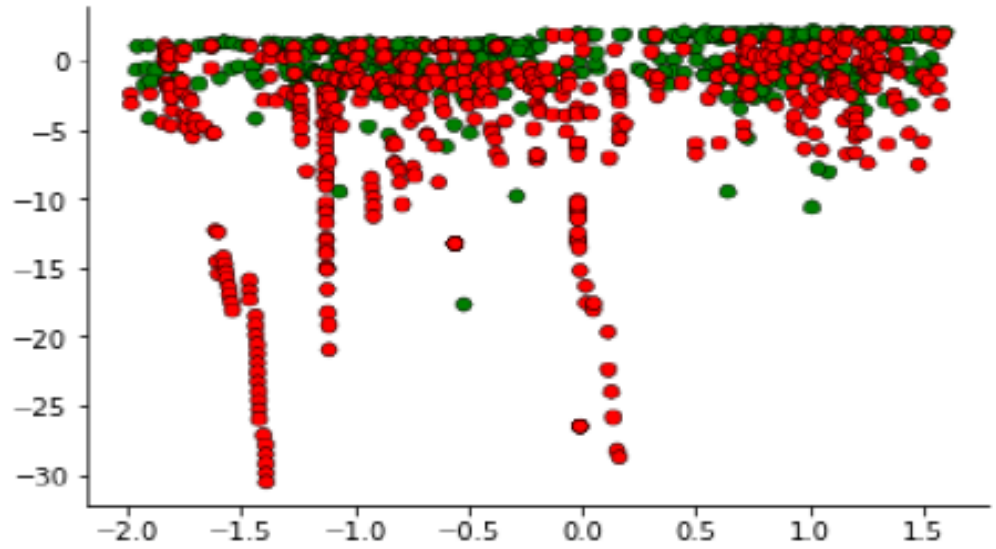


Figure 12: Scatterplot of Random Undersampling on the dataset

3. DATA PREPARATION

DATA BALANCING STRATEGIES

RANDOM OVERSAMPLING

- It is a technique to add samples to the minority class to equal the majority class.

ADVANTAGES: Doesn't lead to information loss and hence better than random Under sampling.

DISADVANTAGES: Can lead to overfitting as only minor class samples are replicated.

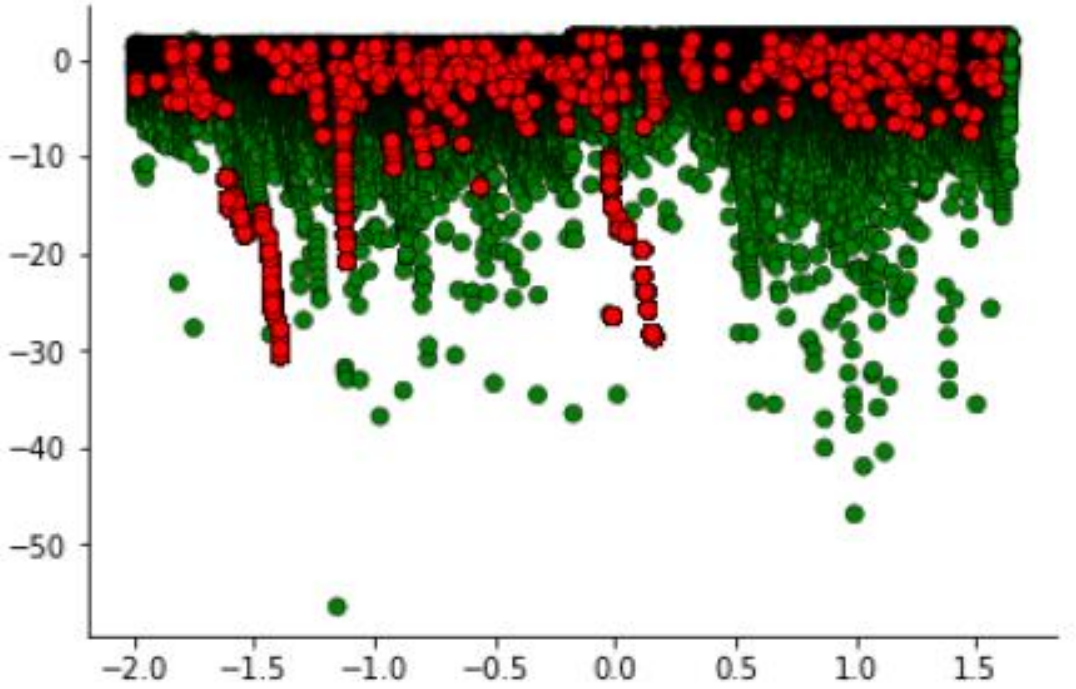


Figure 13: Scatterplot of Random Oversampling on the dataset

3. DATA PREPARATION

DATA BALANCING STRATEGIES

- **Random Oversampling using SMOTE (Synthetic Minority Over-sampling Technique):**
- **Takes a subset of minority class and generates synthetic samples like them and added to the original dataset.**

ADVANTAGES: Removes the issue of overfitting.

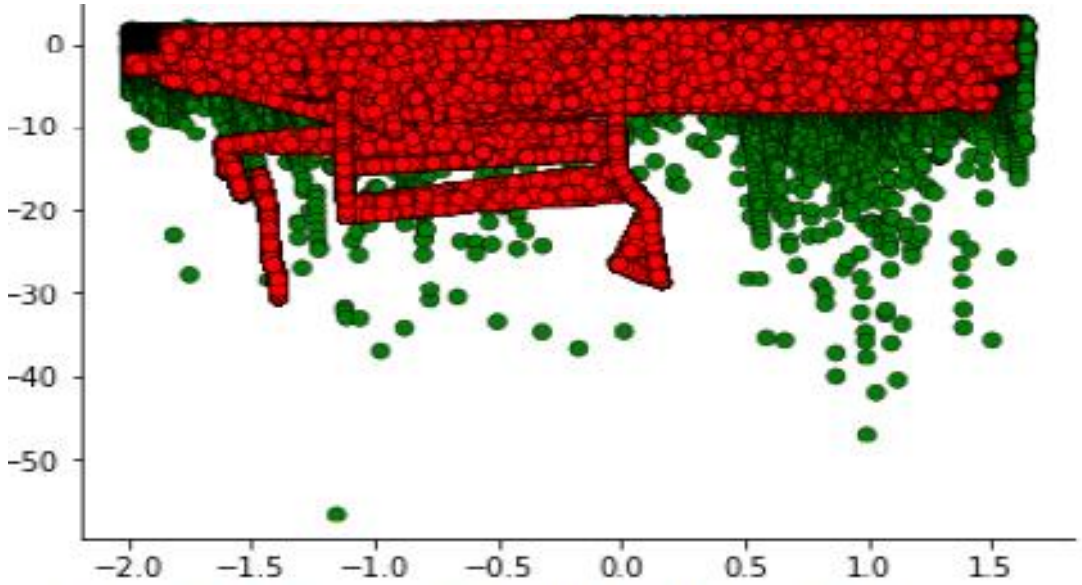
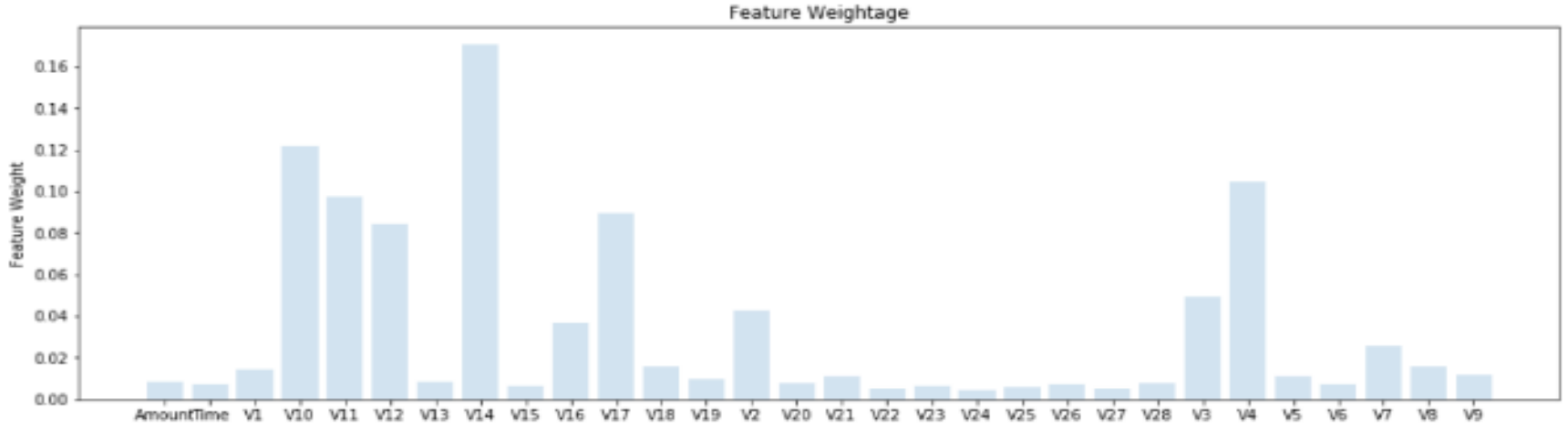


Figure 14: Scatterplot of SMOTE on the dataset

4. MODELLING



- **FEATURE SELECTION**
- **Used Random Forest Classifier, to score the features according to certain weight to get the highly predictive features.**
- **V10, V14, V4 have highest weightage.**

4. PREDICTIVE MODELLING FOR CLASSIFICATION

- **Logistic Regression**
- ✓ Logistic regression models the probability of a class where these values must be transformed to 0 or 1 to make it a binary classifier.
- **K Nearest Neighbour**
- ✓ It uses the majority of the classes of K nearest neighbours as the output for the prediction.
- **Support Vector Machine**
- ✓ **Ensures that examples of different classes can be separated by a hyperplane and are at the maximum distance and hence the predicted variables must be mapped accordingly.**
- **Stochastic Gradient Descent (SGD CLASSIFIER)**
- ✓ Iterative optimizing technique used to minimize the objective function.
- ✓ Samples are randomly selected and the gradient of the loss is estimated each time with a decreasing learning rate.

4. PREDICTIVE MODELLING FOR CLASSIFICATION

Clustering Using t-distributed Stochastic Neighbour Embedding

- t-Distributed Stochastic Neighbor Embedding (t-SNE) is a (prize-winning) technique for dimensionality reduction that is particularly well suited for the **visualization of high-dimensional datasets**.
- It models each high-dimensional object by a two- or three-dimensional point in a way that where **dissimilar objects are modeled by distant points with high probability and similar objects are modeled by nearby points**.
- Is a dimensionality reduction algorithm that clusters fraud and non-fraud samples very clearly after performing dimensionality reduction.
- **Good indicator** at the start to see if predictive models will be able to perform classification better or not as it clearly gives clear distinction between the clusters.

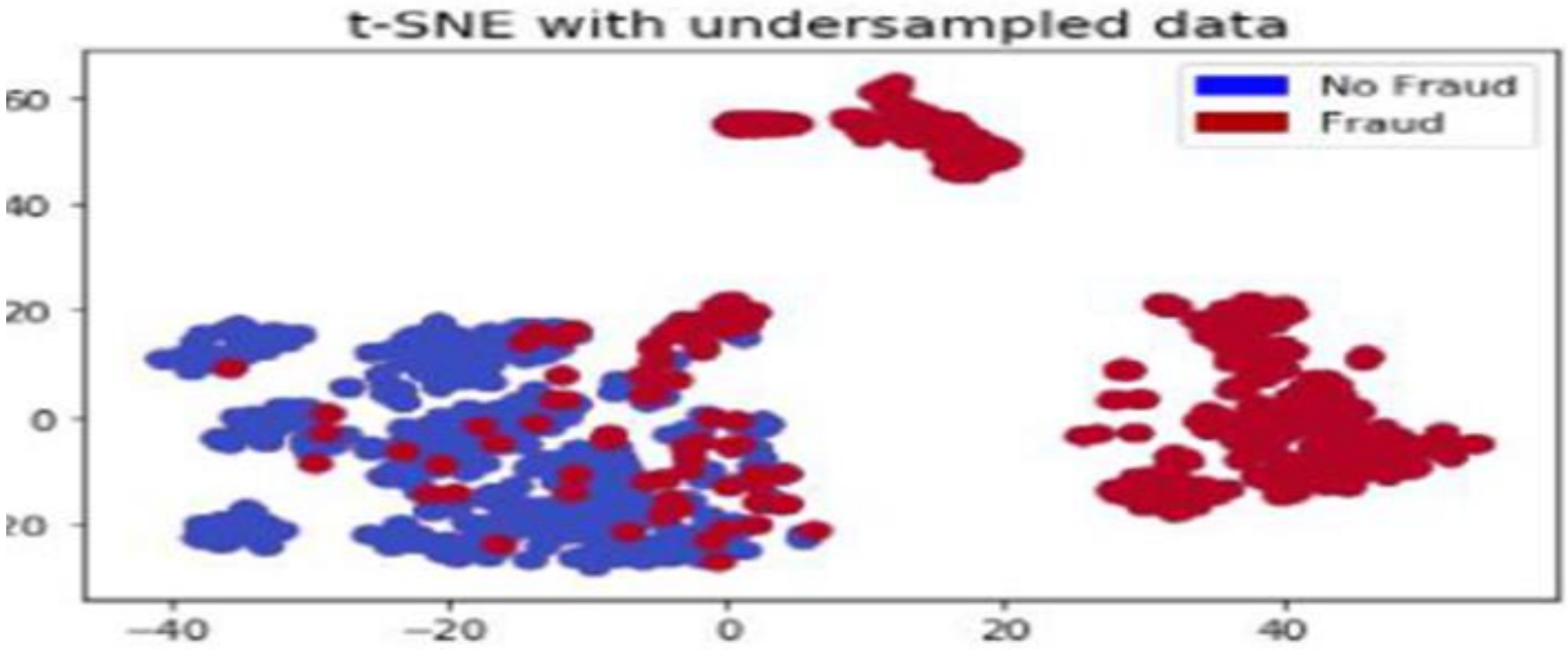
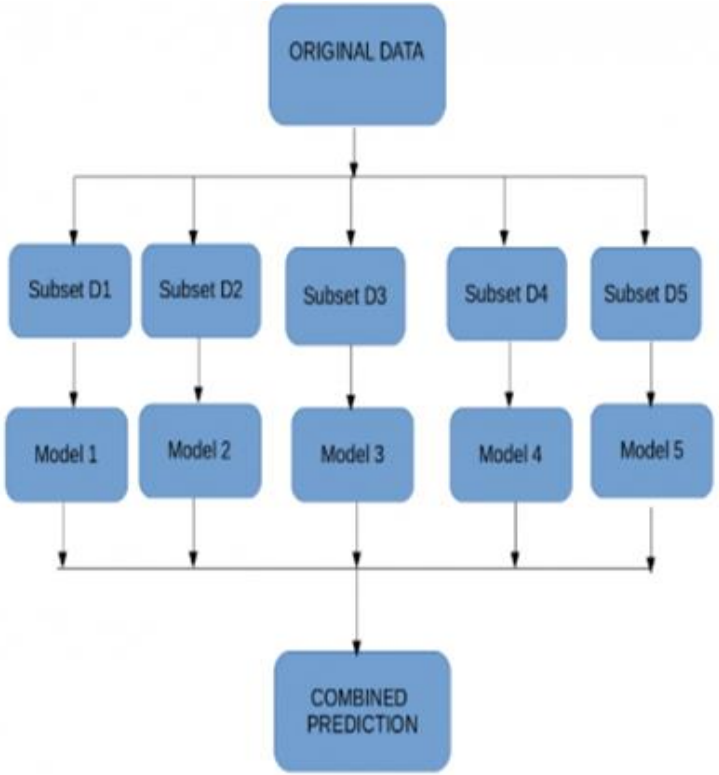


Figure 19: t-SNE on undersampled data

4. ENSEMBLE MODELLING

Ensemble Modelling is a technique where many predictive models are used together to make a **better decision** rather than a single model.

- **Bagging with Random Forest Classifier.**
- Bagging is known as **Bootstrap Aggregating.**
- Instead of applying different models to the same set of data which has a high probability to give the same output
- It vies for creating **multiple subsets** from the original data and applying the predictive model on all of them in **parallel to combine** them in the end to give a strong combined prediction.
- In our project we used Random Forest for Bagging which internally uses decision trees.



https://zhuanlan.zhihu.com/p/40485758

4. BOOSTING

- It is a **sequential process** where the model is applied to subset and the predictions are made on the entire dataset.
- The **error value** is calculated on the predictions and the wrong values are given more weightage.
- Then the process of creating subsets and predicting on the whole dataset is repeated and multiple models are created that try to **minimize the error of the previous model and combines them all in a weighted mean manner** to give the final output.

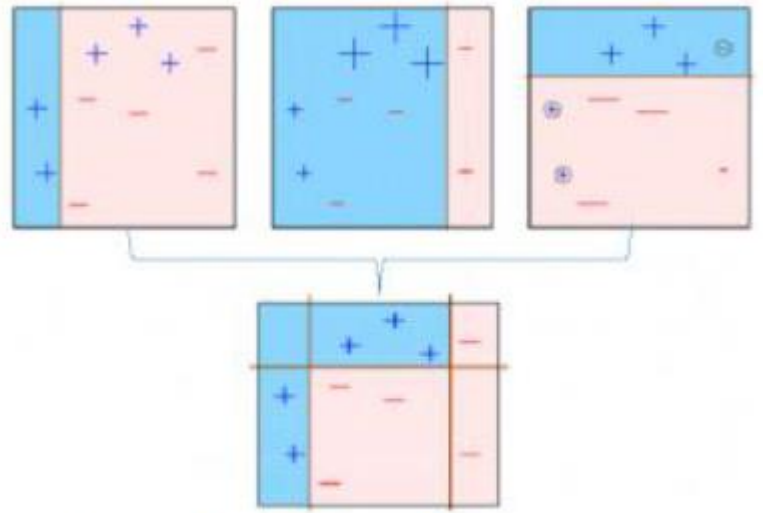


Figure 21: Concept of Boosting

https://www.analyticsvidhya.com/blog/2015/09/questions-ensemble-modeling/

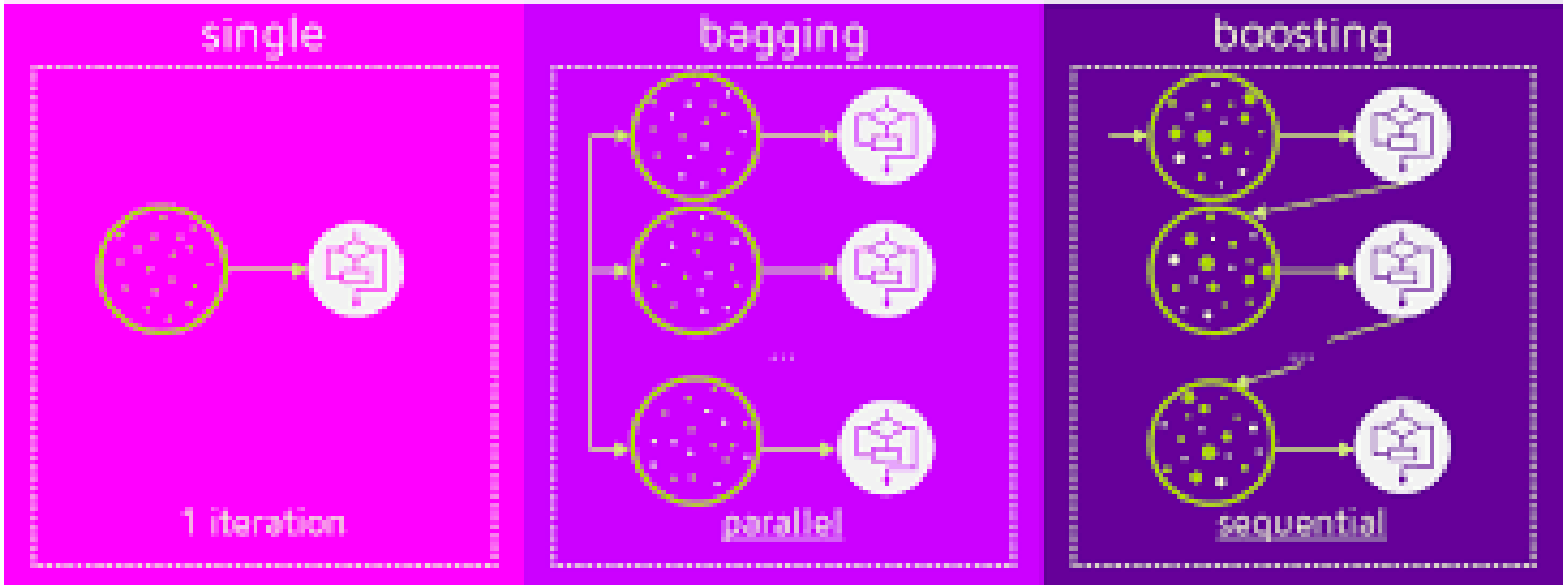
4. BOOSTING

- **XGBoost (Xtreme Gradient Boosting)**

- It can be used for supervised learning tasks such as Regression, Classification, and Ranking. It is built on the principles of gradient boosting framework and designed to "push the **extreme computation limits** of machines to provide a portable, scalable and accurate library."
- Take a weak learner and add to it another weak learner at every step to build a strong learner and increase the performance.
- The new weak learners are added to concentrate on the areas where the **existing learners are performing poorly**.

- **AdaBoost**

- Adaptive boosting also known as AdaBoost is the simplest boosting algorithm that works on improving the areas where the base learner fails.
- The base learner is a machine learning algorithm which is a weak learner and upon which the boosting method is applied to turn it into a strong learner by applying it over and over.
- we **exaggerate the weights of these misclassified samples** so that that they have a better chance of being selected when sampled again.



https://medium.com/greyatom/a-quick-guide-to-boosting-in-ml-acf7c1585cb5

4. VOTING ENSEMBLE

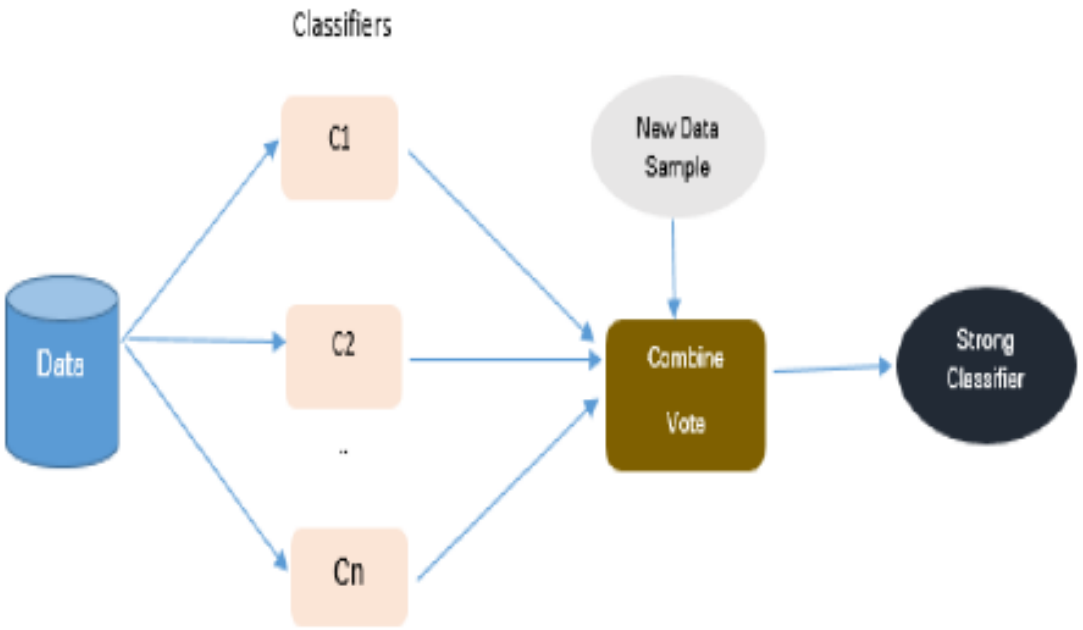


Figure 22: Concept of Voting Ensemble

- In this multiple independent machine learning models are run on the training data and as per the weighted (optional) output is taken into consideration as the number of votes or average of all the classifiers.
- In our project we tried to create a voting ensemble of Logistic regression, K Nearest Neighbour, Stochastic Gradient Descent classifier and SVM classifier.

http://ai.51cto.com/art/201703/535077.htm

5. MODEL EVALUATION

- Model evaluation done with oversampled data with SMOTE.
- Choosing performance metric as accuracy isn't the correct measure and gives the great feeling of your model being good as it majorly predicts the major class as the predicted value.
- We choose **area under the ROC curve (AUC)** as the performance metric for the imbalanced dataset which can test the correct number of classifications for a model.
- A model that has 100% correct classification has an AUC of 1.0 while 0.0 if all are wrong.

5. MODEL EVALUATION

Sr_no	Classification Model	Accuracy (AUC)	Speed (Time)	Average Rank (Accuracy & Speed)
1	Logistic Regression	0.9365	7.6 s	4
2	K Nearest Neighbour	0.9998	1.5e+02 s	1
3	Support Vector Machine	0.9991	2.7e+03 s	5
4	Stochastic Gradient Descent	0.9426	1.2e+02 s	4
5	Random Forest	1.0000	7.6e+02 s	2
6	XGBoost	0.9834	2.4e+02 s	3
7	ADABoost	0.9689	5e+02 s	5
8	Voting Ensemble	0.9536	3e+03 s	6

Accuracy

1. Random Forest
2. K Nearest Neighbour
3. Support Vector Machine
4. XGBoost
5. AdaBoost
6. Voting Ensemble
7. Stochastic Gradient Descent
8. Logistic Regression.

Speed

1. Logistic Regression
2. Stochastic Gradient Descent
3. K Nearest Neighbour
4. XGBoost
5. AdaBoost
6. Random Forest
7. Support Vector Machine
8. Voting Ensemble

Speed and Accuracy

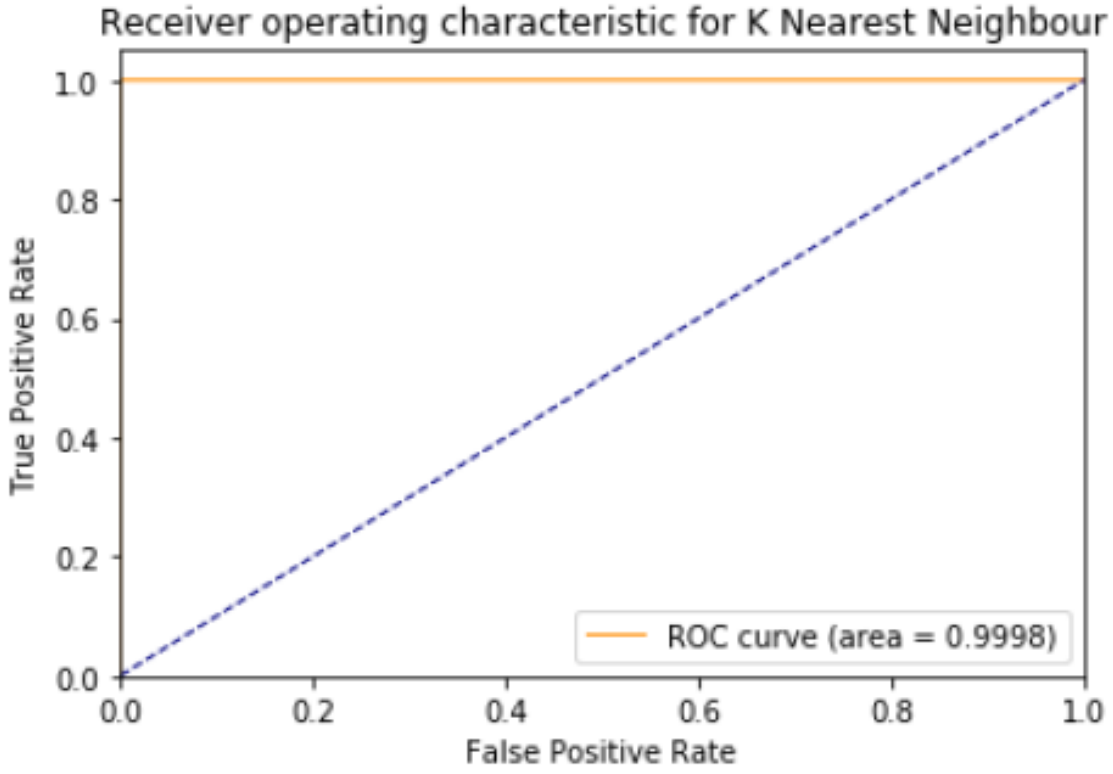
1.K Nearest Neighbour
2. Random Forest
3. XGBoost
4. Logistic Regression
5. Stochastic gradient descent
6. Support Vector Machine
7. AdaBoost
8. Voting Ensemble.

5. MODEL EVALUATION - RESULTS

- It can be seen ensemble models **are not always the best predictive models in terms of speed** mostly as they take time to build though they give good accuracy.
- It's useful when you don't care about speed, and want the best classification performance possible. If you use N models in the ensemble, it will be roughly N times slower to both train and evaluate.
- They are best to be used in certain cases where they give large improvement in accuracy than single classifier (Weak learners).
- The model that is closest to the true data generating process will always be best and will beat most ensemble methods.
- From the above table we can see that **K nearest Neighbour** has the best accuracy and speed of detection.

Figure 25: Comparison of Classification Models with respect to AUC

5. MODEL EVALUATION - RESULTS



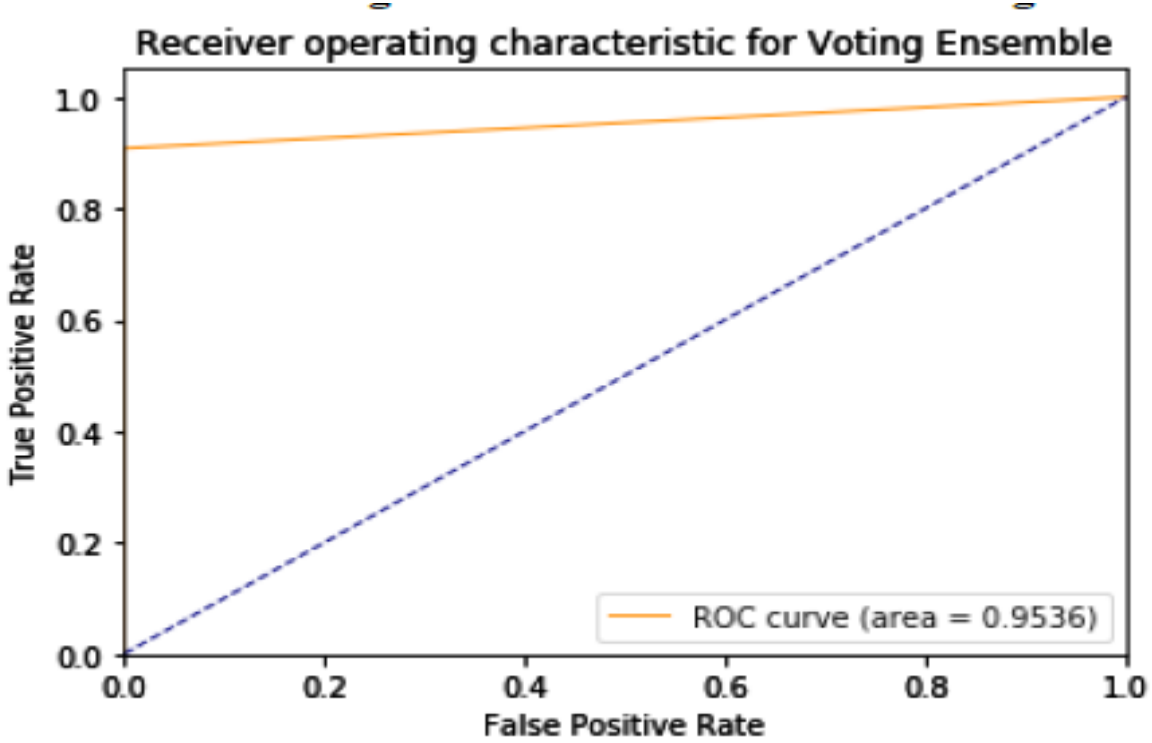


Figure 33: ROC for Voting Ensemble

- An ROC curve plots false positive rate (on the X axis) against true positive rate (on the Y axis).

  false positive rate = FP / (FP + TN)
  true positive rate = TP / (TP + FN)

- It's clear that these rates are irrespective of the actual positive/negative balance on the test set.
- Increasing the number of positive samples in the test set by 2x would increase both TP and FN by 2x, which would not change the true positive rate at any threshold.
- Similarly, increasing the number of negative samples in the test set by 2x would increase both TN and FP by 2x, which would not change the false positive rate at any threshold.
- The shape of the AUC and ROC curve and are not sensitive to the distribution of the class.
- Accuracy = (TP+TN)/(TP+TN+FP+FN) is sensitive to the class distribution.

6. MODEL DEPLOYMENT

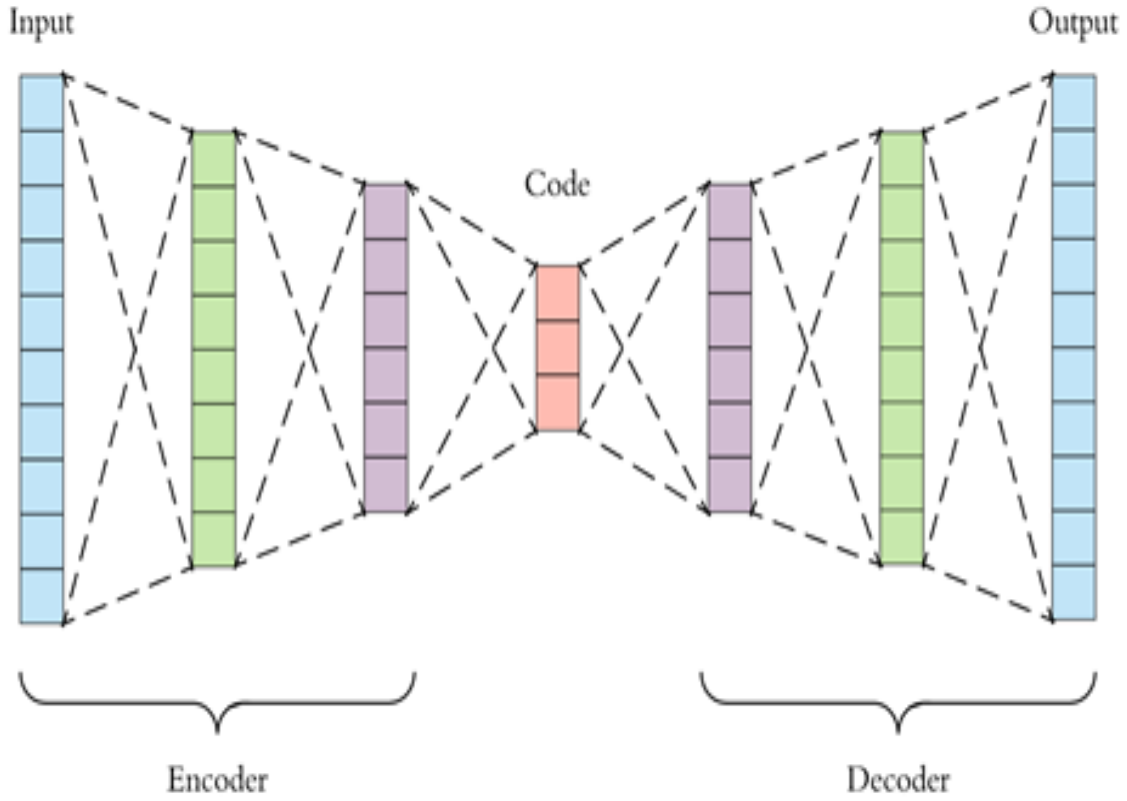
- The project has been developed using **python 3.6 and deployed on Jupyter Notebook.**

Libraries like :

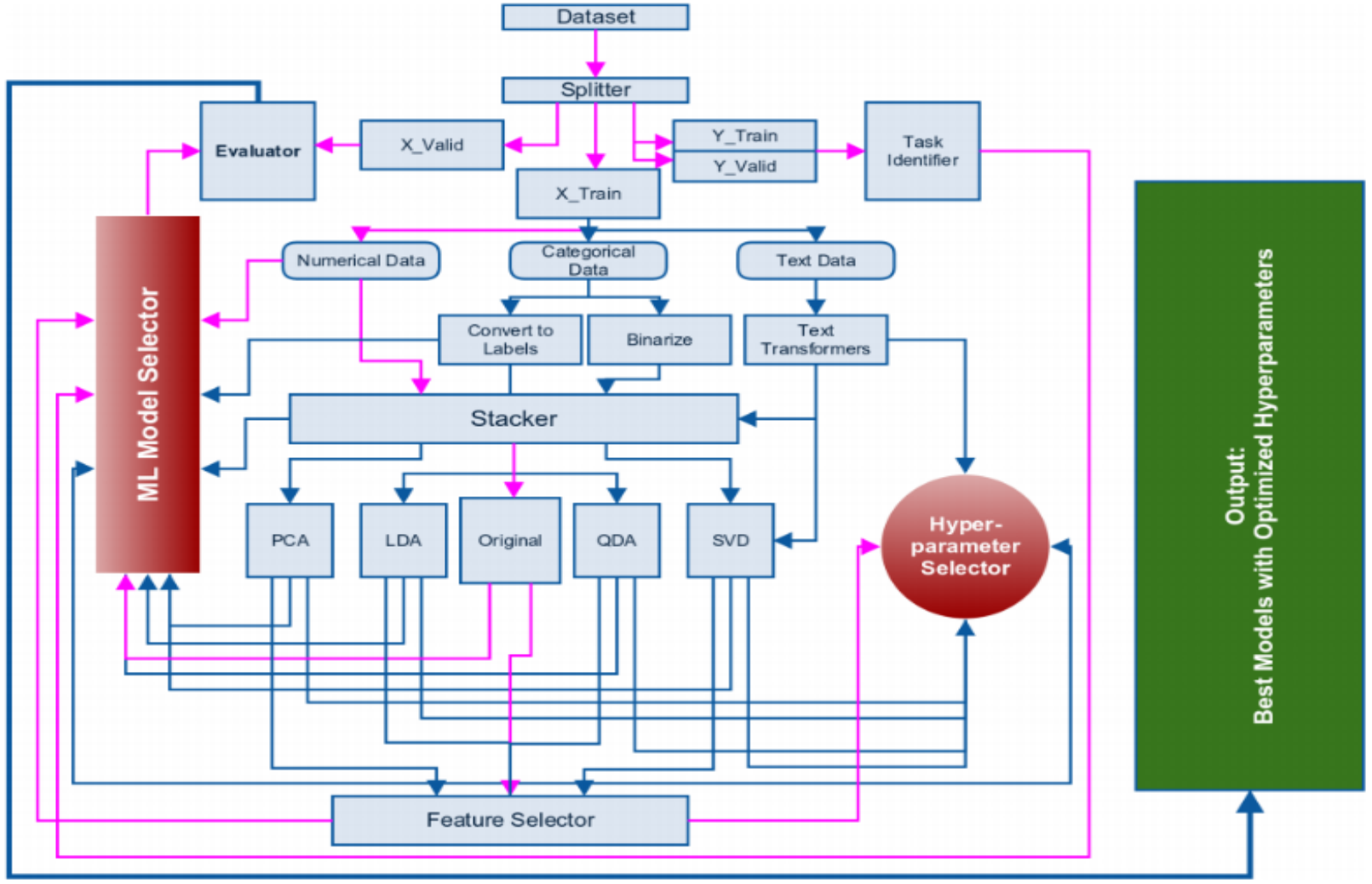
- **Pandas** for **data manipulation**
- **imblearn** for **data balancing**
- **Scikit- Learn** for **predictive modelling for machine learning algorithms and evaluation**
- **NumPy** for **data manipulation**
- **Matplotlib** and **Seaborn** for **Data Visualization.**

CONCLUSION AND FURTHER WORK

- So, when a **dataset is imbalanced,** we need to convert it into a balanced dataset and use area under the curve as the evaluation metric.
- Detecting **fraudulent credit card transactions** is a tedious process which is subjected to change with varying patterns of the fraudsters and building a 100% accurate fraud detection system is in research.
- Various state of art and more powerful approaches are coming into picture like **deep learning like autoencoders** which has the ability to detect features or changes in the data using deep neural networks in the run rather than it being provided features unlike machine learning models.
- So, the future work can be how to improvise the model with deep learning and **build a model that adapts the changes in the data on the go** which then can be useful for building a more powerful credit card fraud detector system.



https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798



ML FRAMEWORK FOR SOLVING ANY PROBLEM

http://blog.kaggle.com/2016/07/21/approaching-almost-any-machine-learning-problem-abhishek-thakur/

THANKYOU

- SREE GOWRI ADDEPALLI (N11837176) [sga297@nyu.edu]
- SREE LAKSHMI ADDEPALLI (N12311918) [sla410@nyu.edu]