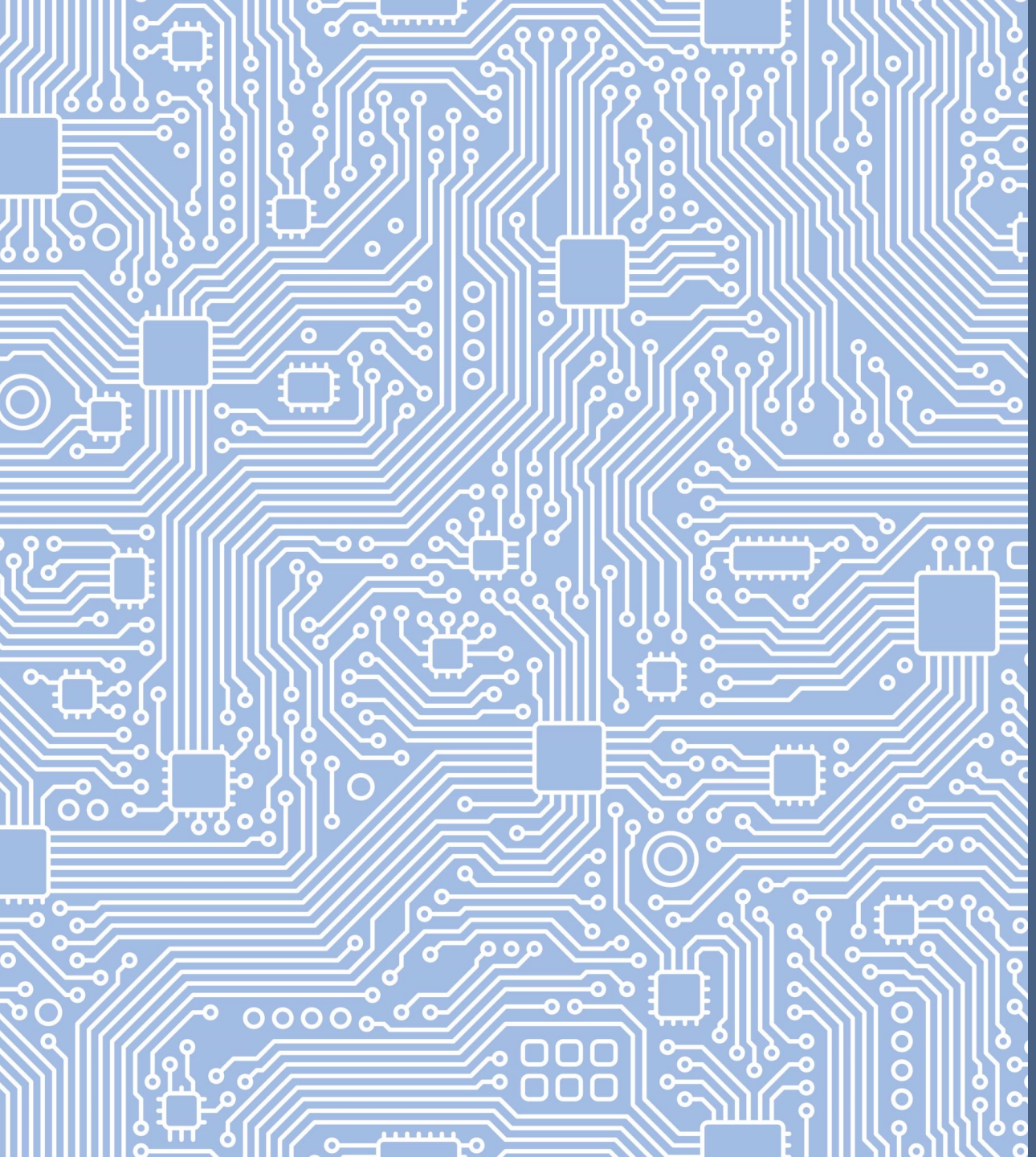


# CREDIT CARD FRAUD DETECTION


*By:*

*Olu Olayeye*

*Daniel Saulsberry*

*Seth Boswell*

*Malli Montano*

# OVERVIEW

- The Annual Data Book compiled by the Federal Trade Commission reports that Credit card fraud accounted for 393,207 of the nearly 1.4 million reports of identity theft in 2020.
- This makes credit card fraud the second most common type of identity theft reported, behind only government documents and benefits fraud for that year.
- Some surveys suggest that a typical organization loses 5% of their yearly revenues to fraud. These numbers can only increase since the number of non-cash transactions increases, providing more opportunities for credit card fraud.

- For retailers and banks to not lose money, procedures must be put in place to detect fraud prior to it occurring.
- Credit card companies must identify fraudulent credit card transactions so that customers are not charged for items that they did not purchase.
  - To combat this problem, financial institution traditionally uses rule-based approaches to identify fraudulent transactions.
  - These algorithms use strict rules to determine when a transaction is fraudulent.

# CHALLENGES OF A STRICT RULED-BASED ALGORITHM INCLUDE:

- Any new scenario that could lead to fraud needs to be manually coded into the algorithm.
- Increases in customers and size of data leads to a corresponding increase in the human effort, time and cost required to track new scenarios and update the algorithm.
- Since the algorithm cannot go beyond defined rules, it cannot dynamically recognize new scenarios that could result in fraudulent transaction.

# HOW DO WE OVERCOME THESE LIMITATIONS?

- Organizations are beginning to utilize machine learning and data science to build fraud detection systems.
- Given the size of available data, computational resources, and powerful machine learning algorithm available today, data science and machine learning processes will be able to find patterns in data and detect fraud easily.

# GOAL

- The goal of this Credit Card Fraud Detection project is to classify a transaction as valid or fraudulent in a large dataset.
- Since we are dealing with discrete values, this is a binary classification problem, and we would employ the use of a supervised machine learning algorithm.

# DATASET USED:

Transactions made by credit cards in September 2013 by European cardholders.

Transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions.

# CONTROL FLOW

1. Understanding the problem
2. Importing required libraries and understanding their use
3. Importing data and learning its structure
4. Performing basic EDA
5. Scaling different variables
6. Outlier treatment
7. Building basic Classification model with Random Forest
8. Nearmiss technique for under sampling data
9. SMOTE for oversampling data

# CONTROL FLOW (CONTINUED):

10. cross validation in the context of under sampling and oversampling

11. Pipelining with sklearn/imblearn

12. Applying Linear model: Logistic Regression

13. Applying Ensemble technique: Random Forest

14. Applying Non-Linear Algorithms: Support Vector Machine, Decision Tree, and k-Nearest Neighbor

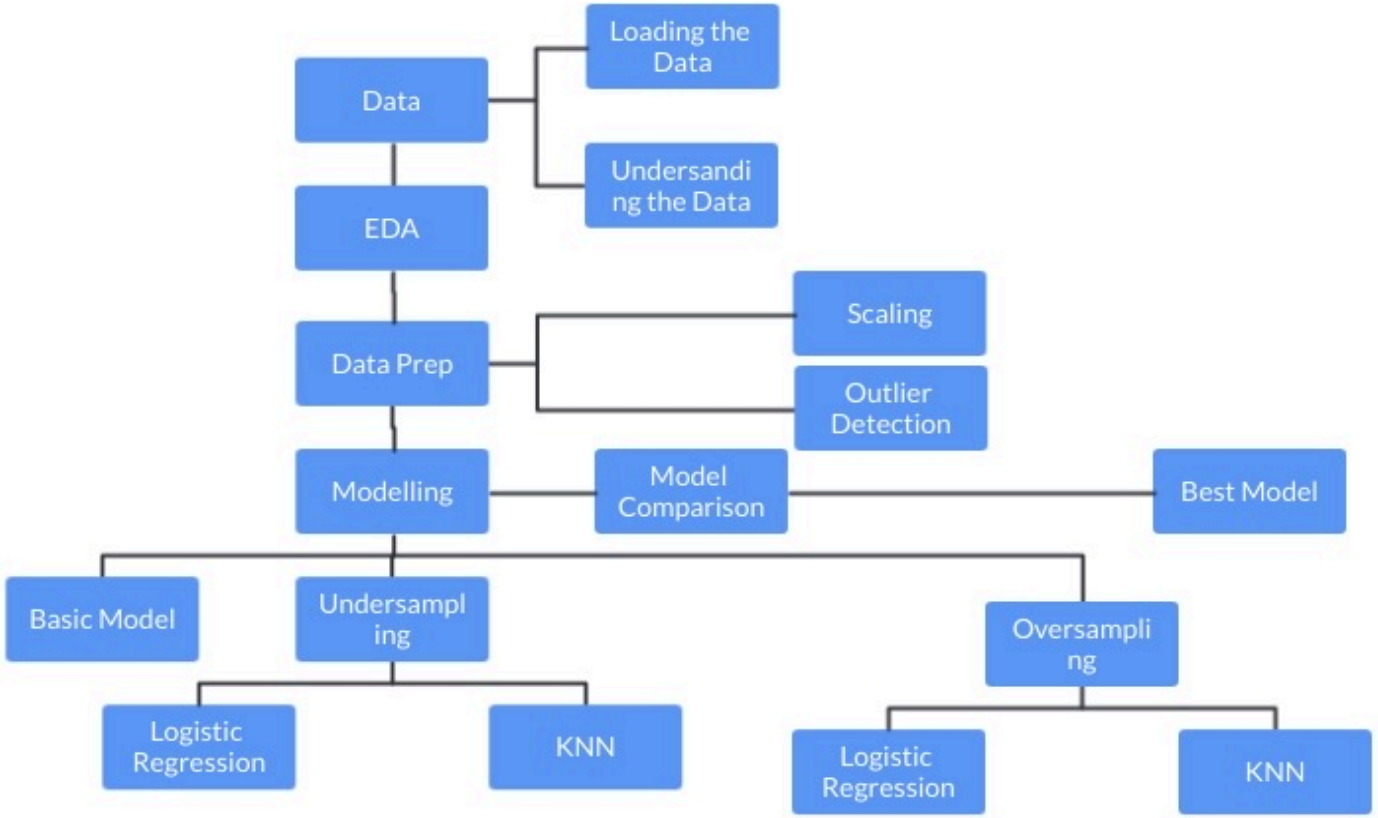
15. Making predictions on test set and computing validation metrics

16. ROC curve and Learning curve

17. Comparison of results and Model Selection

18. Visualization with seaborn and matplotlib

# SOLUTION WORKFLOW




TECHNOLOGY

# LOGISTIC REGRESSION

Logistic regression is a classification algorithm used to find the probability of event success and event failure.

It is used when the dependent variable is binary (0/1, True/False, Yes/No) in nature.

It supports categorizing data into discrete classes by studying the relationship from a given set of labelled data.

It learns a linear relationship from the given dataset and then introduces a non-linearity in the form of the Sigmoid function.

# WHY LOGISTIC REGRESSION?

Logistic regression is easier to implement, interpret, and very efficient to train.

It makes no assumptions about distributions of classes in feature space.

It not only provides a measure of how appropriate a predictor (coefficient size) is, but also its direction of association (positive or negative).

Good accuracy for many simple data sets and it performs well when the dataset is linearly separable.

# RANDOM FOREST

- Random forest is a technique used in modeling predictions and behavior analysis and is built on decision trees.
- It contains many decision trees representing a distinct instance of the classification of data input into the random forest.
- The random forest technique considers the instances individually, taking the one with most votes as the selected prediction.

# WHY RANDOM FOREST?

- It reduces overfitting in decision trees and helps to improve the accuracy.
- It is flexible to both classification and regression problems.
- It works well with both categorical and continuous values.
- It automates missing values present in the data.
- Normalizing of data is not required as it uses a rule-based approach.

# SVM

- While SVMs do a good job recognizing speech, face, and images, they also do a good job at pattern recognition.
- Pattern recognition aims to classify data based on either a priori knowledge or statistical information extracted from raw data, which is a powerful tool in data separation in many disciplines.

# WHY SVM?

SVM works relatively well when there is a clear margin of separation between classes.

SVM is effective in high dimensional spaces.

SVM can be used for other types of machine learning problems, such as regression, outlier detection, and clustering.

SVM is relatively memory efficient

# K-MEANS CLUSTERING

- Is a centroid-based algorithm, or a distance-based algorithm, where we calculate the distances to assign a point to a cluster. In K-Means, each cluster is associated with a centroid

# WHY K-MEANS CLUSTERING?

It is easy to implement k-means and identify unknown groups of data from complex data sets. The results are presented in an easy and simple manner.

K-means algorithm can easily adjust to the changes. If there are any problems, adjusting the cluster segment will allow changes to easily occur on the algorithm.

K-means is suitable for many datasets, and it's computed much faster than the smaller dataset. It can also produce higher clusters.

The results are easy to interpret. It generates cluster descriptions in a form minimized to ease understanding of the data.

Compared to using other clustering methods, a k-means clustering technique is fast and efficient in terms of its computational cost


PGADMIN DATABASE TO STORE OUR DATASET & SOME INTERMEDIATE RESULTS...

# DATABASE APPROACH

Load raw dataset into AWS S3 bucket/PgAdmin.

Connect to AWS S3 bucket/PgAdmin and read data into Pandas.

Load the raw data into a PgAdmin Database Instance located in AWS.

Perform preprocessing steps and store cleaned data in a new table in AWS S3 bucket/PgAdmin.

Store some intermediate results (which can be used later for visualization) in AWS S3 bucket/PgAdmin.

The connection and S3 bucket details are in the Segment_One Jupyter Notebook.

A notebook that contains the code of the above steps is part of this repository.

# DATA CLEANING AND ANALYSIS

- This project will utilize Jupyter notebook and the pandas library to perform data cleaning and analysis.

# DESCRIPTION OF COMMUNICATION PROTOCOLS

Communication for this project was through the Slack Group Chat.

Every team member worked on their individual branch and created a pull request which will collectively approved in the slack group chat before a designated member approves the pull request in GitHub.

The designated team member then created a request to push changes to the main branch.

# RESULTS

# EXPLORATORY DATA ANALYSIS

# UNIVARIATE ANALYSIS

Univariate plots show that the dataset is highly imbalanced. The pie chart shows an imbalance in the data, with only 0.17% of the total cases being fraudulent.

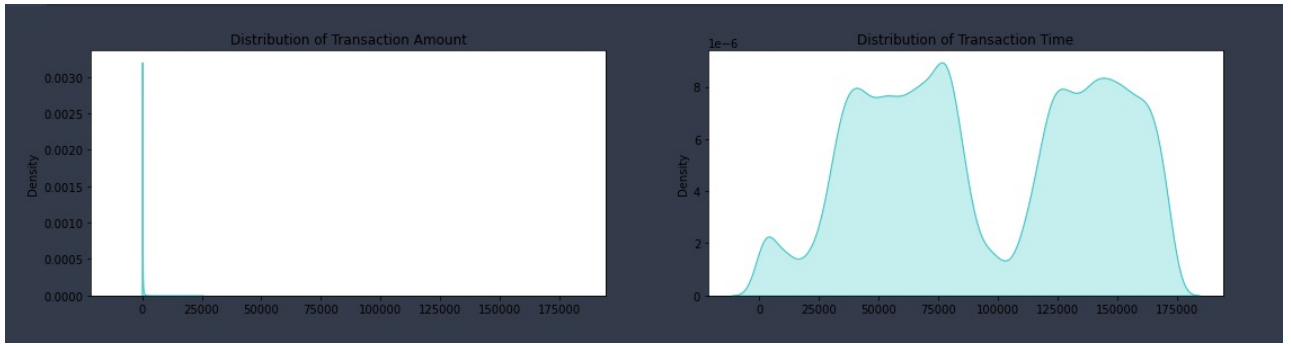
The univariate distribution plot of the time and amount feature showed we have a dataset with some large outlier values for amount, and the time feature is distributed across two days.

Bivariate plots of all features grouped by transaction class, showed that the valid transaction class has a normal distribution shape across most of the features, conversely, the fraud class show long-tailed distribution across many of the features.

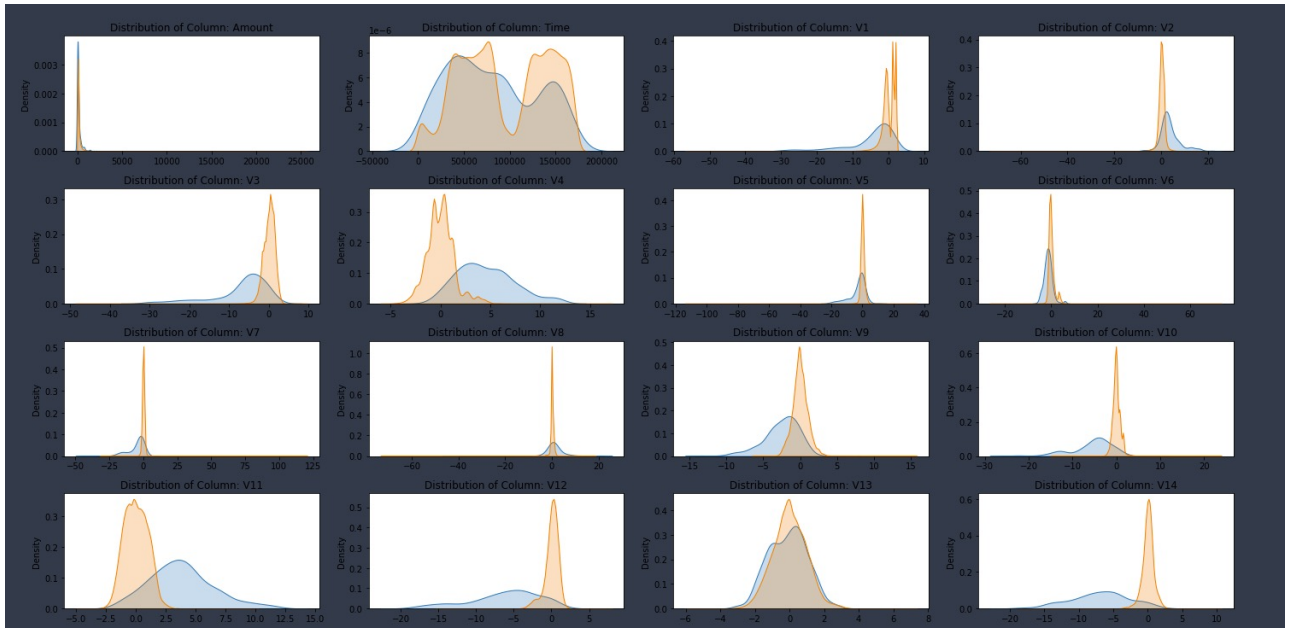
# UNIVARIATE ANALYSIS

```
Fraudulent Transactions: 492
Valid Transactions: 284315
Proportion of Fraudulent Transactions: 0.001727485630620034

<AxesSubplot:ylabel=' '>
```

# BIVARIATE ANALYSIS



# NAÏVE MODEL RESULTS

While the naive logistic classifier accuracy is 100%, our classifier did not do an excellent job at predicting fraudulent transactions. With precision and recall of 0.84 and 0.62, we would need a better understanding of the dataset to determine the best way to improve the recall metric.

While the naive random forest classifier accuracy is 100%, and precision is 95%, our random forest classifier only achieved a 77% recall. We would need a better understanding of the dataset to determine the best way to improve the recall metric.

# NAÏVE MODEL

| | Classifier | Accuracy | Recall Score | Precision Score |
|---|---|---|---|---|
| 0 | Logistic Regression | 0.999143 | 0.617886 | 0.844444 |
| 1 | Random Forest | 0.999551 | 0.772358 | 0.959596 |

# UNDERSAMPLING MODEL RESULTS

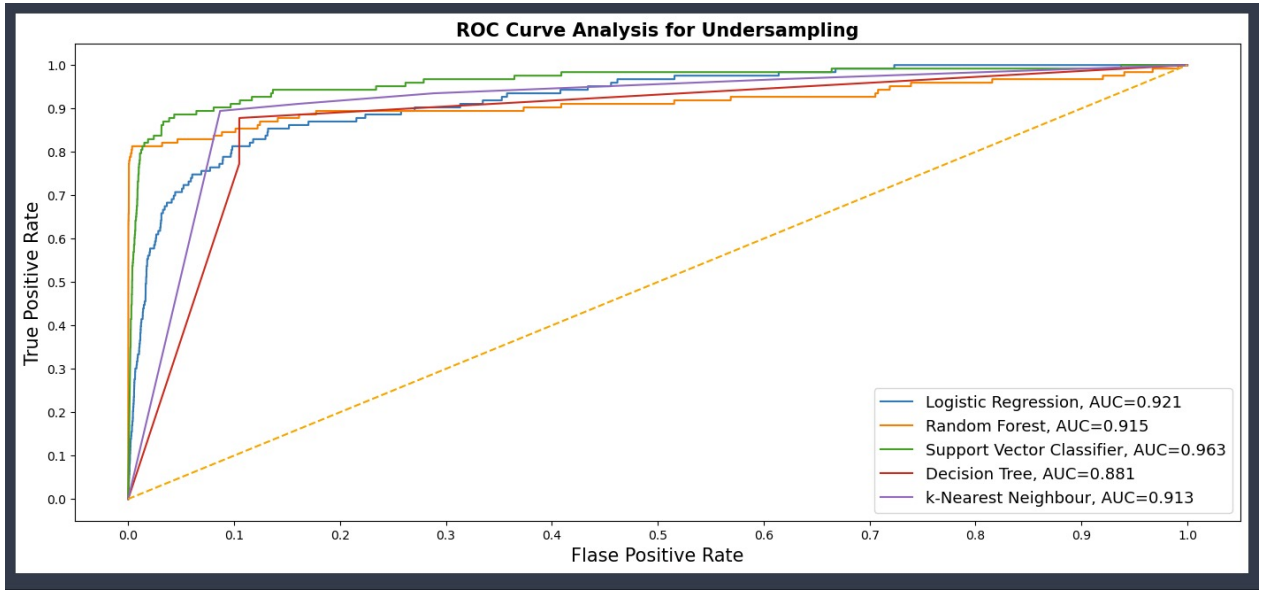
By Undersampling our the majority class in our dataset, all classifiers achieved recall scores greater than 85% with the exception of the Support vector classifier.

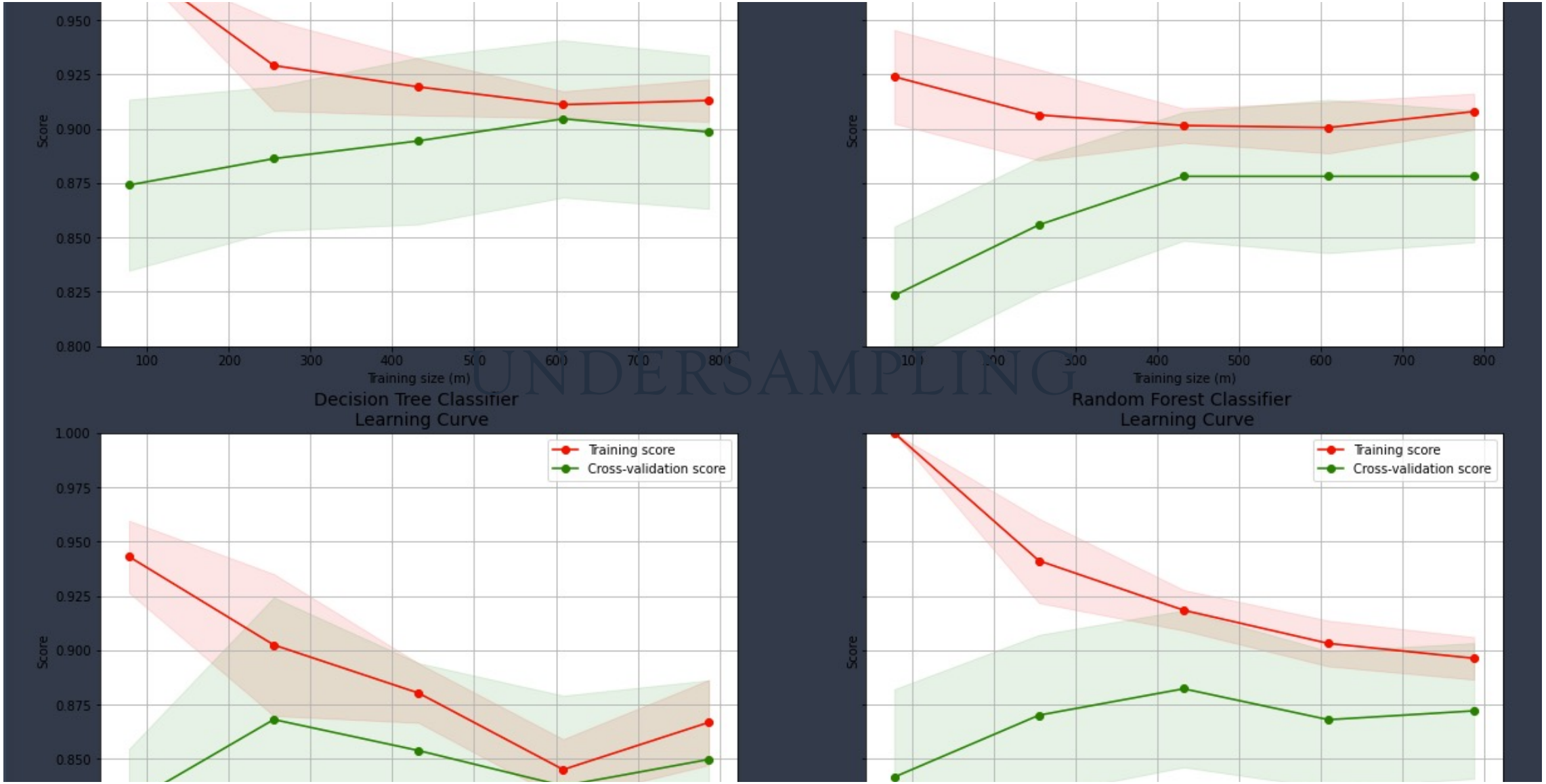
The ROC Curve show that the Support Vector Classifier has the largest area under the curve.

Undersampling Learning Curve

# UNDERSAMPLING

| | Classifier | CV Score | Accuracy | Recall Score | Precision Score |
|---|---|---|---|---|---|
| 0 | Logistic Regression | 0.902526 | 0.656288 | 0.918699 | 0.004598 |
| 1 | Random Forest | 0.945784 | 0.224741 | 0.959350 | 0.002133 |
| 2 | Support Vector | 0.631448 | 0.992093 | 0.642276 | 0.132107 |
| 3 | Decision Tree | 0.937706 | 0.895045 | 0.878049 | 0.014274 |
| 4 | k-Nearest Neighbour | 0.880758 | 0.839738 | 0.910569 | 0.009729 |



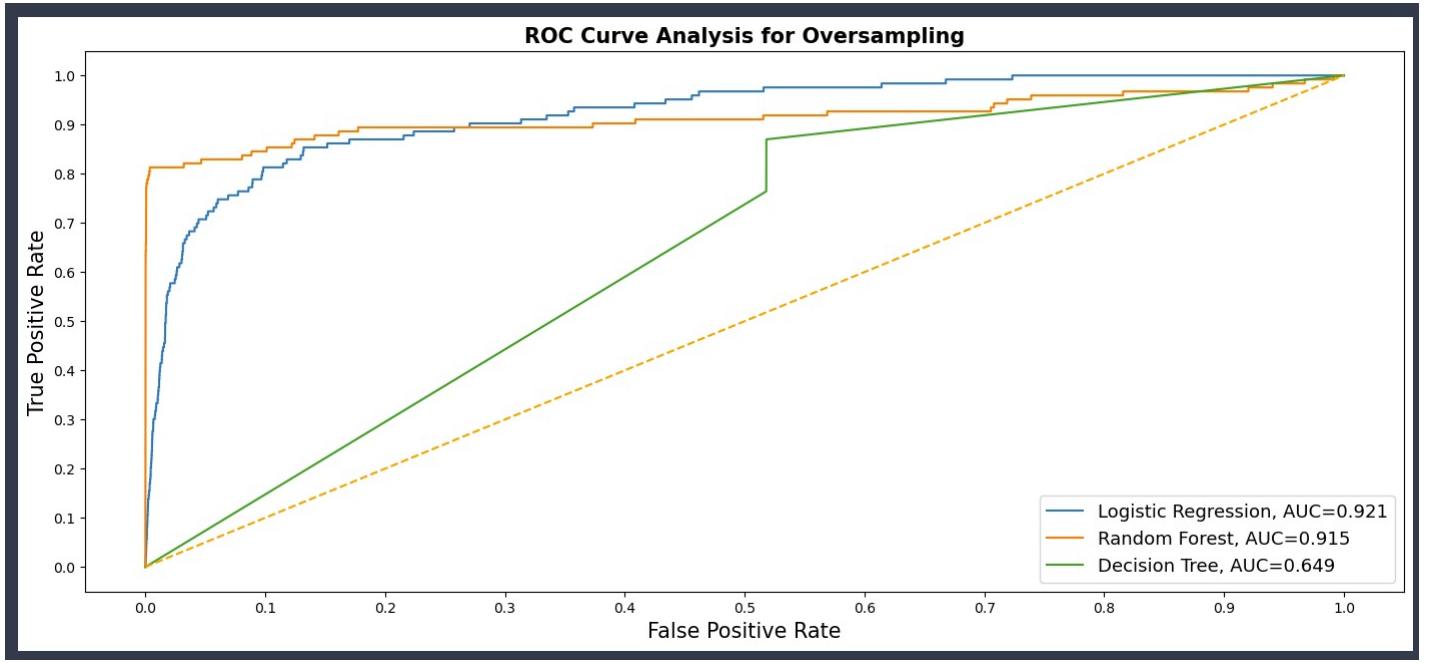

UNDERSAMPLING
Decision Tree Classifier
Learning Curve
Random Forest Classifier
Learning Curve
Training score
Cross-validation score
Score
Training size (m)
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
0.800
100
200
300
400
500
600
700
800

# OVERSAMPLING MODEL RESULTS

- By Oversampling the dataset, we ahieved recall scores greater than 85% for all classifiers. The Random Forest classifier had the best accuracy of 99%.
- The ROC Curve show that the logistic regression had the largest AUC.
- Oversampling Learning Curve

# COMPARING MODEL PERFORMANCE

| | Classifier | Naive - Accuracy | Naive - Recall | Accuracy - Random UnderSampling | Accuracy - Oversampling (SMOTE) | Recall - Random UnderSampling | Recall - Oversampling (SMOTE) |
|---|---|---|---|---|---|---|---|
| 0 | Logistic Regression | 0.999143 | 0.617886 | 0.656288 | 0.976560 | 0.918699 | 0.886179 |
| 1 | Random Forest | 0.999551 | 0.772358 | 0.224741 | 0.995773 | 0.959350 | 0.869919 |
| 2 | Decision Tree | Not Applicable | Not Applicable | 0.482585 | 0.482585 | 0.869919 | 0.869919 |



# SUMMARY

# THE DATASET

The dataset used for this project has 284807 rows of credit card transactions. Exploratory data analysis reveal as expected that we have a highly imbalanced dataset with only 0.17% of all transaction being fraud.

While a large portion of the features have been anonymized with PCA, univariate and bivariate distribution plots show that the genuine transaction class has an approximately normal distribution across all features, and the fraud class was had a left skewed distribution for many of the features.

# NAÏVE MODELS

While naive logistic regression and random forest had an accuracy of 100% and a precisions of 84% and 96% respectively, both classifiers only managed recall scores of 62% and 77% respectively.

This means that, the classifiers would miss fraud transaction almost 25% of the time. This ype of metric would cost an organization alot of money.

# PERFORMANCE METRIC

Since classifying transactions as fraud or genuine is an anomaly detection problem where only a small fraction are the anomalies, measuring model performance with the accuracy metric will not be ideal.

To capture fraud transactions we would require a classifier that has a high recall metric which is the ratio of of True Positives to the total of True Positives and False Positives

# OVERSAMPLING, UNDERSAMPLING, ROC, AND LEARNING CURVE

To improve the recall score of the naive models, we employ oversampling and underampling and with these methods, we achieved recall scores greater than 90% for the undersampling method and recall scores greater than 85% for the oversampling method.

While recall for random forest was highest at 95.9%, the classifier had a lower AUC value (91.5) than the logistic regression classifier with AUC of 92.1.

Analysis of the learning curve show that the logistic regression had a good fit. Increasing our cross-validation folds may make the logistic regression have near perfect without overfitting.

# RECOMMENDATIONS

- One challenge with this project was computation resources required to run the RandomizedGridSearchCV and the model Cross-Validation scores.
  - One way to to mitigate this challenge in the future may be to use the HalvingGridSearchCV which may in some cases may be 30% faster than the RandomizedGridSearchCV.
  - We may also explore using an online environment that has unlimited computation resources that can handle the resource requirements for memory and CPU intensive models and processes. Since feature extraction had been done on, the dataset, visualizing potentially interesting relationships was not possible with this dataset