Credit Card Fraud Detection

Credit card fraud is increasing over the years. In 2020, there were 393,207 reported cases of credit card fraud, which is an increase of 44.7% from 2019. People can imagine the amount of money that people have lost because of these frauds. Machine learning algorithms are used to detect fraud, which we will be using for this project. It is important that credit card companies are able to recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase. In this project we want to answer the question of "Does there exist an opportunity for credit card companies to detect fraud within transactions by using machine learning algorithms."

1. Data

The description of the dataset is explained well on the page where the data was collected from.

"The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, ... V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise."

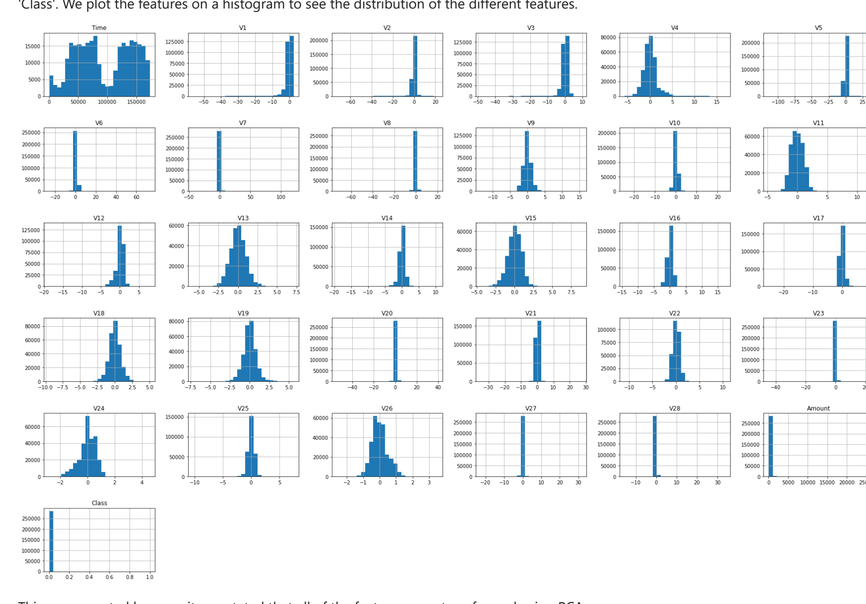
- Credit Card Fraud Dection

2. Data Wrangling

The data set was very clean with zero null values with no duplicates.

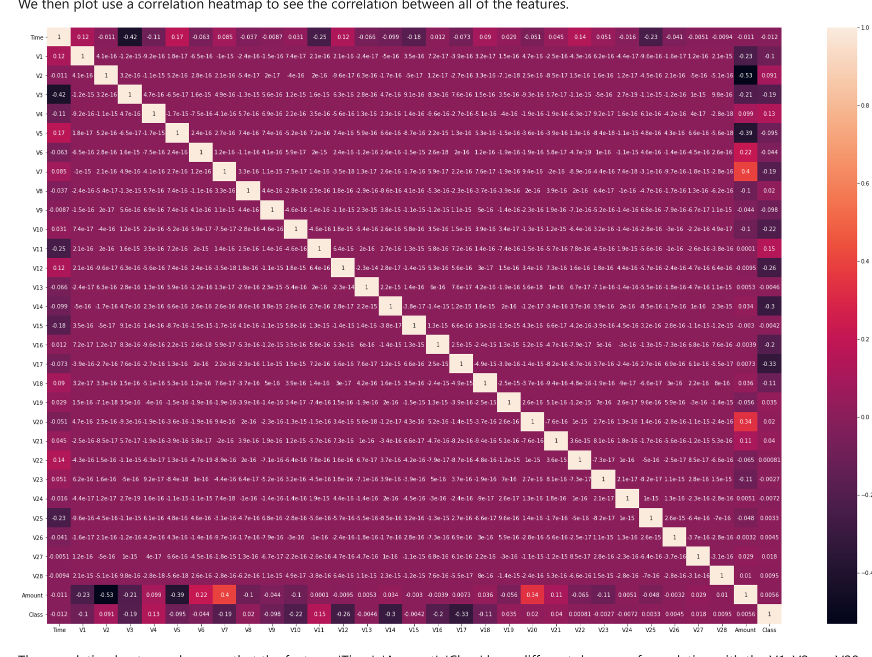
3. EDA

As mentioned above V1, V2, ... , V28 are principal components obtained with PCA. There are other variables such as 'Time', 'Amount', 'Class'. We plot the features on a histogram to see the distribution of the different features.



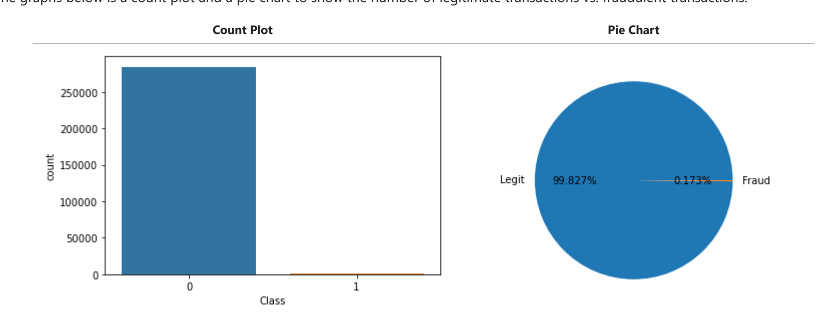
This was expected because it was stated that all of the features were transformed using PCA.

We then plot use a correlation heatmap to see the correlation between all of the features.



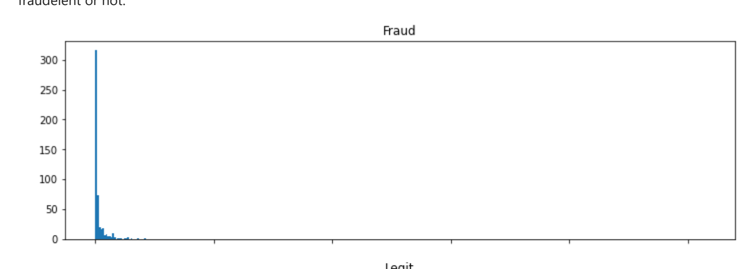
The correlation heatmap shows us that the features 'Time', 'Amount', 'Class' have different degrees of correlation with the V1, V2, ... , V28 features.

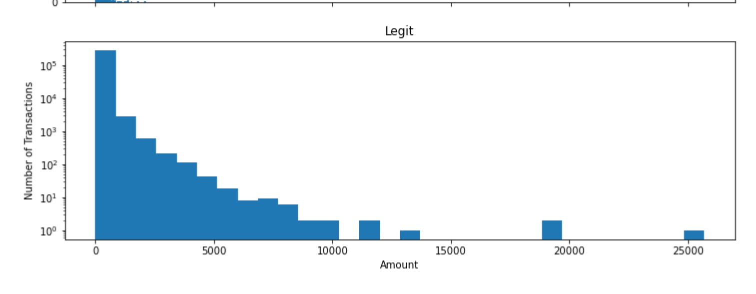
The graphs below is a count plot and a pie chart to show the number of legitimate transactions vs. fraudulent transactions.



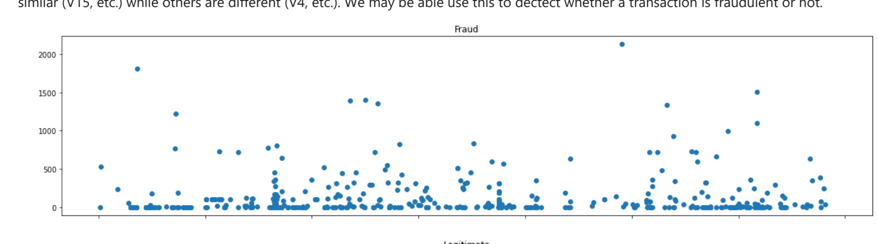
We see here that the data is very unbalanced. Only 0.173% of the transactions are frauds. This will cause a problem with our ML model because ML model do not do very well with unbalanced datasets, but we can fix this problem by undersampling our data, which we will do later.

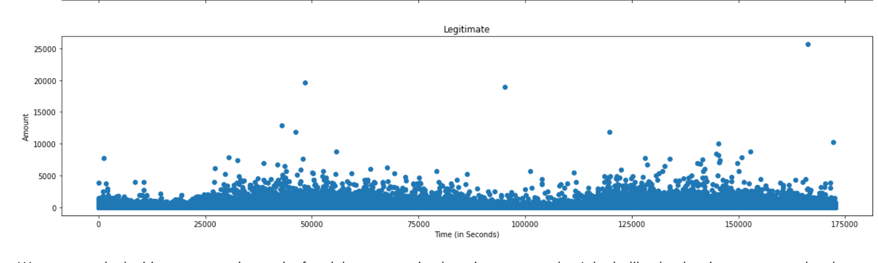
We compare the legitimate amount to the fraud amount. We can see here the there is alot less fraud amount than legit amount in both transactions and amount. This is completely reasonable, but these two variable cannot be the sole bases of whether a transactions is fraudelent or not.





Comparing the kde of legitimate cases and fraudulent cases. We can see here that some of the legitimate and fraudulent distribution are similar (V15, etc.) while others are different (V4, etc.). We may be able use this to dectect whether a transaction is fraudulent or not.

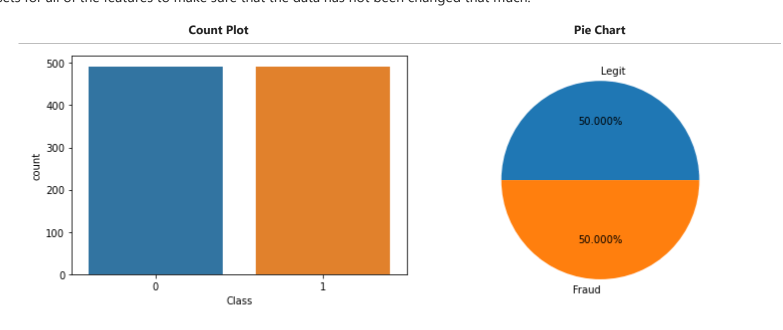




We compare the legitimate transaction vs the fraudulent transaction by using a scatterplot. It looks like the data is very scattered and we may not be able to use it for our judgement.

4. Preprocessing

We are going to undersample because the data is imbalanced. We do this by taking a random sample (n=492) out of the legitimate transaction and then concatenating it with the fraudulent transaction, so that our data has 50% legitmiate transactions and 50% fraudulent transactions. This is so that our ML model will be able to predict the fradulent transactions better. We also check the mean of both of the datasets for all of the features to make sure that the data has not been changed that much.



We split our data 80% training data and 20% testing data. The 80/20 train/test split was chosen because it is often the standard in the industry.

It is important to scale the data when working with the ML models. If the data is not scaled, it can cause a problem with the model and makes it so that the model will not be able to learn properly. For this project, we used the StandardScaler form sklearn.preprocessing to scaled the data.

5. Modeling

We are going to test 4 different models, Logistic Regression, Decision Tree Classifier, Support Vector Classifier, K Nearest Neighbors Classifier. We chose these models because we want to predict whether a certain transaction is fraudulent or not. We also use GridSearchCV to pick out the best parameters for each of the model.

Logistic Regression

Explanation of Logistic Regression can be found in the link below.

- Logistic Regression

```
Tuned Logistic Regression Parameters: {'C': 100, 'penalty': 'l2'}
Best score is 0.9453438684189308
Accuracy is 0.9543147208121827


              precision    recall  f1-score   support

           0       0.97      0.94      0.96       104
           1       0.94      0.97      0.95        93

    accuracy                           0.95       197
   macro avg       0.95      0.96      0.95       197
weighted avg       0.95      0.95      0.95       197
```

Decision Tree Classifier

Explanation of Decision Tree Classifier can be found in the link below.

- Decision Tree Classifier

```
Tuned DecisionTree Parameters: {'criterion': 'entropy', 'max_depth': 3, 'min_samples_leaf': 6}
Best score is 0.9161251310166895
Accuracy is 0.9086294416243654


              precision    recall  f1-score   support

           0       0.90      0.92      0.91        99
           1       0.92      0.90      0.91        98

    accuracy                           0.91       197
   macro avg       0.91      0.91      0.91       197
weighted avg       0.91      0.91      0.91       197
```

Support Vector Classifier

Explanation of Support Vector Classifier can be found in the link below.

- Support Vector Classifier

```
Tuned SVC Parameter: {'C': 0.7, 'kernel': 'linear'}
Best score is 0.9428122228493108
Accuracy is 0.9441624365482234


              precision    recall  f1-score   support

           0       0.95      0.94      0.95       102
           1       0.94      0.95      0.94        95

    accuracy                           0.94       197
   macro avg       0.94      0.94      0.94       197
weighted avg       0.94      0.94      0.94       197
```

K Nearest Neighbors Classifier

Explanation of K Nearest Neighbors can be found in the link below.

- K Nearest Neighbors Classifier

```
Tuned kNN Parameter: {'algorithm': 'auto', 'n_neighbors': 3}
Best score is 0.9072321212609852
Accuracy is 0.934010152284264


              precision    recall  f1-score   support

           0       0.97      0.91      0.94       108
           1       0.90      0.97      0.93        89

    accuracy                           0.93       197
   macro avg       0.93      0.94      0.93       197
weighted avg       0.94      0.93      0.93       197
```

6. Conclusion

The best model here is the Logistic Regression Model. The Logistic Regression Model has the highest best score as well as the highest accuracy out of all of the model. Support Vector Classifier comes in at a close second. I reran the code many times to see if the results changed because of the random sampling. There was difference in results, but Logistic Regression and SVC always had the best results out of all the models.

7. Final Thoughts

We preformed undersampling on the data because the data was imbalanced. Undersampling is a technique that people use to balance uneven datasets by keeping all of the data in the minority class and decreasing the size of the majority class. We then starndarized that data used split the data into training set and testing set. We modeled various classifiers and tuned it using GridSearch and found that Logistic Regression with the best parameters is the best model because because it has best scores when you compare the precision, recall, accuracy.

This project purpose was to teach myself how to use classification models and I would say that it was a success. I gain more confidence in my data science skills.