Credit Fraud Dealing with Imbalanced Datasets

By

1934042
1934015

Problem statement

we will use various predictive models to see how accurate they are in detecting whether a transaction is a normal payment or a fraud.
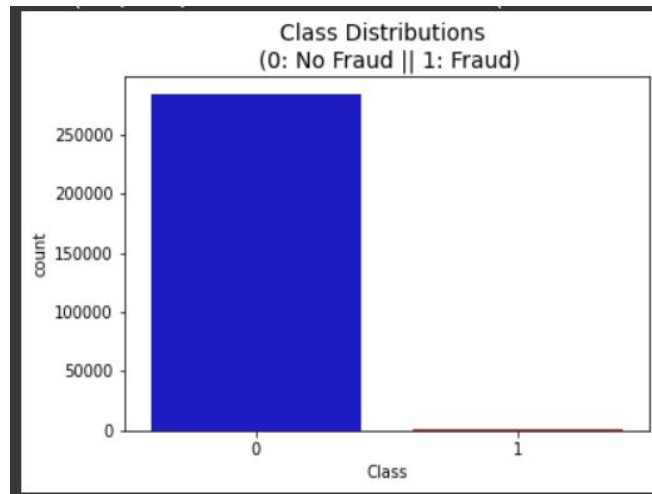
 The features are scaled and the names of the features are not shown due to privacy reasons.

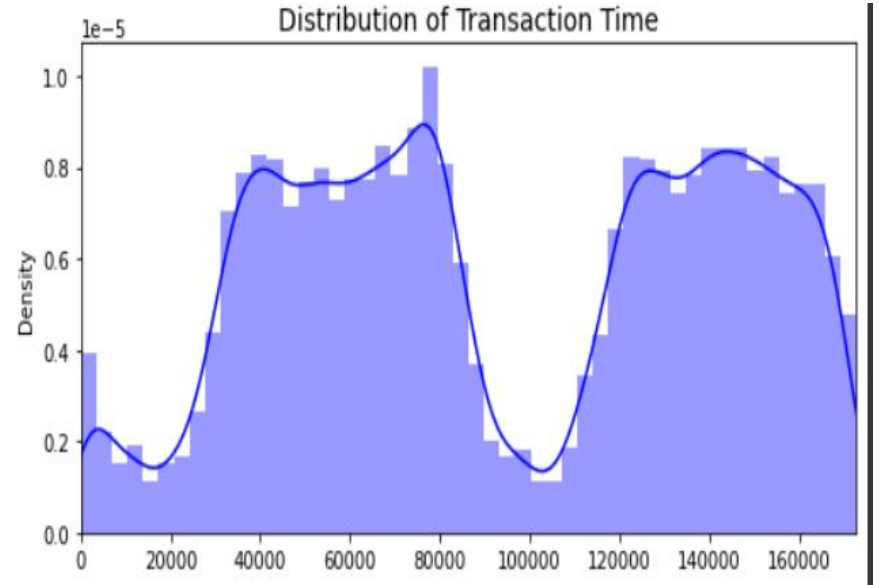The dataset is highly imbalanced hence it has to be treated with some methods say under sampling .
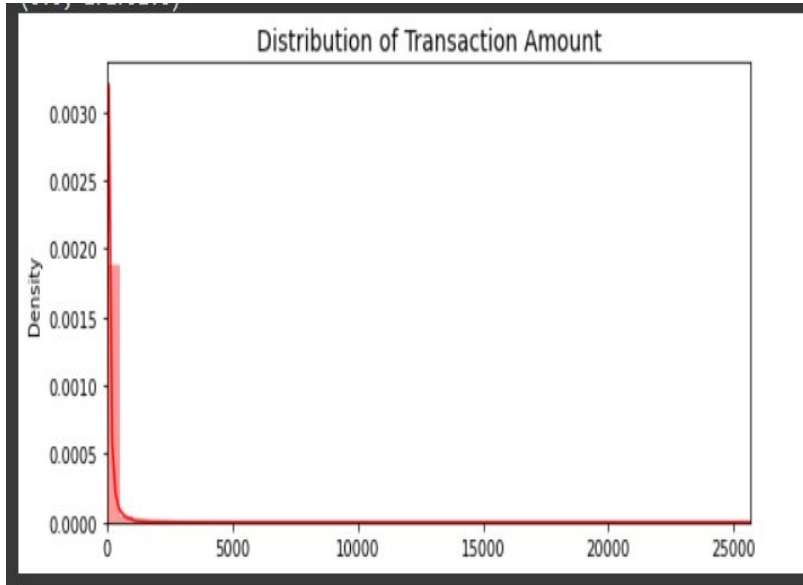
EDA

No Frauds 99.83 % of the dataset
Frauds 0.17 % of the dataset
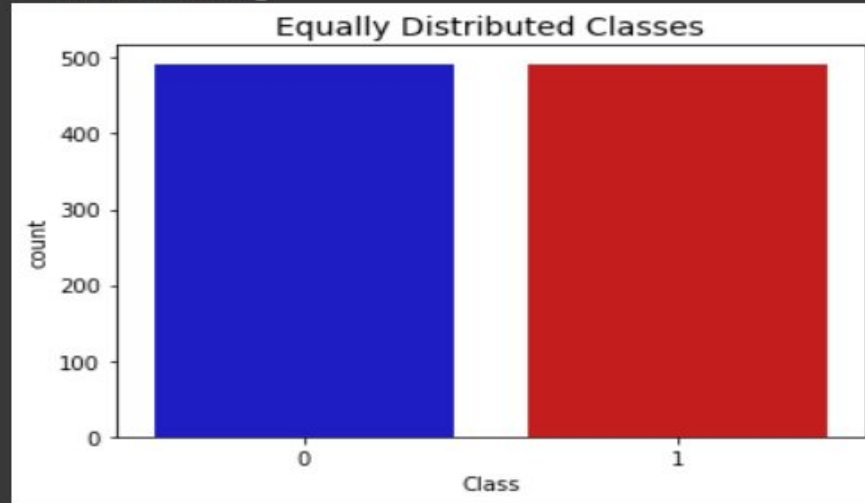
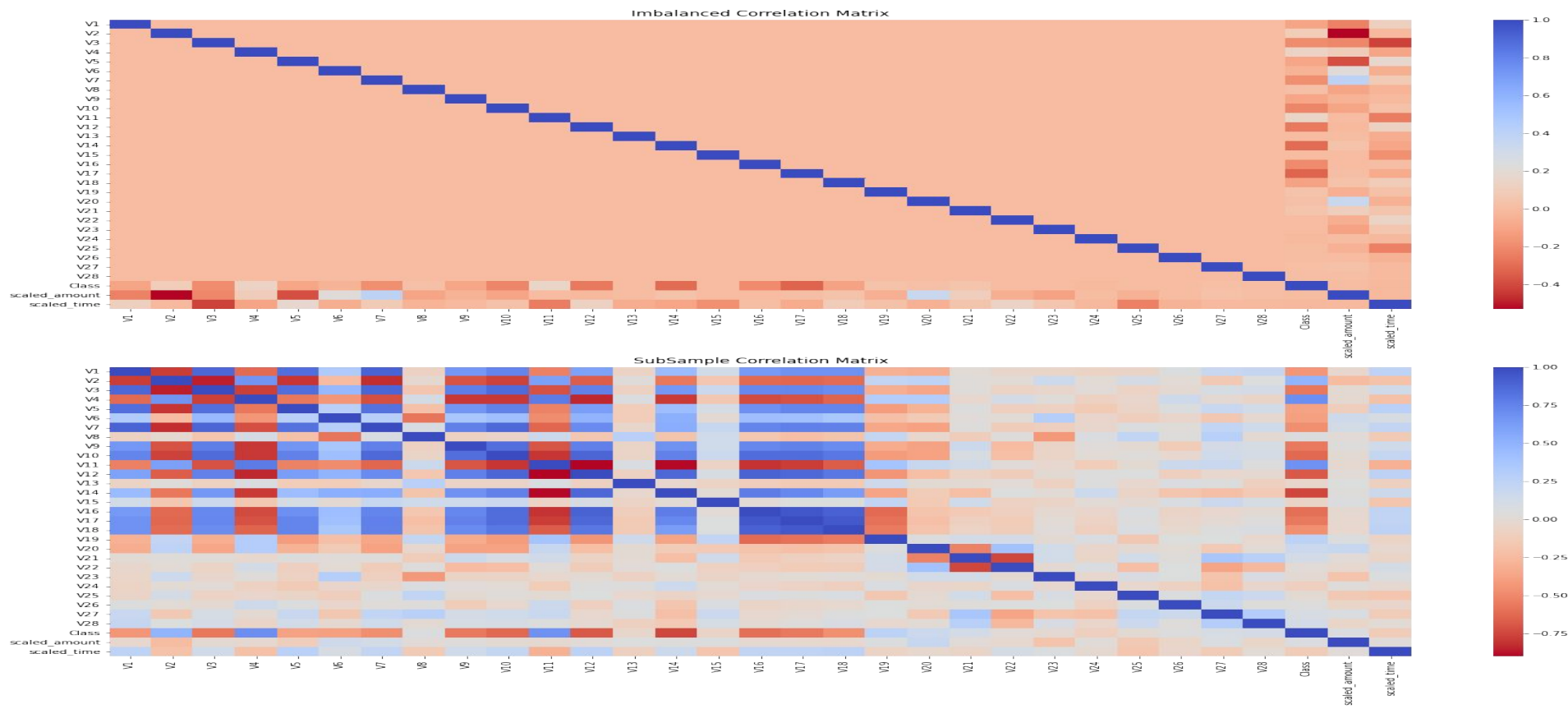As you can see the data is highly imbalanced

Distribution of transactions

Handling imbalance

Random under sampling

Correlation difference in plot

Classifiers

```
classifiers = {
    "LogisiticRegression": LogisticRegression(),
    "KNearest": KNeighborsClassifier(),
    "Support Vector Classifier": SVC(),
    "DecisionTreeClassifier": DecisionTreeClassifier()
}
```

```
Classifiers:    LogisticRegression Has a training score of 93.0 % accuracy score
Classifiers:    KNeighborsClassifier Has a training score of 93.0 % accuracy score
Classifiers:    SVC Has a training score of 93.0 % accuracy score
Classifiers:    DecisionTreeClassifier Has a training score of 90.0 % accuracy score
```

Hyper parameter tuning

```python
log_reg_params = {"penalty": ['l1', 'l2'], 'C': [0.001, 0.01, 0.1, 1, 10, 100, 1000]}

rand_log_reg = RandomizedSearchCV(LogisticRegression(), log_reg_params, n_iter=4)
```

Here we use our top model logistic regression with random search cv hyperparameter tuning technique

SMOTE

```python
pipeline = imbalanced_make_pipeline(SMOTE(sampling_strategy='minority'), rand_log_reg)
model = pipeline.fit(original_Xtrain[train], original_ytrain[train])
```

```
accuracy: 0.9490889526479107
precision: 0.060047142390174876
recall: 0.9137293086660175
f1: 0.11104066800763937
--------------------------------
```

Thank you... :)