APPLIED DATASCIENCE – PHASE 4

CREDIT CARD FRAUD DETECTION

PROBLEM STATEMENT:

The problem is to develop a machine learning-based system for real-time credit card fraud detection. The goal is to create a solution that can accurately identify fraudulent transactions while minimizing false positives. This project involves

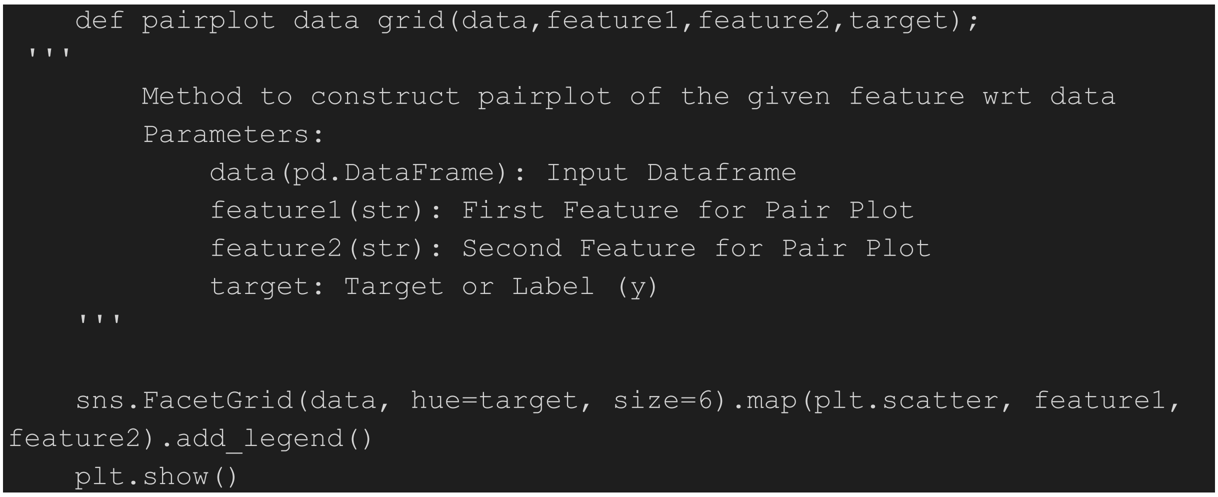
- data preprocessing
- feature engineering
- model selection
- training
- evaluation to create a robust fraud detection system

FEATURE ENGINEERING:

This often involves a combination of

- domain expertise
- data exploration
- experimentation

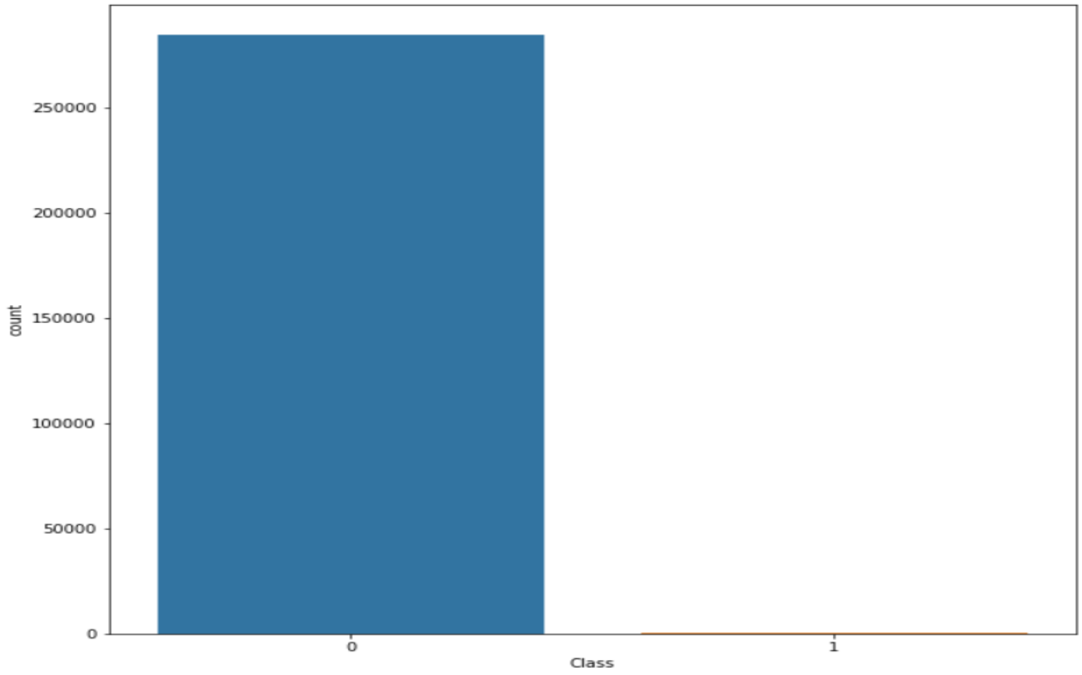
to identify the most relevant and informative feature for input. Once these features have been identified, they may need to be transformed .
The process of feature engineering is crucial for achieving good performance in fraud detection and other machine learning tasks.



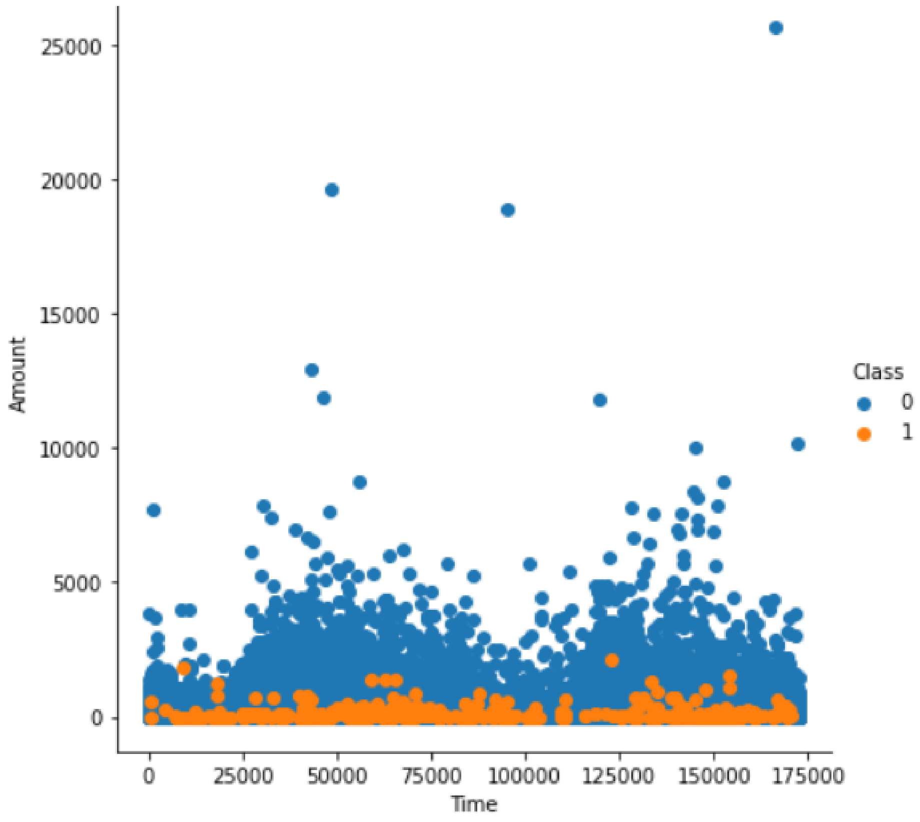
```
    def pairplot data grid(data,feature1,feature2,target);
 '''
        Method to construct pairplot of the given feature wrt data
        Parameters:
            data(pd.DataFrame): Input Dataframe
            feature1(str): First Feature for Pair Plot
            feature2(str): Second Feature for Pair Plot
            target: Target or Label (y)
    '''

    sns.FacetGrid(data, hue=target, size=6).map(plt.scatter, feature1,
feature2).add_legend()
    plt.show()
```

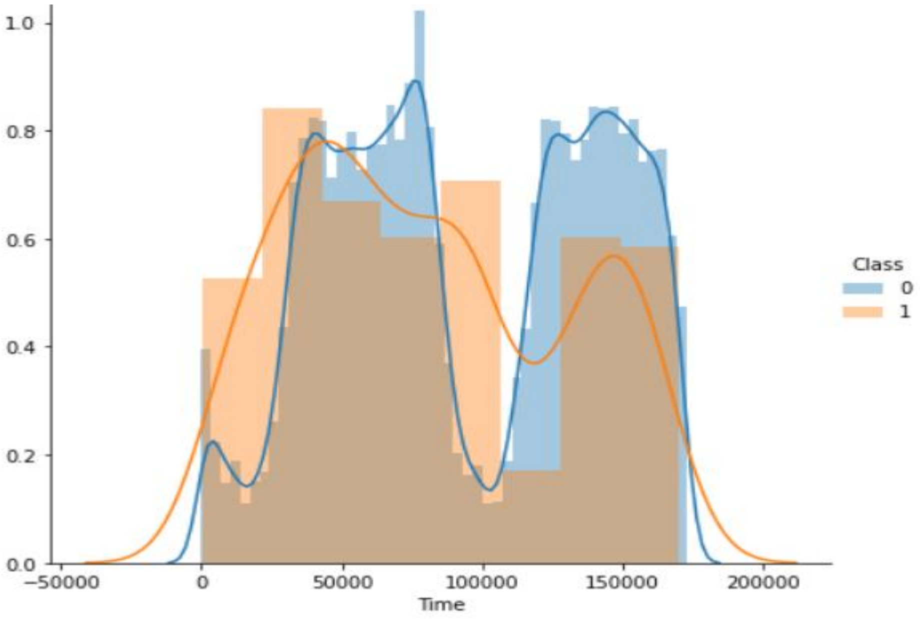
```
countplot_data(df, df.Class)
```



```
pairplot_data_grid(df, "Time", "Amount", "Class")
```



```
sns.FacetGrid(df_refine, hue="Class",
size=6).map(sns.distplot,"Time").add_legend()
plt.show()
```



MODELLING:

- Study the Feature Correlations of the given data
- Plot a Heatmap
- Run GridSearch on the data
- The plan is to train the models on the training data set which we have analyzed above and then use the testing dataset to evaluate the model performance.

- As data models need numeric input, we need to convert some of our categorical observations into numeric ones

```
plt.figure(figsize=(20,20))
df_corr = df.corr()
  sns.heatmap(df_corr)
```

```
 Create Train and Test Data in ratio 70:30
X = df.drop(labels='Class', axis=1) # Features
y = df.loc[:,'Class']               # Target Variable


X_train, X_test, y_train, y_test = train_test_split(X, y, 
test_size=0.3, random_state=1, stratify=y)
```

EVALUATION:

Evaluations of the models by presenting their efficiency, the accuracies of the models will be presented in addition to any comment observed, to find the best and most suited model for detecting the fraud transactions made by credit card.

```
def grid_eval(grid_clf):
    """
        Method to Compute the best score and parameters computed by 
grid search
        Parameter:
            grid_clf: The Grid Search Classifier
    """
    print("Best Score", grid_clf.best_score_)
    print("Best Parameter", grid_clf.best_params_)

def evaluation(y_test, grid_clf, X_test):
    """
        Method to compute the following:
            1. Classification Report
            2. F1-score
            3. AUC-ROC score
            4. Accuracy
        Parameters:
            y_test: The target variable test set
            grid_clf: Grid classifier selected
            X_test: Input Feature Test Set
    """
    y_pred = grid_clf.predict(X_test)
    print('CLASSIFICATION REPORT')
    print(classification_report(y_test, y_pred))

    print('AUC-ROC')
    print(roc_auc_score(y_test, y_pred))

    print('F1-Score')
    print(f1_score(y_test, y_pred))

    print('Accuracy')
    print(accuracy_score(y_test, y_pred))
```