CREDIT CARD FRAUD DETECTION

The "**Credit Card Fraud Detection**" project aims to identify fraudulent transactions based on various features available in the dataset. The ultimate goal is to build a model that can accurately distinguish between legitimate and fraudulent transactions, thereby helping financial institutions in preventing and minimizing fraudulent activities.

DATASET:

The dataset provided contain information related to credit card transactions and associated attributes.

The dataset contains the following columns:

1. Merchant_id: Unique identifier for the merchant involved in the transaction.
2. Transaction date: Date of the transaction (however, in this dataset, this column seems to have missing values for all entries).
3. Average Amount/transaction/day: The average amount spent in transactions per day.
4. Transaction_amount: The amount of the specific transaction.
5. Is declined: Indicates whether the transaction was declined (Categorical - "Y" or "N").
6. Total Number of declines/day: Total number of transaction declines in a day.
7. isForeignTransaction: Indicates if the transaction is foreign (Categorical - "Y" or "N").
8. isHighRiskCountry: Indicates if the transaction involves a high-risk country (Categorical - "Y" or "N").
9. Daily_chargeback_avg_amt: The average chargeback amount per day.
10. 6_month_avg_chbk_amt: The average chargeback amount over six months.
11. 6-month_chbk_freq: Frequency of chargebacks over six months.
12. isFradulent: The target variable indicating whether the transaction is fraudulent or not (Categorical - "Y" or "N").

STEPS TAKEN:

1. Importing Libraries
2. Data Loading and Initial Examination
3. Data Cleaning
4. Exploratory Data Analysis(EDA)
5. Data Preprocessing
6. Train/Test Split

7. Model Building and Evaluation
8. Hyperparameter Tuning
9. Final Model Assessment

DATA LOADING AND INITIAL EXAMINATION:

Load the dataset using **pd.read_csv('creditcardcsvpresent.csv')**.

Display the initial rows using **data.head()**.

Check for missing values using **data.isnull().sum()** and handle missing data by dropping the "Transaction date" column.

Check for duplicates using **data.duplicated().sum()**.

DATA CLEANING:

1. Handling Missing Values

The initial examination of the dataset reveals missing values in the "Transaction date" column.

This was identified using **'data.isnull().sum()'**.

As all entries in the "Transaction date" column were missing, this column was dropped entirely using **'data.drop("Transaction date", axis=1, inplace=True)'**.

2. Handling Duplicates

'data.duplicated().sum()' was used to check for duplicate values in the dataset but none was found.

EXPLORATORY DATA ANALYSIS(EDA):

Qualitative variables were visualized using pie charts and distributions for various columns.

Using histograms, box plots, pair plots, and other visualizations we try to understand the distribution and relationships between variables, grouped by fraudulent status.

Some insights gained from EDA:

From visualization of qualitative variables using pie plot:

-Distribution of is declined

>Declined transactions are relatively rare: Only 1.9% of transactions in the dataset were declined.

>Approved transactions dominate: The vast majority (98.1%) of transactions were approved.

-Distribution of isForeignTransaction

>77.0% of the transactions were domestic (not foreign).

>23.0% of the transactions were foreign.

-Distribution of isHighRiskCountry

>Most transactions originate from low-risk countries: 93.3% of transactions in the dataset are associated with countries not considered high-risk.

>A smaller portion involve high-risk countries: 6.7% of transactions are linked to countries deemed high-risk.

-Distribution of isFradulent

>Fraudulent transactions are a minority: Only 14.6% of the transactions in the dataset are labeled as fraudulent.

>Most transactions are legitimate: The vast majority (85.4%) of transactions are considered non-fraudulent.

From Distribution of Transaction_amount for Fraudulent vs. Non-fraudulent Transactions using histogram:

>The distribution for non-fraudulent transactions (blue) appears to be relatively normal or bell-shaped, with a peak around 20,000 and gradually decreasing tails on either side.

>The distribution for fraudulent transactions (orange) is more skewed to the right, with a longer tail extending towards higher transaction amounts. This suggests that fraudulent transactions tend to involve larger amounts of money compared to non-fraudulent ones.

>While there is no clear separation between the two distributions, the tail of the orange distribution extends beyond the blue distribution, particularly around 50,000 and above. This suggests that transactions exceeding this amount might be more likely to be fraudulent.

From Box Plots for Average Amount/transaction/day Across Different Levels of isFradulent:

>Higher average transaction amounts per day could be a potential indicator of fraudulent activity

DATA PREPROCESSING:

Qualitative variables ("N"/"Y" columns) are encoded into numerical representations (0 and 1).

Categorical variables in the dataset are handled by mapping their values to numeric representations.

TRAIN/TEST SPLIT:

Using **train_test_split** from **sklearn.model_selection** training and testing sets were created.

MODEL BUILDING AND EVALUATION:

Pipelines are initialized for Support Vector Classifier (SVC), K-Nearest Neighbors (KNN), and Decision Tree models using **make_pipeline** from **sklearn.pipeline**.

An evaluation function is created to fit the models, predict on the test set, display confusion matrix, classification report, and plot learning curves.

Models are then build(StandardScalar() is used to standardize features) and evaluated using the evaluation function.

SVM:

True Positives (TP) : 78

True Negatives (TN) : 527

False Positives (FP): 5

False Negatives (FN): 5

Precision: For class 1 (fraudulent transactions), it's 0.94, indicating 94% of transactions predicted as fraudulent were actually fraudulent.

Recall : For class 1, it's 0.94, indicating that 94% of actual fraudulent transactions were correctly predicted.

F1-score : For class 1, it's 0.94, indicating high balance between precision and recall.

Accuracy : Overall, the model's accuracy is 98%. It correctly predicted 98% of the total instances in the dataset.

KNN:

True Positives (TP) : 76

True Negatives (TN) : 526

False Positives (FP): 6

False Negatives (FN): 7

Precision: For class 1 (fraudulent transactions), it's 0.93, indicating 93% of transactions predicted as fraudulent were actually fraudulent.

Recall : For class 1, it's 0.92, indicating that 92% of actual fraudulent transactions were correctly predicted.

F1-score : For class 1, it's 0.92, indicating high balance between precision and recall.

Accuracy :Overall, the model's accuracy is 98%. It correctly predicted 98% of the total instances in the dataset.

DECESION TREE:

True Positives (TP): 79

True Negatives (TN): 524

False Positives (FP): 8

False Negatives (FN): 4

Precision: For class 1 (fraudulent transactions), it's 0.91, indicating 91% of transactions predicted as fraudulent were actually fraudulent.

Recall (Sensitivity): For class 1, it's 0.95, indicating that 95% of actual fraudulent transactions were correctly predicted.

F1-score : For class 1, it's 0.93, indicating high balance between precision and recall.

Accuracy: Overall, the model's accuracy is 98%. It correctly predicted 98% of the total instances in the dataset.

HYPERPARAMETER TUNING:

For the SVC model, hyperparameter tuning is performed using **GridSearchCV** from **sklearn.model_selection** to optimize the model for better performance.

FINAL MODEL ASSESMENT:

The tuned SVC model is evaluated on the test set using the best estimator obtained from grid search.

The accuracy score on the test set is calculated using the score method of the best estimator.

True Positives (TP): 79

True Negatives (TN): 529

False Positives (FP): 3

False Negatives (FN): 4

Precision: For class 1 (fraudulent transactions), it's 0.96, indicating 96% of transactions predicted as fraudulent were actually fraudulent.

Recall (Sensitivity): For class 1, it's 0.95, indicating that 95% of actual fraudulent transactions were correctly predicted.

F1-score : For class 1, it's 0.96, indicating high balance between precision and recall.

Accuracy: Overall, the model's accuracy is 99%. It correctly predicted 99% of the total instances in the dataset.

CONCLUSION:

The project successfully demonstrates the feasibility of using machine learning models to detect credit card fraud with high accuracy. The project's models, including Support Vector Classifier (SVC), K-Nearest Neighbors (KNN), and Decision Trees, exhibited strong performance in identifying fraudulent transactions within the credit card dataset. Particularly noteworthy was the SVC model, achieving an exceptional accuracy of approximately 99% after rigorous hyperparameter tuning. These high accuracy rates showcase the models' robustness in distinguishing between legitimate and fraudulent transactions, highlighting their potential efficacy in bolstering fraud detection systems for financial security.