

# Credit Card Fraud Detection Using logistic regression

Purpose : The primary purpose of this project is to develop and evaluate a machine learning model capable of accurately identifying fraudulent credit card transactions.

The steps involved are as follows:

- 1) **Importing Libraries:** Imported essential libraries for data manipulation, visualization, and model building: numpy, pandas, matplotlib.pyplot, seaborn, and various modules from sklearn.
- 2) **Loading the Dataset:** Loaded the credit card transactions dataset using `pd.read_csv('creditcard.csv')`.
- 3) **Exploratory Data Analysis (EDA):**
  - Used `data.info()` to understand the structure of the dataset.
  - Counted the number of fraudulent and non-fraudulent transactions using `data['Class'].value_counts()`.
  - Created separate DataFrames for fraudulent (`fraud`) and non-fraudulent (`Notfraud`) transactions.
  - Describe the transaction amounts for both fraudulent and non-fraudulent transactions using `fraud['Amount'].describe()` and `Notfraud['Amount'].describe()`.
- 4) **Balancing the Dataset:** Sampled an equal number of non-fraudulent transactions (492) and combined them with all the fraudulent transactions to create a balanced dataset (`new_data`).
- 5) **Correlation Analysis:**
  - Calculated the correlation matrix of the balanced dataset using `new_data.corr(method='pearson')`.
  - Visualized the correlation matrix using a heatmap with `seaborn`.
- 6) **Feature Selection:** Identified features with an absolute correlation of at least 0.1 with the `Class` variable and removed the `Class` column itself from this list, resulting in a list of high-correlation features (`h_feature`).
- 7) **Data Preparation:**
  - Created feature (`x`) and target (`y`) variables from the balanced dataset.
  - Split the data into training and testing sets using `train_test_split()`, ensuring the splits had the same class distribution as the original data (`stratify=y`).

- Standardized the features using `StandardScaler()` to ensure they have a mean of 0 and a standard deviation of 1.

#### **8) Model Training:**

- Initialized a logistic regression model with a maximum of 10,000 iterations.
- Trained the logistic regression model on the training data using `logistic.fit(x_train, y_train)`.

#### **9) Model Evaluation:**

- Made predictions on the test set using `logistic.predict(x_test)`.
- Calculated the accuracy of the model's predictions with `accuracy_score(y_test, y_pred)`.
- Displayed the accuracy score using a formatted print statement.

## **Conclusion**

By following these steps, we successfully built and evaluated a logistic regression model to detect fraudulent credit card transactions. The project included data loading, exploratory data analysis, dataset balancing, feature selection, data preparation, model training, and evaluation, providing a comprehensive workflow for a binary classification problem. The final accuracy score provided an initial measure of the model's performance.