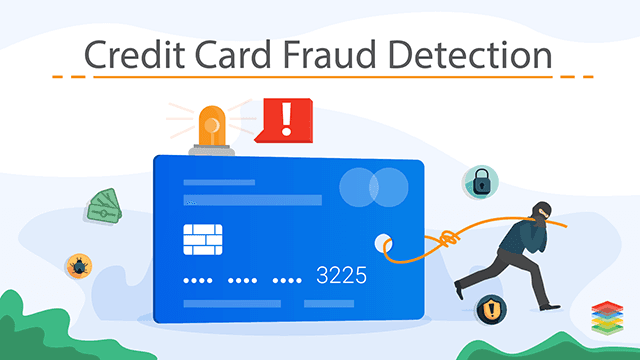
CREDIT CARD FRAUD DETECTION

PHASE-5

DOCUMENTATION



**DESIGN THINKING:**

* Data Source
* Data Preprocessing
* Feature Engineering
* Model Selection
* Model Training

**Data Source:**

Data collection is the process of acquiring, collecting, extracting, and storing the voluminous amount of data which may be in the structured or unstructured form like text, video, audio, XML files, records, or other image files used in later stages of data analysis.  
In the process of big data analysis, “Data collection” is the initial step before starting to analyze the patterns or useful information in data. The data which is to be analyzed must be collected from different valid sources.

1.In the data source we have to utilize some dataset on it.Here we will get the dataset from the webites they have the complete transaction data.

2.Including all the  transaction amount, timestamp, merchant information, and card details.

3.Download the .CSV datasets and use in the program.

**Data Preprocessing:**

* Gathering the data
* Import the dataset & Libraries
* Dealing with Missing Values
* Divide the dataset into Dependent & Independent variable
* dealing with Categorical values
* Split the dataset into training and test set
* Feature Scaling.

**DATA CLEAN:**

[Data cleaning](https://monkeylearn.com/data-cleaning/) is the process of editing, correcting, and structuring data within a data set so that it’s generally uniform and prepared for analysis. This includes removing corrupt or irrelevant data and formatting it into a language that computers can understand for optimal analysis.

**Data Cleaning Steps & Techniques:**

* **Step 1:**[**Remove irrelevant data**](https://monkeylearn.com/blog/data-cleaning-steps/)
* **Step 2:**[**Deduplicate your data**](https://monkeylearn.com/blog/data-cleaning-steps/)
* **Step 3:**[**Fix structural errors**](https://monkeylearn.com/blog/data-cleaning-steps/)
* **Step 4:**[**Deal with missing data**](https://monkeylearn.com/blog/data-cleaning-steps/)
* **Step 5:**[**Filter out data outliers**](https://monkeylearn.com/blog/data-cleaning-steps/)
* **Step 6:**[**Validate your data**](https://monkeylearn.com/blog/data-cleaning-steps/)

**This article covers 7 ways to handle missing values in the dataset:**

* Deleting Rows with missing values
* Impute missing values for continuous variable
* Impute missing values for categorical variable
* Other Imputation Methods
* Using Algorithms that support missing values
* Prediction of missing values
* Imputation using Deep Learning Library — Datawig.

**Feature Engineering:**

Feature engineering is the process of **transforming raw data into features that are suitable for machine learning models**. In other words, it is the process of selecting, extracting, and transforming the most relevant features from the available data to build more accurate and efficient machine learning models.

**Why do we Engineer Features?**

* Improve User Experience
* Competitive Advantage
* Meet Customer Needs
* Increase Revenue
* Future-Proofing.

**Model Selection:**

* The complexity of the issue
* Data Availability & Quality
* Interpretability
* Model Assumptions
* Scalability and Efficiency
* Regularisation and Generalisation
* Domain Expertise
* Resource Constraints
* Ensemble Methods
* Evaluation and Experimentation.
* **Model Training:**
* Model training is the phase in the data science development lifecycle where practitioners try to fit the best combination of weights and bias to a machine learning algorithm to minimize a loss function over the prediction range.
* We train our model over a subset of the original dataset, i.e., the training dataset, and then evaluate whether it can generalize well to the new or unseen dataset or test set.

**How to Train ML Models:**

* Exploring the data.
* Cleaning the data.
* Engineering new features.

**Evaluation:**

* Accuracy measures how often the classifier makes the correct predictions, as it is the ratio between the number of correct predictions and the total number of predictions.
* Precision measures the proportion of predicted Positives that are truly Positive. Precision is a good choice of evaluation metrics when you want to be very sure of your prediction. For example, if you are building a system to predict whether to decrease the credit limit on a particular account, you want to be very sure about the prediction or it may result in customer dissatisfaction.

**Example of Confusion Matrix on Iris Flower Dataset:**

* AUC (Area Under the ROC Curve) is a performance measurement for classification problems at various thresholds settings. It tells how much a model is capable of distinguishing between classes. The higher the AUC, better the model is at predicting when a 0 is actually a 0 and a 1 is actually a 1.

**Feature Importance:**

**The Credit Card Fraud Detection Problem includes modeling past credit card transactions with the knowledge of the ones that turned out to be fraud. This model is then used to identify whether a new transaction is fraudulent or not.**

**Data Integration:**

**This involves combining data from multiple sources to create a unified dataset. Data integration can be challenging as it requires handling data with different formats, structures, and semantics. Techniques such as record linkage and data fusion can be used for data integration.**

**Data Transformation:**

**This involves converting the data into a suitable format for analysis. Common techniques used in data transformation include normalization, standardization, and discretization. Normalization is used to scale the data to a common range, while standardization is used to transform the data to have zero mean and unit variance. Discretization is used to convert continuous data into discrete categories.**

**Data Reduction:**

**This involves reducing the size of the dataset while preserving the important information. Data reduction can be achieved through techniques such as feature selection and feature extraction. Feature selection involves selecting a subset of relevant features from the dataset, while feature extraction involves transforming the data into a lower-dimensional space while preserving the important information.**

**PROGRAM:**

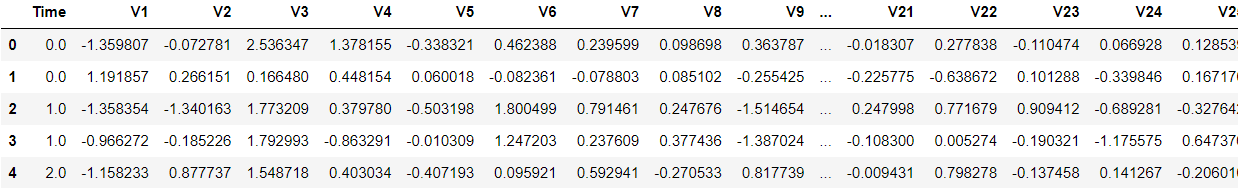
**import pandas as pd**

**from collections import Counter**

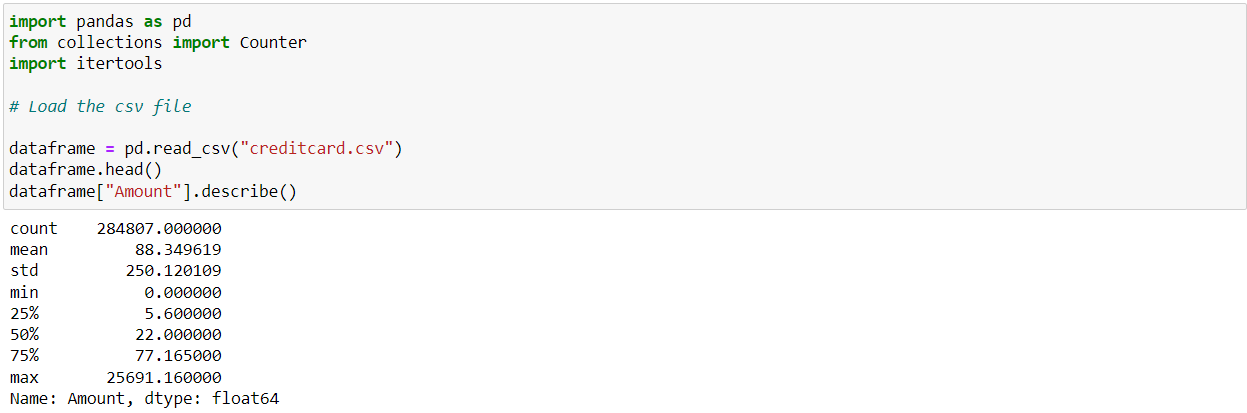
**import itertools**

**dataframe = pd.read\_csv("creditcard.csv")**

**dataframe.head()**

****

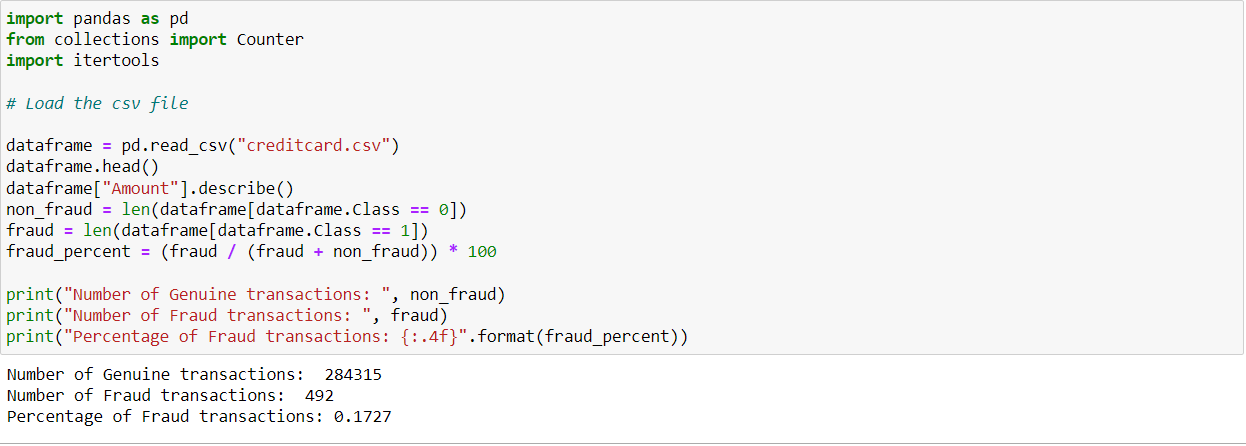
**dataframe["Amount"].describe()**



**non\_fraud = len(dataframe[dataframe.Class == 0])**

**fraud = len(dataframe[dataframe.Class == 1])**

**fraud\_percent = (fraud / (fraud + non\_fraud)) \* 100**

****

**import matplotlib.pyplot as plt**

**labels = ["Genuine", "Fraud"]**

**count\_classes = dataframe.value\_counts(dataframe['Class'], sort= True)**

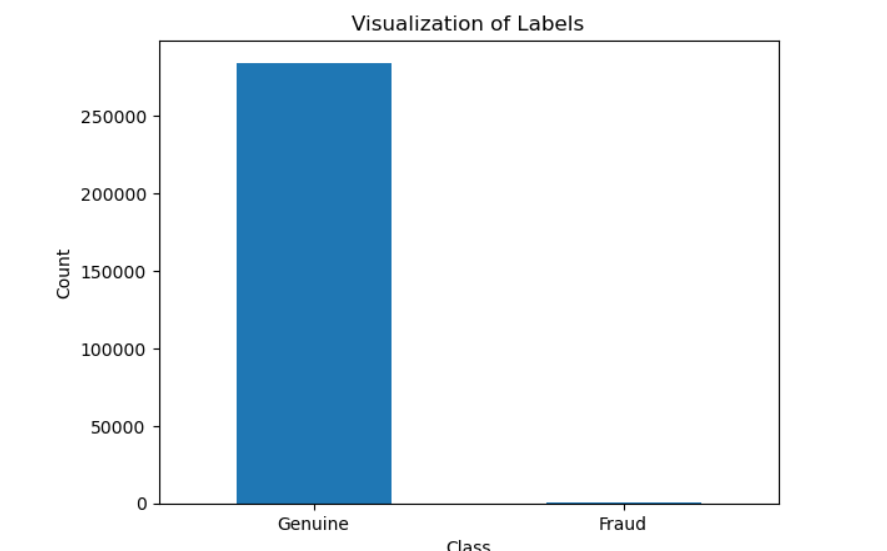
**count\_classes.plot(kind = "bar", rot = 0)**

**plt.title("Visualization of Labels")**

**plt.ylabel("Count")**

**plt.xticks(range(2), labels)**

**plt.show()**

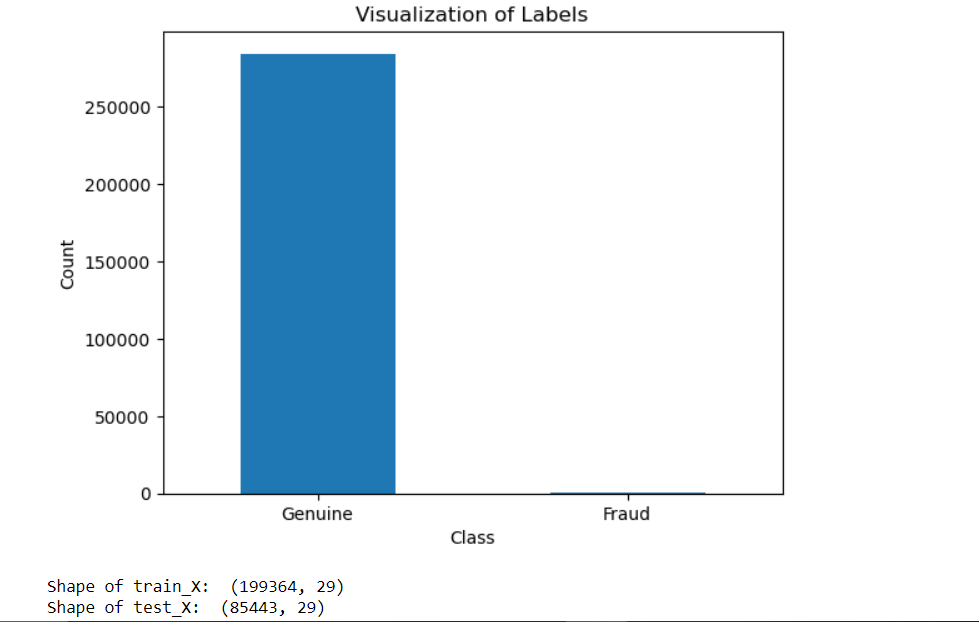
****

**from sklearn.model\_selection import train\_test\_split**

**(train\_X, test\_X, train\_Y, test\_Y) = train\_test\_split(X, Y, test\_size= 0.3, random\_state= 42)**

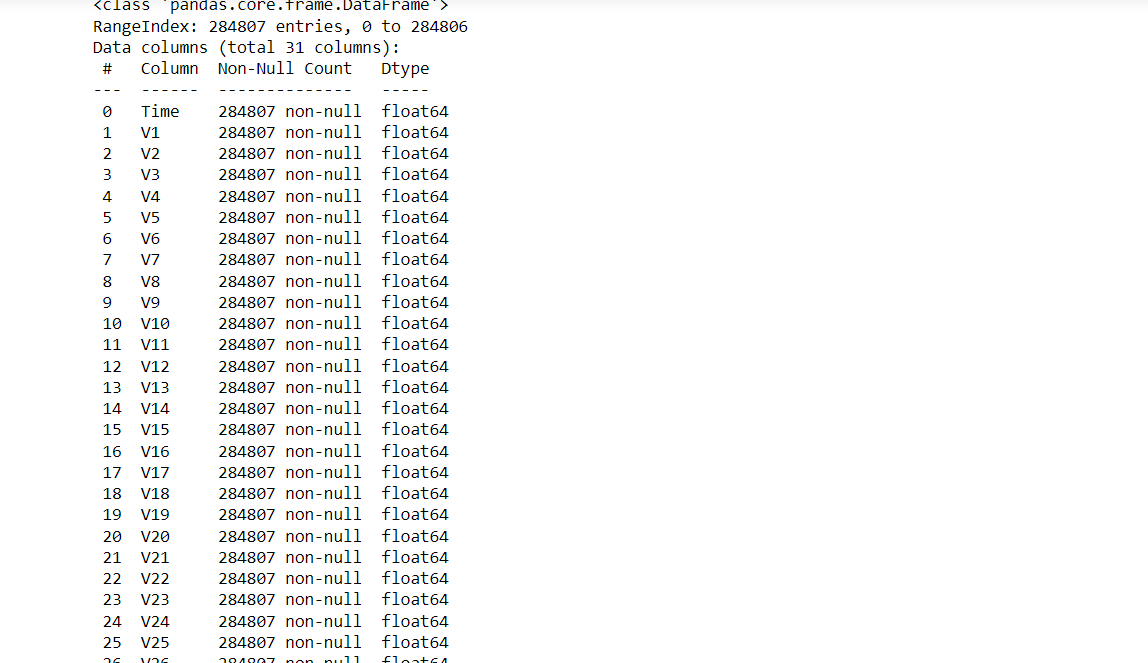
**print("Shape of train\_X: ", train\_X.shape)**

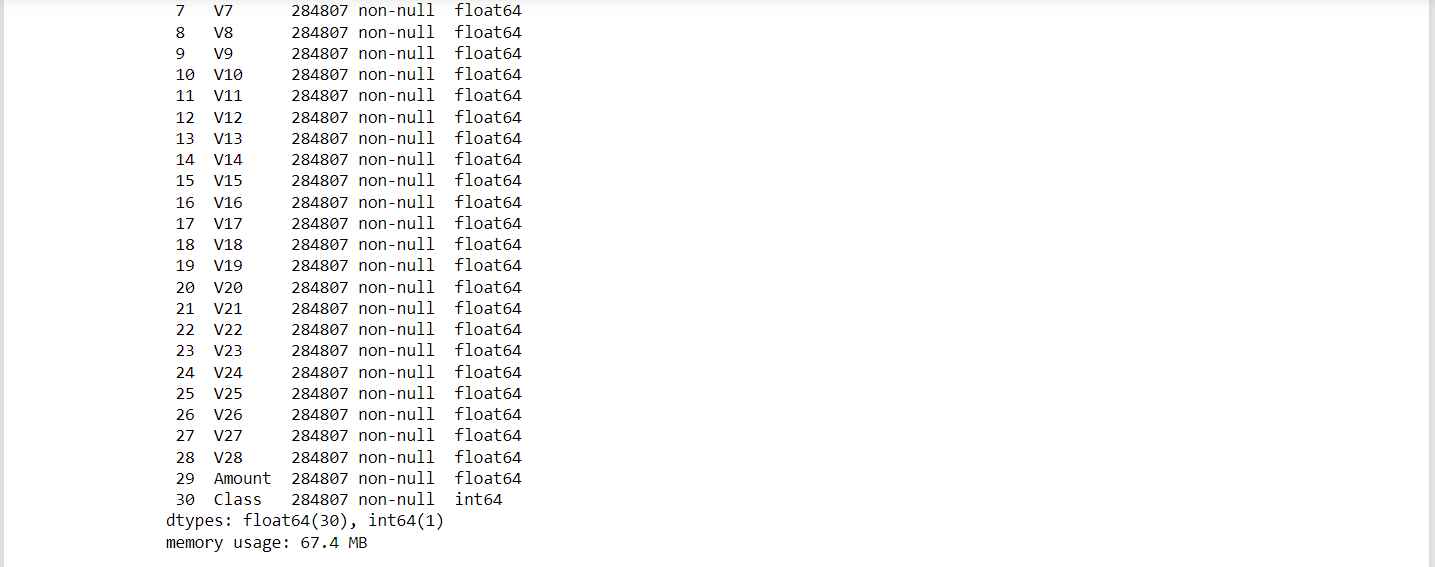
**print("Shape of test\_X: ", test\_X.shape)**



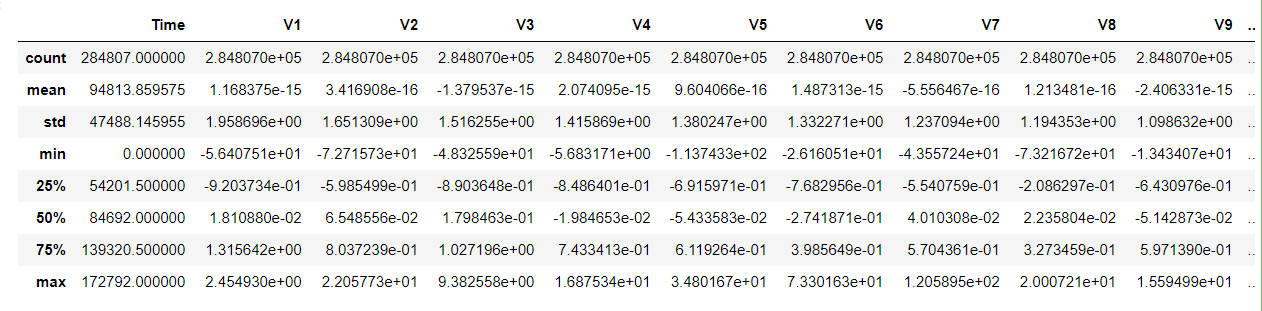
**dataframes.head()**

**dataframes.info()**

****

****

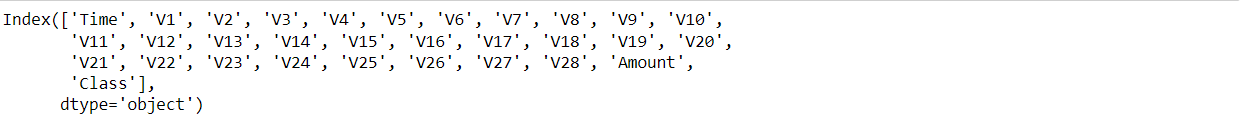
**dataframe.describe()**

****

**dataframe.isnull().sum()**

****

**dataframe.columns**

****

**dataframe.shape**

(284807, 31)

**dataframe.duplicated().value\_counts()**

False 283726

True 1081

Name: count, dtype: int64

**dataframe.Class.unique()**

array([0, 1], dtype=int64)

**print(legitimate)**

**print(fraud)**

284315

492

# **Handling Imbalance of the data using SMOTE:**

x=dataframe.drop("Class",axis=1)

y=dataframe["Class"]

linkcode

smote = SMOTE(random\_state=42)

x\_smote, y\_smote = smote.fit\_resample(x, y)

## Data Visualization:

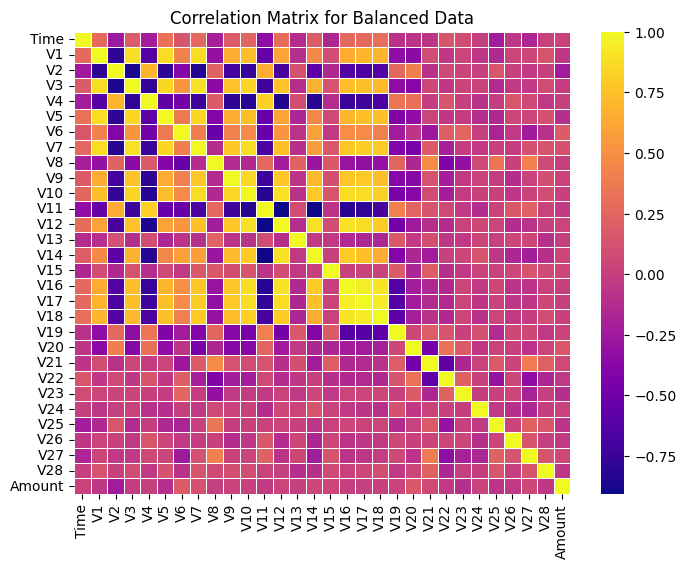
corr\_balanced = x\_smote.corr()

plt.figure(figsize=(8, 6))

sns.heatmap(corr\_balanced, annot=False, cmap="plasma", linewidths=0.5)

plt.title("Correlation Matrix for Balanced Data")

plt.show()

****