DATA SET

# CREDIT CARD FRAUD DETECTION

DEVELOPING A CREDIT CARD FRAUD DETECTION SYSTEM INVOLVES IMPLEMENTING SOPHISTICATED ALGORITHMS AND STRATEGIES TO IDENTIFY AND PREVENT FRAUDULENT TRANSACTIONS

DATA SET

# INTRODUCTION

Loading And Preprocessing Data Are Critical Steps In Credit Card Fraud Detection. The Quality And Suitability Of Your Data Greatly Impact The Performance Of The Fraud Detection Model. Here Are Steps For Loading And Preprocessing Data

Developing A Credit Card Fraud Detection System Involves Implementing Sophisticated Algorithms And Strategies To Identify And Prevent Fraudulent Transactions

DATA SET

## LOADING DATA:

1. **Data Sources:**
   - Identify and collect data from various sources. In credit card fraud detection, this typically includes transaction records.
2. **Data Format:**
   - Ensure that the data is available in a format compatible with your chosen programming language and libraries (e.g., CSV, JSON, SQL databases).
3. **Loading Libraries:**
   - Use relevant libraries in your programming language (e.g., Pandas for Python) to load and manipulate the data efficiently.
4. **Data Exploration:**
   - Explore the dataset to gain insights into the distribution of features, identify missing values, and understand the overall structure.

DATA SET

## PROGRAM:

```
import pandas as pd

# Load data
df = pd.read_csv('credit_card_data.csv')

# Explore the data
print(df.head())
print(df.info())
print(df.describe())
```

DATA SET

## Preprocessing Data:

5. **Handling Missing Values:**
    - Identify and handle missing values. Options include imputation or removal of rows/columns with missing values.

```
# Handling missing values
df = df.dropna()  # or df.fillna(value)
```

**6.Handling Imbalanced Data:**

- In fraud detection, the data is often imbalanced, with normal transactions outnumbering fraudulent ones. Consider strategies like oversampling, undersampling, or using synthetic data.

```
# Handling imbalanced data
from imblearn.over_sampling import SMOTE

smote = SMOTE(sampling_strategy='auto')
X_resampled, y_resampled = smote.fit_resample(X, y)
```

DATA SET

1. **Feature Engineering:**
   - Create new features that might improve the model's ability to distinguish between normal and fraudulent transactions.

```
# Feature engineering
df['hour'] = df['timestamp'].dt.hour
```

1. **Scaling/Normalization:**
   - Scale numerical features to bring them to a standard range, ensuring that no feature dominates others.

1. **Handling Categorical Variables:**
   - Encode categorical variables using techniques like one-hot encoding.

```
# One-hot encoding
df = pd.get_dummies(df, columns=['category'])
```

**Data Splitting:**

- Split the data into training, validation, and test sets.

DATA SET

1. **Data Splitting:**
   - Split the data into training, validation, and test sets

```
from sklearn.model_selection import train_test_split

X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.3, random_state=42)

X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5, random_state=42)
```

2. **Data Balancing Techniques:**
   - Use techniques such as undersampling or oversampling to handle imbalanced datasets.

```
from imblearn.under_sampling import RandomUnderSampler

rus = RandomUnderSampler(sampling_strategy='auto', random_state=42)

X_resampled, y_resampled = rus.fit_resample(X_train, y_train)
```

DATA SET

1. **Handling Time-Based Data:**
   - For credit card transactions, time is often a crucial factor. Consider how time-related features can be utilized, and if necessary, preprocess the data accordingly.
2. **Data Privacy:**
   - Anonymize or mask sensitive information to comply with data privacy regulations.
3. **Data Serialization:**
   - Serialize the preprocessed data for easy storage and retrieval.

```
import joblib

# Save preprocessed data
joblib.dump(df, 'preprocessed_data.joblib')

# Load preprocessed data
df = joblib.load('preprocessed_data.joblib')
```

# DATA SET

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | -1.35981 | -0.07278 | 2.536347 | 1.378155 | -0.33832 | 0.462388 | 0.239599 | 0.098698 | 0.363787 | 0.090794 | -0.5516 | -0.6178 | -0.99139 | -0.31117 | 1.468177 | -0.4704 | 0.207971 | 0.025791 | 0.4039 |
| 0 | 1.191857 | 0.266151 | 0.16648 | 0.448154 | 0.060018 | -0.08236 | -0.0788 | 0.085102 | -0.25543 | -0.16697 | 1.612727 | 1.065235 | 0.489095 | -0.14377 | 0.635558 | 0.463917 | -0.1148 | -0.18336 | -0.145 |
| 1 | -1.35835 | -1.34016 | 1.773209 | 0.37978 | -0.5032 | 1.800499 | 0.791461 | 0.247676 | -1.51465 | 0.207643 | 0.624501 | 0.066084 | 0.717293 | -0.16595 | 2.345865 | -2.89008 | 1.109969 | -0.12136 | -2.261 |
| 1 | -0.96627 | -0.18523 | 1.792993 | -0.86329 | -0.01031 | 1.247203 | 0.237609 | 0.377436 | -1.38702 | -0.05495 | -0.22649 | 0.178228 | 0.507757 | -0.28792 | -0.63142 | -1.05965 | -0.68409 | 1.965775 | -1.232 |
| 2 | -1.15823 | 0.877737 | 1.548718 | 0.403034 | -0.40719 | 0.095921 | 0.592941 | -0.27053 | 0.817739 | 0.753074 | -0.82284 | 0.538196 | 1.345852 | -1.11967 | 0.175121 | -0.45145 | -0.23703 | -0.03819 | 0.8034 |
| 2 | -0.42597 | 0.960523 | 1.141109 | -0.16825 | 0.420987 | -0.02973 | 0.476201 | 0.260314 | -0.56867 | -0.37141 | 1.341262 | 0.359894 | -0.35809 | -0.13713 | 0.517617 | 0.401726 | -0.05813 | 0.068653 | -0.033 |
| 4 | 1.229658 | 0.141004 | 0.045371 | 1.202613 | 0.191881 | 0.272708 | -0.00516 | 0.081213 | 0.46496 | -0.09925 | -1.41691 | -0.15383 | -0.75106 | 0.167372 | 0.050144 | -0.44359 | 0.002821 | -0.61199 | -0.045 |
| 7 | -0.64427 | 1.417964 | 1.07438 | -0.4922 | 0.948934 | 0.428118 | 1.120631 | -3.80786 | 0.615375 | 1.249376 | -0.61947 | 0.291474 | 1.757964 | -1.32387 | 0.686133 | -0.07613 | -1.22213 | -0.35822 | 0.3245 |
| 7 | -0.89429 | 0.286157 | -0.11319 | -0.27153 | 2.669599 | 3.721818 | 0.370145 | 0.851084 | -0.39205 | -0.41043 | -0.70512 | -0.11045 | -0.28625 | 0.074355 | -0.32878 | -0.21008 | -0.49977 | 0.118765 | 0.5703 |
| 9 | -0.33826 | 1.119593 | 1.044367 | -0.22219 | 0.499361 | -0.24676 | 0.651583 | 0.069539 | -0.73673 | -0.36685 | 1.017614 | 0.83639 | 1.006844 | -0.44352 | 0.150219 | 0.739453 | -0.54098 | 0.476677 | 0.4517 |
| 10 | 1.449044 | -1.17634 | 0.91386 | -1.37567 | -1.97138 | -0.62915 | -1.42324 | 0.048456 | -1.72041 | 1.626659 | 1.199644 | -0.67144 | -0.51395 | -0.09505 | 0.23093 | 0.031967 | 0.253415 | 0.854344 | -0.221 |
| 10 | 0.384978 | 0.616109 | -0.8743 | -0.09402 | 2.924584 | 3.317027 | 0.470455 | 0.538247 | -0.55889 | 0.309755 | -0.25912 | -0.32614 | -0.09005 | 0.362832 | 0.928904 | -0.12949 | -0.80998 | 0.359985 | 0.7076 |
| 10 | 1.249999 | -1.22164 | 0.38393 | -1.2349 | -1.48542 | -0.75323 | -0.6894 | -0.22749 | -2.09401 | 1.323729 | 0.227666 | -0.24268 | 1.205417 | -0.31763 | 0.725675 | -0.81561 | 0.873936 | -0.84779 | -0.683 |
| 11 | 1.069374 | 0.287722 | 0.828613 | 2.71252 | -0.1784 | 0.337544 | -0.09672 | 0.115982 | -0.22108 | 0.46023 | -0.77366 | 0.323387 | -0.01108 | -0.17849 | -0.65556 | -0.19993 | 0.124005 | -0.9805 | -0.982 |
| 12 | -2.79185 | -0.32777 | 1.64175 | 1.767473 | -0.13659 | 0.807596 | -0.42291 | -1.90711 | 0.755713 | 1.151087 | 0.844555 | 0.792944 | 0.370448 | -0.73498 | 0.406796 | -0.30306 | -0.15587 | 0.778265 | 2.2218 |
| 12 | -0.75242 | 0.345485 | 2.057323 | -1.46864 | -1.15839 | -0.07785 | -0.60858 | 0.003603 | -0.43617 | 0.747731 | -0.79398 | -0.77041 | 1.047627 | -1.0666 | 1.106953 | 1.660114 | -0.27927 | -0.41999 | 0.4325 |
| 12 | 1.103215 | -0.0403 | 1.267332 | 1.289091 | -0.736 | 0.288069 | -0.58606 | 0.18938 | 0.782333 | -0.26798 | -0.45031 | 0.936708 | 0.70838 | -0.46865 | 0.354574 | -0.24663 | -0.00921 | -0.59591 | -0.575 |
| 13 | -0.43691 | 0.918966 | 0.924591 | -0.72722 | 0.915679 | -0.12787 | 0.707642 | 0.087962 | -0.66527 | -0.73798 | 0.324098 | 0.277192 | 0.252624 | -0.2919 | -0.18452 | 1.143174 | -0.92871 | 0.68047 | 0.0254 |
| 14 | -5.40126 | -5.45015 | 1.186305 | 1.736239 | 3.049106 | -1.76341 | -1.55974 | 0.160842 | 1.23309 | 0.345173 | 0.91723 | 0.970117 | -0.26657 | -0.47913 | -0.52661 | 0.472004 | -0.72548 | 0.075081 | -0.406 |
| 15 | 1.492936 | -1.02935 | 0.454795 | -1.43803 | -1.55543 | -0.72096 | -1.08066 | -0.05313 | -1.97868 | 1.638076 | 1.077542 | -0.63205 | -0.41696 | 0.052011 | -0.04298 | -0.16643 | 0.304241 | 0.554432 | 0.054 |
| 16 | 0.694885 | -1.36182 | 1.029221 | 0.834159 | -1.19121 | 1.309109 | -0.87859 | 0.44529 | -0.4462 | 0.568521 | 1.019151 | 1.298329 | 0.42048 | -0.37265 | -0.80798 | -2.04456 | 0.515663 | 0.625847 | -1.300 |