DSC550_Milestone2_Moussadeq

May 10, 2024

Said Moussadeq

16-April-2024

DSC550, Milestone 1

1) Businesss problem, complications and objective.

Business Problem Overview:

In the digital age, credit cards have become the most common form of payment across various sectors, processing large volumes of transactions daily. This widespread use has, unfortunately, also made them a prime target for cybercriminals. The digital economy faces significant threats from credit card fraud, impacting consumers and businesses. Financial institutions and retailers are under continuous pressure to effectively enhance their cybersecurity measures to detect and prevent fraudulent activities.

Complications in Fraud Detection:

- **High Volume of Transactions:** Daily, vast amounts of data are processed, requiring the fraud detection model to swiftly respond to potential fraud in real-time.
- **Imbalanced Data:** A significant challenge in fraud detection is the imbalanced nature of transaction data, where typically, a vast majority (e.g., 99.8%) of transactions are legitimate. This imbalance makes detecting the few fraudulent transactions challenging without a high rate of false positives.
- **Data Privacy:** Transaction data is often private and sensitive, limiting the availability of such data for model training and testing.
- **Misclassification Issues:** Not every fraudulent transaction is caught and reported; some are misclassified as legitimate, which can lead to inaccuracies in model training and subsequent predictions.
- **Adaptive Threats:** Scammers continually evolve their techniques to circumvent detection measures, requiring adaptive models that dynamically learn from new fraud patterns.

The objective:

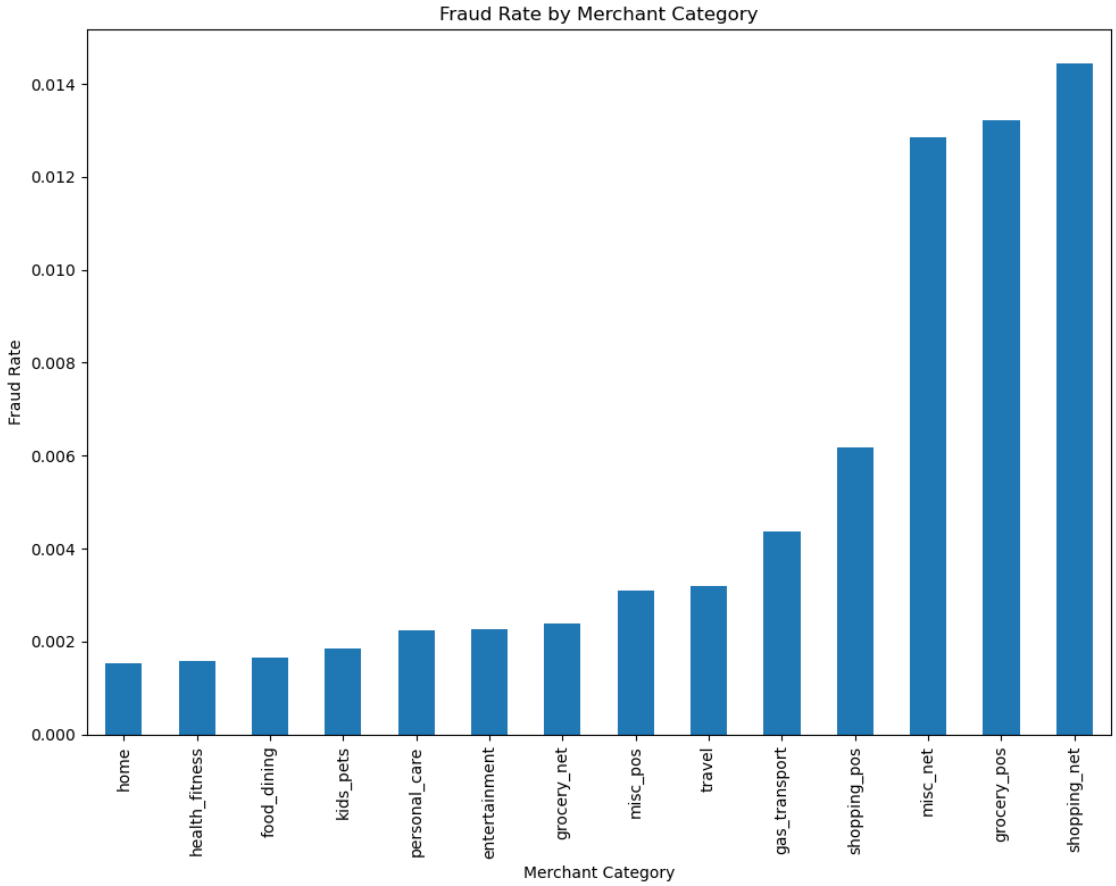
This project aims to develop a predictive model that can determine the likelihood of fraud in credit card transactions based on critical indicators such as merchant category, transaction amount, and demographic data of the cardholder. Financial institutions can effectively be more alert in their fraud detection strategies to specific transaction types by pinpointing that these indicators are more prone to fraud. Using the **'is_fraud'** variable, which flags transactions as fraudulent (1) or non-fraudulent (0), our model can flag and deny suspicious transactions.

2) Then, do a graphical analysis creating a minimum of four graphs.

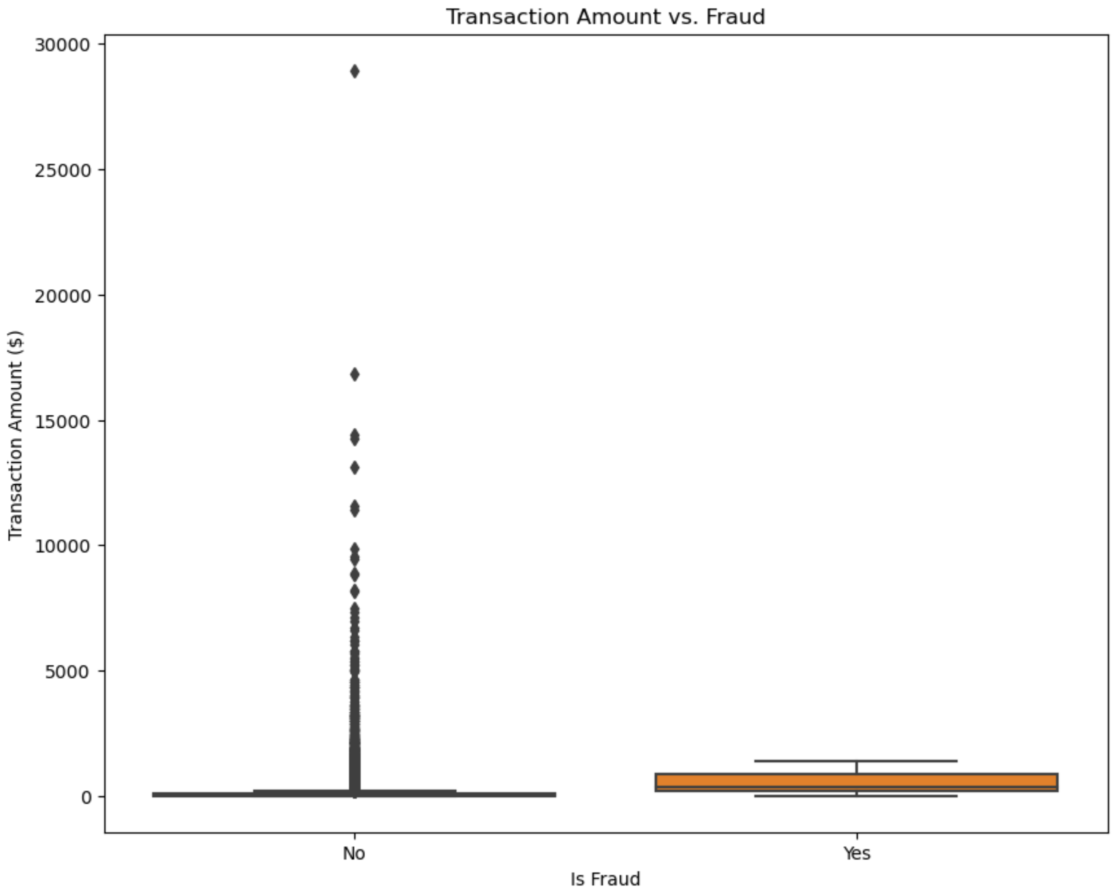
[5]:
```
!pip install geopy
```

```
Requirement already satisfied: geopy in
c:\users\thearchitect\anaconda3\lib\site-packages (2.4.1)
Requirement already satisfied: geographiclib<3,>=1.52 in
c:\users\thearchitect\anaconda3\lib\site-packages (from geopy) (2.0)
```

[6]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime

from sklearn.preprocessing import LabelEncoder
from geopy.distance import geodesic

# Load the dataset
data = pd.read_csv('credit_card_fraud.csv')

# Calculate the fraud rate by category
category_fraud_rate = data.groupby('category')['is_fraud'].mean().sort_values()

# Plot the fraud rate by merchant category
plt.figure(figsize=(10, 8))
category_fraud_rate.plot(kind='bar')
plt.title('Fraud Rate by Merchant Category')
plt.xlabel('Merchant Category')
plt.ylabel('Fraud Rate')
plt.tight_layout()
plt.show()
```



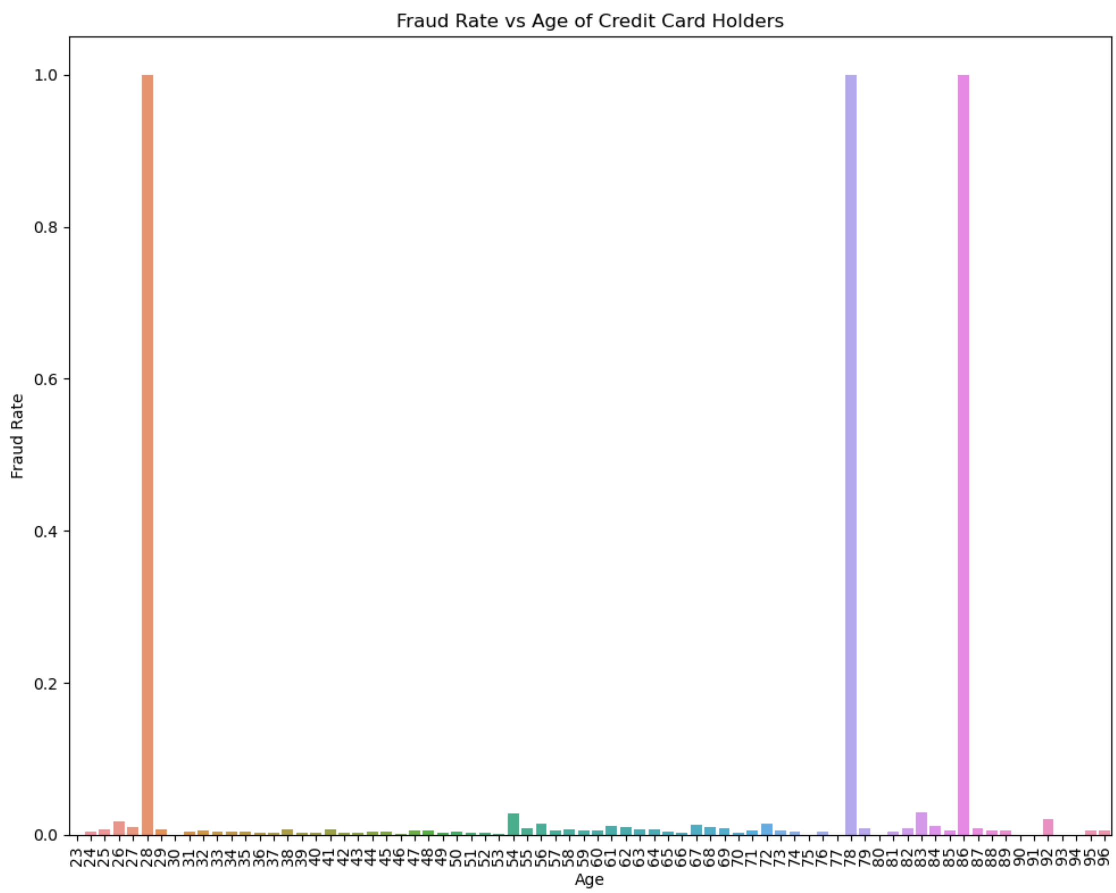
[7]:
```python
# Plot 2: Transaction Amount vs is_fraud
plt.figure(figsize=(10, 8))
sns.boxplot(x='is_fraud', y='amt', data=data)
plt.title('Transaction Amount vs. Fraud')
plt.xlabel('Is Fraud')
plt.ylabel('Transaction Amount ($)')
plt.xticks([0, 1], ['No', 'Yes'])  # Clearly label the x-axis for fraud status
plt.show()
```



[8]:

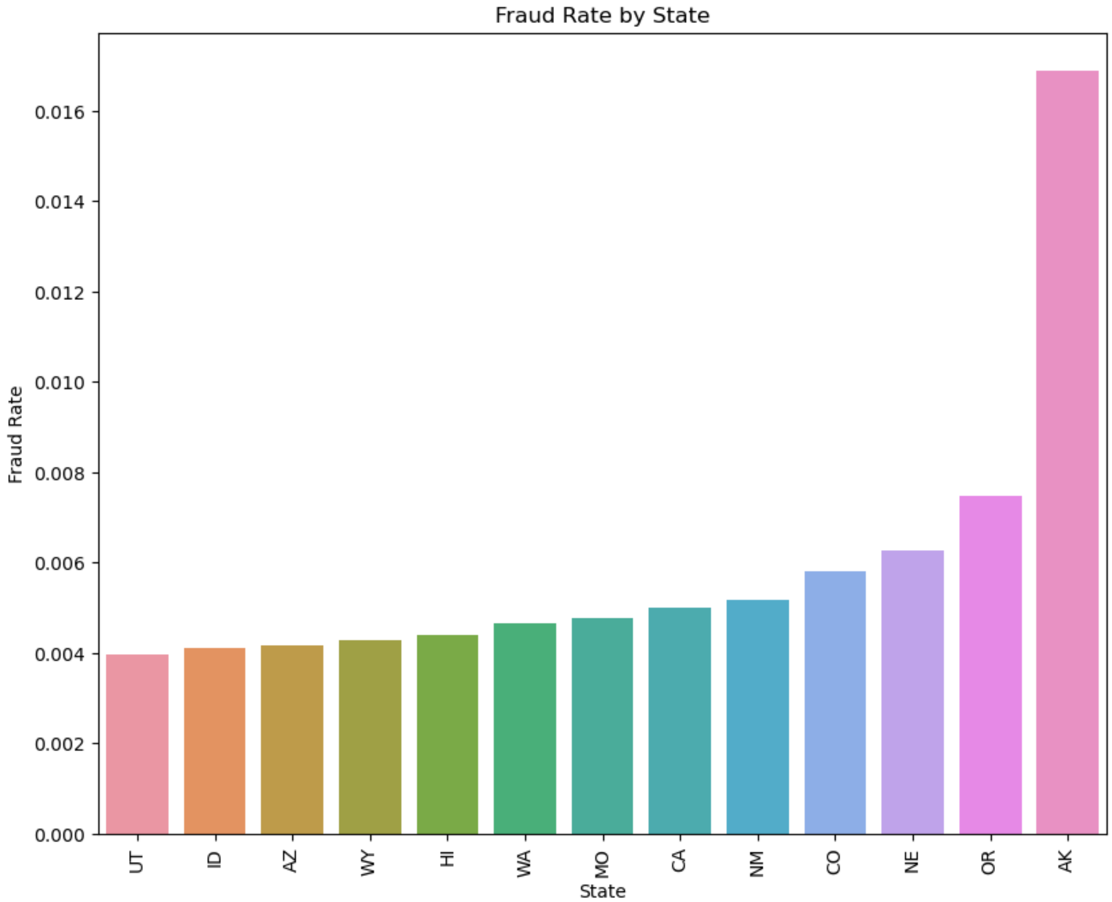
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime


# Calculate the current year
current_year = datetime.now().year

# Convert 'dob' to datetime and calculate age
data['dob'] = pd.to_datetime(data['dob'])
data['age'] = current_year - data['dob'].dt.year

# Create age bins
bins = list(range(data['age'].min(), data['age'].max() + 1, 1))  # 1-year bins
data['age_bin'] = pd.cut(data['age'], bins=bins, right=False, labels=bins[:-1])

# Calculate fraud rate by age
fraud_rate_by_age = data.groupby('age_bin')['is_fraud'].mean()
```

```
# Plot the fraud rate by age
plt.figure(figsize=(10, 8))
sns.barplot(x=fraud_rate_by_age.index, y=fraud_rate_by_age.values)
plt.title('Fraud Rate vs Age of Credit Card Holders')
plt.xlabel('Age')
plt.ylabel('Fraud Rate')
plt.xticks(rotation=90)  # Rotate the x-axis labels for better readability
plt.tight_layout()  # Adjust layout to fit the plot and labels
plt.show()
```



[9]:
```
# Plot 4: Fraud Rate by State
plt.figure(figsize=(10, 8))
state_fraud_rate = data.groupby('state')['is_fraud'].mean().sort_values()
sns.barplot(x=state_fraud_rate.index, y=state_fraud_rate.values)
plt.title('Fraud Rate by State')
plt.xlabel('State')
plt.ylabel('Fraud Rate')
plt.xticks(rotation=90)  # Rotate the x labels for better readability
```

```
plt.show()
```



3) Write a short overview/conclusion of the insights gained from your graphical analysis.

The graphical analysis reveals key trends in fraud patterns across various factors. Merchant categories with frequent online purchases, gas station purchases, or luxury purchases show a higher incidence of fraud, indicating a need for stricter transaction verification in those areas. Fraudulent transaction amounts are small amounts in general. More significant transactions are often strictly verified. Moreover, some states show a higher risk of credit card fraud. Age doesn't seem to have fraud susceptibility.

Term Project Milestone 2: Data Preparation

1) Feature Selection

From the initial dataset review, the following features may not be very useful:

- trans_num (Transaction Number): While unique identifiers are necessary for record-keeping,

they do not provide predictive power for the model.
- city: can lead to model overfitting. Instead, broader geographical indicators like state can be more useful.

[10]:
```
data.head()
```

[10]:
```
  trans_date_trans_time                   merchant       category     amt  \
0          1/1/2019 0:00  Heller, Gutmann and Zieme    grocery_pos  107.23
1          1/1/2019 0:00             Lind-Buckridge  entertainment  220.11
2          1/1/2019 0:07                 Kiehn Inc    grocery_pos   96.29
3          1/1/2019 0:09               Beier-Hyatt   shopping_pos    7.77
4          1/1/2019 0:21                Bruen-Yost       misc_pos    6.85

                      city state      lat      long  city_pop  \
0                   Orient    WA  48.8878 -118.2105       149
1               Malad City    ID  42.1808 -112.2620      4154
2                  Grenada    CA  41.6125 -122.5258       589
3  High Rolls Mountain Park    NM  32.9396 -105.8189       899
4                  Freedom    WY  43.0172 -111.0292       471

                                job         dob  \
0  Special educational needs teacher 1978-06-21
1        Nature conservation officer 1962-01-19
2                    Systems analyst 1945-12-21
3                    Naval architect 1967-08-30
4          Education officer, museum 1967-08-02

                           trans_num  merch_lat  merch_long  is_fraud  age  \
0  1f76529f8574734946361c461b024d99  49.159047 -118.186462         0   46
1  a1a22d70485983eac12b5b88dad1cf95  43.150704 -112.154481         0   62
2  413636e759663f264aae1819a4d4f231  41.657520 -122.230347         0   79
3  8a6293af5ed278dea14448ded2685fea  32.863258 -106.520205         0   57
4  f3c43d336e92a44fc2fb67058d5949e3  43.753735 -111.454923         0   57

  age_bin
0      46
1      62
2      79
3      57
4      57
```

[24]:
```
# Drop less useful features
data_cleaned = data.drop([ 'trans_num', 'city'], axis=1)
data_cleaned.head()
```

[24]:
```
  trans_date_trans_time                   merchant       category     amt  \
0          1/1/2019 0:00  Heller, Gutmann and Zieme    grocery_pos  107.23
1          1/1/2019 0:00             Lind-Buckridge  entertainment  220.11
```

```
2         1/1/2019 0:07                    Kiehn Inc    grocery_pos   96.29
3         1/1/2019 0:09                  Beier-Hyatt   shopping_pos    7.77
4         1/1/2019 0:21                   Bruen-Yost       misc_pos    6.85

  state      lat      long  city_pop                                job  \
0    WA  48.8878 -118.2105       149  Special educational needs teacher
1    ID  42.1808 -112.2620      4154        Nature conservation officer
2    CA  41.6125 -122.5258       589                    Systems analyst
3    NM  32.9396 -105.8189       899                    Naval architect
4    WY  43.0172 -111.0292       471          Education officer, museum

         dob  merch_lat  merch_long  is_fraud  age
0 1978-06-21  49.159047 -118.186462         0   46
1 1962-01-19  43.150704 -112.154481         0   62
2 1945-12-21  41.657520 -122.230347         0   79
3 1967-08-30  32.863258 -106.520205         0   57
4 1967-08-02  43.753735 -111.454923         0   57
```

2) Data Extraction and Transformation

- Extract time-based features from trans_date_trans_time, such as day of the week or hour of the day, which might correlate with fraud occurrences.

[18]:
```
# Convert 'trans_date_trans_time' to datetime and extract time-based features
data_cleaned['trans_date_trans_time'] = pd.
 ↪to_datetime(data_cleaned['trans_date_trans_time'])
data_cleaned['transaction_hour'] = data_cleaned['trans_date_trans_time'].dt.hour
data_cleaned['day_of_week'] = data_cleaned['trans_date_trans_time'].dt.
 ↪day_name()
data_cleaned.head()
```

[18]:
```
  trans_date_trans_time                    merchant       category     amt  \
0   2019-01-01 00:00:00  Heller, Gutmann and Zieme    grocery_pos  107.23
1   2019-01-01 00:00:00             Lind-Buckridge  entertainment  220.11
2   2019-01-01 00:07:00                  Kiehn Inc    grocery_pos   96.29
3   2019-01-01 00:09:00                Beier-Hyatt   shopping_pos    7.77
4   2019-01-01 00:21:00                 Bruen-Yost       misc_pos    6.85

  state      lat      long  city_pop                                job  \
0    WA  48.8878 -118.2105       149  Special educational needs teacher
1    ID  42.1808 -112.2620      4154        Nature conservation officer
2    CA  41.6125 -122.5258       589                    Systems analyst
3    NM  32.9396 -105.8189       899                    Naval architect
4    WY  43.0172 -111.0292       471          Education officer, museum

         dob  merch_lat  merch_long  is_fraud  age age_bin  transaction_hour  \
0 1978-06-21  49.159047 -118.186462         0   46      46                 0
```

```
1 1962-01-19  43.150704 -112.154481         0    62       62                   0
2 1945-12-21  41.657520 -122.230347         0    79       79                   0
3 1967-08-30  32.863258 -106.520205         0    57       57                   0
4 1967-08-02  43.753735 -111.454923         0    57       57                   0

  day_of_week
0     Tuesday
1     Tuesday
2     Tuesday
3     Tuesday
4     Tuesday
```

3) Data Cleaning and Feature Engineering

- Distance between customer and merchant: Calculate the geographic distance using (lat, long) and (merch_lat, merch_long), as transactions with unusual distances might indicate fraud.

[19]:
```
# Calculate distance between customer and merchant
def calculate_distance(row):
    return geodesic((row['lat'], row['long']), (row['merch_lat'],␣
 ↪row['merch_long'])).miles

data_cleaned['distance'] = data.apply(calculate_distance, axis=1)
```

[20]:
```
# Remove quotation marks from 'merchant' and 'job'
data_cleaned['merchant'] = data_cleaned['merchant'].str.replace('"', '')
data_cleaned['job'] = data_cleaned['job'].str.replace('"', '')
```

4) Handling Missing Data

[25]:
```
# Check for missing values
missing_data = data_cleaned.isnull().sum()
```

There were no missing values identified in the critical features after the initial cleaning steps.