Artificial Intelligence for Cybersecurity

Project

Sofia Prokhorova, Alexandra Pavlova

University of Pisa
2024/2025

Goal

The goal of this project is to analyze credit card transaction data using machine learning techniques to identify and classify fraud transactions.

Analysis

1

Preprocessing

2

Processing

3

Validation

4

Dataset

Description

The dataset we are working with contains 1,296,675 rows and 24 columns, split into:

- 11 numeric columns
- 2 datetime columns
- 7 text columns
- 4 categorical columns

trans_date_trans_...	merchant	amt	gender	city	zip	lat
Timestamp of the transaction.	Merchant or store where the transaction occurred.	Amount of the transaction.	Gender of the cardholder.	Address details of the cardholder.	Address details of the cardholder.	Geographical coordinates of the transaction.
2019-01-01 2020-06-21	693 unique values	1 28.9k	F 55% M 45%	894 unique values	1257 99.8k	20 6
2019-01-01 00:00:18	fraud_Rippin, Kub and Mann	4.97	F	Moravian Falls	28654	36.0788
2019-01-01 00:00:44	fraud_Heller, Gutmann and Zieme	107.23	F	Orient	99160	48.8878
2019-01-01 00:00:51	fraud_Lind-Buckridge	220.11	M	Malad City	83252	42.1808
2019-01-01 00:01:16	fraud_Kutch, Hermiston and Farrell	45.0	M	Boulder	59632	46.2306
2019-01-01 00:03:06	fraud_Keeling-Crist	41.96	M	Doe Hill	24433	38.4207
2019-01-01 00:04:08	fraud_Stroman, Hudson and Erdman	94.63	F	Dublin	18917	40.375
2019-01-01 00:04:42	fraud_Rowe-Vandervort	44.54	F	Holcomb	67851	37.9931

The file size is 354.15 MB. The transactions are recorded from 2019 to 2020.
We began by focusing on a subset of 10,000 rows to test and validate our methods before scaling up.

Dataset

Description

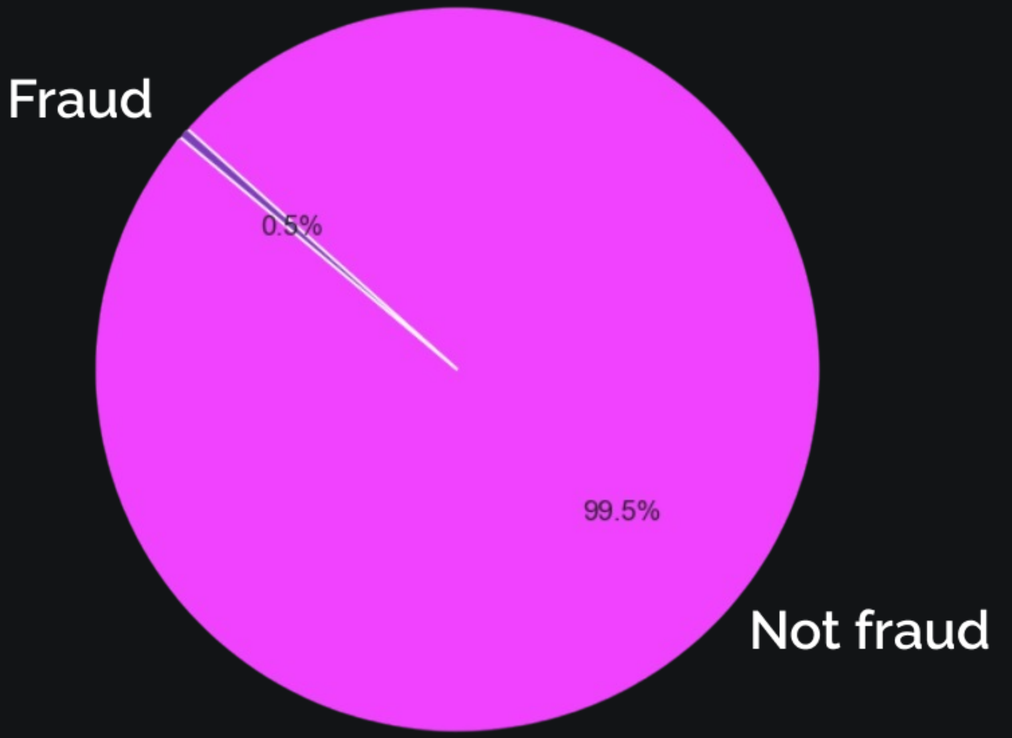
Our dataset is imbalanced. We have 7,506 fraud transactions in full dataset and 47 in small dataset.

```
df_fraud.is_fraud.value_counts()
```

```
is_fraud
0    1289169
1       7506
Name: count, dtype: int64
```

```
df_small.is_fraud.value_counts()
```

```
is_fraud
0    9953
1      47
Name: count, dtype: int64
```



Preprocessing | Feature Transformation

In original database we had columns “dob” (date of birth) and “trans_date_trans_time” (time of transaction) of type object.

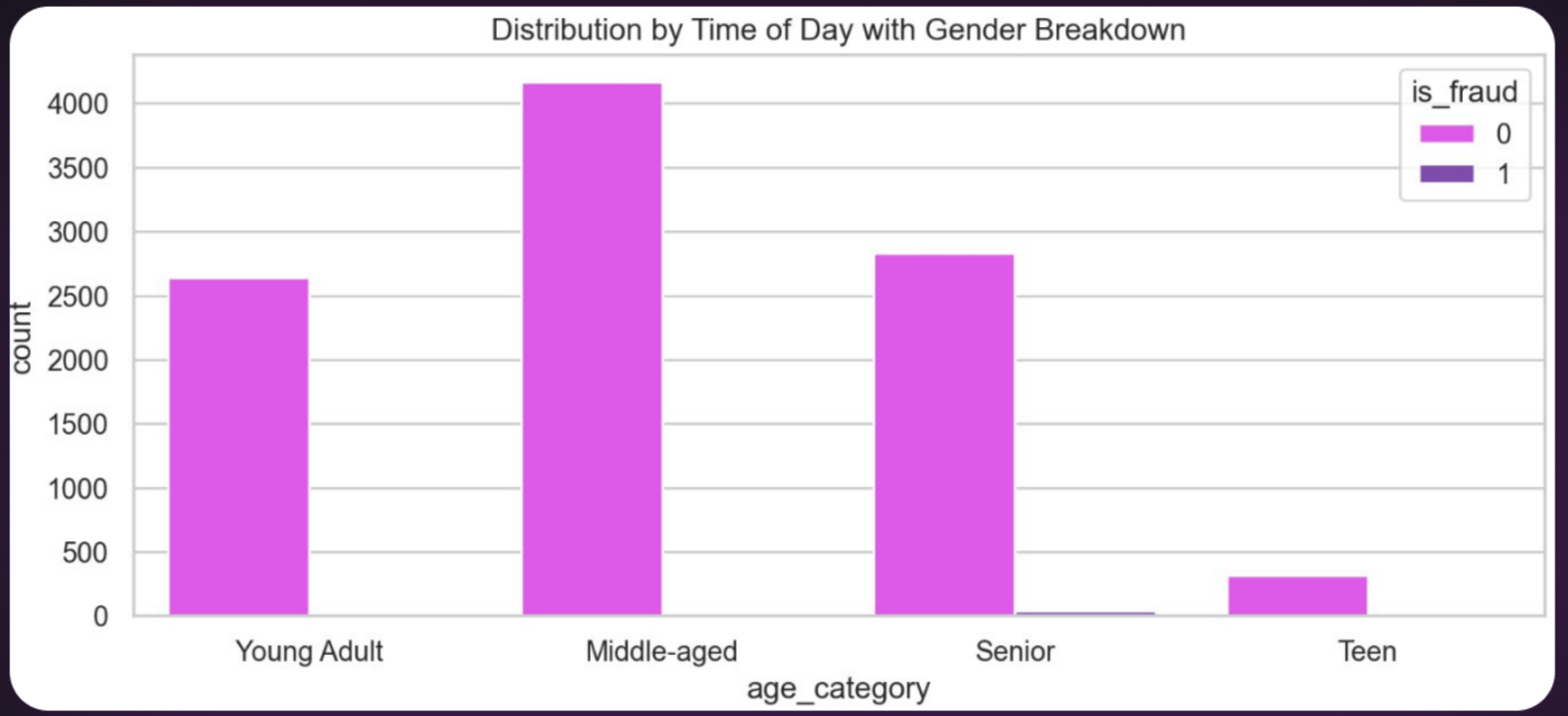
We created new columns and transformed data using OrdinalEncoder:

- time_of_day - nominal
- time_of_day_encoded - numerical
- age - numerical
- age_category - nominal
- age_category_encoded - numerical

Preprocessing

Feature Transformation

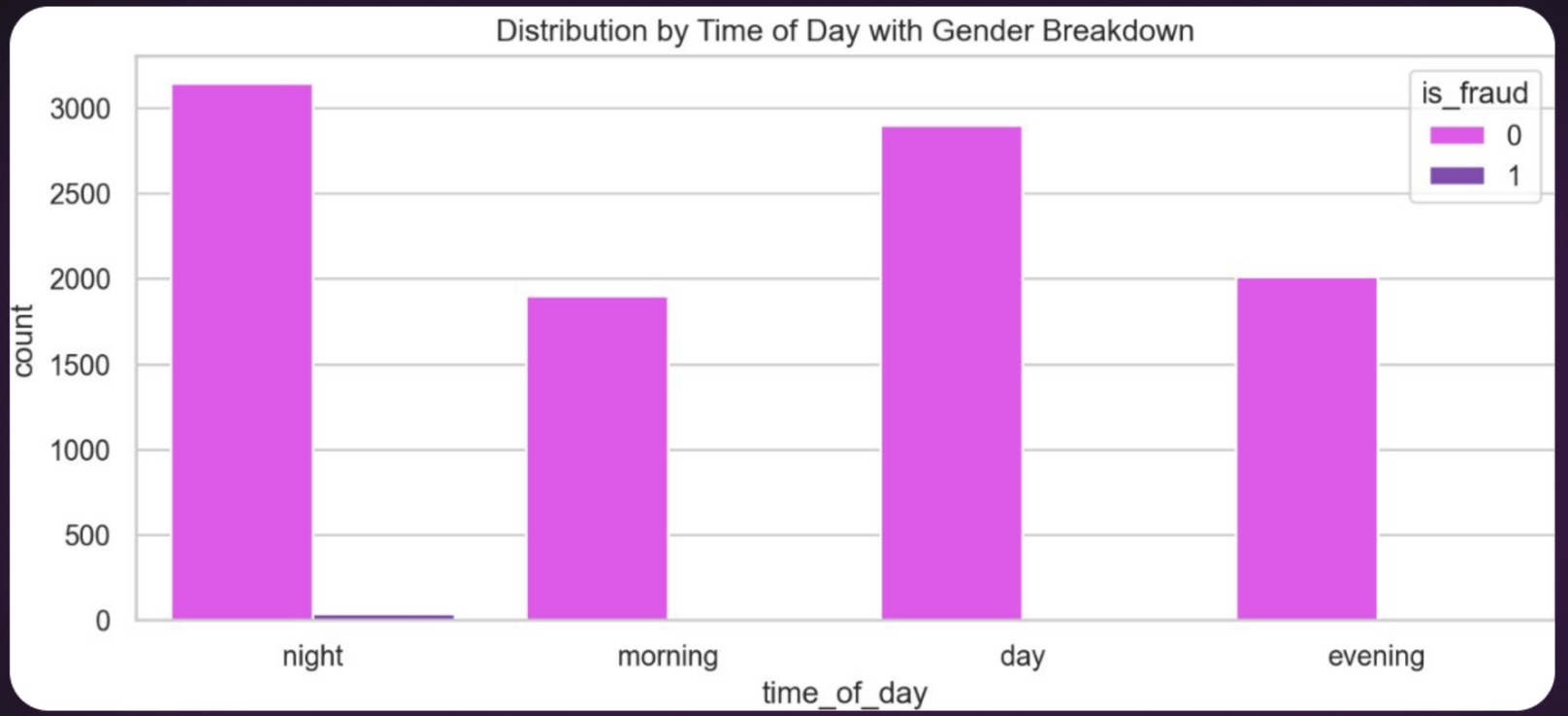
We found that fraud transactions were made with cards belonging to people aged 55 and above. Unfortunately, we notice that seniors are being targeted by criminals.



Preprocessing

Feature Transformation

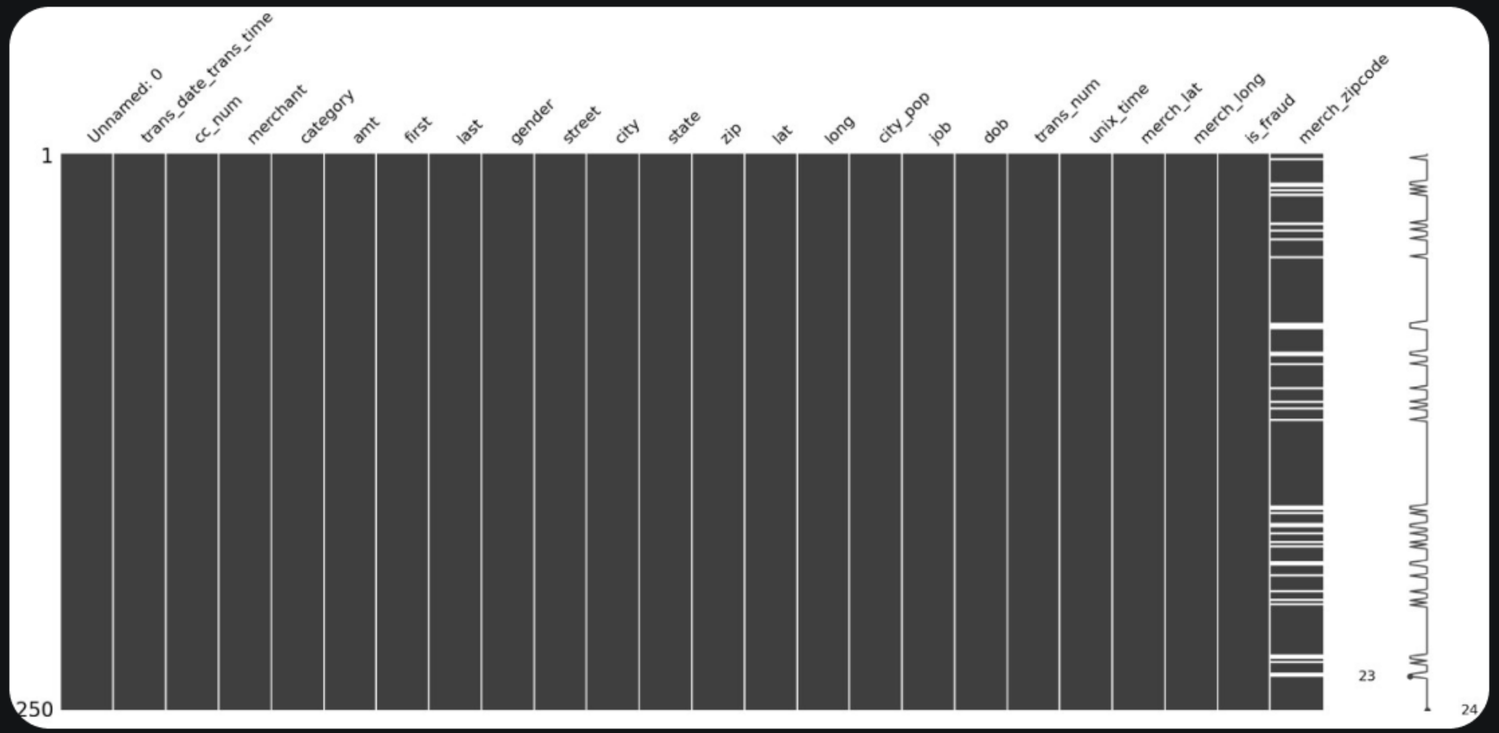
We also found that fraud transactions were detected only during nighttime.



Preprocessing

Missing data

We have missing data in the “merch_zipcode” field. However, due to the high correlation with the “zip” field, we decided to remove this column instead of filling in the missing values, in order to speed up and optimize our work.

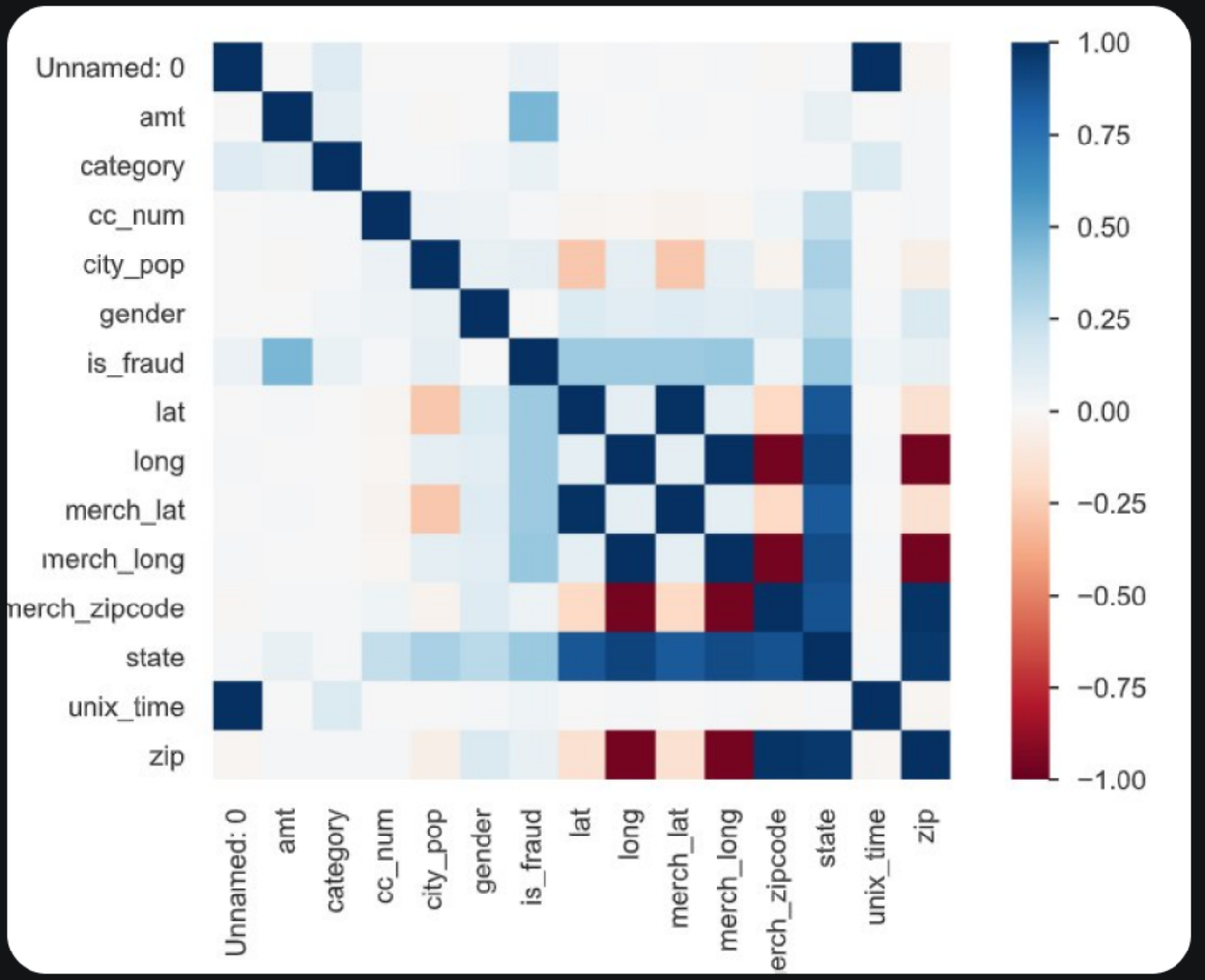


Preprocessing

Feature Selection

We built a ProfileReport and recieved correlation matrix.

We see that pairs long and merch_long, lat and merch_lat, zip and merch_zipcode has high correlation.

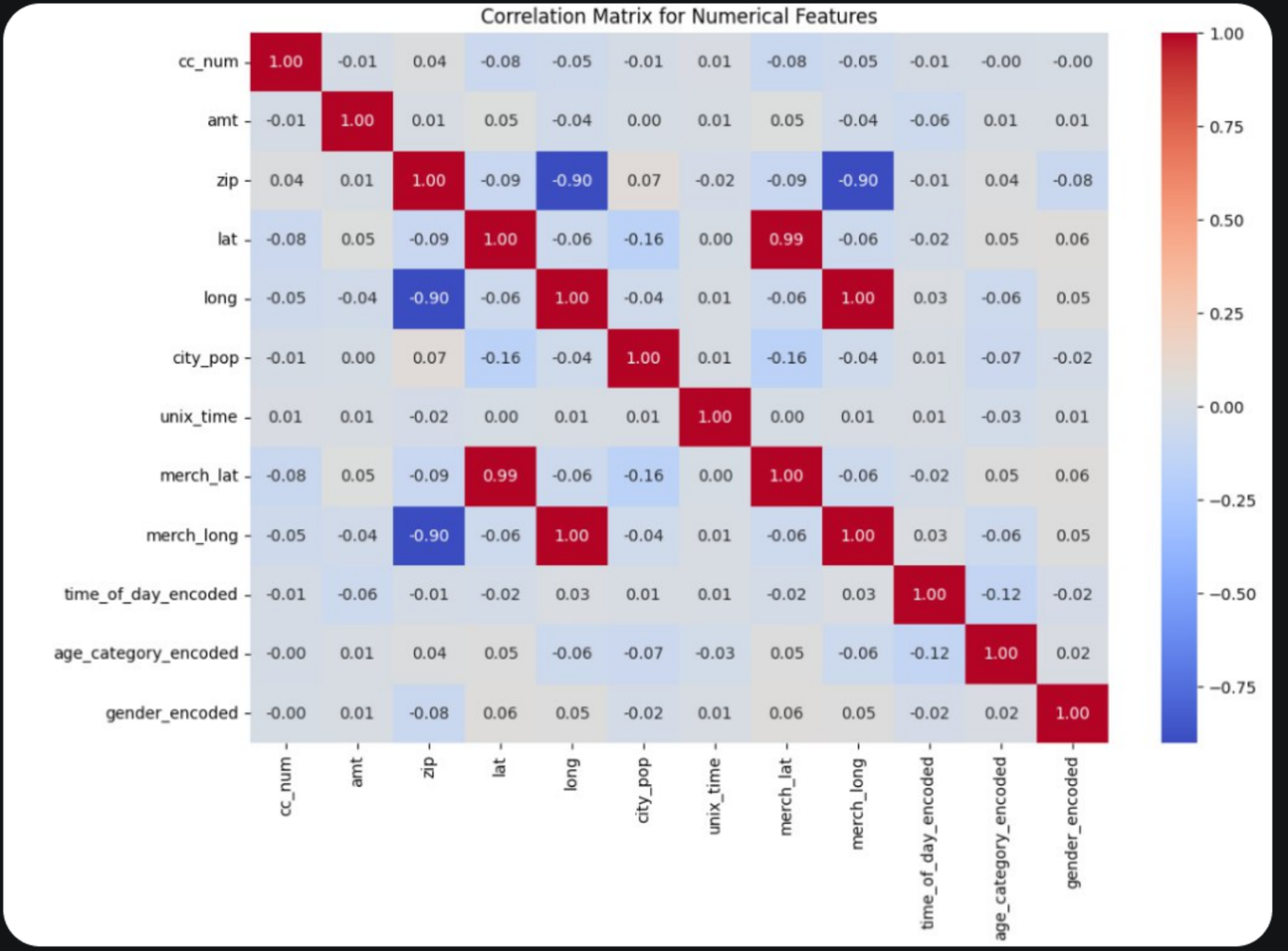


Preprocessing

Feature Selection

We also used StratifiedKFold method for two feature selection methods for already transformed data.

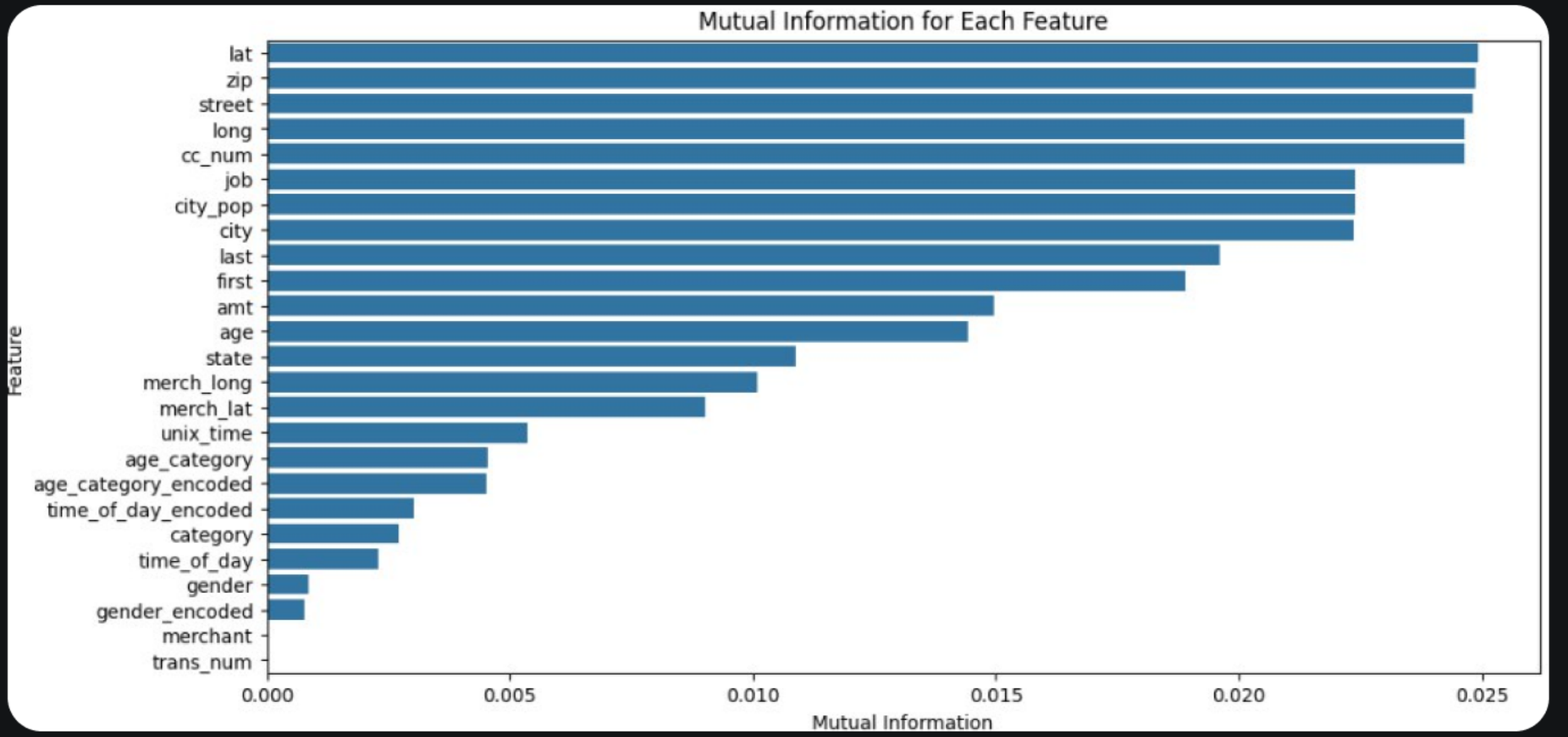
Correlation matrix:



Preprocessing

Feature Selection

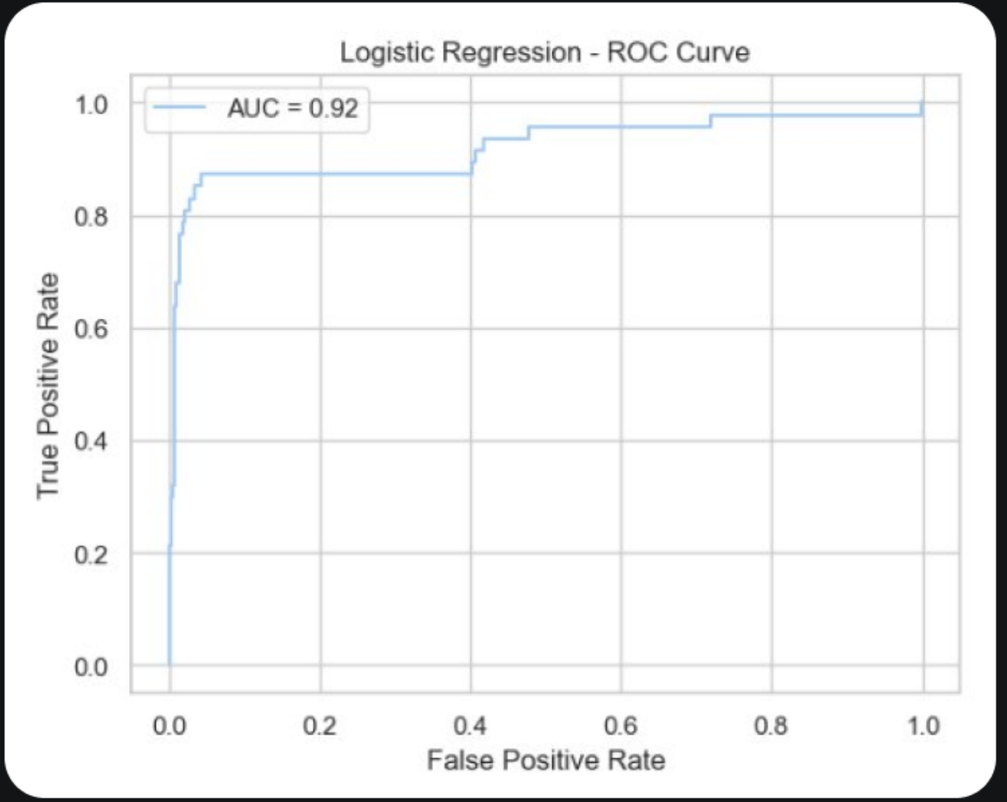
Univariante feature selection

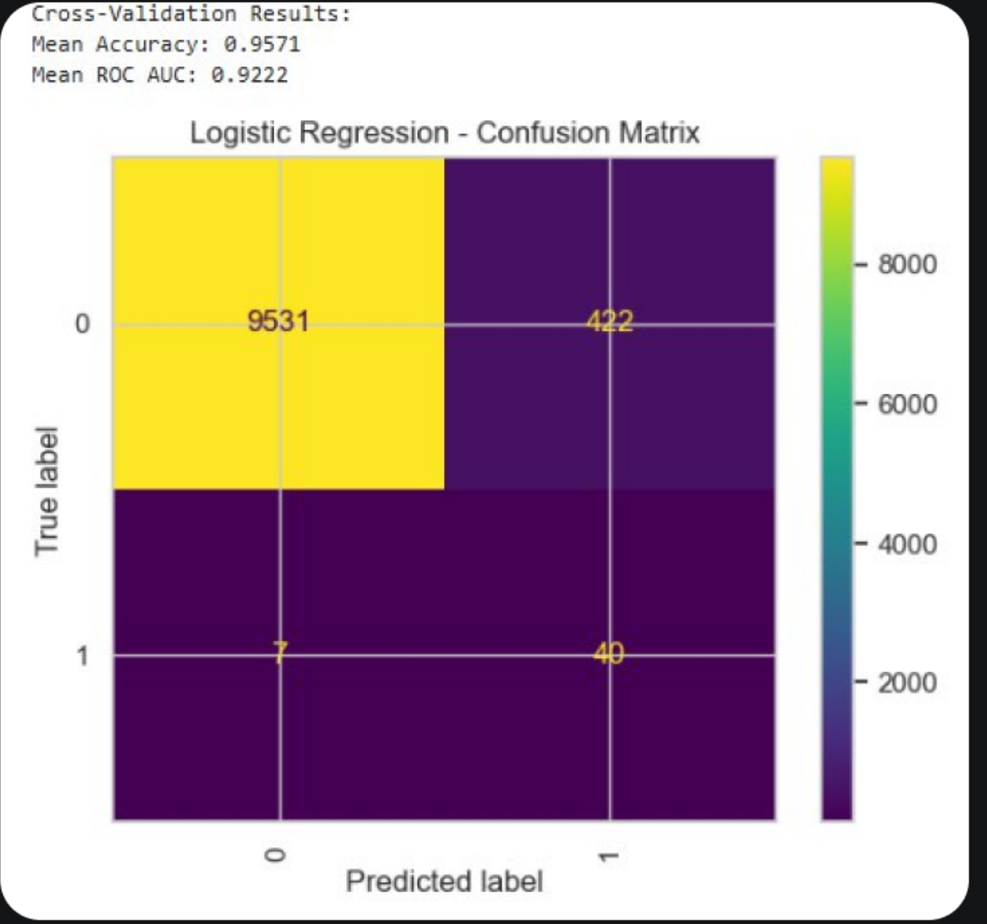


Processing

Classification

Logistic regression

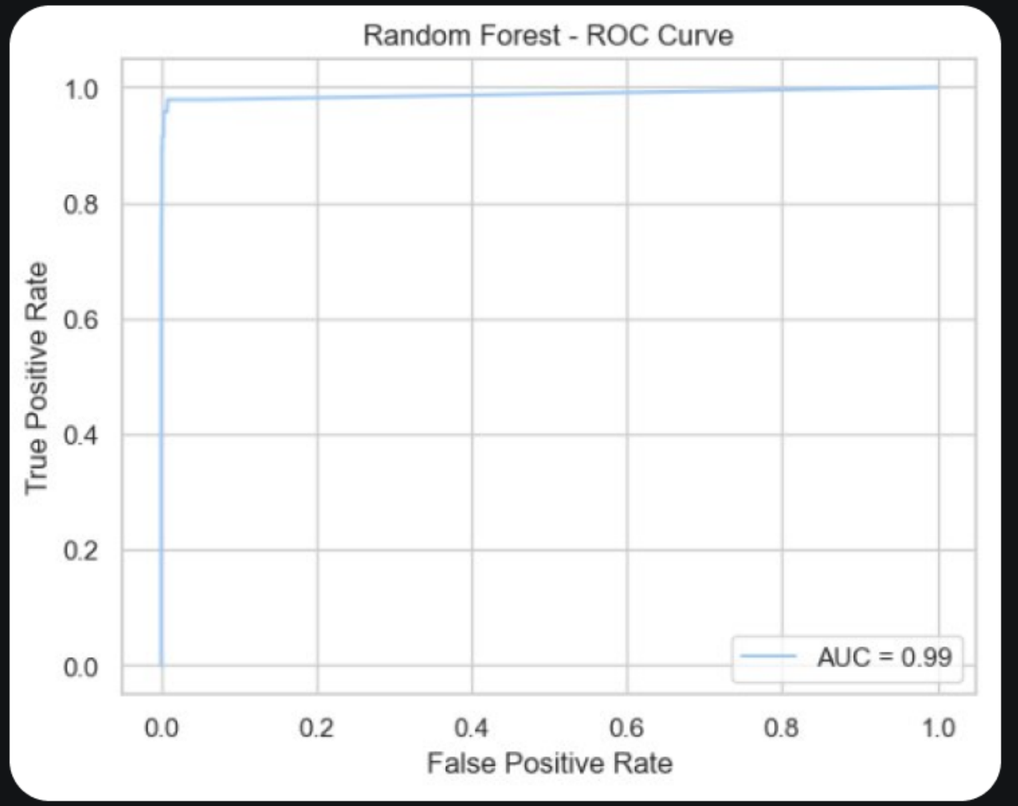


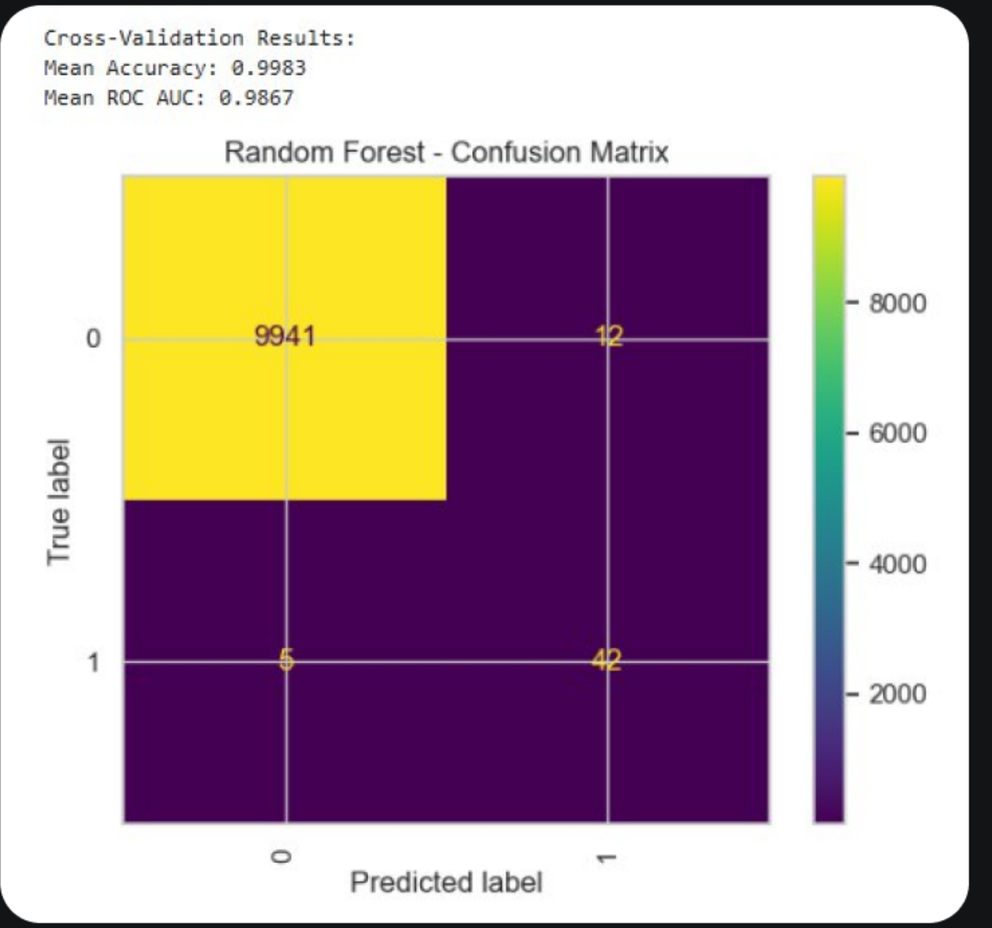


Processing

Classification

Random forest

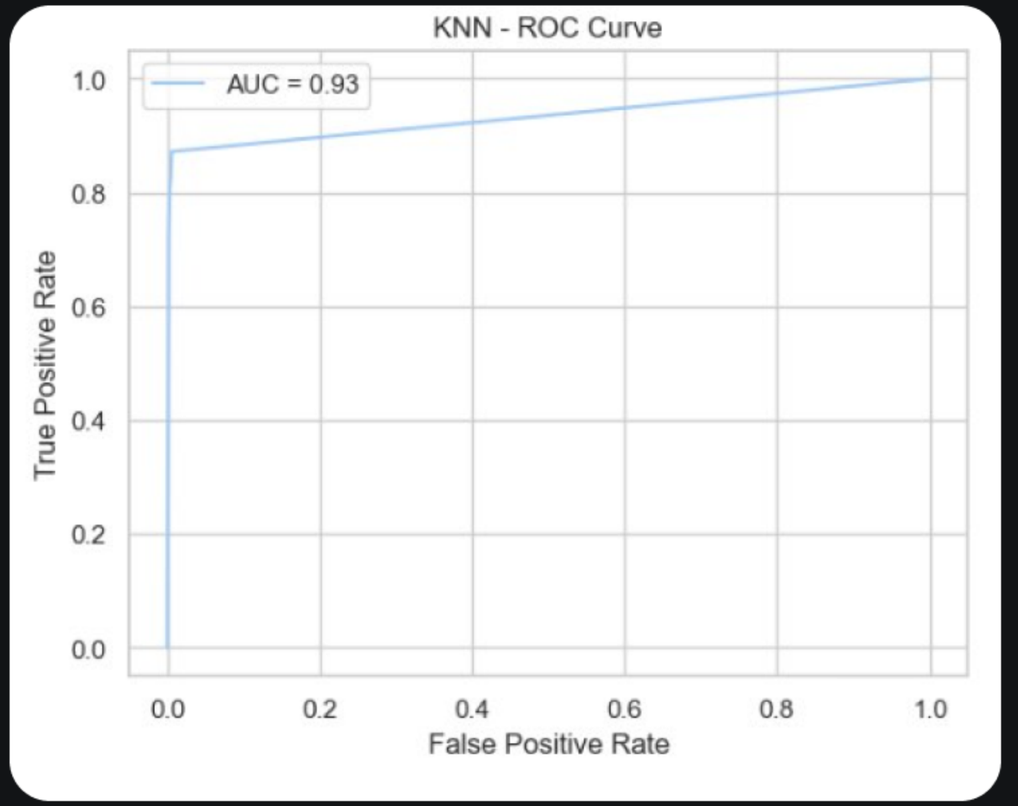


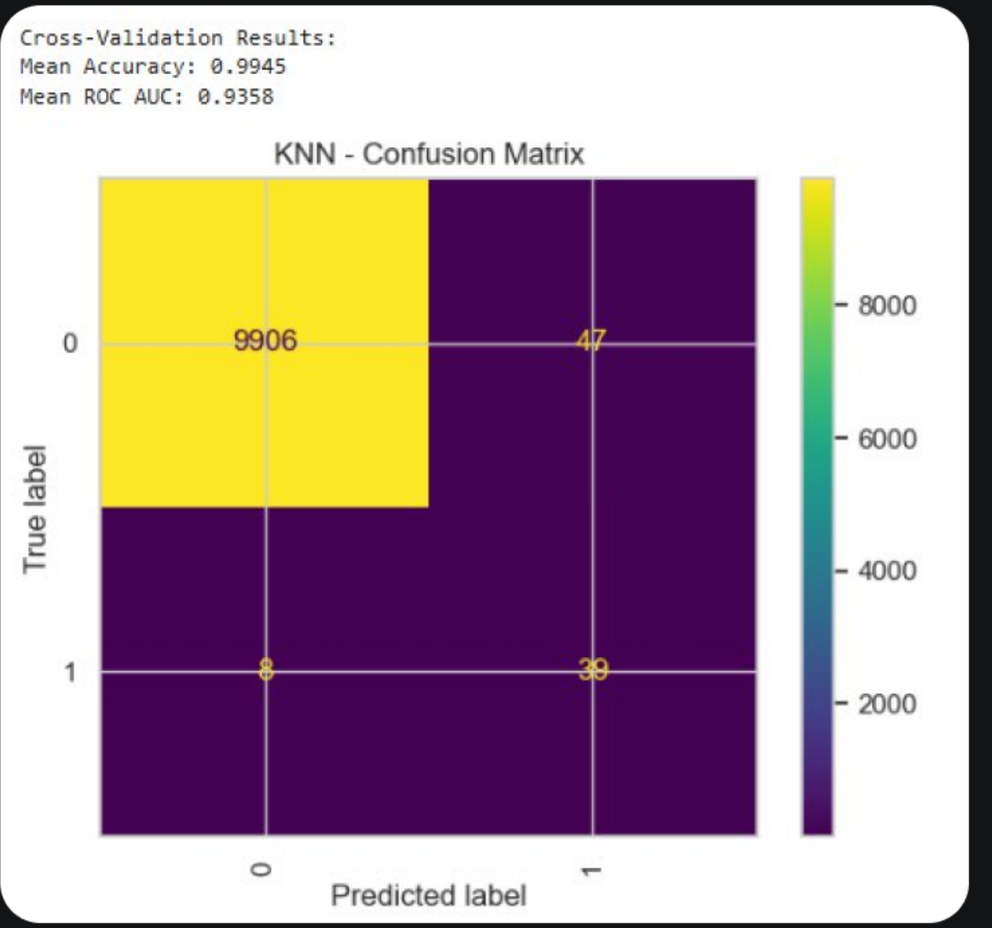


Processing

Classification

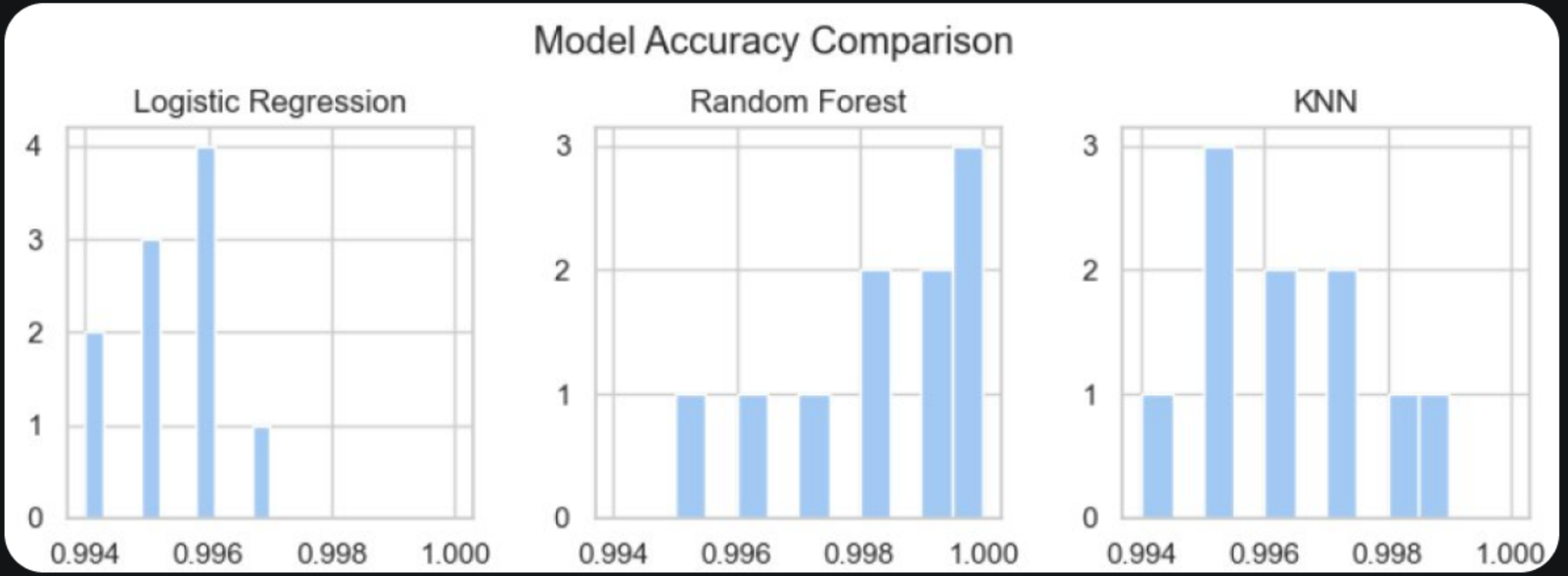
KNN

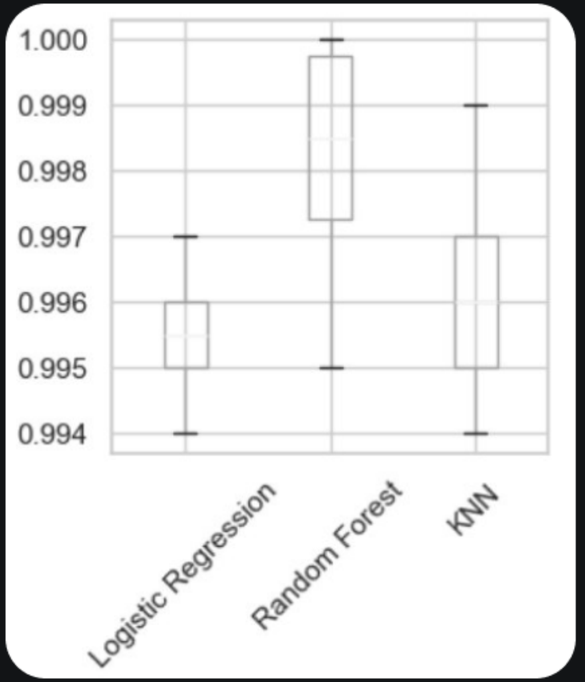




Validation

We generated histograms to analyze the accuracy comparison.






```
Wilcoxon Test Results:
Logistic Regression vs Random Forest: WilcoxonResult(statistic=np.float64(0.0), pvalue=np.float64(0.001953125))
Logistic Regression vs KNN: WilcoxonResult(statistic=np.float64(0.0), pvalue=np.float64(0.06559969214707187))
Random Forest vs KNN: WilcoxonResult(statistic=np.float64(4.0), pvalue=np.float64(0.026479069642187852))
```

Conclusion

- To summarize, we used machine learning techniques to analyze credit card transaction data, helping to detect and classify fraudulent transactions.
- In the analysis phase, we found that fraud transactions happened at night with the cards of seniors over 55. This shows that older people are being targeted by criminals.
- In the processing phase, we compared Logistic Regression, Random Forest, and KNN, and found that Random Forest gave the best results for fraud detection.

References

- Link on the dataset on Kaggle: https://www.kaggle.com/code/momotokd/prophet-arima-cc-transactions-analysis/input
- Our work was based on lectures from professors

Thank you!