HEALTHCARE ANALYSIS

INTRODUCTION

- Blood datasets are essential tools in the field of medical research, clinical diagnostics, and public health initiatives. These datasets typically contain a variety of hematological and blood chemistry information, including blood cell counts, hemoglobin levels, glucose, cholesterol, and electrolyte values. These data points are invaluable for analyzing various health conditions, identifying risk factors, and assessing treatment efficacy. With advancements in machine learning, blood datasets are increasingly being used to develop predictive models for diagnosing diseases, predicting patient outcomes, and enhancing personalized care.
- The dataset under analysis in this study includes attributes such as Recency, Frequency, Monetary, Time, and Class. These attributes measure different aspects of blood donation behavior, making the dataset useful for analyzing donation patterns and classifying donors based on various parameters. This can provide insights into donor retention and the overall effectiveness of blood donation campaigns.

OBJECTIVE

•**Donor Segmentation**: Grouping donors based on their behavior (recency, frequency, and monetary contributions) to identify patterns and trends in blood donations.

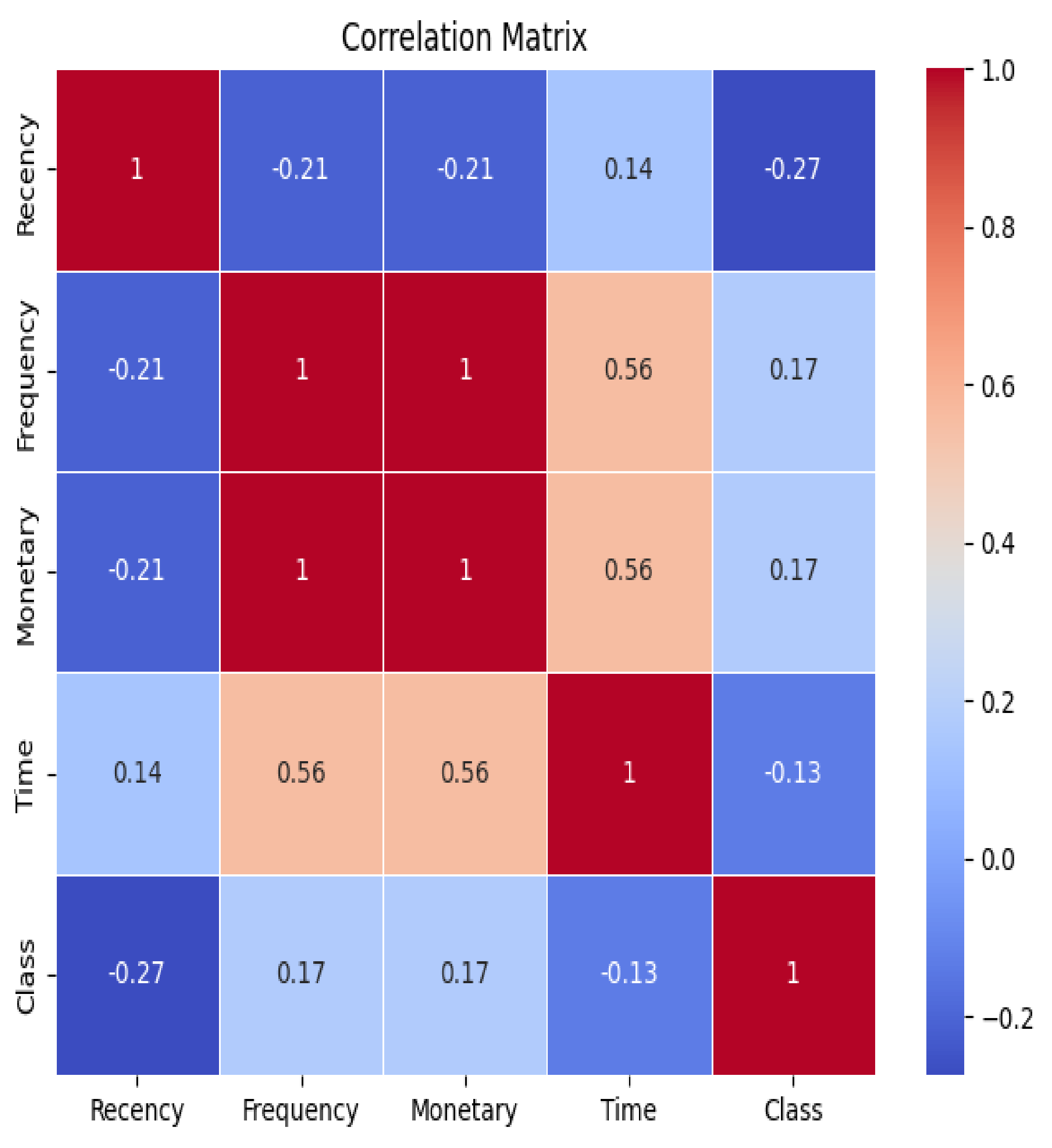
•**Predictive Modeling**: Using machine learning techniques to predict whether a donor will donate blood again in the future based on historical data.

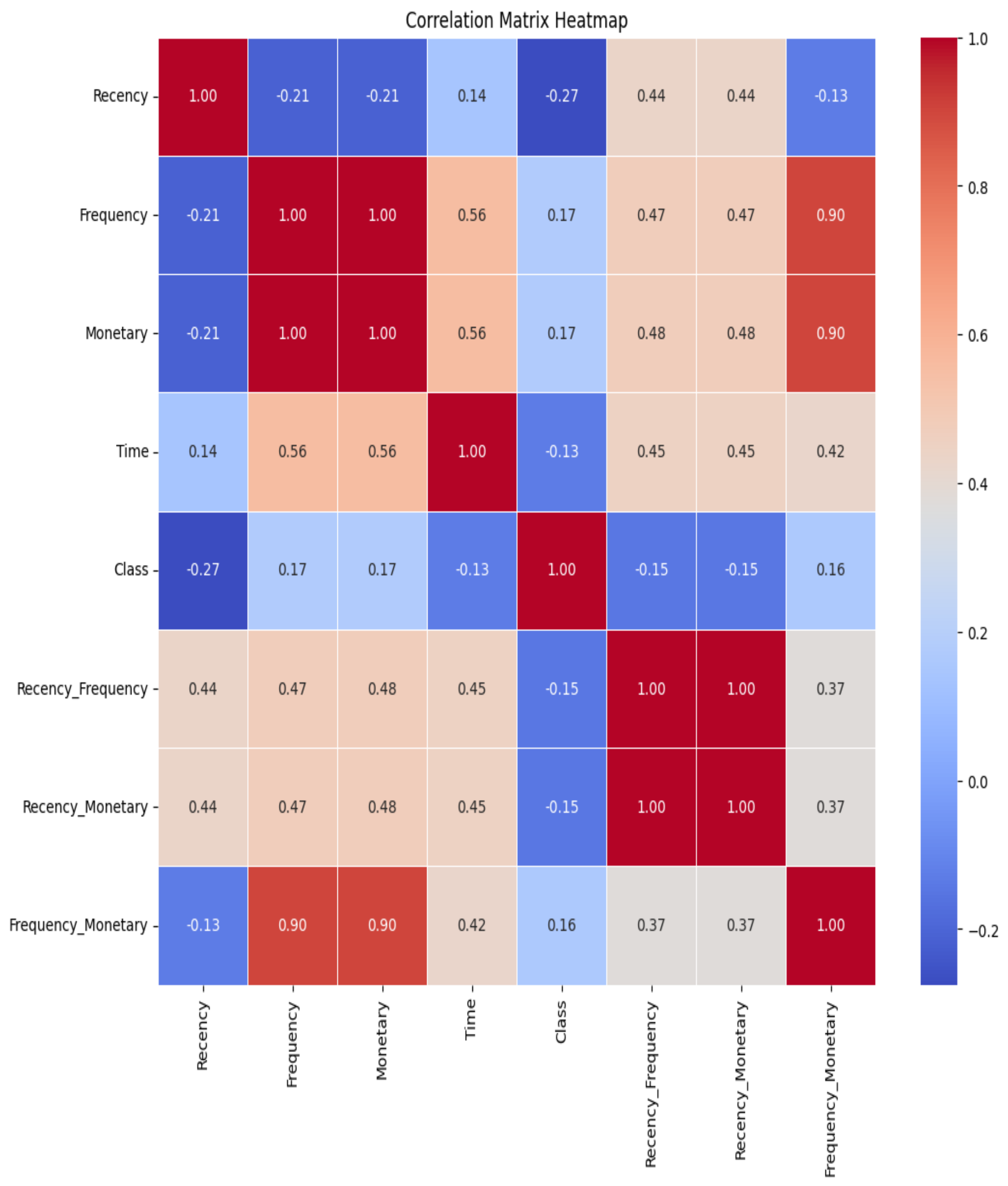
•**Insight Generation**: Analyzing the dataset to generate actionable insights, such as which groups of donors are
more likely to be retained and how donor engagement can be improved.

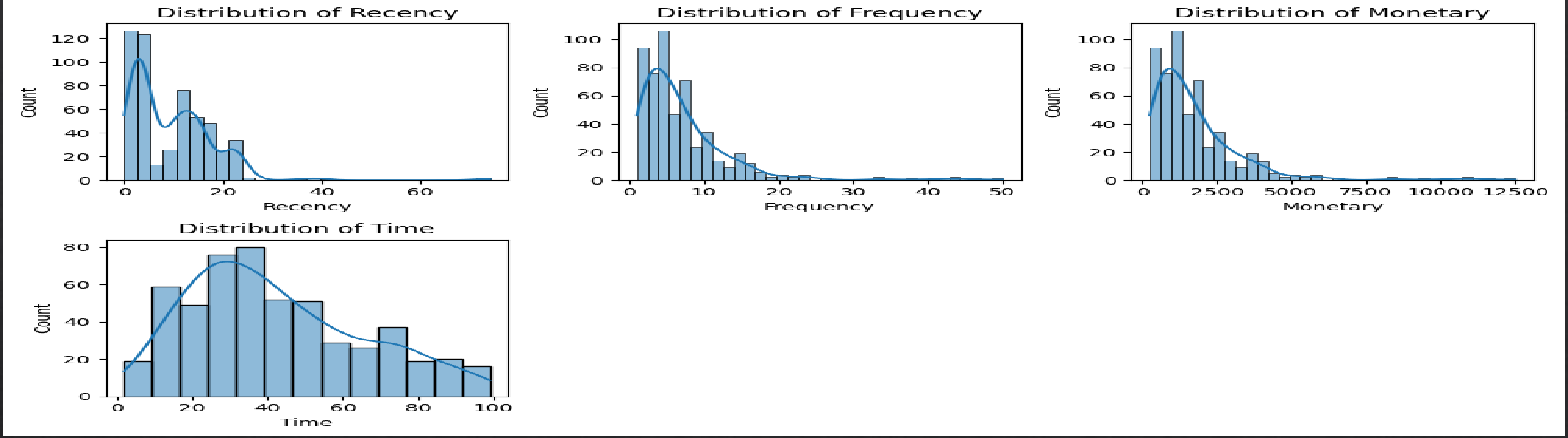
•**Improving Healthcare Outcomes**: Leveraging the findings to assist in the development of targeted strategies to
enhance blood donation rates, ensuring an adequate and continuous supply of blood for medical needs.

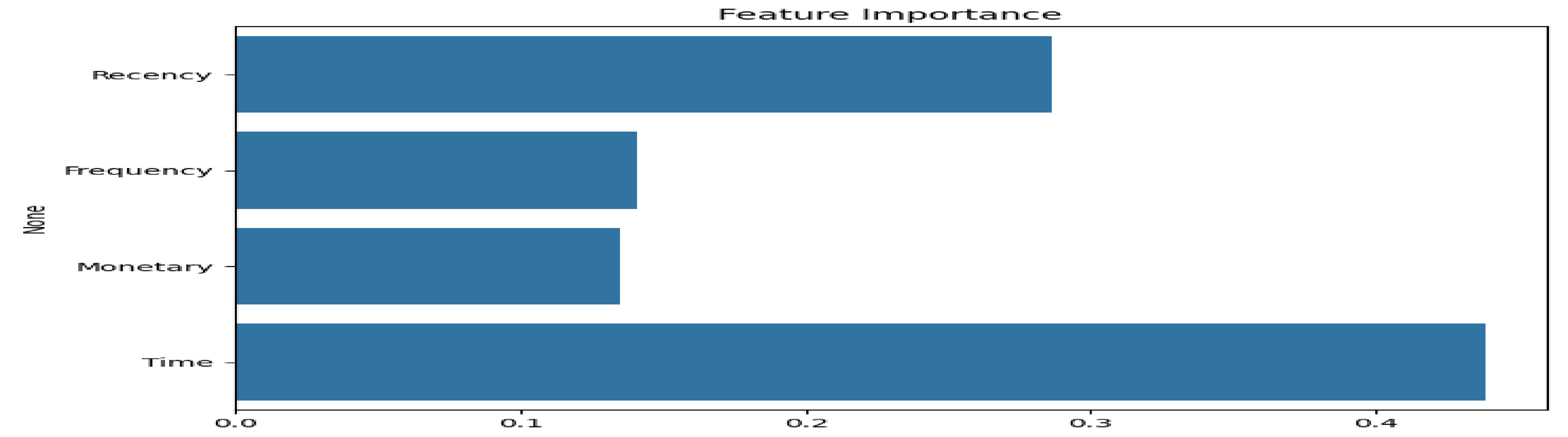
PYTHON

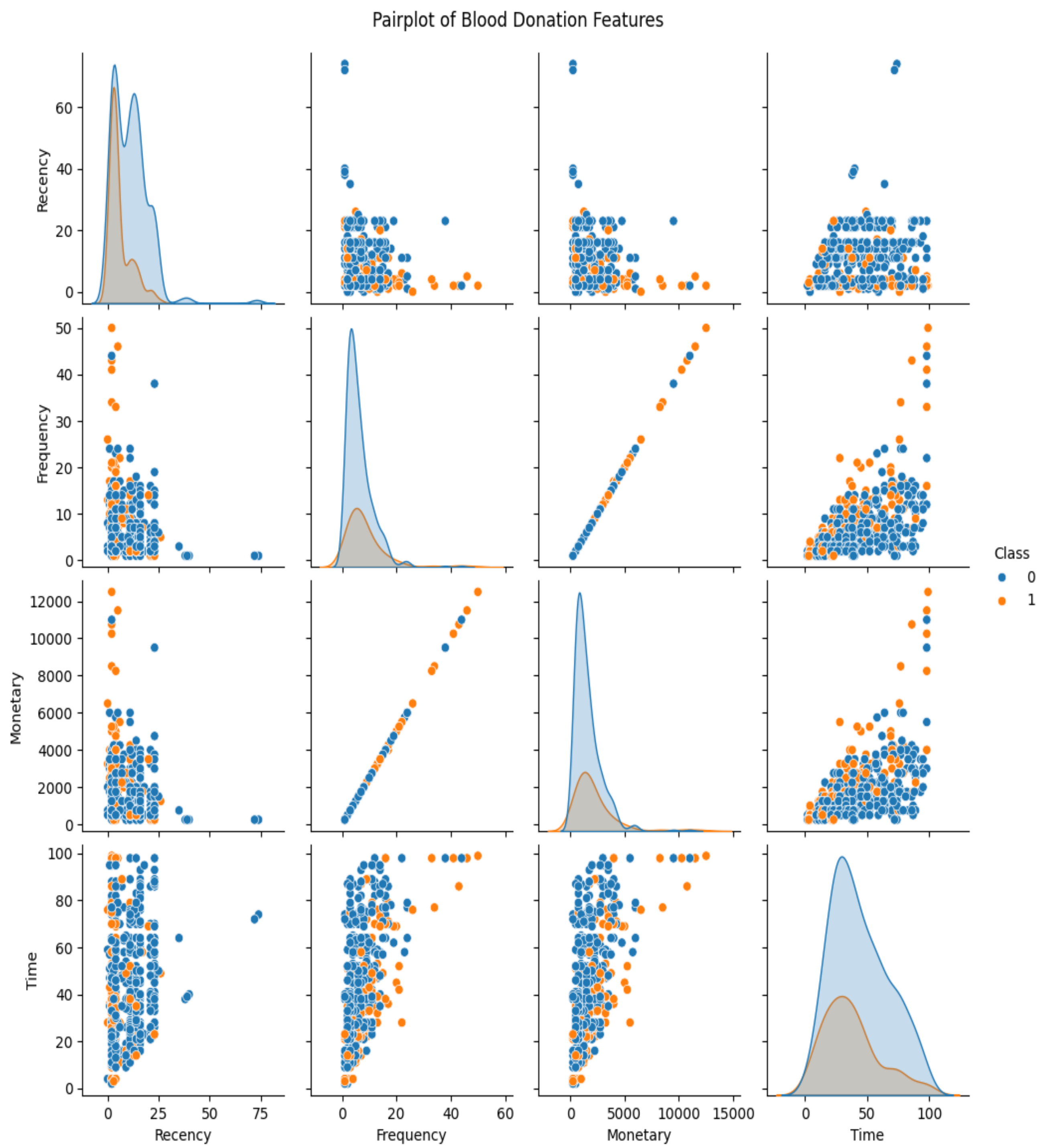
In this project, Python played a crucial role in data analysis and machine learning. I utilized libraries such as **Pandas** for data manipulation and cleaning, allowing me to preprocess the blood donor dataset efficiently. For exploratory data analysis (EDA), I employed **Seaborn** and **Matplotlib** to visualize trends and relationships within the data.
Additionally, I implemented various **machine learning algorithms** from **Scikit-learn**, including Logistic Regression, Decision Trees, and Random Forest, to build predictive models that classify donor behavior. This integration of Python's powerful data analysis and machine learning capabilities enabled me to derive meaningful insights and improve blood donation predictions effectively.

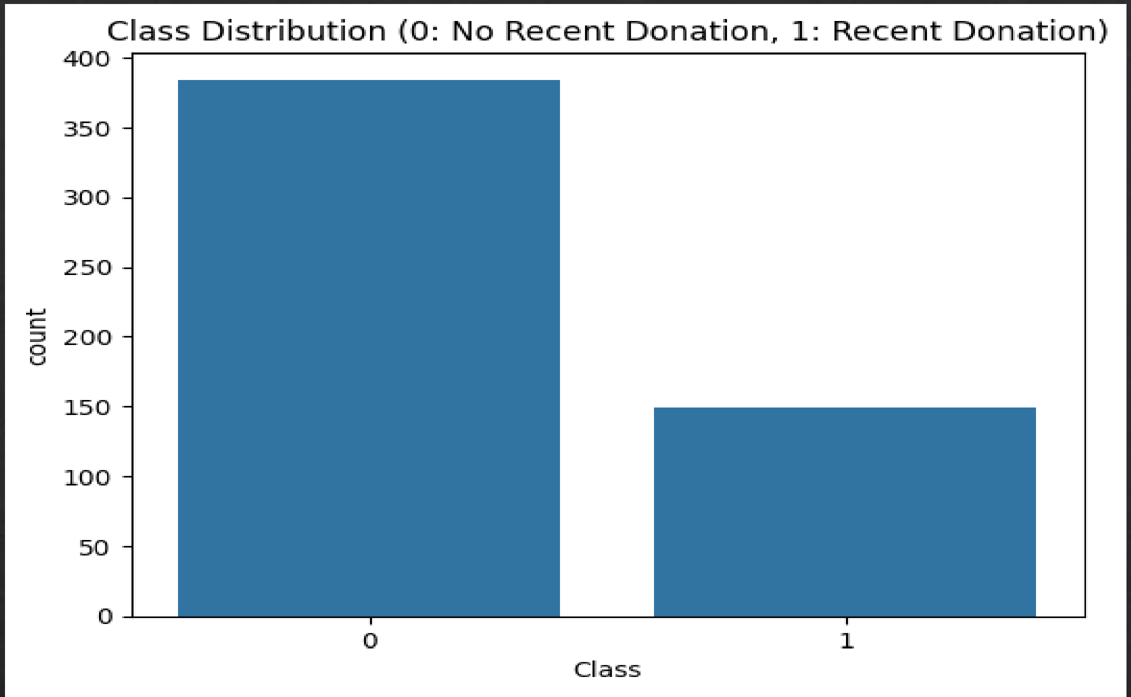

Correlation Matrix

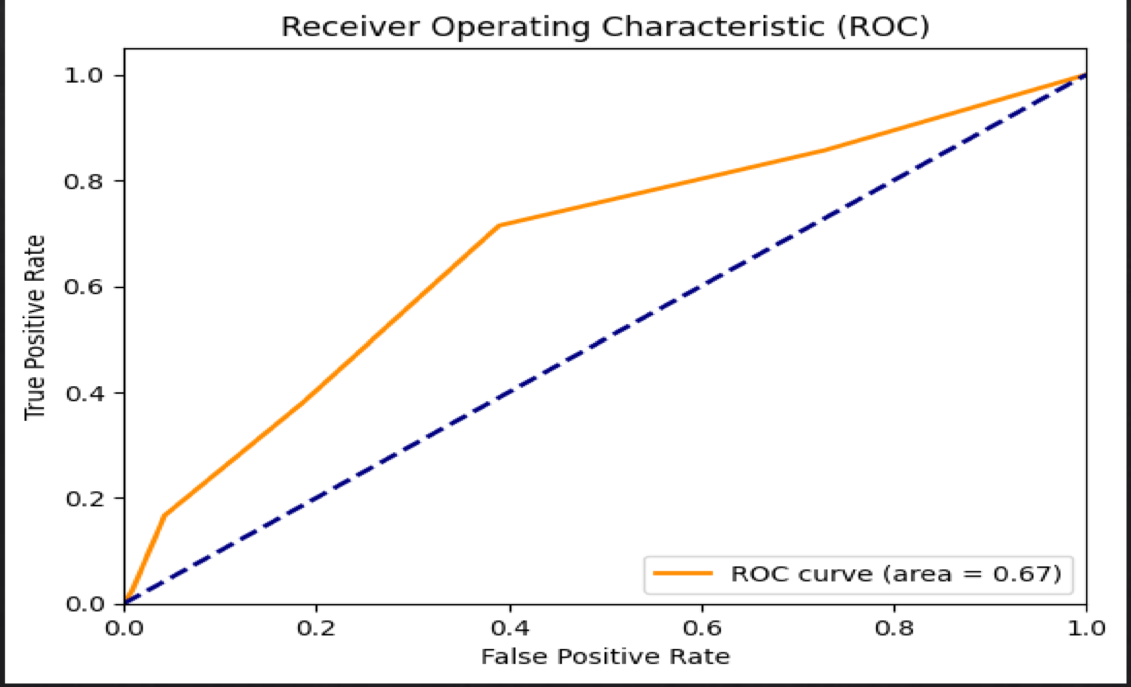

Correlation Matrix Heatmap

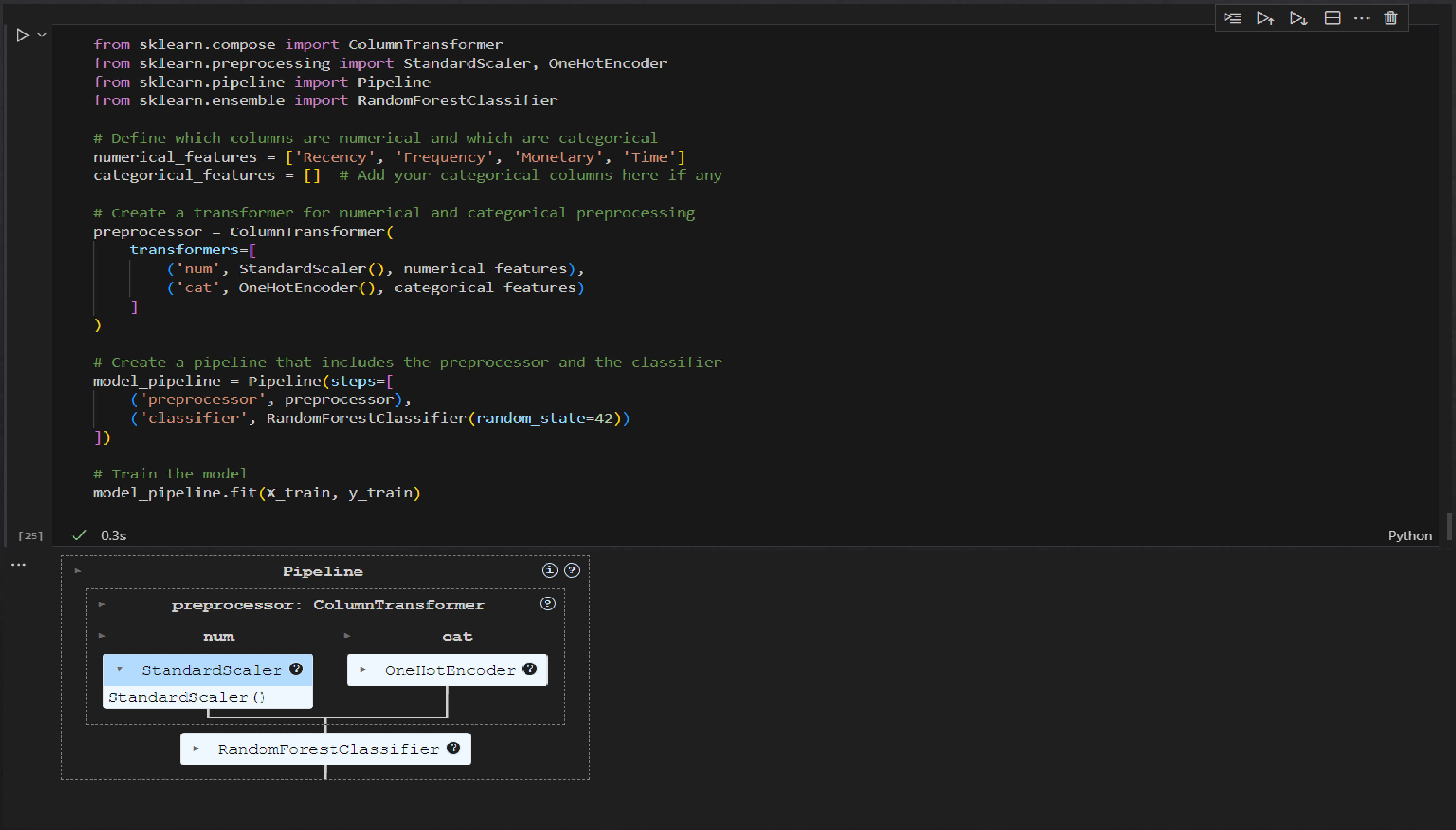

Distribution of Recency
Count
120
100
80
60
40
20
0
0
20
40
60
Recency
Distribution of Frequency
Count
100
80
60
40
20
0
0
10
20
30
40
50
Frequency
Distribution of Monetary
Count
100
80
60
40
20
0
0
2500
5000
7500
10000
12500
Monetary
Distribution of Time
Count
80
60
40
20
0
0
20
40
60
80
100
Time


Feature Importance
None
Recency
Frequency
Monetary
Time
0.0
0.1
0.2
0.3
0.4


Pairplot of Blood Donation Features
Recency
Frequency
Monetary
Time
Class
0
1
0
20
40
60
10
30
50
2000
4000
6000
8000
10000
12000
80
100
25
75
5000
15000


Class Distribution (0: No Recent Donation, 1: Recent Donation)
count
Class
0
1
0
50
100
150
200
250
300
350
400


Receiver Operating Characteristic (ROC)
True Positive Rate
False Positive Rate
ROC curve (area = 0.67)
0.0
0.2
0.4
0.6
0.8
1.0

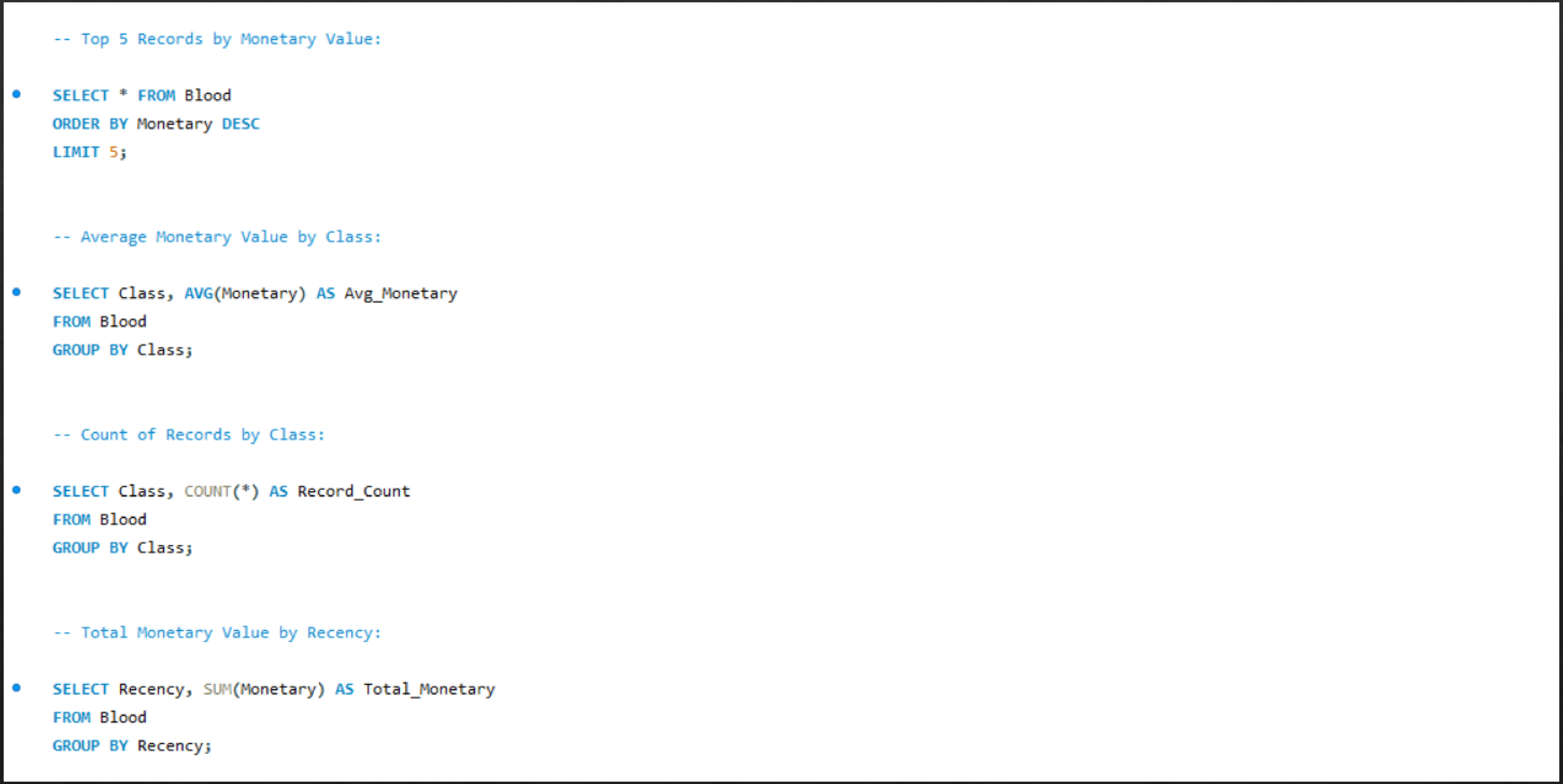


```python
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestClassifier

# Define which columns are numerical and which are categorical
numerical_features = ['Recency', 'Frequency', 'Monetary', 'Time']
categorical_features = []  # Add your categorical columns here if any

# Create a transformer for numerical and categorical preprocessing
preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numerical_features),
        ('cat', OneHotEncoder(), categorical_features)
    ]
)

# Create a pipeline that includes the preprocessor and the classifier
model_pipeline = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('classifier', RandomForestClassifier(random_state=42))
])

# Train the model
model_pipeline.fit(X_train, y_train)
```

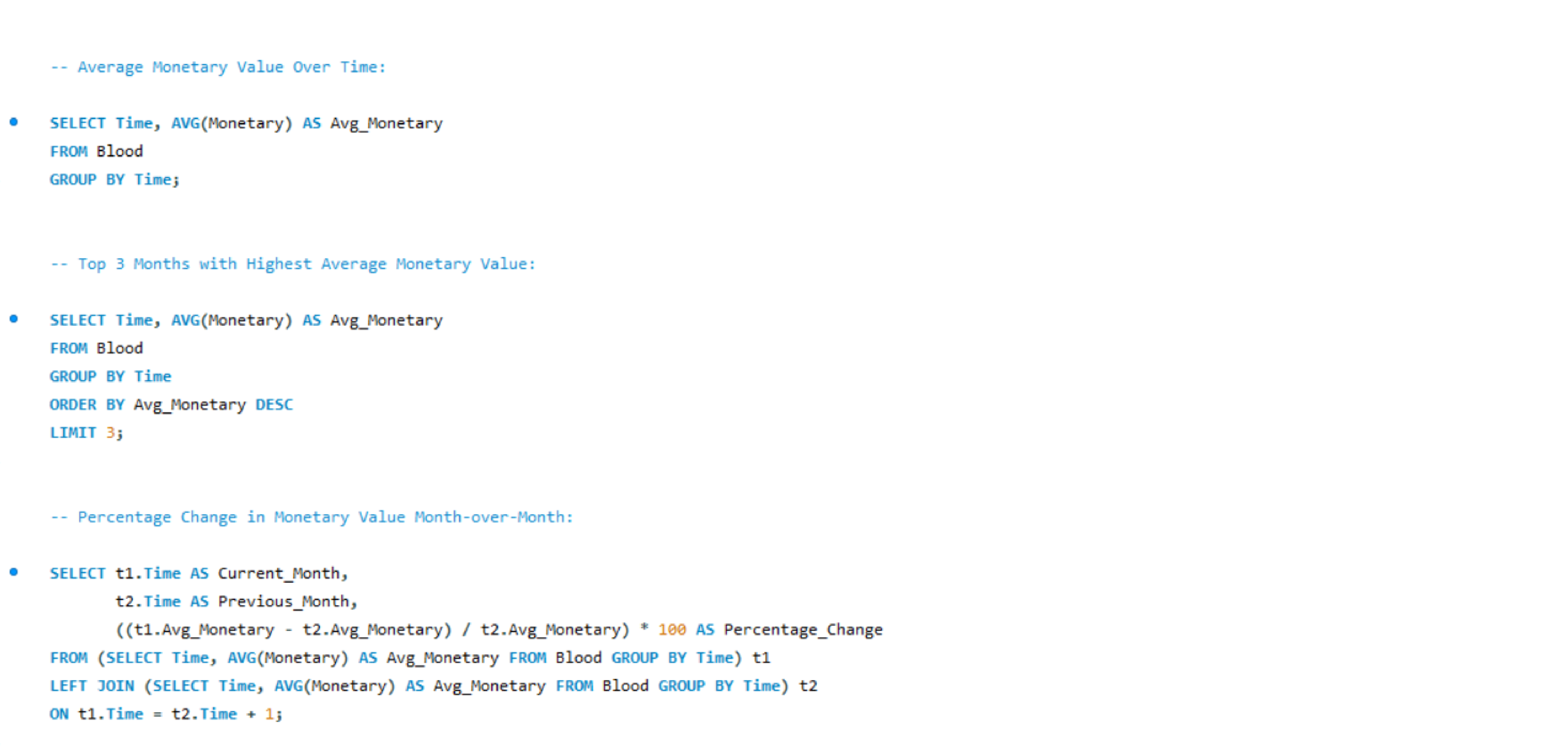
```
StandardScaler()
```

MY SQL

In this project, I utilized **MySQL** to efficiently manage and analyze the blood donor dataset. My SQL queries facilitated complex data manipulations, allowing me to perform intricate joins, aggregations, and filtering to extract meaningful insights. By structuring the database effectively, I ensured seamless data retrieval and improved performance for exploratory data analysis. My experience with SQL enhanced my ability to derive actionable insights and support data-driven decision-making within the context of blood donation predictions.



```
-- Top 5 Records by Monetary Value:

SELECT * FROM Blood
ORDER BY Monetary DESC
LIMIT 5;


-- Average Monetary Value by Class:

SELECT Class, AVG(Monetary) AS Avg_Monetary
FROM Blood
GROUP BY Class;


-- Count of Records by Class:

SELECT Class, COUNT(*) AS Record_Count
FROM Blood
GROUP BY Class;


-- Total Monetary Value by Recency:

SELECT Recency, SUM(Monetary) AS Total_Monetary
FROM Blood
GROUP BY Recency;
```



```sql
-- Average Monetary Value Over Time:
```

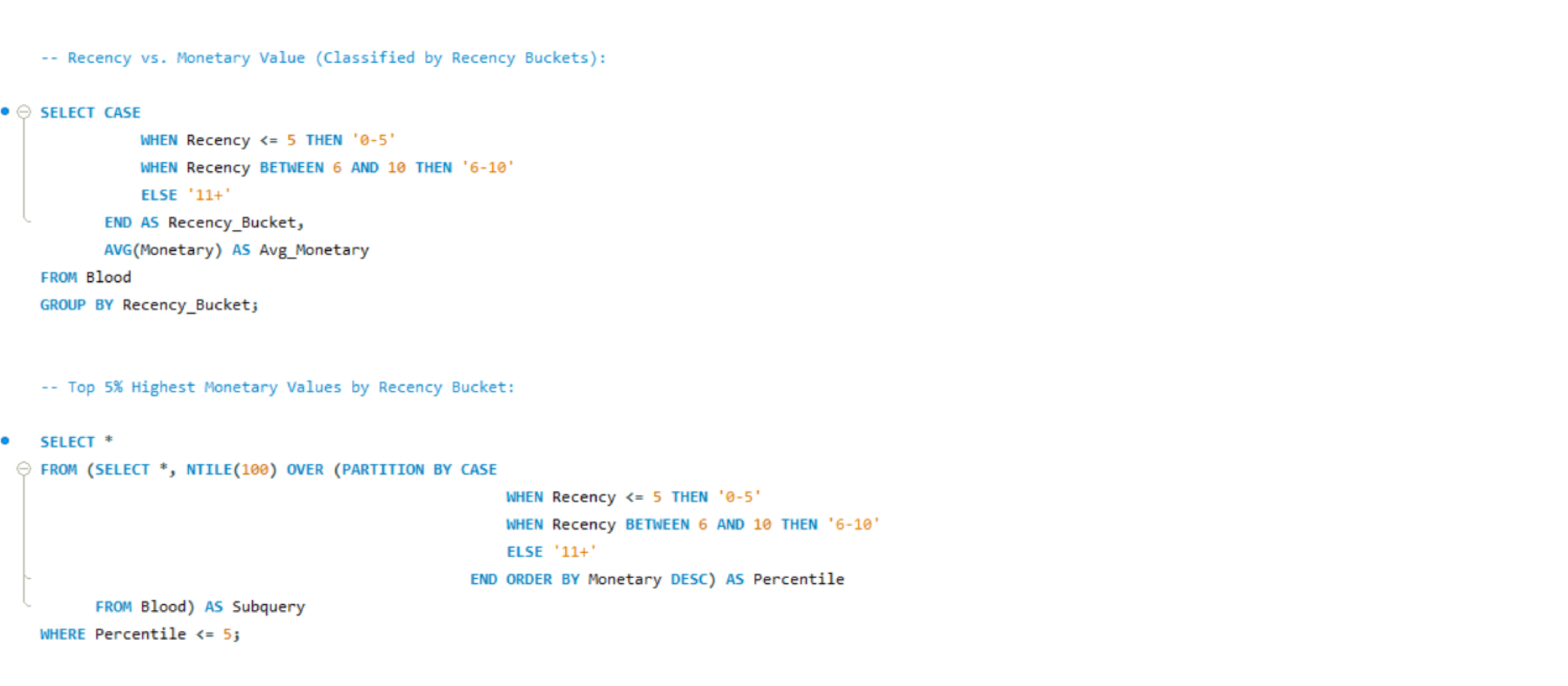
-
  ```sql
  SELECT Time, AVG(Monetary) AS Avg_Monetary
  FROM Blood
  GROUP BY Time;
  ```

```sql
-- Top 3 Months with Highest Average Monetary Value:
```

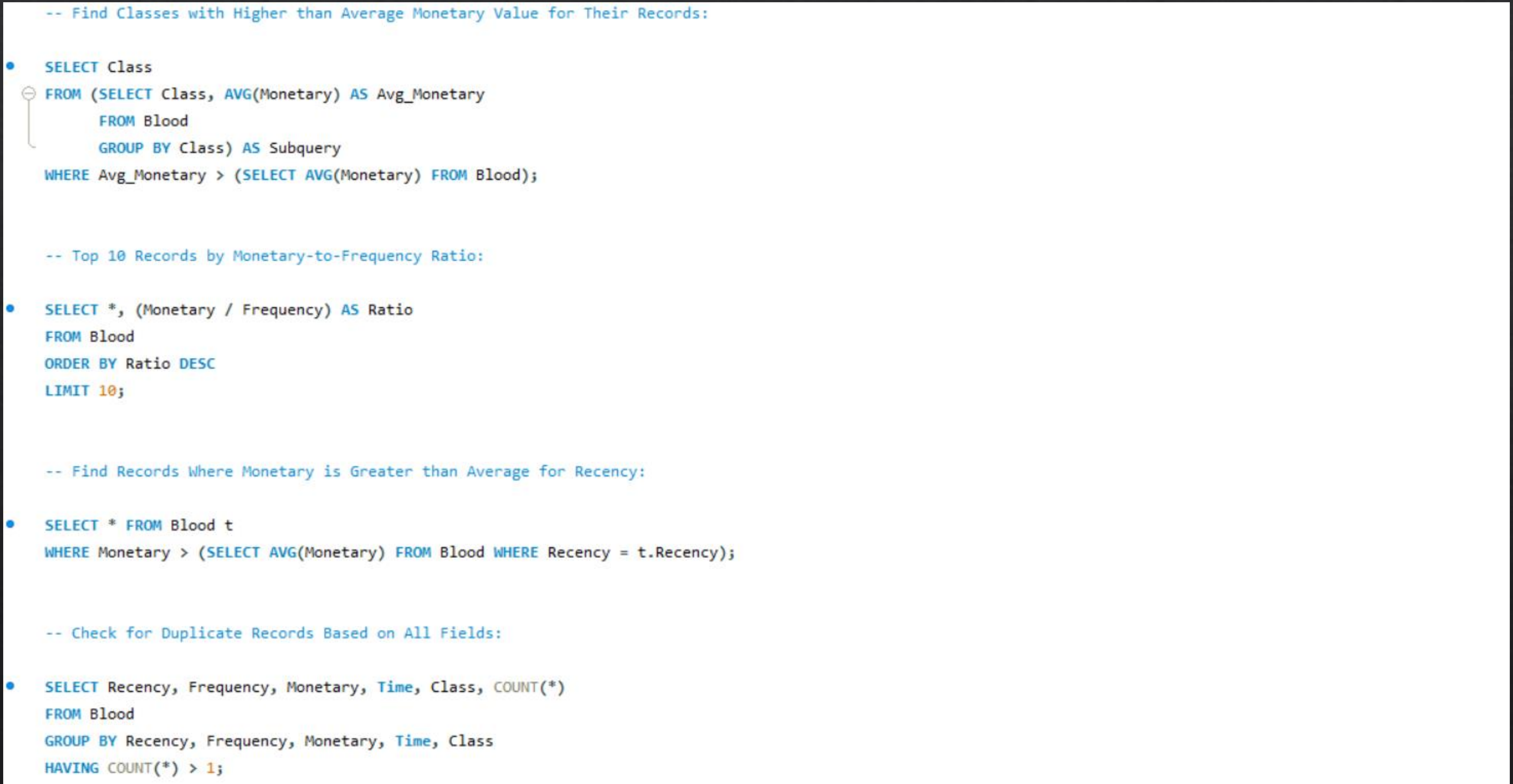
-
  ```sql
  SELECT Time, AVG(Monetary) AS Avg_Monetary
  FROM Blood
  GROUP BY Time
  ORDER BY Avg_Monetary DESC
  LIMIT 3;
  ```

```sql
-- Percentage Change in Monetary Value Month-over-Month:
```

-
  ```sql
  SELECT t1.Time AS Current_Month,
         t2.Time AS Previous_Month,
         ((t1.Avg_Monetary - t2.Avg_Monetary) / t2.Avg_Monetary) * 100 AS Percentage_Change
  FROM (SELECT Time, AVG(Monetary) AS Avg_Monetary FROM Blood GROUP BY Time) t1
  LEFT JOIN (SELECT Time, AVG(Monetary) AS Avg_Monetary FROM Blood GROUP BY Time) t2
  ON t1.Time = t2.Time + 1;
  ```



```sql
-- Recency vs. Monetary Value (Classified by Recency Buckets):

SELECT CASE
           WHEN Recency <= 5 THEN '0-5'
           WHEN Recency BETWEEN 6 AND 10 THEN '6-10'
           ELSE '11+'
       END AS Recency_Bucket,
       AVG(Monetary) AS Avg_Monetary
FROM Blood
GROUP BY Recency_Bucket;


-- Top 5% Highest Monetary Values by Recency Bucket:

SELECT *
FROM (SELECT *, NTILE(100) OVER (PARTITION BY CASE
                                                   WHEN Recency <= 5 THEN '0-5'
                                                   WHEN Recency BETWEEN 6 AND 10 THEN '6-10'
                                                   ELSE '11+'
                                               END ORDER BY Monetary DESC) AS Percentile
      FROM Blood) AS Subquery
WHERE Percentile <= 5;
```



```sql
-- Find Classes with Higher than Average Monetary Value for Their Records:

SELECT Class
FROM (SELECT Class, AVG(Monetary) AS Avg_Monetary
      FROM Blood
      GROUP BY Class) AS Subquery
WHERE Avg_Monetary > (SELECT AVG(Monetary) FROM Blood);


-- Top 10 Records by Monetary-to-Frequency Ratio:

SELECT *, (Monetary / Frequency) AS Ratio
FROM Blood
ORDER BY Ratio DESC
LIMIT 10;


-- Find Records Where Monetary is Greater than Average for Recency:

SELECT * FROM Blood t
WHERE Monetary > (SELECT AVG(Monetary) FROM Blood WHERE Recency = t.Recency);


-- Check for Duplicate Records Based on All Fields:

SELECT Recency, Frequency, Monetary, Time, Class, COUNT(*)
FROM Blood
GROUP BY Recency, Frequency, Monetary, Time, Class
HAVING COUNT(*) > 1;
```

CONCLUSION

In summary, blood datasets offer a wealth of information that contributes to advancing medical research, improving clinical practices, and promoting better health outcomes. They are pivotal for understanding hematological health and for the development of innovative healthcare solutions through data-driven insights.

VISIT FOR DETAILED PROJECT

Click on linkedin logo to visit my Profile

Click on Github logo to visit the portfolio of the project