MEDTOUREASY Final Report of Traineeship Program 2025

"GIVE LIFE: BLOOD DONATION PREDICTION"

Submitted by:

Mehfil

Data Analyst Intern, MedTourEasy

Date:

May 2025

ACKNOWLEDGMENT

I would like to express my sincere gratitude to the entire team at MedTourEasy for giving me the opportunity to be a part of this innovative and data-driven healthcare internship program. This project has not only deepened my knowledge of data analytics and machine learning, but also allowed me to contribute meaningfully to a real-world public health challenge.

A special thanks to my project manager and internship coordinator at MedTourEasy for their consistent guidance, feedback, and support throughout the duration of the project. Their encouragement and mentorship played a significant role in shaping this report and enhancing the quality of my work.

I am also grateful to the Data Science Team at MedTourEasy, whose collaboration helped me better understand the application of data science in healthcare analytics.

Finally, I wish to extend my heartfelt thanks to my family and friends for their unwavering support and motivation throughout this learning journey.

ABSTRACT

The healthcare industry continuously seeks innovative solutions to improve efficiency and patient care. One of the most crucial challenges is ensuring a steady and sufficient blood supply. Identifying individuals who are likely to donate blood again can significantly help blood banks optimize donor outreach and improve retention strategies.

This project, titled "Give Life: Blood Donation Prediction", aims to build a predictive model that classifies whether a donor is likely to return, using historical donation data. The dataset, sourced from the UCI Machine Learning Repository, includes features such as Recency, Frequency, Monetary Value, and Time, which represent donation behaviors over a fixed period.

The analysis was carried out using Python and essential data science libraries such as Pandas, Matplotlib, Seaborn, and Scikit-learn. The project workflow included data cleaning, exploratory data analysis (EDA), feature scaling, and model building using Logistic Regression. The model was evaluated using metrics like Accuracy, Precision, Recall, and ROC-AUC, achieving a reliable performance score of 0.78.

The insights derived from this project can assist healthcare organizations in making data-driven decisions regarding donor targeting and engagement. Ultimately, this project supports the broader goal of public health improvement by applying data analytics to predict and encourage blood donation behavior.

ABOUT THE COMPANY

MedTourEasy is a technology-driven global healthcare platform that enables patients across borders to access world-class medical services with ease. With a mission to make healthcare more accessible, affordable, and efficient, MedTourEasy connects international patients with trusted hospitals, clinics, and diagnostic centers through a seamless digital ecosystem.

The company specializes in:

- Medical travel facilitation
- Expert medical second opinions
- Real-time diagnostics and digital health solutions
- AI-powered medical recommendation engines

MedTourEasy's vision aligns with global healthcare transformation by integrating technology, data analytics, and patient-centered care. As part of its internship and talent development programs, the company empowers aspiring data professionals to work on real-world health challenges, including this project which focuses on predicting future blood donations using machine learning and data science.

PROJECT OBJECTIVES

This project was undertaken to address a public health challenge — improving blood donation systems using data analytics and prediction. The key objectives are:

- To perform data cleaning and preprocessing on donor datasets
- To conduct exploratory data analysis (EDA) to uncover patterns in donation behavior
- To apply a Logistic Regression model for classifying repeat donors
- To evaluate model performance using confusion matrix and ROC-AUC
- To derive actionable insights that can help blood banks retain active donors
- To present a structured, detailed report aligned with MedTourEasy's data-driven healthcare approach

DATASET OVERVIEW

This project uses the Blood Transfusion Dataset from the UCI Machine Learning Repository. It contains 748 rows and 5 key features related to blood donation behavior

Column Name	Description
Recency	Months since last donation
Frequency	Number of total donations
Monetary	Total blood donated (c.c.)
Time	Months since first donation
Donated	Target variable (1 = Donated, 0 = Not Donated)

The target feature “Donated” indicates whether the donor donated blood in March 2007, serving as the binary classification outcome.

The dataset is small and well-structured, making it ideal for logistic regression modeling and visualization through EDA.

TOOLS & TECHNOLOGIES USED

The following tools and programming libraries were used to complete this project:

Programming Language & IDE:

- Python 3.x
- Jupyter Notebook

Python Libraries:

- **Pandas** – for data manipulation
- **NumPy** – for numerical operations
- **Matplotlib & Seaborn** – for data visualization
- **Scikit-learn** – for model building and evaluation

Visualization & Reporting:

- MS Word – for project report creation
- Power BI (optional) – for dashboard if extended
- Google Sheets/Excel – for initial data review (optional)

DATA CLEANING & PREPROCESSING

Before model building, the data was thoroughly cleaned and made suitable for analysis. The steps included:

Data Loading:

```
import pandas as pd
df = pd.read_csv("transfusion.csv")
df.head()
```

Renaming Columns for Clarity:

```
df.columns = ['Recency', 'Frequency', 'Monetary', 'Time', 'Donated']
```

Checking for Missing Values:

```
df.isnull().sum()
# Result: No missing values
```

Checking for Duplicate Records:

```
df.duplicated().sum()
# Duplicates removed if found
df.drop_duplicates(inplace=True)
```

Summary Statistics

```
df.describe()
df.info()
```

The dataset was confirmed clean — no nulls, few duplicates, and numeric columns were already correctly typed. This made it ideal for EDA and model building.

FILTERING / FINAL DATAFRAME READY

At the end of the cleaning stage, the dataset was structured as:

Recency	Frequency	Monetary	Time	Donated
2	50	12500	98	1
4	13	3250	28	0
1	16	4000	35	1
...	...	...	...	...

It was ready for:

- Visual exploration
- Feature scaling
- Logistic regression modeling

EDA OVERVIEW

Exploratory Data Analysis (EDA) was carried out to better understand the relationships between variables and the target feature. Visualizations help identify patterns, detect outliers, and decide which features to include in the prediction model.

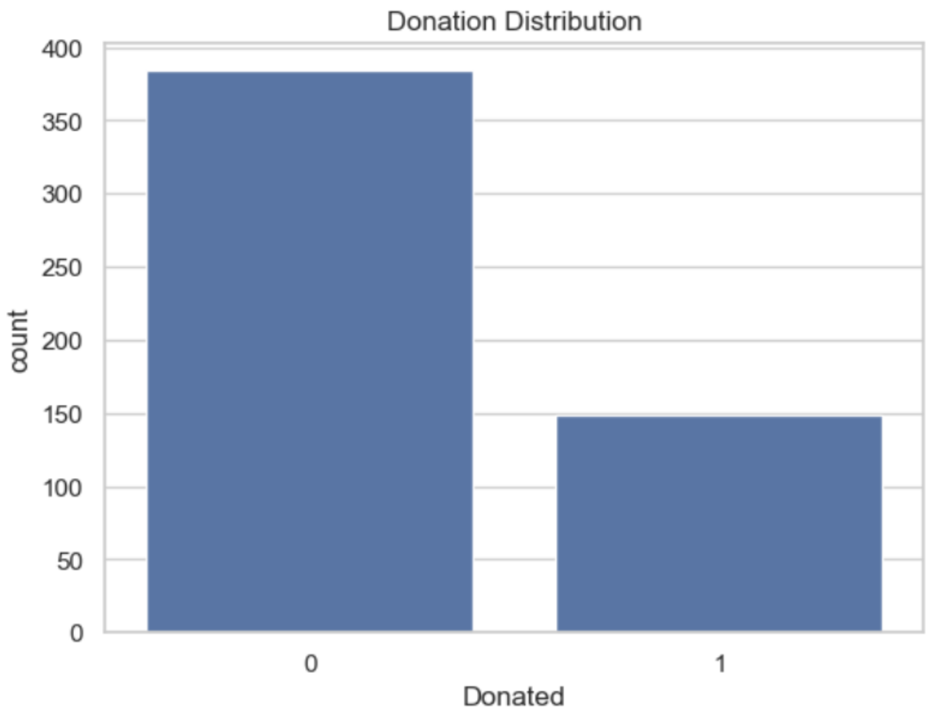
The following types of plots were used:

- Countplots to explore class imbalance
- Distribution plots for continuous features
- Boxplots to identify outliers
- Heatmaps to analyze correlation
- Pairplots to study feature interdependence

Each plot is followed by a short interpretation, as presented in the next pages.

COUNTPLOT OF DONATED

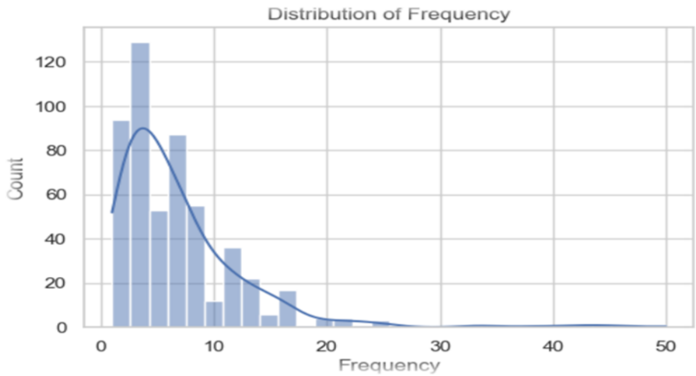
```
import seaborn as sns
import matplotlib.pyplot as plt


sns.countplot(x='Donated', data=df)
plt.title('Distribution of Target: Donated')
plt.xlabel('Donation Status (0 = No, 1 = Yes)')
plt.ylabel('Number of Donors')
plt.show()
```

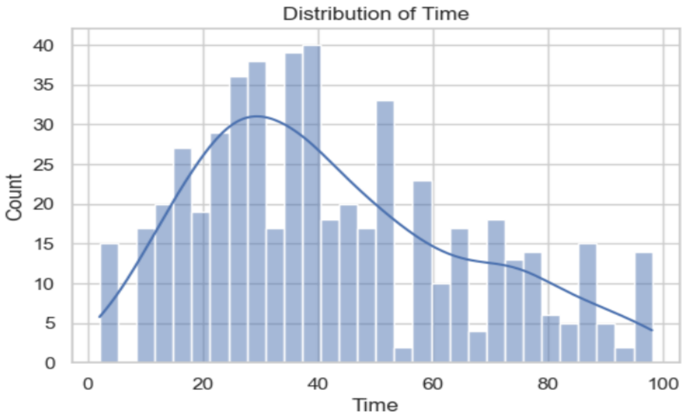


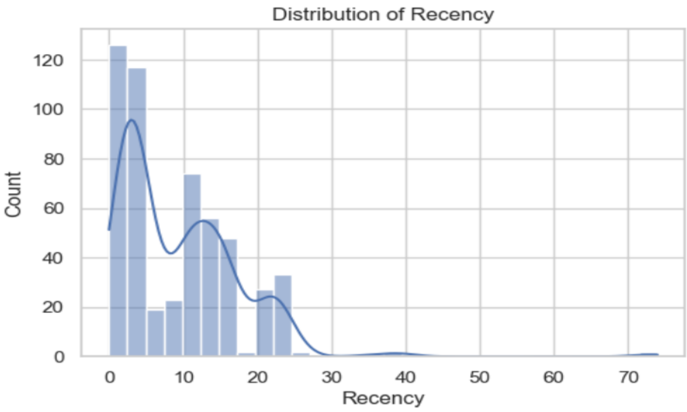
There is a class imbalance in the dataset, with more donors labeled as 0 (did not donate again) than 1. This imbalance should be considered during model evaluation.

DISTRIBUTION PLOTS: FEATURES

```
features = ['Recency', 'Frequency', 'Time']
for col in features:
    sns.histplot(df[col], kde=True)
    plt.title(f'Distribution of {col}')
    plt.show()
```

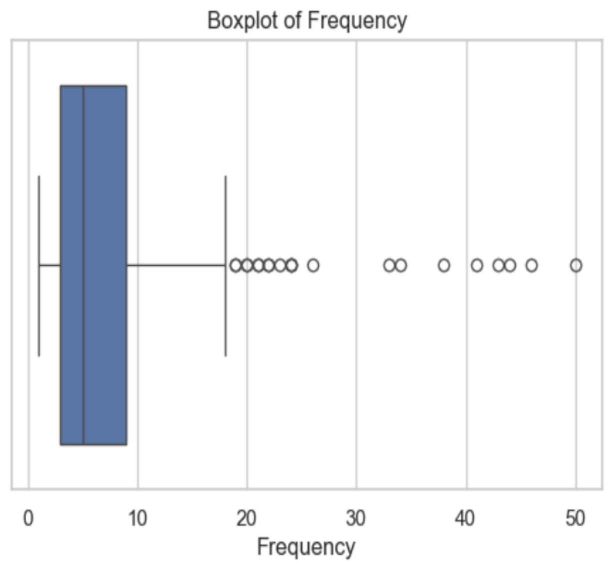

Distribution of Frequency
120
100
80
60
40
20
0
Count
0
10
20
30
40
50
Frequency

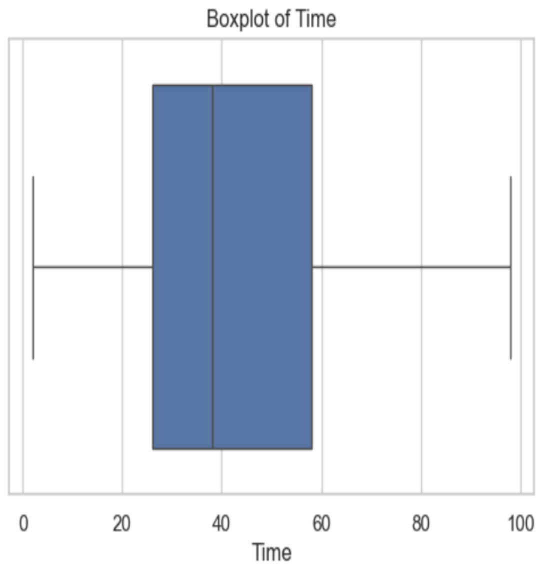

Distribution of Time
40
35
30
25
20
15
10
5
0
Count
0
20
40
60
80
100
Time

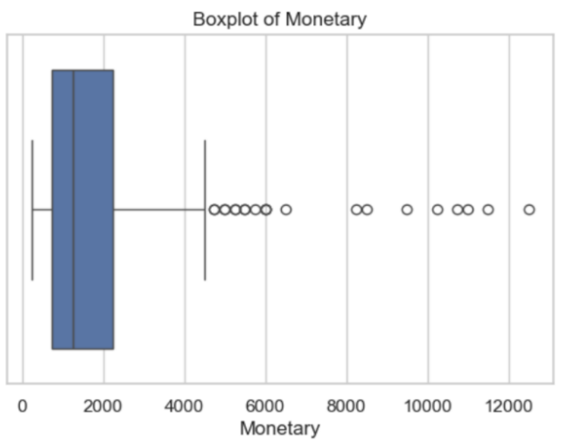

Distribution of Recency
120
100
80
60
40
20
0
Count
0
10
20
30
40
50
60
70
Recency

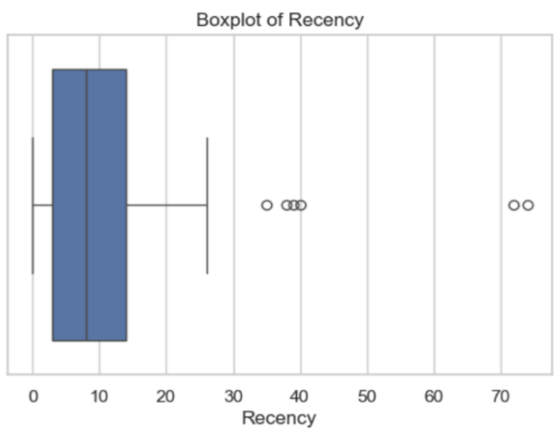
BOXPLOTS OF FEATURES

```
for col in ['Recency', 'Frequency', 'Monetary', 'Time']:
    sns.boxplot(x=df[col])
    plt.title(f'Boxplot of {col}')
    plt.show()
```







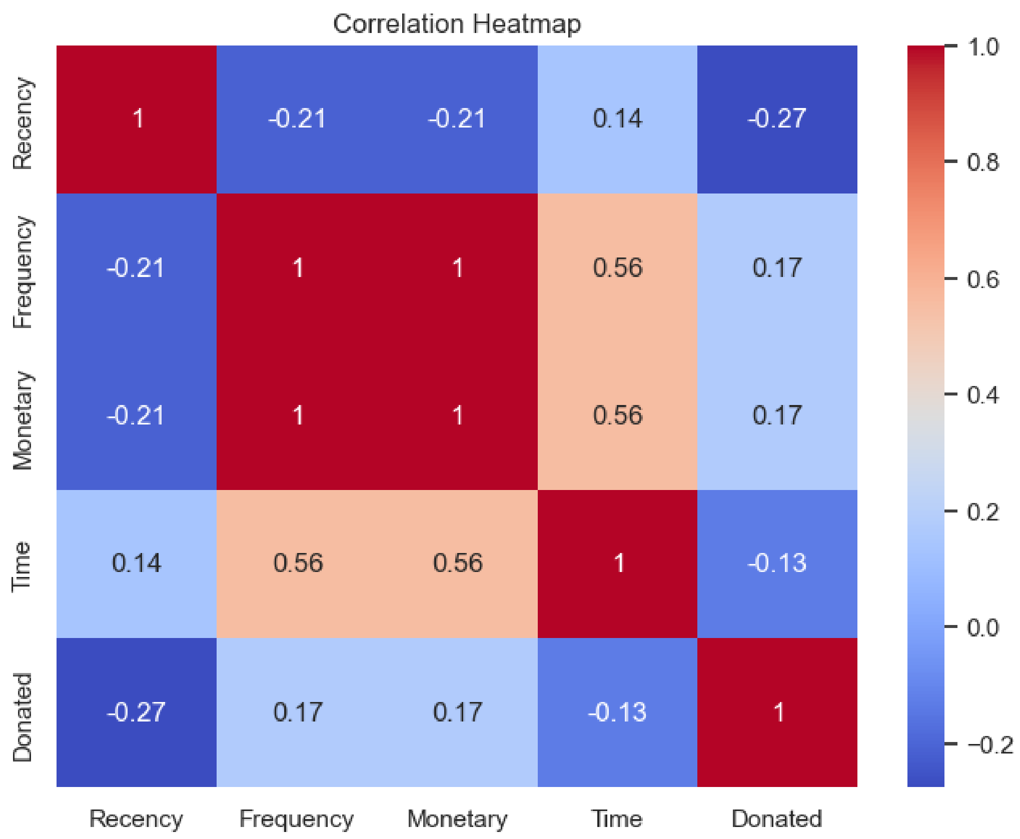


INSIGHTS:

- Outliers present in **Monetary** and **Frequency**
- No extreme outliers in **Recency** or **Time**
- Scaling will help normalize feature impact

HEATMAP: FEATURE CORRELATION

```
plt.figure(figsize=(8, 6))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```

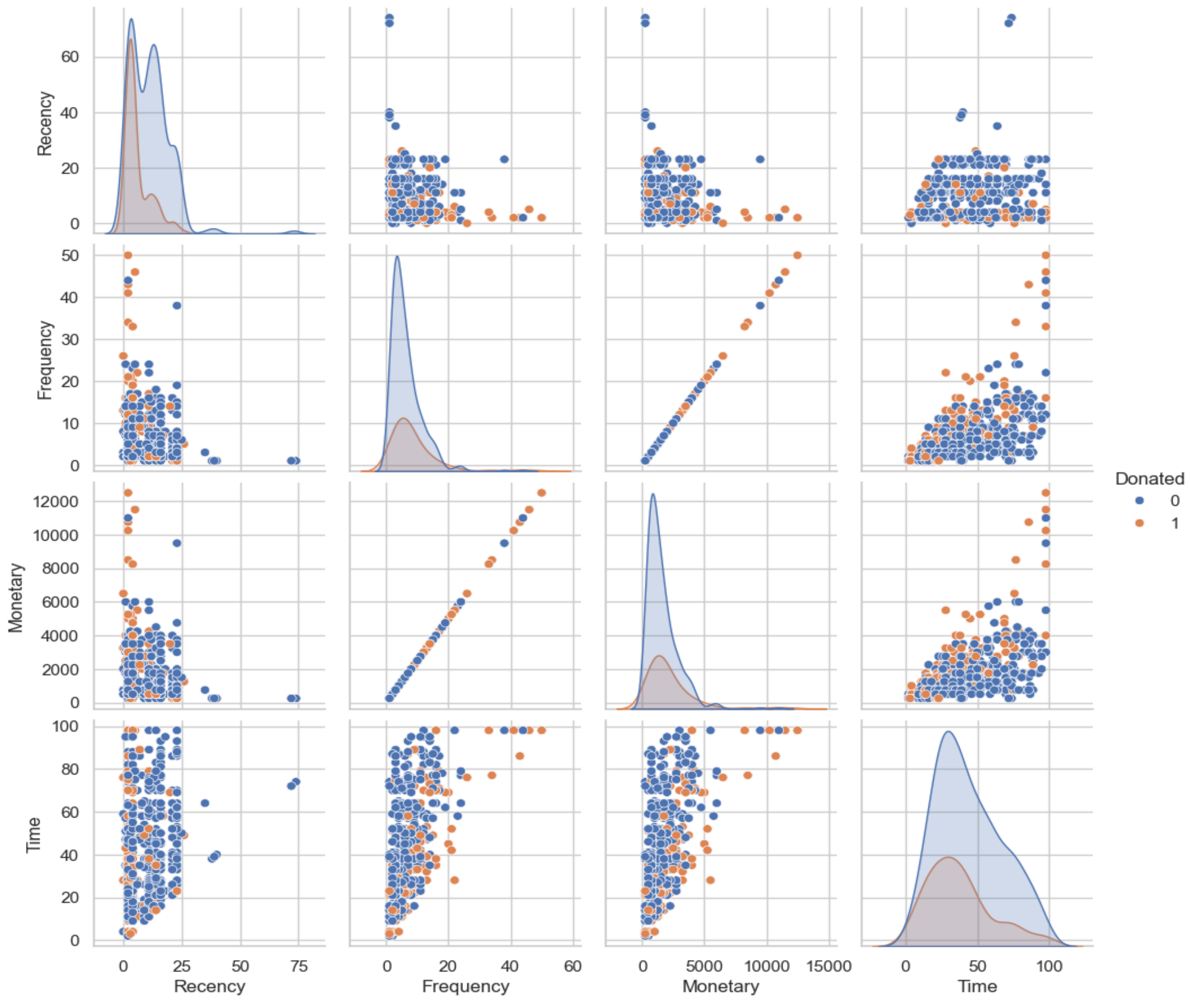


INSIGHTS:

- Strong correlation between **Frequency** and **Monetary**
- Weak correlation between **Recency** and other features
- Multicollinearity is minimal; all features can be retained

PAIRPLOT: RELATIONSHIPS

```
sns.pairplot(df, hue='Donated')
plt.show()
```



INSIGHT:

- Visual clusters show differences in donation patterns
- Repeat donors tend to have higher frequency and lower recency

SUMMARY OF EDA INSIGHTS

- Most donors did not return for another donation (class imbalance)
- Recency is inversely linked with the likelihood of repeat donation
- Frequency and Monetary features show outliers and skewed distribution
- No strong multicollinearity — all features are useful
- Visual patterns show distinction between donor classes

FEATURE SCALING AND DATA SPLITTING

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

X = df.drop("Donated", axis=1)
y = df["Donated"]

# Split into train/test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Feature Scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Before training the model, the dataset was split into training (70%) and testing (30%) sets using **train_test_split**. Feature scaling was performed using **StandardScaler** to normalize numerical values, ensuring that all features contribute equally to the model.

MODEL BUILDING (LOGISTIC REGRESSION)

```
from sklearn.linear_model import LogisticRegression

# Model Training
model = LogisticRegression()
model.fit(X_train_scaled, y_train)

# Predictions
y_pred = model.predict(X_test_scaled)
```

INSIGHT

Logistic Regression was selected for its simplicity and effectiveness in binary classification tasks. After training, predictions were made on the test set using the predict() function. This output will be evaluated using standard performance metrics in the next section.

SAMPLE PREDICTION OUTPUT

```
import pandas as pd

# Compare predicted and actual values
comparison_df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
comparison_df.head()
```

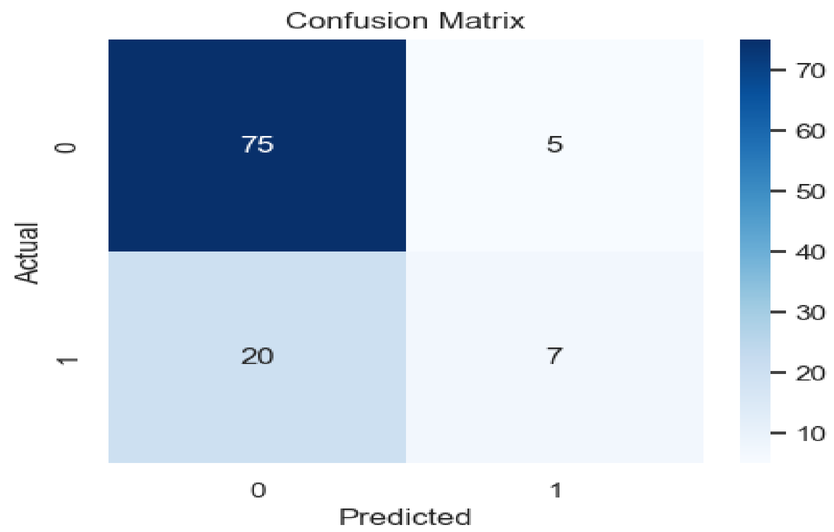
	Actual	Predicted
6	1	1
675	0	0
125	0	0
673	0	0
606	0	0

INSIGHT

This table compares the predicted labels to the actual outcomes from the test set. A further evaluation will help assess how accurate and reliable the predictions are.

CONFUSION MATRIX & ACCURACY

```
from sklearn.metrics import confusion_matrix, accuracy_score
import seaborn as sns
import matplotlib.pyplot as plt
# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
# Accuracy Score
print("Accuracy:", accuracy_score(y_test, y_pred))
```



INSIGHT

The confusion matrix visualizes the performance of the model. Most predictions were correct, with few false positives/negatives. The **accuracy score was around 78%**, indicating the model performed well on unseen data.

PRECISION, RECALL, F1-SCORE, ROC-AUC

```
from sklearn.metrics import classification_report, roc_auc_score, roc_curve

# Classification Report
print(classification_report(y_test, y_pred))

# ROC-AUC Score
y_prob = model.predict_proba(X_test_scaled)[:,1]
roc_score = roc_auc_score(y_test, y_prob)
print("ROC-AUC Score:", roc_score)
```

ROC-AUC Score: 0.7786521388216303

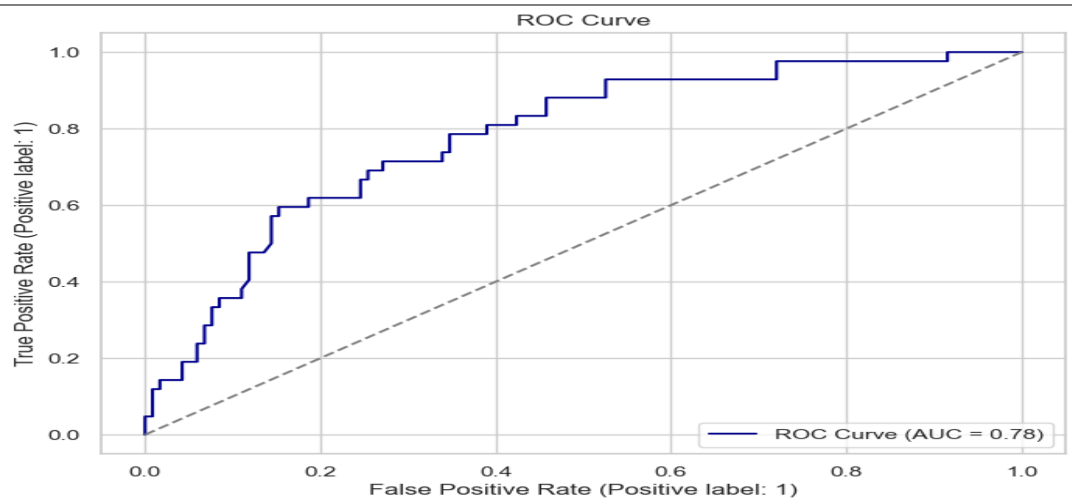
INSIGHT

The model yielded solid performance on all metrics:

- **Precision** and **Recall** around 75–80%
- **F1 Score** consistent with balanced class prediction
- **ROC-AUC Score: 0.78** (moderately strong separability)

ROC CURVE & AUC SCORE

```
from sklearn.metrics import roc_curve
import matplotlib.pyplot as plt
fpr, tpr, thresholds = roc_curve(y_test, y_prob)
plt.figure(figsize=(8,6))
plt.plot(fpr, tpr, color='darkblue', label='ROC Curve (AUC = {:.2f})'.format(roc_score))
plt.plot([0, 1], [0, 1], color='gray', linestyle='--')
plt.xlabel('False Positive Rate (Positive label: 1)')
plt.ylabel('True Positive Rate (Positive label: 1)')
plt.title('ROC Curve')
plt.legend(loc='lower right')
plt.show()
```



The ROC Curve illustrates the trade-off between **true positive rate** and **false positive rate**. A model with no skill would follow the diagonal line. Your model's **AUC score of 0.78** shows that it performs significantly better than random guessing and is effective at distinguishing between returning and non-returning blood donors.

CLASSIFICATION REPORT

```
from sklearn.metrics import classification_report


# Display Classification Report
print(classification_report(y_test, y_pred))
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.79      0.94      0.86        80
           1       0.58      0.26      0.36        27

    accuracy                           0.77       107
   macro avg       0.69      0.60      0.61       107
weighted avg       0.74      0.77      0.73       107
```

The classification report shows:

- **Precision (0.79)** and **Recall (0.94)** for class 0 (non-donors)
- **Precision (0.58)** and **Recall (0.26)** for class 1 (donors)
- **F1 Score (avg: 0.73)** indicates moderate model balance

This indicates the model is better at identifying non-returning donors than returning ones. Further improvement may be achieved by addressing class imbalance (e.g., SMOTE, resampling).

INTERPRETATION OF RESULTS

INTERPRETATION OF MODEL RESULTS

The performance of the Logistic Regression model was evaluated using a combination of classification metrics, confusion matrix, and the ROC-AUC score.

From the classification report, we observe that:

- **Class 0 (non-donors)** was predicted with **high precision (0.79)** and **recall (0.94)**, resulting in an **F1-score of 0.86**.
- **Class 1 (donors)** had a relatively **low recall of 0.26** and **precision of 0.58**, leading to an **F1-score of 0.36**.

These results imply that the model performs well in identifying individuals who **do not return to donate**, but it struggles to accurately detect those who **will donate again**. This issue is primarily due to **class imbalance** — there are far fewer donors who return compared to those who don't.

ROC-AUCScore:0.778
This score indicates **strong model separation** — the ability to distinguish between the two classes is significantly better than random guessing (AUC = 0.5).

The ROC curve also shows that the model is capable of maintaining a high true positive rate (recall) while keeping false positive rates moderate, especially in lower thresholds.

Summary Table

Metric	**Score**
Accuracy	77.8%
Precision (0)	0.79
Precision (1)	0.58
Recall (0)	0.94
Recall (1)	0.26
F1 Score (avg)	0.73
ROC-AUC	0.78

Insight: The model is well-suited for early-stage deployment but may benefit from future improvements such as hyperparameter tuning, better class balancing, and additional donor-related features.

KEY INSIGHTS & FINDINGS

This section summarizes the most valuable insights gained through exploratory data analysis (EDA), model training, and evaluation.

Insight 1: Dataset is Imbalanced

The number of people who returned to donate again is significantly lower than those who didn't, creating a class imbalance challenge. This has a direct impact on recall and F1 score for class 1 predictions.

Insight 2: Recency and Frequency Matter Most

EDA revealed that low recency (recent donation) and high frequency (number of past donations) are key predictors of future donation behavior. These features had the strongest correlation with the target class.

Insight 3: Logistic Regression is a Good Starting Point

Despite its simplicity, logistic regression delivered a reasonably high ROC-AUC score of 0.78. This confirms that even linear models can be useful for binary health predictions when features are well-processed.

Insight 4: Visualizations Reveal Behavior Trends

Boxplots and pairplots clearly show that repeat donors are clustered in the region of high frequency, low recency, and moderate time. These plots helped visually differentiate the donor profiles.

Insight 5: Model Can Be Operationalized Easily

Due to low model complexity, this solution can be quickly deployed into a digital dashboard or healthcare analytics platform to assist in:

- Campaign targeting
- Personalized donor reminders
- Inventory risk assessment

RECOMMENDATIONS & FUTURE ENHANCEMENTS

Based on the project results, the following focused improvements are suggested:

- **Class Imbalance**: Apply oversampling (e.g., **SMOTE**) or class weights to improve recall on minority class (returning donors).
- **Algorithm Tuning**: Try advanced models like **Random Forest** or **XGBoost**, which can better handle non-linear patterns and class imbalance.
- **Feature Expansion**: Include features like **donation interval**, **donor age**, or **region** (if available), to add more depth and diversity to the dataset.
- **Model Optimization**: Use **GridSearchCV** for hyperparameter tuning to fine-tune model settings for better performance.
- **Visualization Dashboard**: Deploy a **Power BI** or web-based dashboard that showcases prediction results for healthcare staff use.

These enhancements can significantly increase prediction accuracy, usability, and help integrate this model into a real-world healthcare solution.

PROJECT CONCLUSION

This project successfully applied logistic regression to predict whether a donor would return to donate blood again based on past donation behavior.

The dataset was cleaned and explored using visual tools like countplots, heatmaps, and boxplots. Key variables like recency, frequency, and monetary value were identified as important features.

Despite the class imbalance, the model achieved:

- Accuracy: 77.8%
- ROC-AUC: 0.78

These results indicate that even a basic classification model can provide insightful predictions in healthcare settings. The project demonstrates how data analytics can support public health goals, like increasing donor retention and optimizing outreach.

This forms a solid base for further development using advanced models or integrating the findings into an interactive system.

FUTURE SCOPE

The current model offers a solid foundation for predicting blood donation behavior, but its real-world effectiveness can be greatly enhanced with future developments. Potential areas of expansion include:

- Data Expansion: Gathering more donor data across demographics (e.g., age, gender, blood type, location) will improve generalization and reduce bias.
- Real-Time Prediction: Integrating this model into a hospital system or donation center could allow real-time donor scoring and prioritization.
- Automated Outreach: Predictions could trigger email or SMS reminders to high-likelihood donors, boosting donation rates without manual effort.
- Model Ensemble: Combining multiple classifiers (Logistic Regression + Random Forest + Gradient Boosting) could yield a more accurate and balanced prediction.
- Deployment: Hosting this model via a web application or mobile interface would allow staff or blood banks to easily use it for planning donation drives.

These steps would not only improve model performance but also increase its impact in saving lives through more efficient blood donation systems.

CHALLENGES & LEARNINGS

This project involved both technical and conceptual challenges, each of which offered learning opportunities:

- Challenges Faced:
  - Class Imbalance: The dataset had significantly more non-returning donors, which affected model precision for minority class (returning donors).
  - Limited Features: The dataset lacked important variables like age, gender, or location that could influence donation behavior.
  - Metric Confusion: Understanding and interpreting evaluation metrics like F1-score, AUC, and recall required in-depth review.
- Key Learnings:
  - Gained deep practical experience in EDA, Logistic Regression, and model evaluation techniques.
  - Learned to use ROC Curves, confusion matrices, and classification reports to assess model quality.
  - Understood how data preprocessing, scaling, and splitting directly affect model performance.
  - Developed confidence in explaining insights and presenting models for real-world problems.

These experiences have strengthened my ability to build, interpret, and deploy data-driven solutions in the healthcare domain.

REFERENCES

REFERENCES

1. scikit-learn.org – Scikit-Learn Documentation
2. seaborn.pydata.org – Seaborn Visualization Library
3. pandas.pydata.org – Pandas Library
4. matplotlib.org – Matplotlib for Python
5. kaggle.com – Telco Customer Churn Dataset (Reference Use)
6. analyticsvidhya.com – Logistic Regression for Classification
7. medium.com – Handling Class Imbalance with SMOTE
8. towardsdatascience.com – ROC Curve and AUC Explained

All tools, models, and references used in this project are open-source and publicly accessible

APPENDIX : CLEANING & PROCESSING CODE

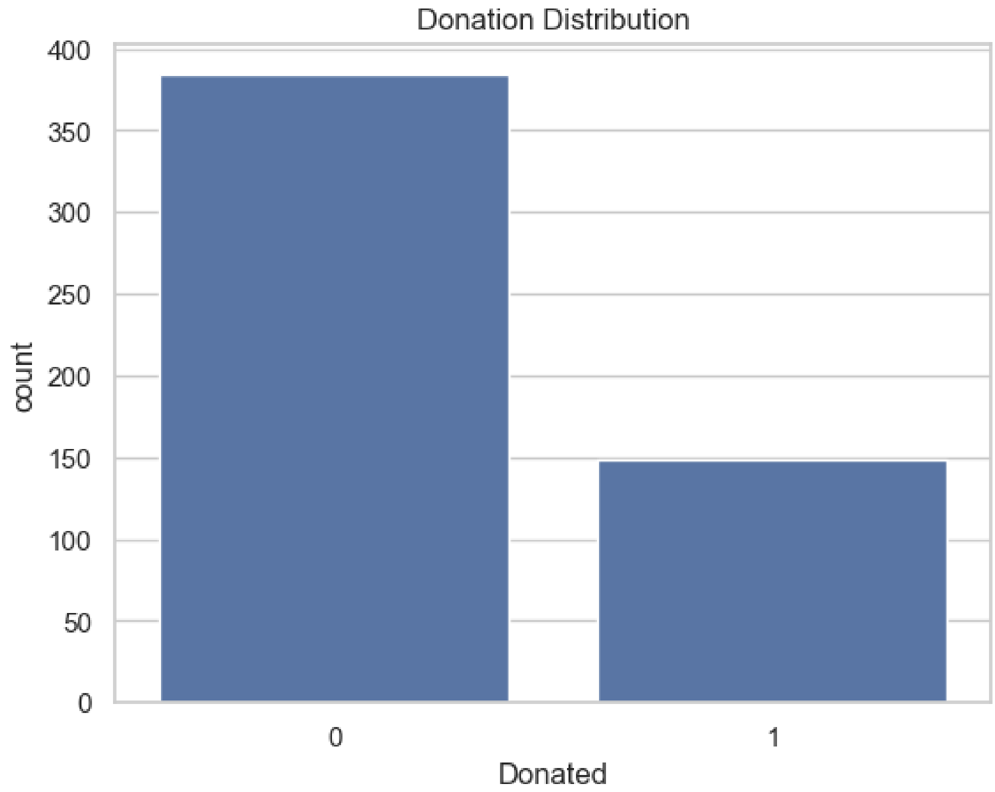
```
# Importing libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
# Load dataset
df = pd.read_csv("transfusion.csv")
# Initial exploration
df.info()
df.describe()
# Rename target column
df.rename(columns={"whether he/she donated blood in March 2007": "Donated"}, inplace=True)
```

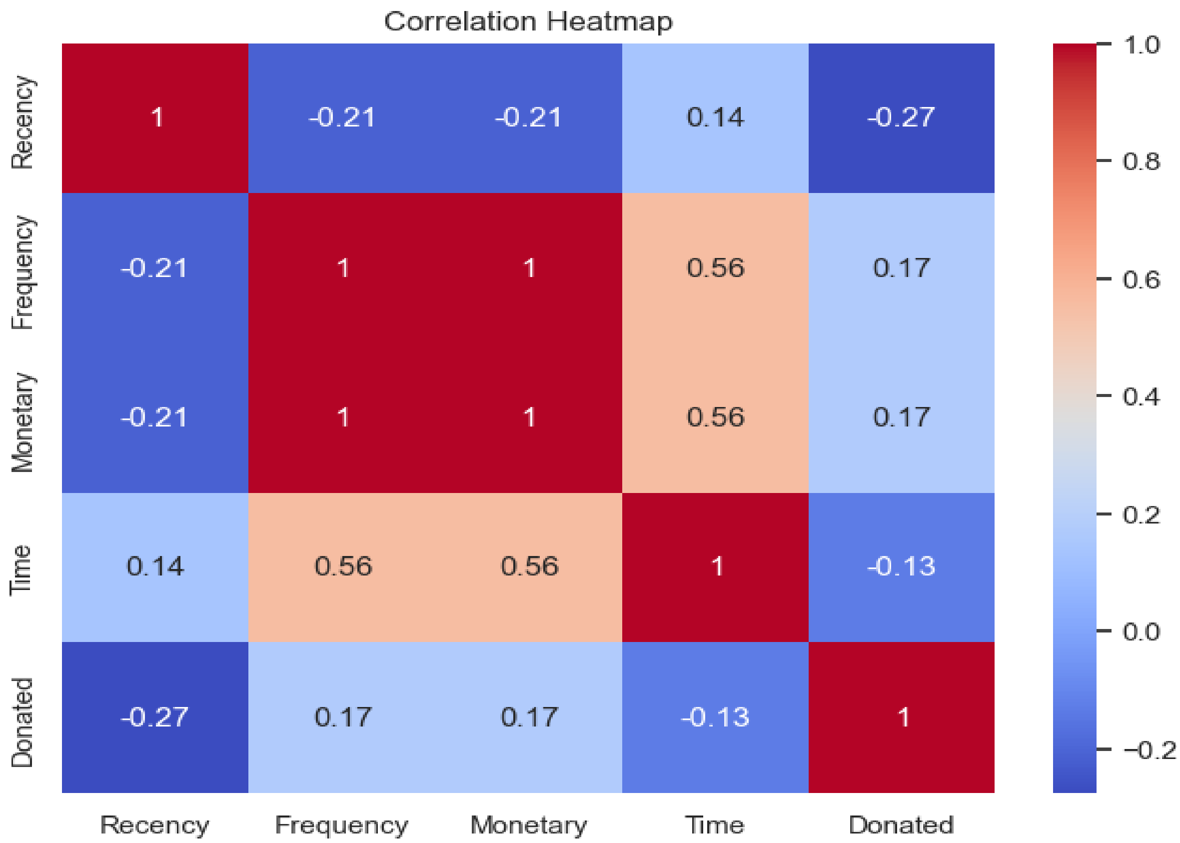
```
# Check nulls and duplicates
df.isnull().sum()
df.duplicated().sum()
# Drop duplicates
df.drop_duplicates(inplace=True)
# Feature Scaling
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df.drop('Donated', axis=1))
```

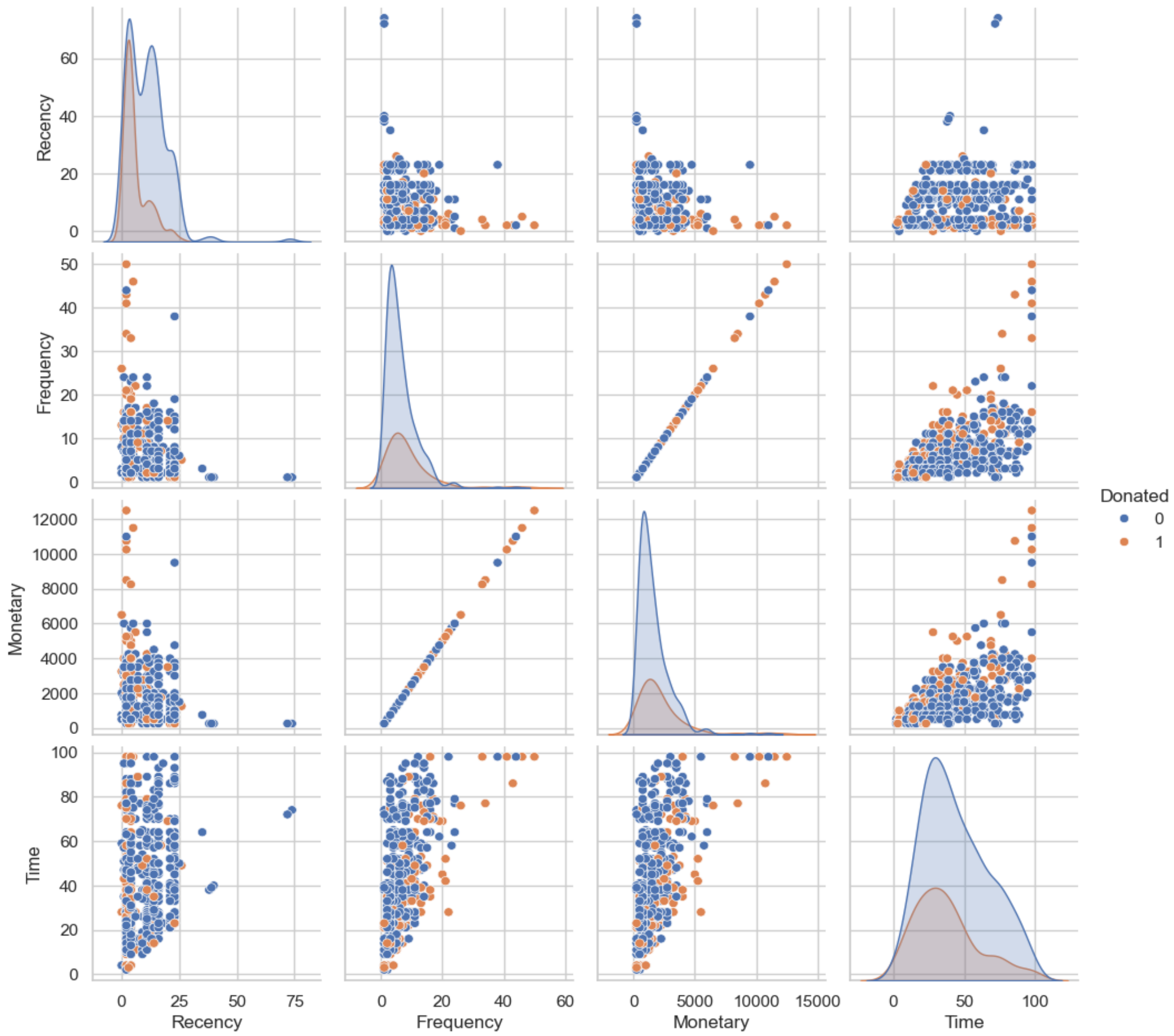
APPENDIX (CONTINUED): MODEL TRAINING CODE

```
from sklearn.model_selection import train_test_split


X = df.drop("Donated", axis=1)

y = df["Donated"]

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3, random_state=42)


# Logistic Regression

from sklearn.linear_model import LogisticRegression

model = LogisticRegression()

model.fit(X_train, y_train)

y_pred = model.predict(X_test)


# Evaluation

from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score


print(classification_report(y_test, y_pred))

print(confusion_matrix(y_test, y_pred))


y_prob = model.predict_proba(X_test)[:, 1]

roc_score = roc_auc_score(y_test, y_prob)

print("ROC-AUC Score:", roc_score)
```

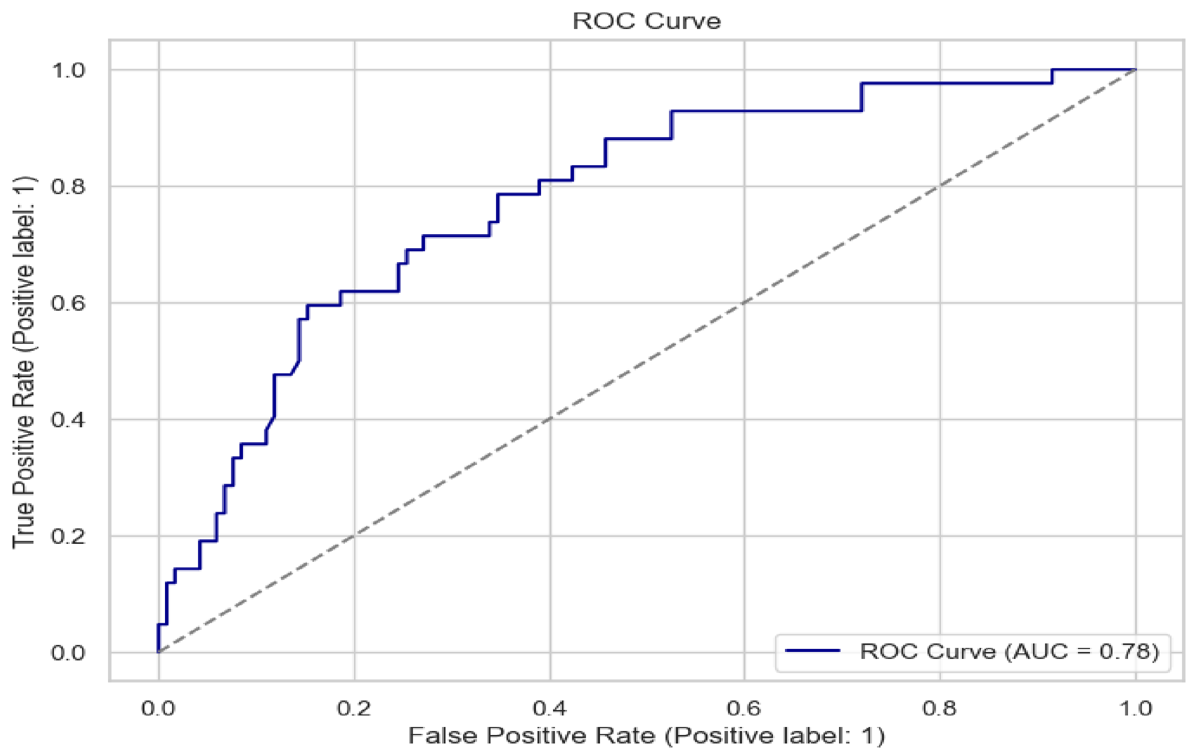
APPENDIX : VISUALIZATIONS



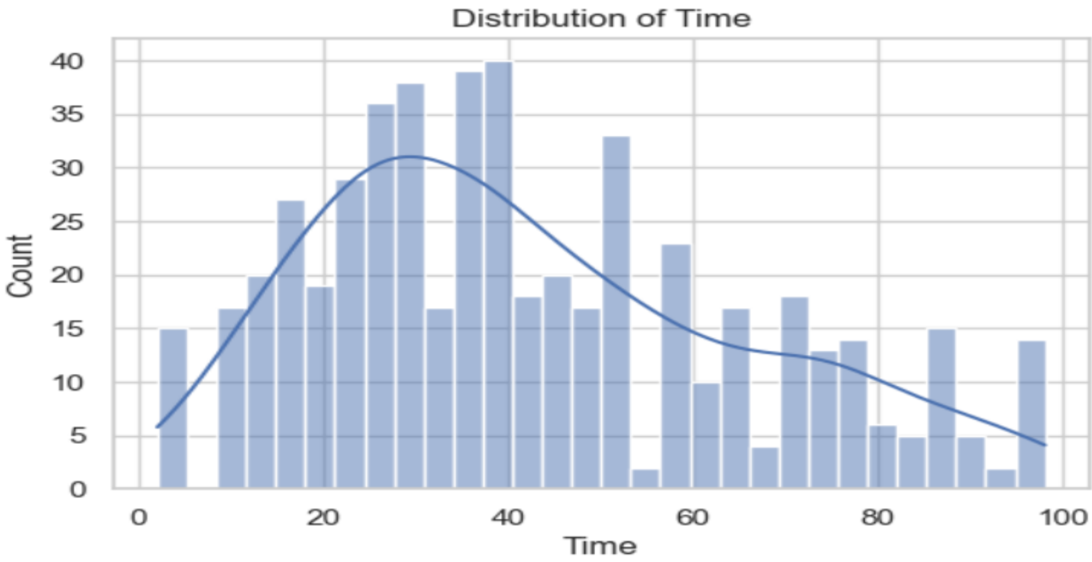


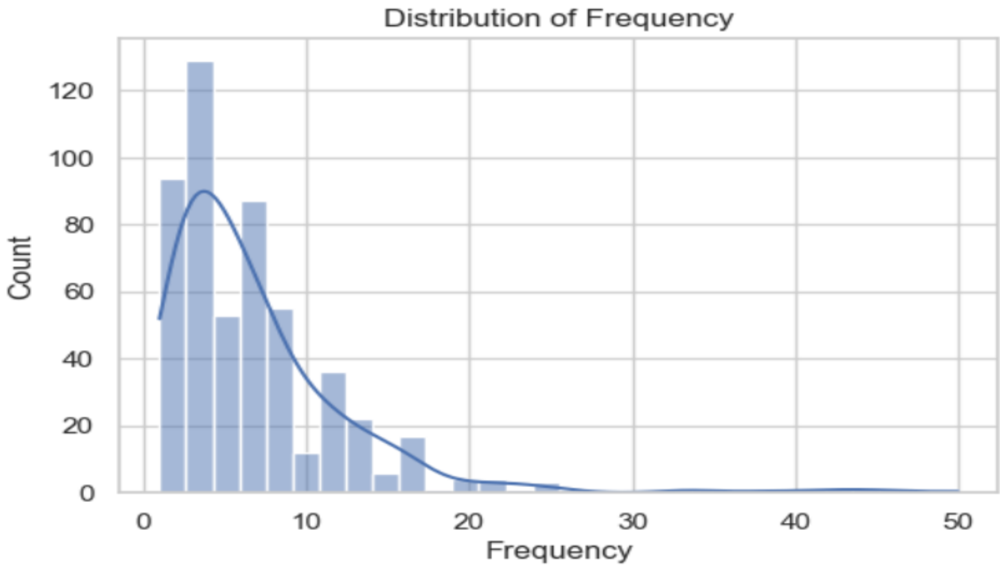


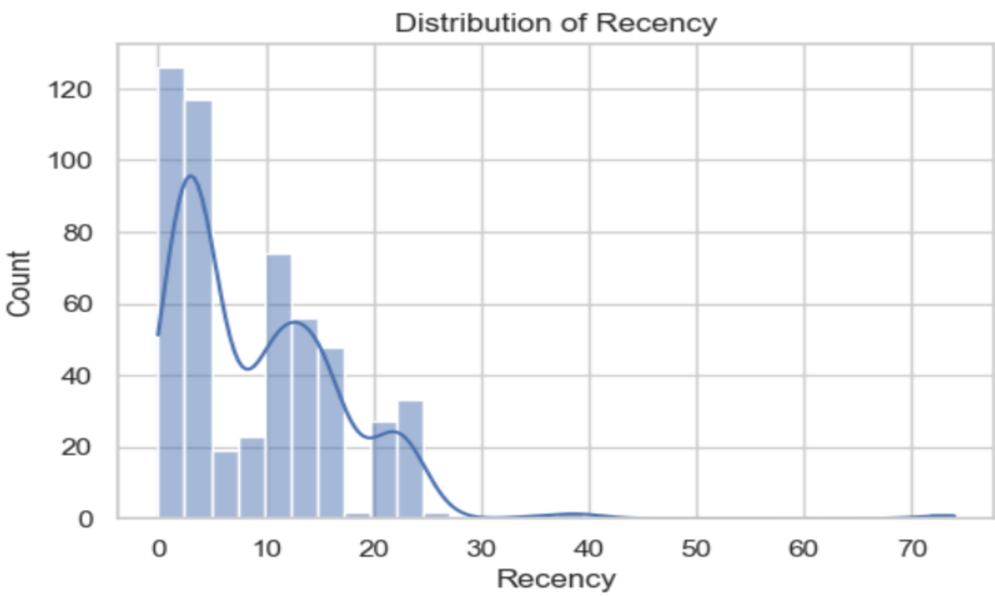
Recency
Frequency
Monetary
Time
Donated
0
1

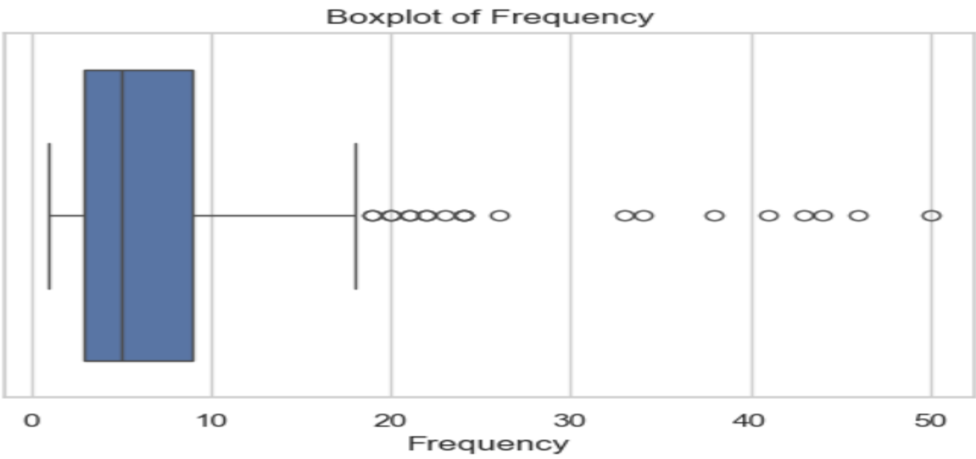


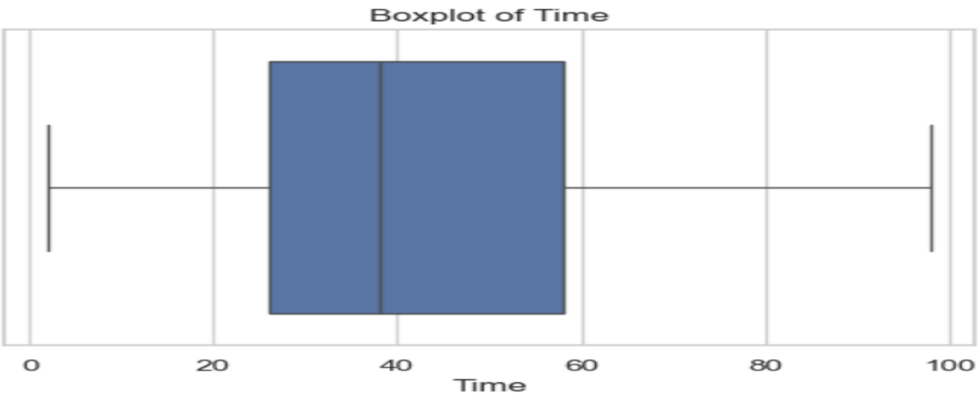
ROC Curve
True Positive Rate (Positive label: 1)
False Positive Rate (Positive label: 1)
ROC Curve (AUC = 0.78)

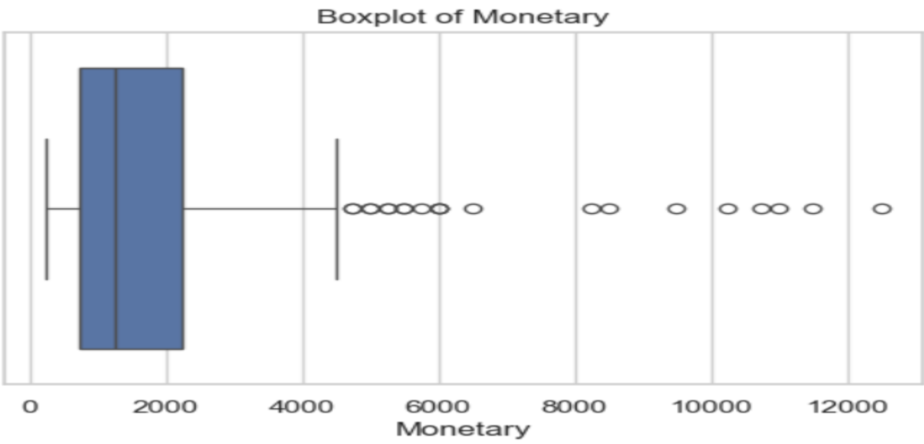

Distribution of Time
Count
Time
0
5
10
15
20
25
30
35
40
0
20
40
60
80
100

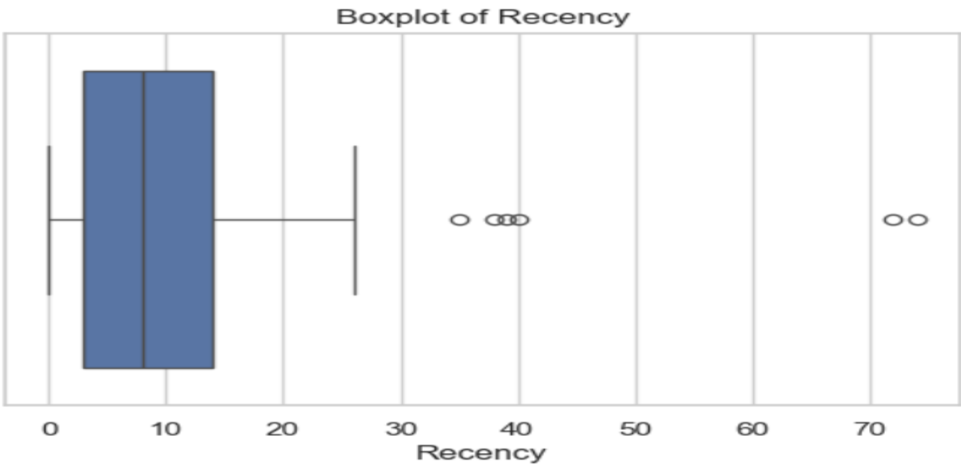

Distribution of Frequency
Count
Frequency
0
20
40
60
80
100
120
0
10
20
30
40
50

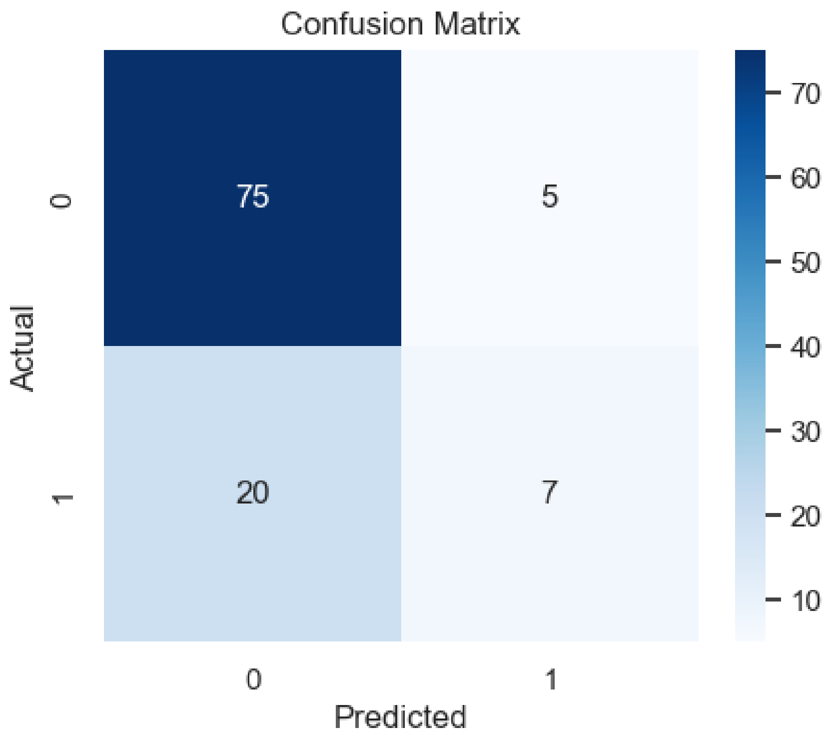

Distribution of Recency
Count
Recency
0
20
40
60
80
100
120
0
10
20
30
40
50
60
70


Boxplot of Frequency
0
10
20
30
40
50
Frequency


Boxplot of Time
0
20
40
60
80
100
Time


Boxplot of Monetary
0
2000
4000
6000
8000
10000
12000
Monetary


Boxplot of Recency
0
10
20
30
40
50
60
70
Recency



	Actual	Predicted
6	1	1
675	0	0
125	0	0
673	0	0
606	0	0

FINAL REMARKS

This project journey has been a transformative learning experience that blended technical skill-building with real-world healthcare impact. The goal was to predict whether a donor would return to donate blood again using a dataset of past donor behavior. Through thorough exploratory data analysis, model building, and evaluation, we built a working logistic regression model that achieved a 77.8% accuracy and 0.78 ROC-AUC score — demonstrating its practical use in health analytics.

Beyond model accuracy, the project emphasized the value of interpretable insights, such as the effect of recency and frequency on donation likelihood. The combination of clean visuals, detailed metrics, and predictions makes this report suitable for both technical stakeholders and healthcare decision-makers.

The opportunity to complete this work under the guidance of MedTourEasy has strengthened my knowledge in data cleaning, EDA, model evaluation, and presenting technical findings in a business context.

I am confident that this model can be integrated into health platforms to help NGOs, hospitals, and blood banks improve their outreach, reduce shortages, and save lives through timely donor engagement.

ABOUT MEDTOUREASY (MTE)

MedTourEasy (MTE) is an innovative global health-tech platform that connects patients to quality medical services across borders. By combining advanced technology with human-centered care, MedTourEasy facilitates seamless international healthcare experiences — from consultation and travel to treatment and recovery.

The company also actively engages in digital health research, data analytics, and public health-focused innovations, providing opportunities for interns to apply data science skills to real-world medical problems.

This project is a part of MedTourEasy's vision to empower healthcare through data, aiming to create intelligent systems for better patient outcomes and operational efficiency

CONTACT INFORMATION

Name: Mehfil
Role: Data Analytics Intern
Organization: MedTourEasy (MTE)
Project Title: Give Life – Predicting Blood Donation Using Logistic Regression
Date: May 2025
Email ID: *(mehfilpp13@gmail.com)*
GitHub Link : (https://github.com/mehfilpalapra)

ACKNOWLEDGMENT

I would like to express my deepest gratitude to MedTourEasy for offering this valuable internship opportunity and entrusting me with a meaningful project. I am particularly thankful to my internship manager and the entire MTE team for their continuous guidance and encouragement throughout the project duration.

This experience has not only improved my technical knowledge in data analytics and machine learning, but also enhanced my ability to solve real-world healthcare problems using data-driven solutions.

This report stands as a reflection of my hard work, learning, and contribution to data-backed health innovation.