Predicting Heart Failure Readmission: Data Analysis and Machine Learning Insights

- Onyinyechukwu Obijiofor

Agenda Overview

- Introduction and Objectives
- Exploratory Data Analysis
- Data Preprocessing
- Machine Learning Algorithms Applied
- Recommendations and Future Work

Overview of Heart Failure Readmission

Heart failure readmission occurs when patients return to the hospital shortly after being discharged, indicating potential issues in care.

Importance of Prediction

Predicting heart failure readmissions is crucial for improving patient outcomes and reducing hospital costs.

1. **Resource Allocation:** Predicting readmission helps healthcare providers allocate resources more effectively, ensuring that patients receive timely care.

2. **Targeted Interventions:** By identifying patients at high risk of readmission, healthcare providers can implement targeted interventions to prevent unnecessary hospital stays.

3. **Enhanced Patient Care:** Accurate readmission predictions lead to better patient care outcomes and improved quality of healthcare delivery overall.

Objectives of the Analysis

The dataset consists of **heart failure patient records** collected from various hospitals. It includes demographic details, medical history, lab test results, and hospital visit records. The primary objective is to analyze this data and build a model to predict whether a patient will be readmitted after being discharged.

Description of the Dataset

Target Variable: Readmission Status (1 = Readmitted, 0 = Not Readmitted) •

FEATURES

- Demographics: Age, Gender, Race
- Medical History: Hypertension, Diabetes, Chronic Diseases
- Lab Results & Vitals: Blood Pressure, Creatinine Levels, Sodium Levels
- Hospitalization Details: Length of Stay, Number of Prior Admissions, Discharge Type
- Medications & Treatment: Prescribed Drugs, Procedures

Description of the Dataset

- The data contained 1000 entries and 20 columns
- There were no missing data, and all columns were in their appropriate datatype.
- The data contained 1000 unique patients.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 20 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Patient_ID             1000 non-null   int64
 1   Age                    1000 non-null   int64
 2   Gender                 1000 non-null   object
 3   Ethnicity              1000 non-null   object
 4   Length_of_Stay         1000 non-null   int64
 5   Previous_Admissions    1000 non-null   int64
 6   Discharge_Disposition  1000 non-null   object
 7   Pulse                  1000 non-null   int64
 8   Temperature            1000 non-null   float64
 9   Heart_Rate             1000 non-null   int64
 10  Systolic_BP            1000 non-null   int64
 11  Diastolic_BP           1000 non-null   int64
 12  Respiratory_Rate       1000 non-null   int64
 13  BUN                    1000 non-null   int64
 14  Creatinine             1000 non-null   float64
 15  Sodium                 1000 non-null   int64
 16  Hemoglobin             1000 non-null   float64
 17  NT_proBNP              1000 non-null   int64
 18  Ejection_Fraction      1000 non-null   int64
 19  Readmission_30Days     1000 non-null   int64
dtypes: float64(3), int64(14), object(3)
memory usage: 156.4+ KB
```

There are no missing data and all the columns are in their appropraite datatype.

```
hdata.shape  # There are 1000 entries and 20 columns in the dataset.
```

```
(1000, 20)
```

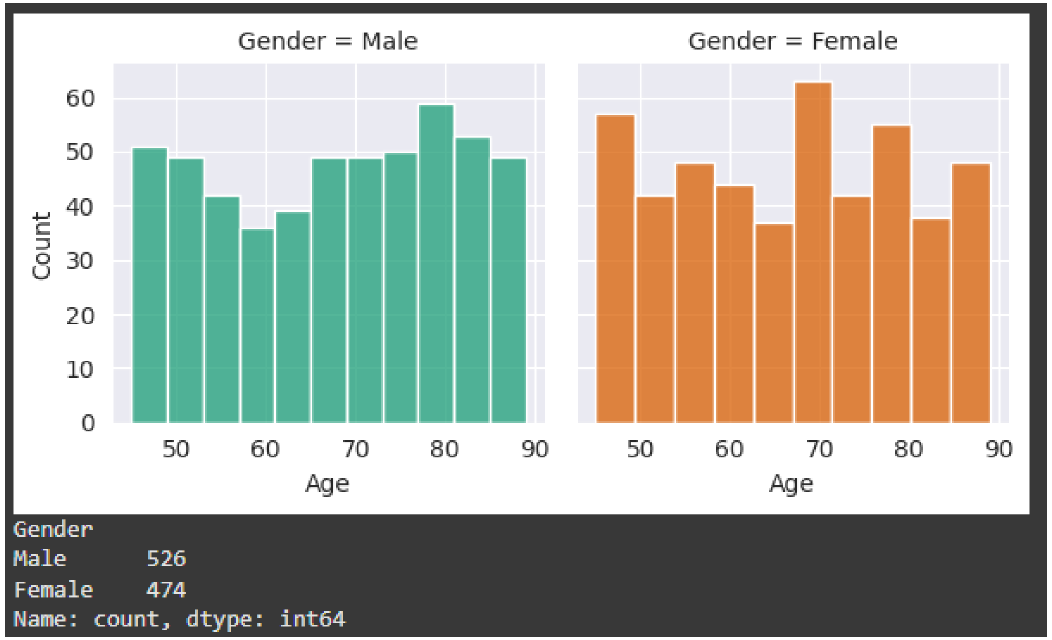
```
# Checking number of unique patients

print('There are', len(hdata['Patient_ID'].unique()), 'unique patients in the data.')
```

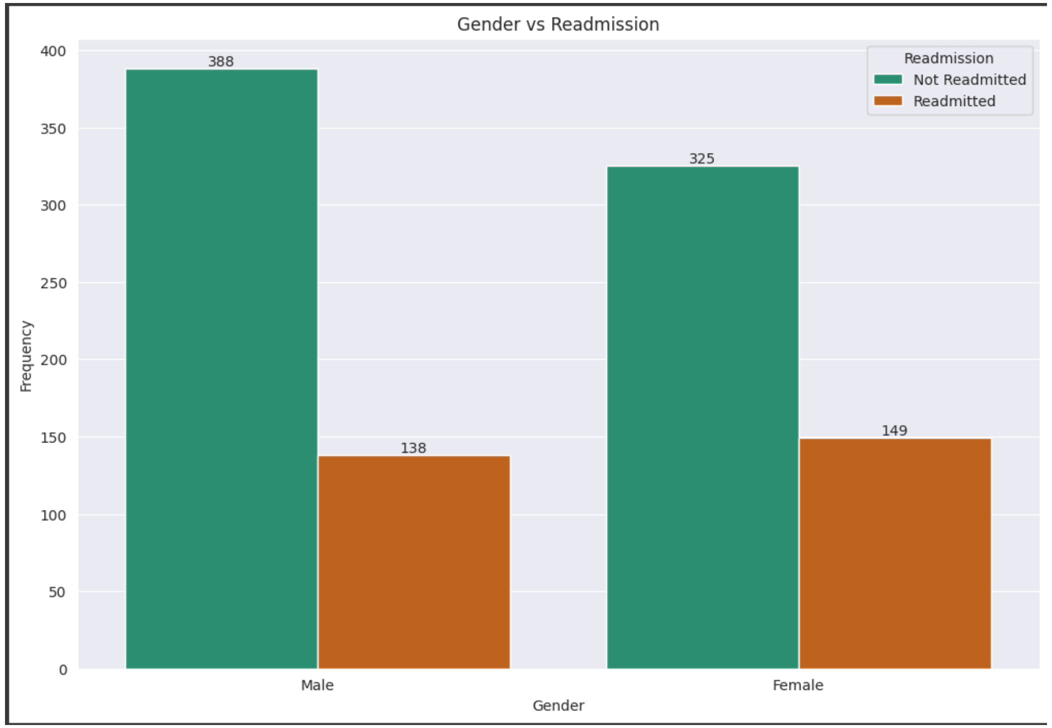
```
There are 1000 unique patients in the data.
```

	count	mean	std	min	25%	50%	75%	max
Patient_ID	1000.0	500.50000	288.819436	1.0	250.75	500.50	750.25	1000.0
Age	1000.0	67.00000	12.945562	45.0	56.00	68.00	78.00	89.0
Length_of_Stay	1000.0	7.40700	4.086325	1.0	4.00	7.00	11.00	14.0
Previous_Admissions	1000.0	1.94800	1.429454	0.0	1.00	2.00	3.00	4.0
Pulse	1000.0	84.71400	20.022465	50.0	67.00	85.00	102.00	119.0
Temperature	1000.0	37.71530	1.001438	36.0	36.80	37.70	38.60	39.5
Heart_Rate	1000.0	98.77000	29.208530	50.0	74.00	97.00	125.00	149.0
Systolic_BP	1000.0	135.49300	25.956303	90.0	112.00	136.00	159.00	179.0
Diastolic_BP	1000.0	79.28900	17.348327	50.0	65.00	79.00	94.25	109.0
Respiratory_Rate	1000.0	20.56300	5.103732	12.0	16.00	21.00	25.00	29.0
BUN	1000.0	23.13900	9.381241	7.0	15.00	23.00	31.00	39.0
Creatinine	1000.0	1.77273	0.715125	0.5	1.19	1.77	2.38	3.0
Sodium	1000.0	137.09700	7.019178	125.0	131.00	137.00	143.00	149.0
Hemoglobin	1000.0	12.53170	2.588240	8.0	10.40	12.60	14.80	17.0
NT_proBNP	1000.0	2552.54700	1416.044376	100.0	1352.75	2546.00	3747.25	4997.0
Ejection_Fraction	1000.0	44.14900	14.733699	20.0	32.00	43.50	57.00	69.0
Readmission_30Days	1000.0	0.28700	0.452588	0.0	0.00	0.00	1.00	1.0

Statistical Summary of Variables

- Patients in the dataset are within 45 - 89 years old, with an average of 67 years.
- Class Imbalance: Around 28.7% (287) of patients were readmitted within 30 days, while 71.3% were not.
- The number of days (Length of stay) the patients stayed in the hospital was an average of 7 days. This ranged from 1 day to 14 days.
- BUN (Blood Urea Nitrogen) and Creatinine levels show high variation, which might indicate kidney function issues.





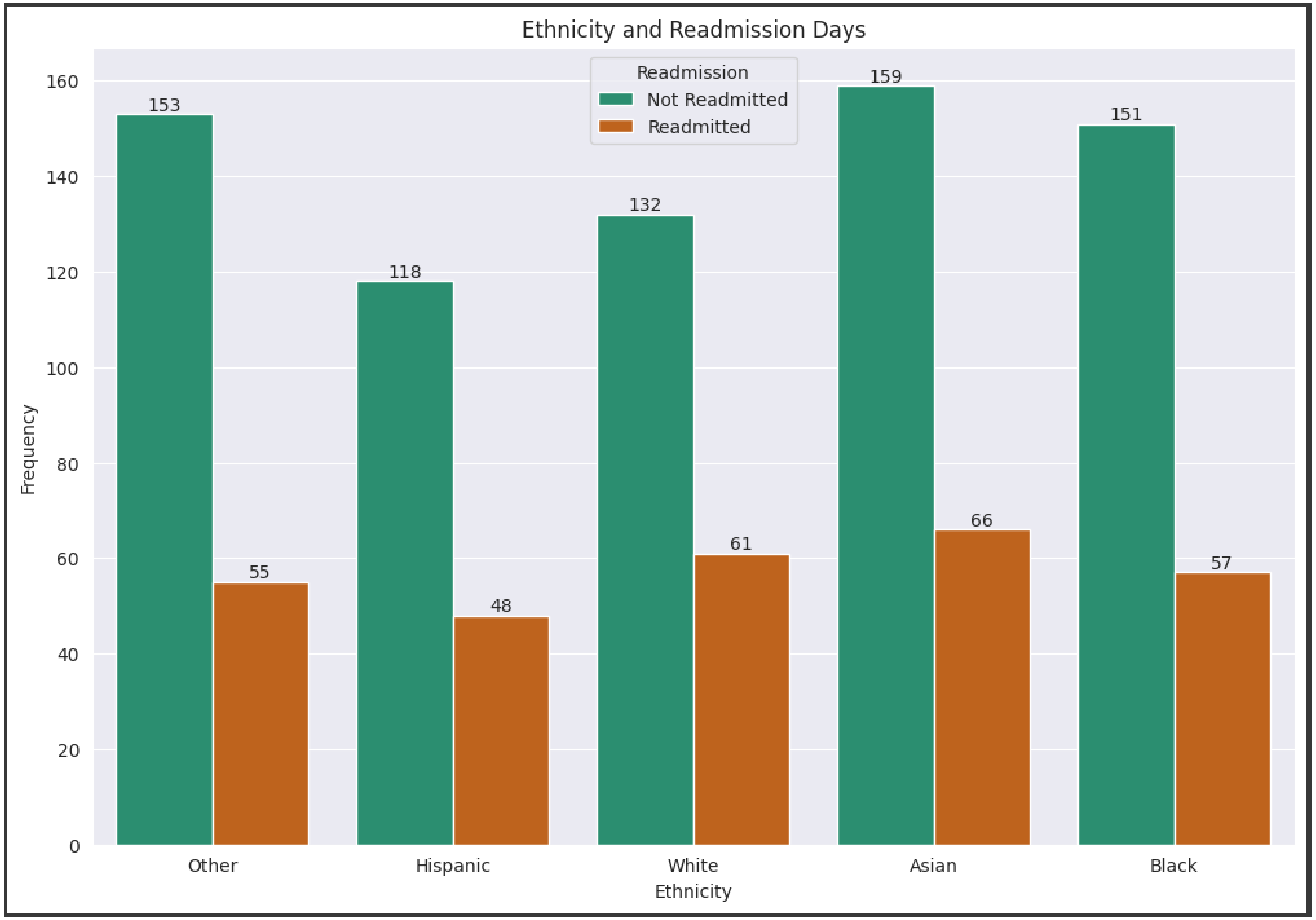
Demographics Summary

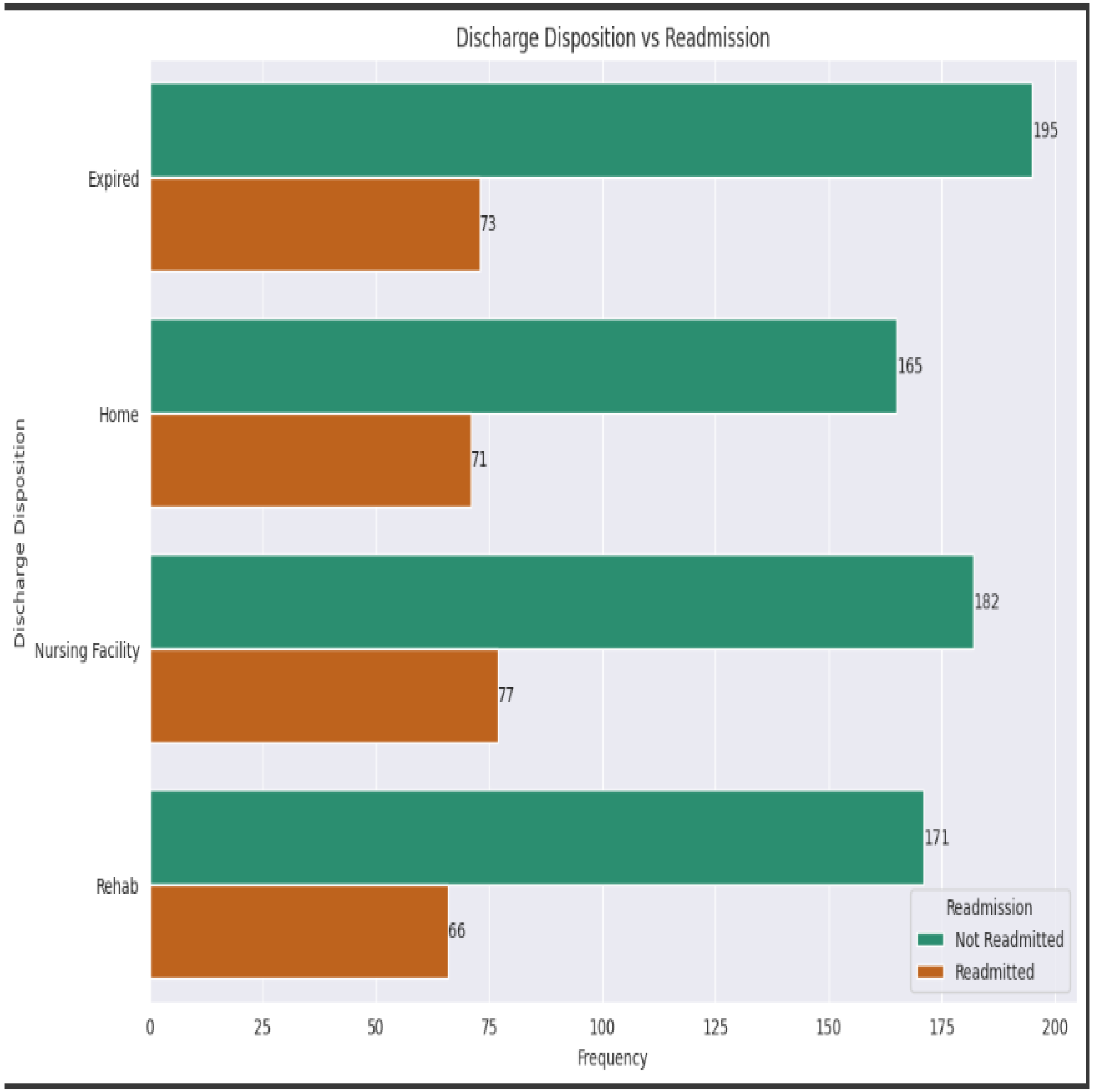
- There are more Males than Females in this data, however there are more females within the ages of 45 and 70 than males.
- Females were readmitted within 30 days more than Males

Ethnicity vs. Readmission Days

The chart shows Asian had higher readmission times than any other ethnicity followed by Whites.

Hispanic had a lower readmission count than the rest of the group.

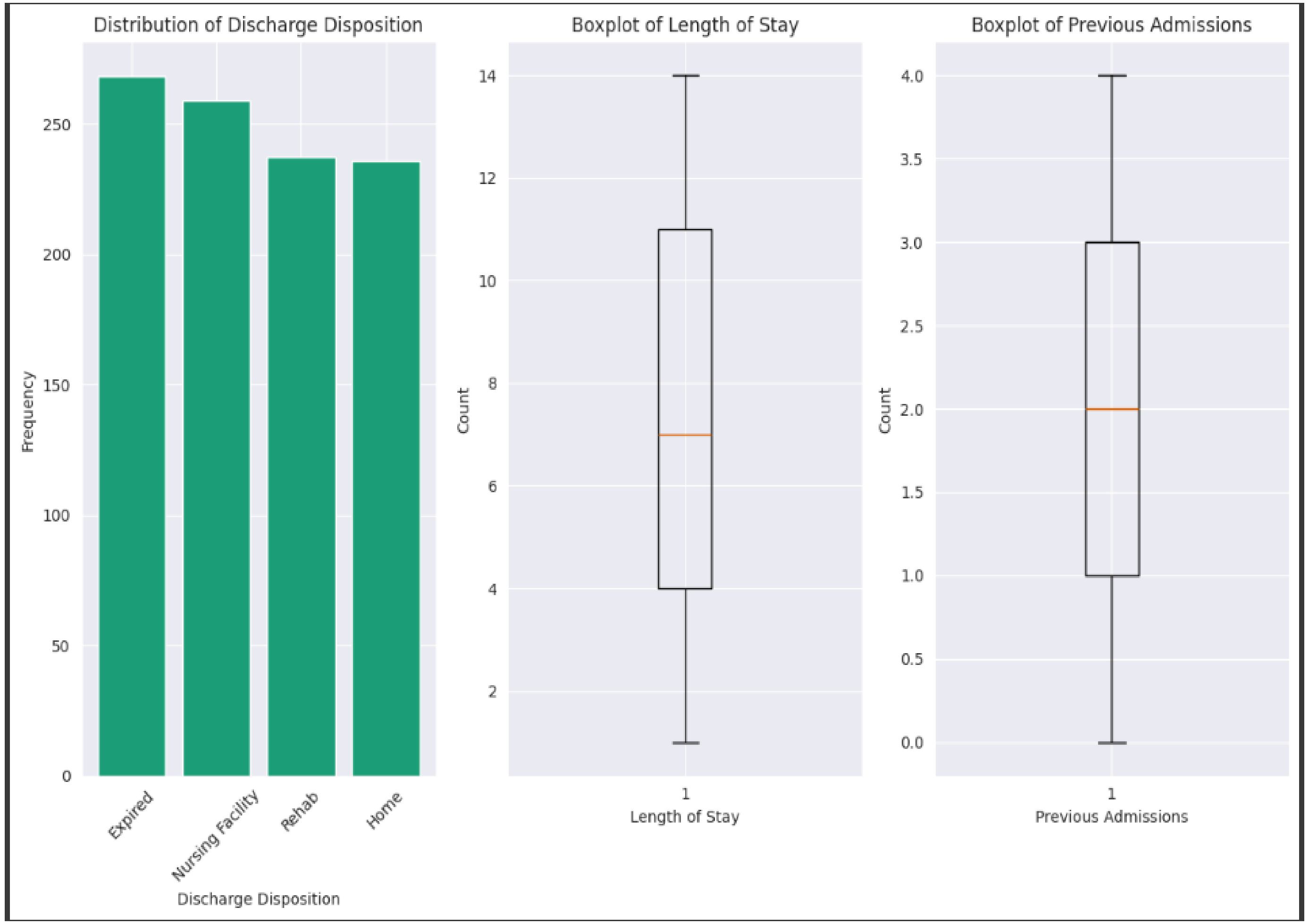




Discharged Distribution vs. Readmission Days

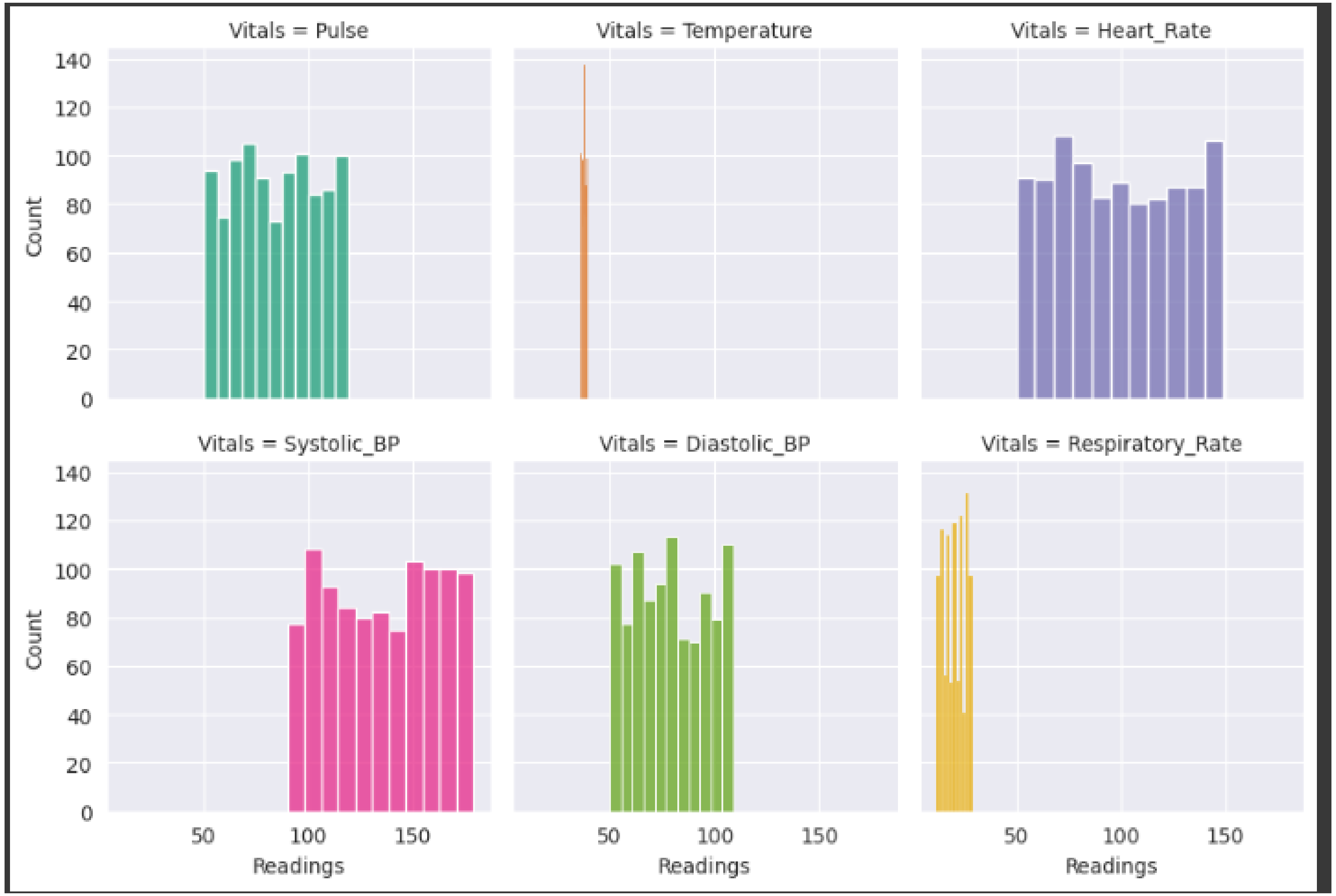
According to the bar chart, those in the Nursing Facility were readmitted within 30 days more than the rest of the group. Conversely, the expired group had higher "Not Readmitted" cases.

Hospitalisation History



- Based on the **subplots**, a lot of patients were expired and following the numbers were those who were taking to a nursing facility.
- The distribution of Length of stay was between 1 day and 14 days. The average day stay in the hospital was 7 days
- Previous admissions for patients was up to 4 times with an average of 2.

Vitals Readings



Adult Vital Signs

Normal vital signs vary based on your age, BMI, sex and overall health.

Vital Sign	Adults
Temperature	97.8 F to 99.1 F (36.5 C to 37.3 C).
Blood pressure	90/60 mm Hg to 120/80 mm Hg.
Pulse	60 to 100 beats per minute.
Respiratory rate	12 to 18 breaths per minute.

Cleveland Clinic

The vitals of the patients (Pulse, Temperature, Heart_Rate, Systolic_BP, Diastolic_BP, and Respiratory_Rate) were within normal on an average, except Blood pressure (Systolic_BP and Diastolic_BP) which were over the recommended 120 /80 mm Hg and 90 / 60 mm Hg respectively.

Data Preprocessing

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report, roc_auc_score
```

```
cat_cols = ['Gender', 'Ethnicity', 'Discharge_Disposition']

hdata_encoded = hdata.copy()

label_encoders = {}

for col in cat_cols:
  le = LabelEncoder()
  hdata_encoded[col] = le.fit_transform(hdata_encoded[col])
  label_encoders[col] = le

# Defining features and target

X = hdata_encoded.drop(columns={"Patient_ID", "Readmission_30Days", 'Readmission'})
y = hdata_encoded["Readmission_30Days"]
```

Training Data for Modeling

```
# Split into training and test data (80% train, 20% test)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state=42)


# Standardize numeric features

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Training RandomForest Classifier

model = RandomForestClassifier(n_estimators = 100, random_state = 42)
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)
y_prob = model.predict_proba(X_test)[:, 1]


# Evaluate Model
report = classification_report(y_test, y_pred)
roc_auc = roc_auc_score(y_test, y_prob)

print(report)
print(roc_auc)
```

Random Forest was used to predict heart failure readmission because:

- It combines multiple trees to reduce overfitting

- Readmission risk is influenced by complex interactions between features (for example, age, gender, discharge disposition, etc)

Performance Metrics and Evaluation

	precision	recall	f1-score	support
0	0.72	0.99	0.84	143
1	0.67	0.04	0.07	57
accuracy			0.72	200
macro avg	0.69	0.51	0.45	200
weighted avg	0.71	0.72	0.62	200

0.46963562753036436

Model Accuracy

Accuracy measures the overall correctness of the model in making predictions. It is the ratio of correct predictions to total predictions.

Precision

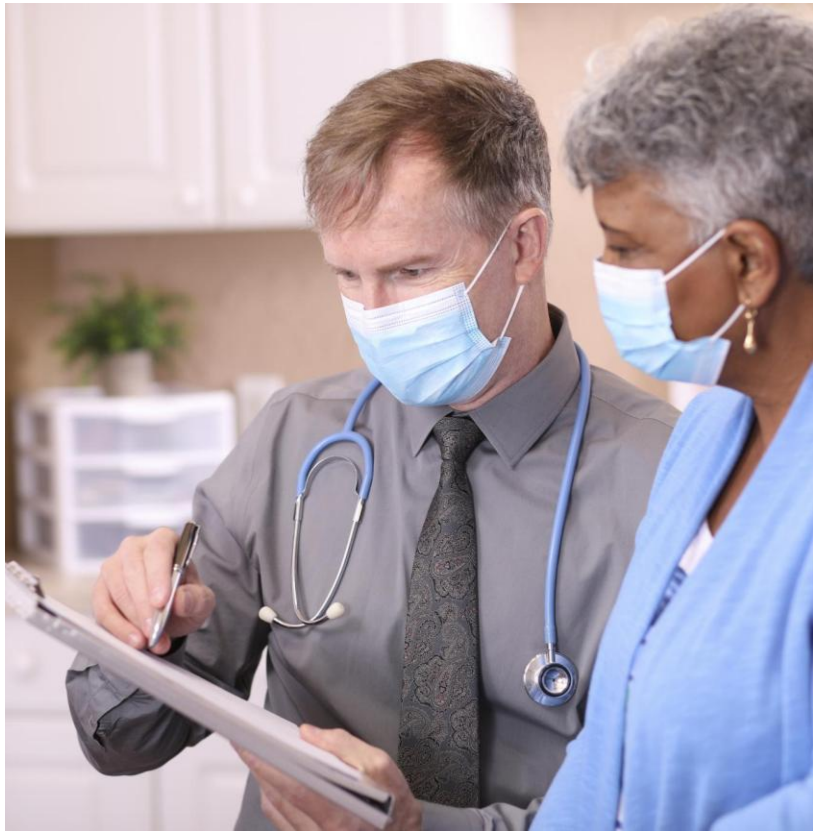
Precision indicates the quality of positive predictions made by the model. It shows how many of the predicted positives are actually positive.

Recall

Recall measures the model's ability to identify all relevant instances. It reflects the ratio of true positives to the actual positives.

F1 Score

The F1 score is the harmonic mean of precision and recall. It balances the trade-offs between precision and recall in evaluating model performance.



Insights From the Prediction Model

Classification Report:

- The model correctly predicts 72% of all cases, however this is misleading due to the class imbalance (there are more "Not Readmitted" cases).
- Precision & Recall for Class 1 (Readmitted Patients) is 67%: This means when the model predicts "readmitted," it is correct 67% of the time.
- Also, the recall for class 1 is only 4%; this is a poor recall as it fails to detect most patients who will be readmitted.
- The F1-Score is 0.07: This is the balance between precision and recall. It has a score of 0.07 which means the model is not effective at predicting readmissions.
- ROC-AUC Score (0.47)
- A good model should have a score above 70%
- The ROC-AUC is 47% which is lower than random guessing (50%)
- The model does not effectively separate readmitted vs. non-readmitted patients.

Recommendations for Improvement

- Though the RandomForest model is good at predicting patients who won't be readmitted (99% recall at Class 1), it fails to predict for those who would be readmitted (4% recall for Class 1).
- For further analysis, handling class imbalance will be considered as well as using more advanced models (e.g, XGBoost or Logistic Regression)

Conclusion

Implementing strategies such as improved discharge planning, patient education, and follow-up care can significantly lower readmission rates and improve patient outcomes.

Thank You