Heart Disease Data Analysis Report

1. Introduction

1.1 Project Overview

The objective of this project is to analyse a heart disease dataset to understand the distribution of heart disease among individuals based on various factors such as age, gender, and other medical attributes. The analysis also involves building a predictive model to identify the key features influencing heart disease.

1.2 Dataset Description

The dataset used for this analysis is sourced from the `Heart Disease data.csv` file. The dataset includes various attributes of patients, including age, sex, chest pain type, resting blood pressure, cholesterol levels, and more, along with the target variable indicating the presence or absence of heart disease.

---

2. Data Loading and Preparation

2.1 Data Import

The dataset was imported into the analysis environment using the pandas library. The first few rows of the dataset were inspected to verify successful data loading.

```
import pandas as pd

data = pd.read_csv('/content/Data Science/Heart Disease data.csv')
print(data.head())
```

2.2 Data Inspection

Column names and unique values in critical columns (`sex` and `target`) were checked to understand the data structure and categories.

```
print(data.columns)
print(data['sex'].unique())
print(data['target'].unique())
```

2.3 Missing Values

Missing values were checked to identify any gaps in the dataset.

```
print(data.isnull().sum())
```

---

3. Data Analysis

3.1 Gender Distribution Analysis

Total individuals and those with heart disease were counted by gender. The results were visualized using a stacked bar chart.

```
import matplotlib.pyplot as plt

gender_counts = data['sex'].value_counts()
gender_heart_disease_counts = data[data['target'] ==
1]['sex'].value_counts()

gender_df = pd.DataFrame({
    'Total': gender_counts,
    'With Heart Disease': gender_heart_disease_counts
}).fillna(0)

gender_df.plot(kind='bar', stacked=True)
plt.title('Heart Disease by Gender')
plt.xlabel('Gender (0: Female, 1: Male)')
plt.ylabel('Count')
plt.show()
```

3.2 Age Distribution of Heart Disease Patients

The age distribution of patients with heart disease was visualized using a histogram.

```
plt.figure(figsize=(10, 6))
sns.histplot(data[data['target'] == 1]['age'], bins=20, kde=True)
plt.title('Distribution of Age in Heart Disease Patients')
```

```
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```

3.3 Feature Relationships

Pairplots were created to visualize relationships between features, coloured by the target variable.

```
sns.pairplot(data, hue='target')
plt.show()
```

3.4 Correlation Matrix

A correlation matrix was created and visualized to understand the relationships between different features.

```
correlation_matrix = data.corr()
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
linewidths=0.2)
plt.title('Correlation Matrix')
plt.show()
```

---

4. Model Training and Evaluation

4.1 Data Splitting

The data was split into training and testing sets.

```
from sklearn.model_selection import train_test_split

X = data.drop('target', axis=1)
y = data['target']
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

4.2 Model Training

A Random Forest classifier was trained on the training data.

```
from sklearn.ensemble import RandomForestClassifier

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

4.3 Model Evaluation

The trained model was evaluated on the test data.

```
from sklearn.metrics import classification_report, accuracy_score

y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
print('Accuracy:', accuracy_score(y_test, y_pred))
```

4.4 Feature Importance

The importance of each feature was extracted from the trained model and visualized.

```
feature_importances = model.feature_importances_
features = X.columns
importance_df = pd.DataFrame({'Feature': features, 'Importance':
feature_importances}).sort_values(by='Importance', ascending=False)
print(importance_df)

plt.figure(figsize=(10, 6))
sns.barplot(x='Importance', y='Feature', data=importance_df)
plt.title('Feature Importance')
plt.show()
```

---

5. Dashboard Creation in Power BI

5.1 Importing Data

The dataset was imported into Power BI Desktop for visualization.

5.2 Creating Visuals

- **Gender Distribution:** A stacked bar chart was created to show the total number of individuals and those with heart disease by gender.
- **Age Distribution:** A histogram was created to show the distribution of age among heart disease patients.
- **Feature Relationships and Correlation:** Scatter plot matrices and a correlation matrix were created using custom visuals from the Power BI marketplace.

5.3 Dashboard Design

The visuals were arranged on the Power BI canvas to create an informative and interactive dashboard. Slicers were added to allow dynamic filtering by gender and age.

---

6. Conclusion

The analysis provided insights into the distribution and key factors influencing heart disease. The predictive model achieved satisfactory accuracy, and feature importance analysis highlighted the most significant attributes. The Power BI dashboard offers an interactive way to explore the data and gain further insights.

This structured report covers the entire process from data loading to visualisation in Power BI, providing a comprehensive overview of the analysis and findings.