title	Heart Disease Prediction Analysis
author	
email	
date	2024-05-20
output	Notebook, Markdown

Load the data set + Imports

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report

print('pandas:',pd. __version__)
print('seaborn',sns.__version__)


# Load dataset
df = pd.read_csv('.venv/heart_disease_uci.csv')
```

```
pandas: 2.2.2
seaborn 0.13.2
```

remove missing values

```
# Check for missing values
print(df.isnull().sum())

# Drop or impute missing values
df = df.dropna()
```

```
id            0
age           0
sex           0
dataset       0
cp            0
trestbps     59
chol         30
fbs          90
restecg       2
thalch       55
exang        55
oldpeak      62
slope       309
ca          611
thal        486
```

```
num          0
dtype: int64
```

Remove Duplicates

```
# Check for duplicates
print(df.duplicated().sum())

# Remove duplicates
df = df.drop_duplicates()

0
```

Data Type Conversion

```
# Convert 'fbs' column to boolean
df['fbs'] = df['fbs'].astype(bool)

# Standardize column names
df.columns = df.columns.str.lower().str.replace(' ', '_')

# Convert categorical columns to numeric using one-hot encoding
df = pd.get_dummies(df, columns=['sex', 'cp', 'restecg', 'slope',
'thal', 'dataset'], drop_first=True)
```

Standardize Column Names

```
# Standardize column names
df.columns = df.columns.str.lower().str.replace(' ', '_')
```

START

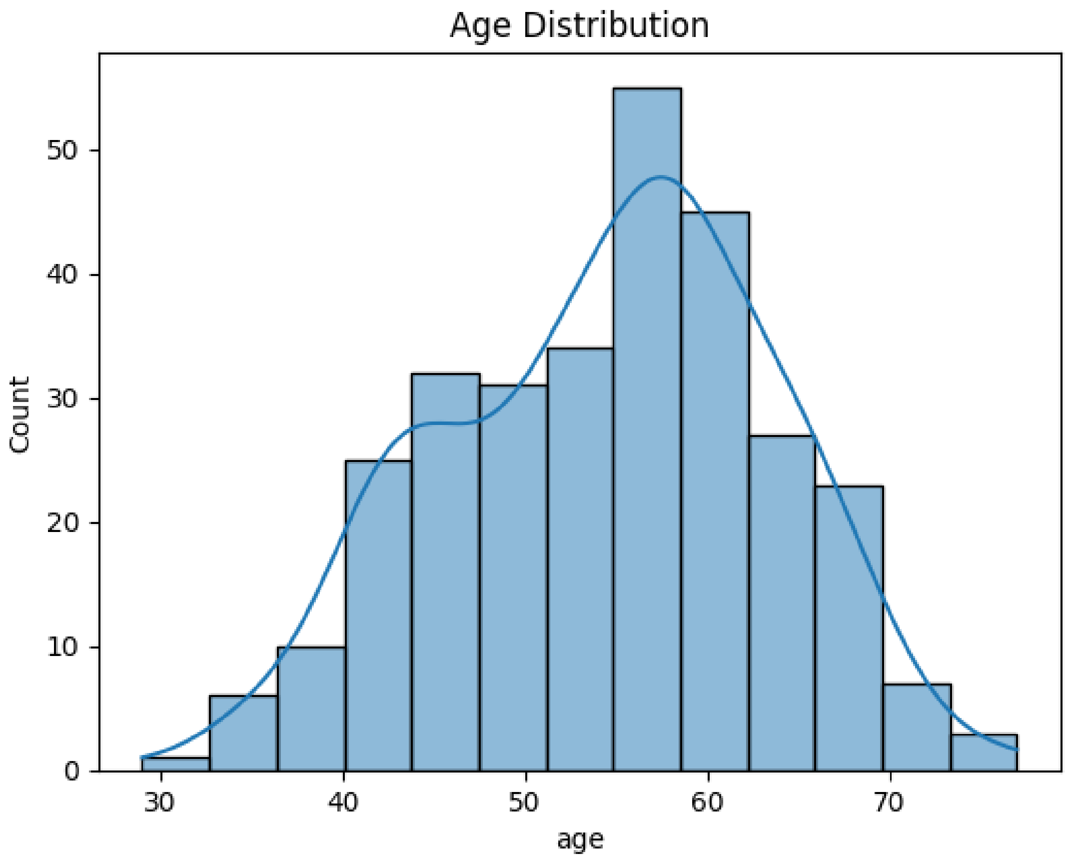
Descriptive Statistics

```
# Get summary statistics
print(df.describe())

               id         age    trestbps        chol      thalch
oldpeak  \
count  299.000000  299.000000  299.000000  299.000000  299.000000
299.000000
mean   153.872910   54.521739  131.715719  246.785953  149.327759
1.058528
std     95.896287    9.030264   17.747751   52.532582   23.121062
1.162769
min      1.000000   29.000000   94.000000  100.000000   71.000000
0.000000
```

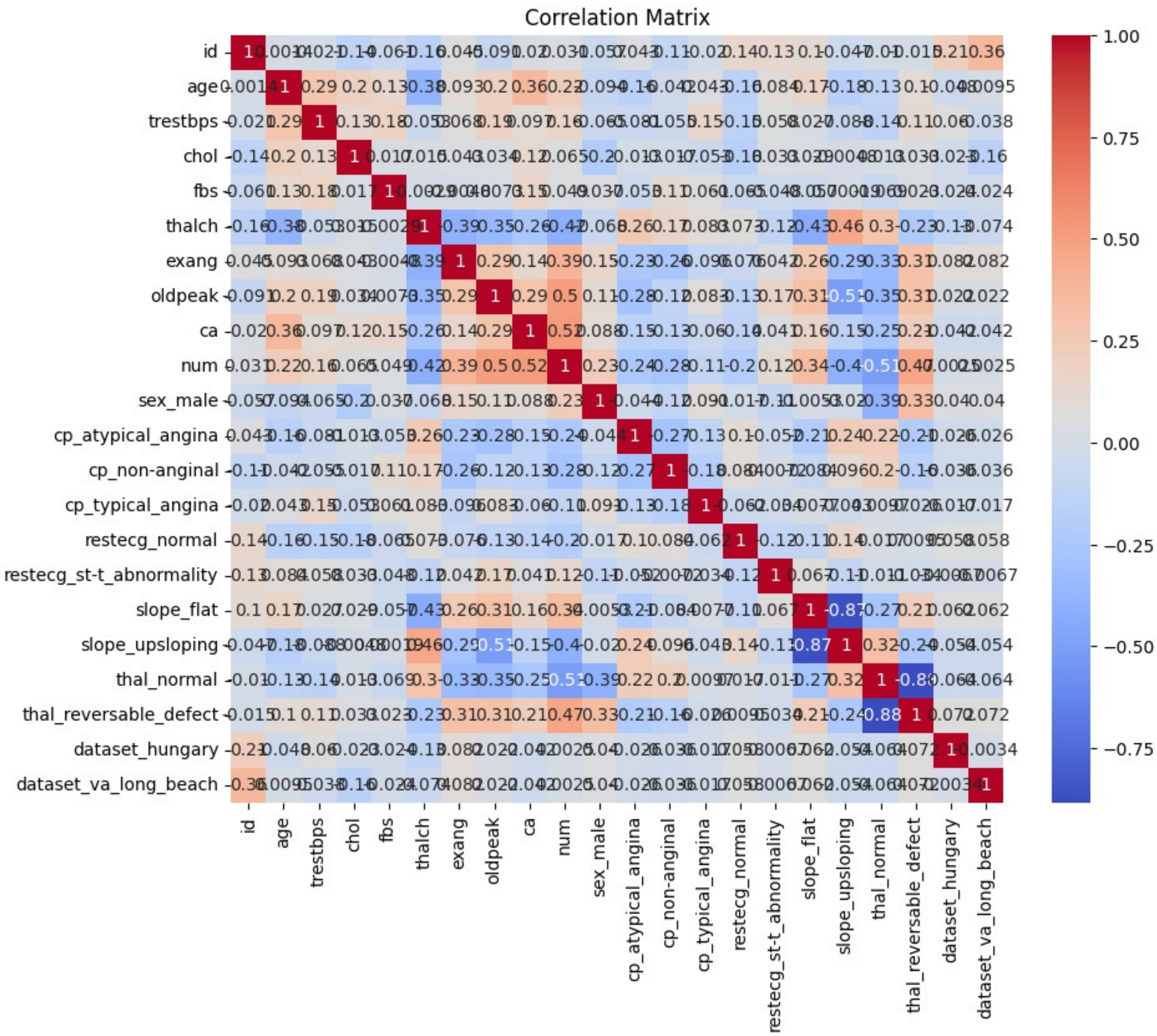
```
25%     75.500000   48.000000  120.000000  211.000000  132.500000
0.000000
50%    151.000000   56.000000  130.000000  242.000000  152.000000
0.800000
75%    227.500000   61.000000  140.000000  275.500000  165.500000
1.600000
max    749.000000   77.000000  200.000000  564.000000  202.000000
6.200000

               ca         num
count  299.000000  299.000000
mean     0.672241    0.946488
std      0.937438    1.230409
min      0.000000    0.000000
25%      0.000000    0.000000
50%      0.000000    0.000000
75%      1.000000    2.000000
max      3.000000    4.000000
```

Data Distribution

```
# Plot distribution of age
sns.histplot(df['age'], kde=True)
plt.title('Age Distribution')
plt.show()
```




```
# Compute correlation matrix
corr = df.corr()

# Plot heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```

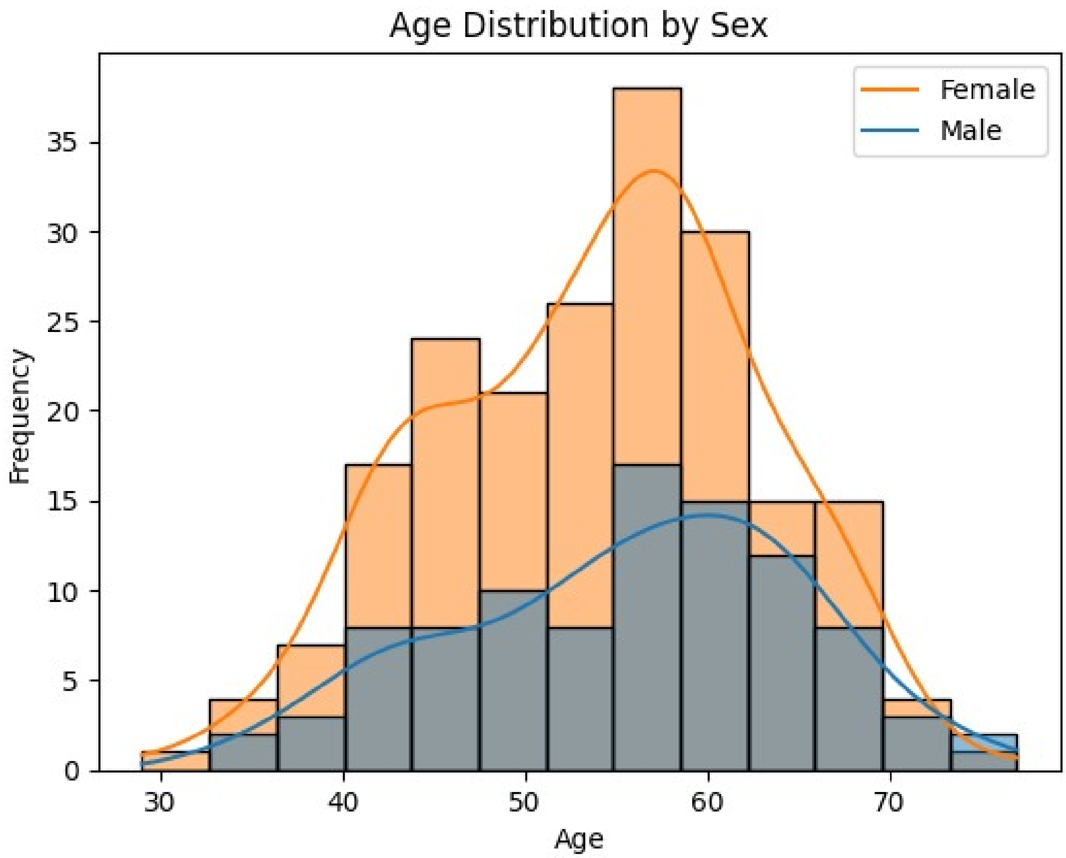



```python
# Define features and target
X = df.drop('num', axis=1)
y = df['num']

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Train a Random Forest model
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Predict and evaluate
y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```

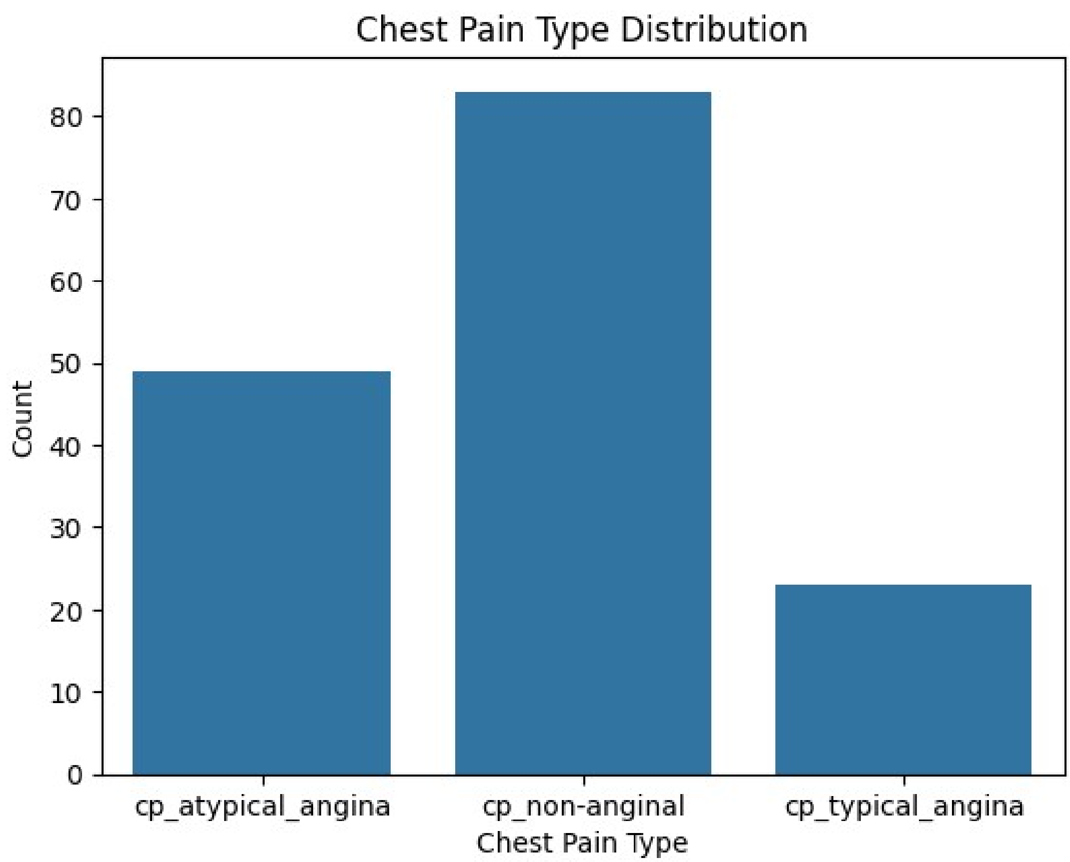
```
              precision    recall  f1-score   support

           0       0.79      0.94      0.86        35
           1       0.12      0.08      0.10        13
           2       0.25      0.20      0.22         5
           3       0.17      0.25      0.20         4
           4       0.00      0.00      0.00         3

    accuracy                           0.60        60
   macro avg       0.27      0.29      0.27        60
weighted avg       0.52      0.60      0.55        60


/Users/jimmytran/Downloads/project/pythonProject/.venv/lib/
python3.12/site-packages/sklearn/metrics/_classification.py:1517:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0
in labels with no predicted samples. Use `zero_division` parameter to
control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is",
len(result))
/Users/jimmytran/Downloads/project/pythonProject/.venv/lib/python3.12/
site-packages/sklearn/metrics/_classification.py:1517:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0
in labels with no predicted samples. Use `zero_division` parameter to
control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is",
len(result))
/Users/jimmytran/Downloads/project/pythonProject/.venv/lib/python3.12/
site-packages/sklearn/metrics/_classification.py:1517:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0
in labels with no predicted samples. Use `zero_division` parameter to
control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is",
len(result))
```

Age Distribution by Sex

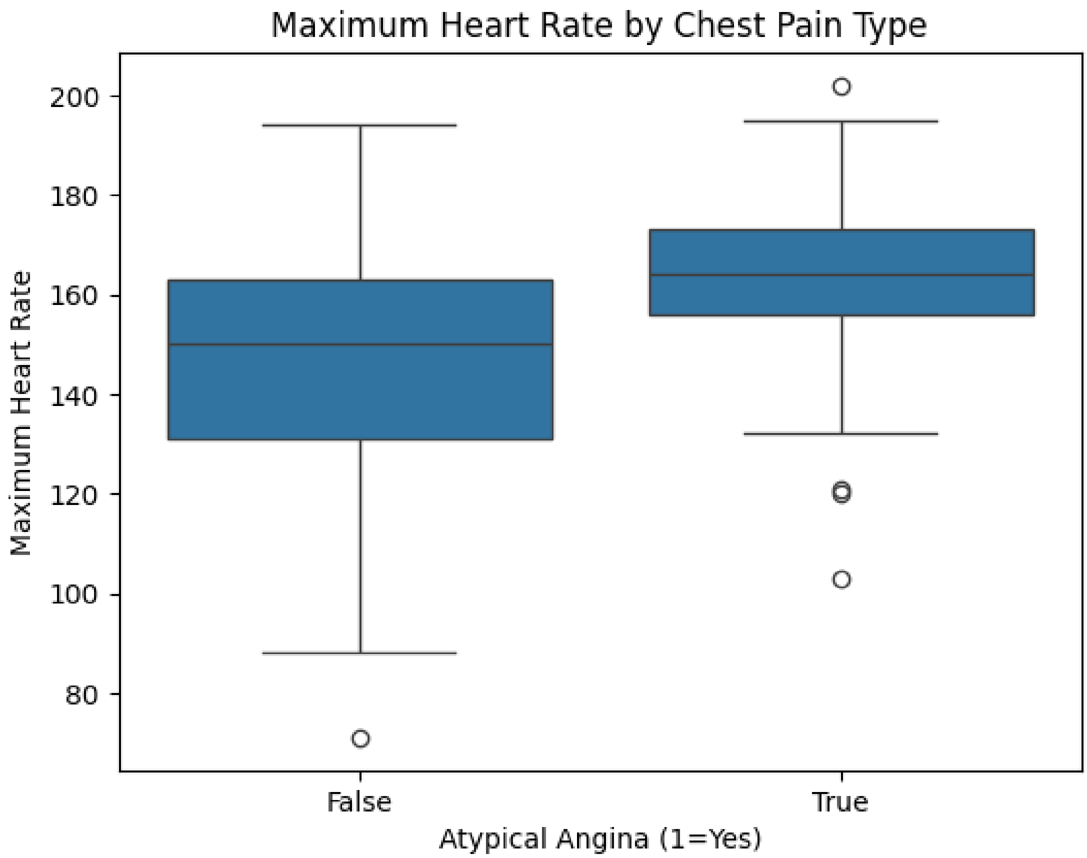
```python
# Age Distribution by Sex
sns.histplot(data=df, x='age', hue='sex_male', kde=True)
plt.title('Age Distribution by Sex')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.legend(['Female', 'Male'])
plt.show()
```



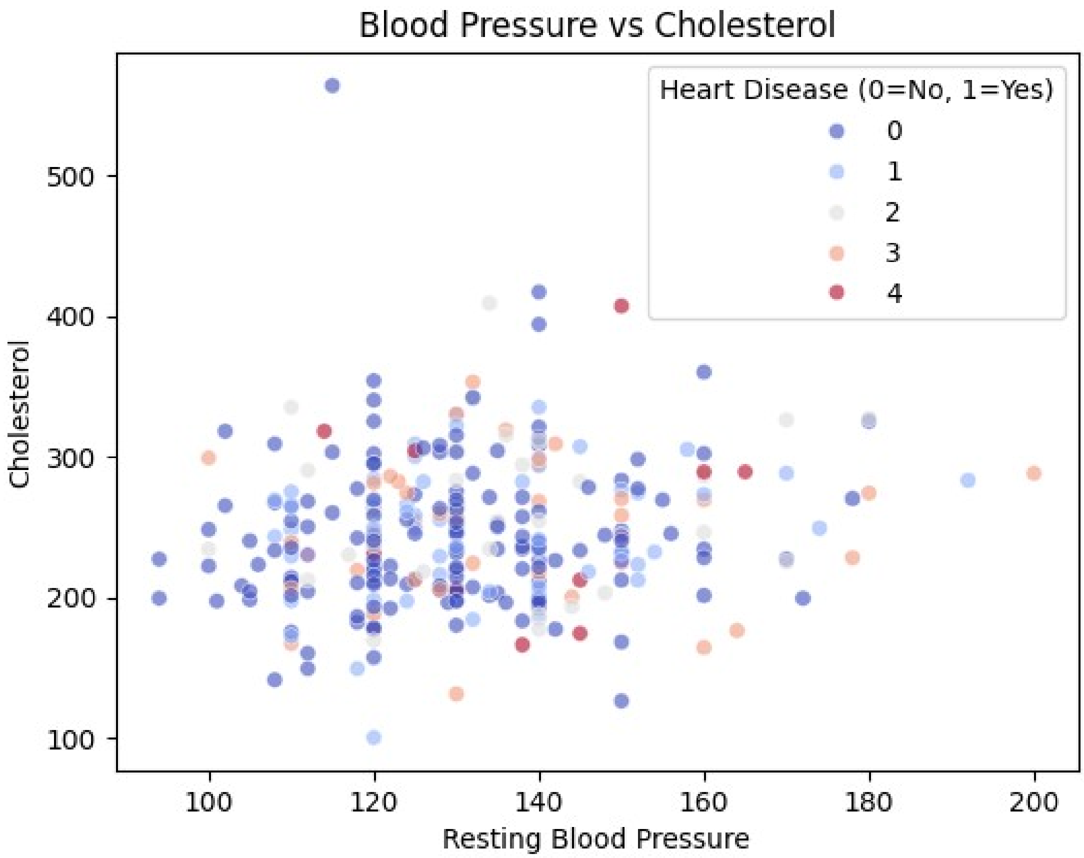

```
# Chest Pain Type Distribution
cp_cols = [col for col in df.columns if 'cp_' in col]
df_cp = df[cp_cols].sum().reset_index()
df_cp.columns = ['Chest Pain Type', 'Count']

sns.barplot(data=df_cp, x='Chest Pain Type', y='Count')
plt.title('Chest Pain Type Distribution')
plt.xlabel('Chest Pain Type')
plt.ylabel('Count')
plt.show()
```



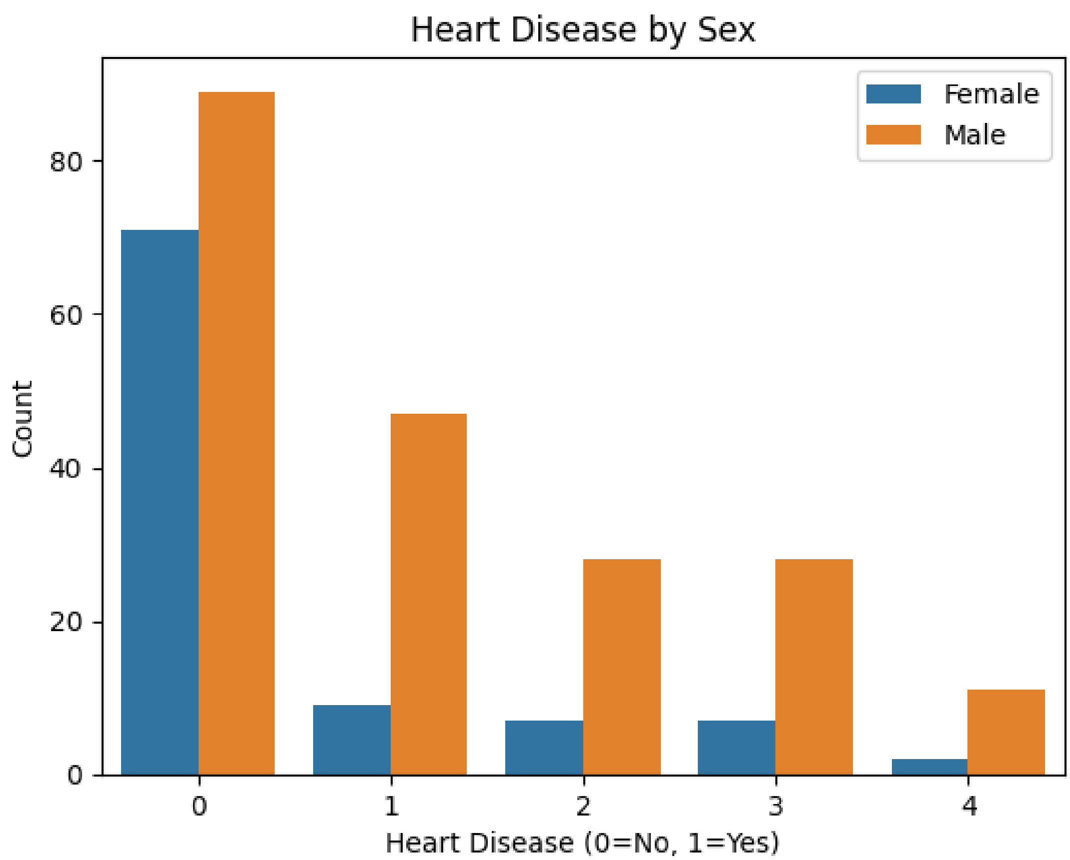

```
# Heart Rate by Chest Pain Type
sns.boxplot(data=df, x='cp_atypical_angina', y='thalch')
plt.title('Maximum Heart Rate by Chest Pain Type')
plt.xlabel('Atypical Angina (1=Yes)')
plt.ylabel('Maximum Heart Rate')
plt.show()
```



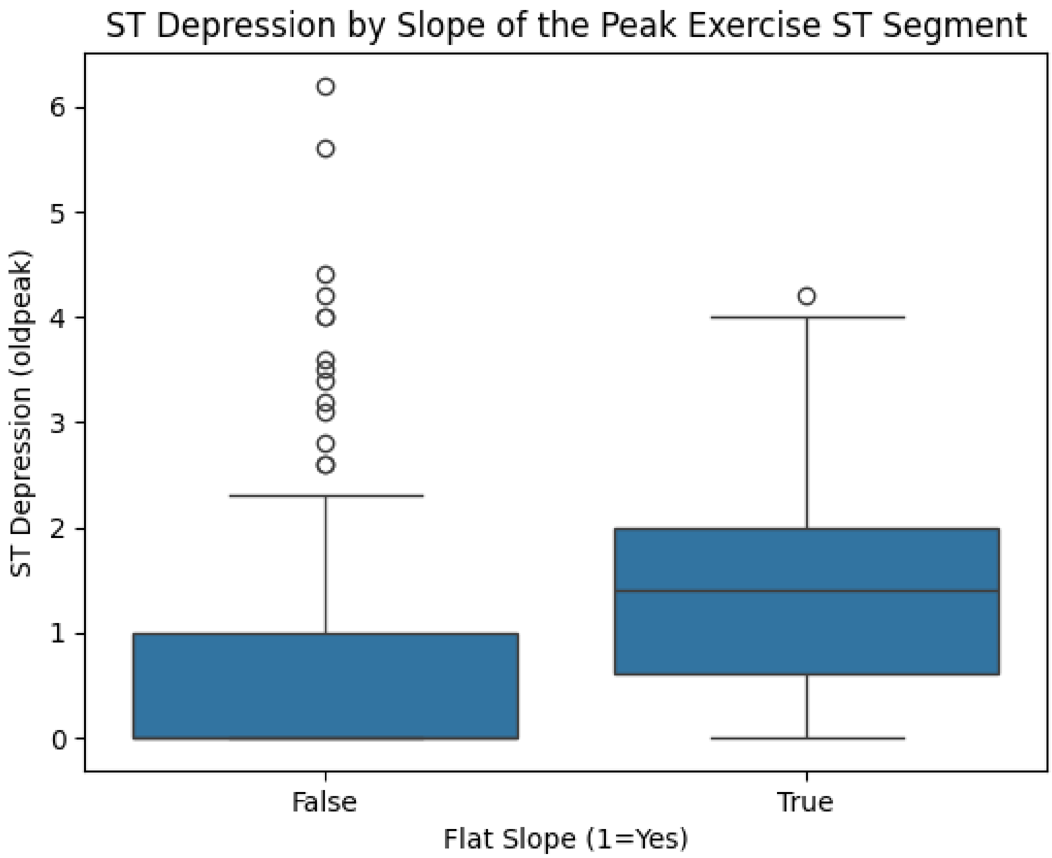

```python
# Blood Pressure vs Cholesterol
sns.scatterplot(data=df, x='trestbps', y='chol', hue='num',
palette='coolwarm', alpha=0.6)
plt.title('Blood Pressure vs Cholesterol')
plt.xlabel('Resting Blood Pressure')
plt.ylabel('Cholesterol')
plt.legend(title='Heart Disease (0=No, 1=Yes)')
plt.show()
```



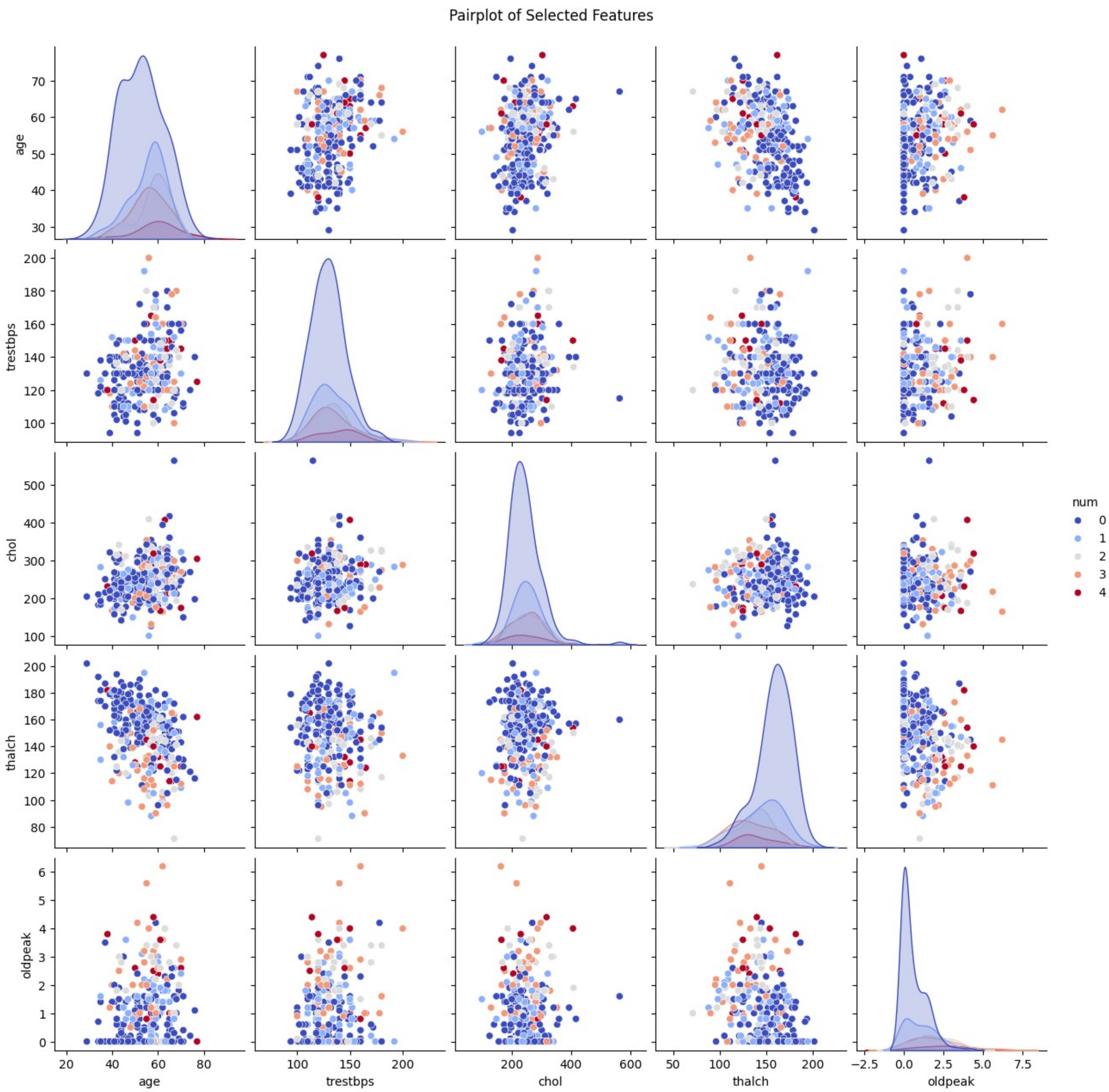

```python
# Heart Disease by Sex
sns.countplot(data=df, x='num', hue='sex_male')
plt.title('Heart Disease by Sex')
plt.xlabel('Heart Disease (0=No, 1=Yes)')
plt.ylabel('Count')
plt.legend(['Female', 'Male'])
plt.show()
```




```python
# Oldpeak (ST Depression) by Slope of the Peak Exercise ST Segment
sns.boxplot(data=df, x='slope_flat', y='oldpeak')
plt.title('ST Depression by Slope of the Peak Exercise ST Segment')
plt.xlabel('Flat Slope (1=Yes)')
plt.ylabel('ST Depression (oldpeak)')
plt.show()
```




```
selected_features = ['age', 'trestbps', 'chol', 'thalch', 'oldpeak',
'num']
sns.pairplot(df[selected_features], hue='num', palette='coolwarm')
plt.suptitle('Pairplot of Selected Features', y=1.02)
plt.show()
```

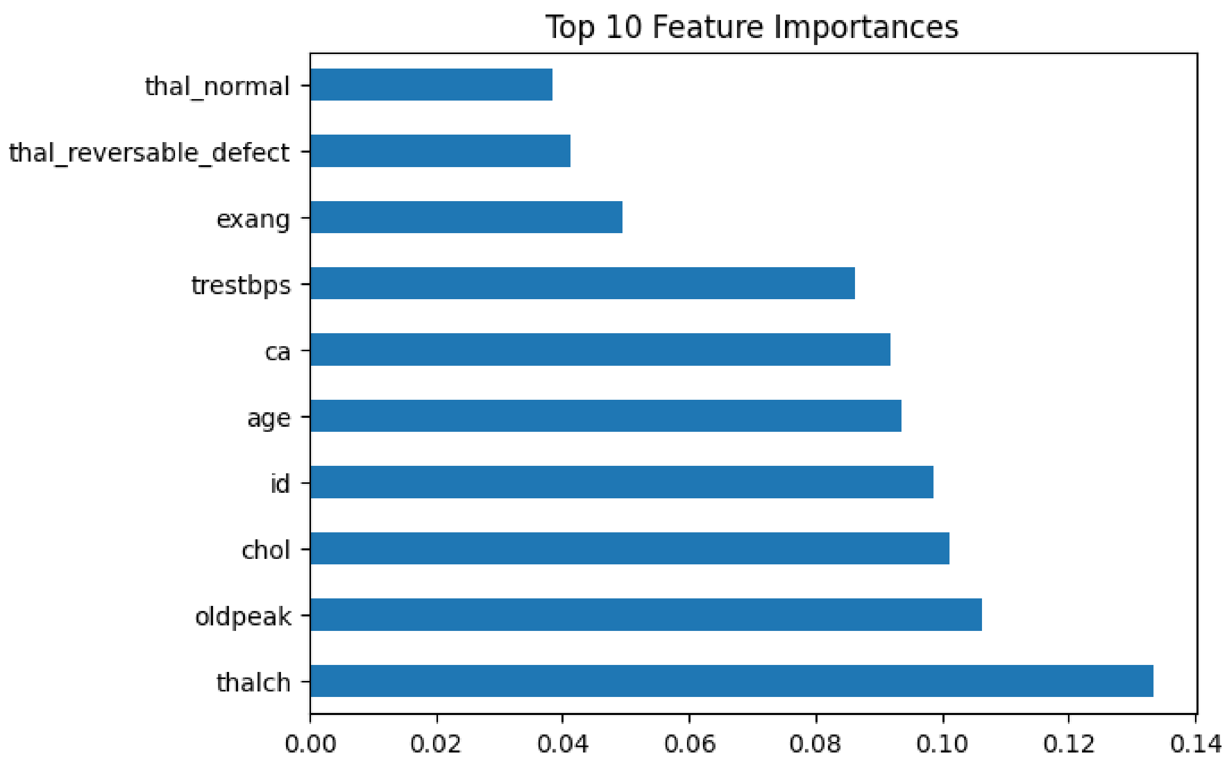



```python
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report

# Define features and target
X = df.drop('num', axis=1)
y = df['num']

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Train a Random Forest model
```

```
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Predict and evaluate
y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.78      1.00      0.88        35
           1       0.00      0.00      0.00        13
           2       0.17      0.20      0.18         5
           3       0.00      0.00      0.00         4
           4       0.00      0.00      0.00         3

    accuracy                           0.60        60
   macro avg       0.19      0.24      0.21        60
weighted avg       0.47      0.60      0.53        60


/Users/jimmytran/Downloads/project/pythonProject/.venv/lib/
python3.12/site-packages/sklearn/metrics/_classification.py:1517:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0
in labels with no predicted samples. Use `zero_division` parameter to
control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is",
len(result))
/Users/jimmytran/Downloads/project/pythonProject/.venv/lib/python3.12/
site-packages/sklearn/metrics/_classification.py:1517:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0
in labels with no predicted samples. Use `zero_division` parameter to
control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is",
len(result))
/Users/jimmytran/Downloads/project/pythonProject/.venv/lib/python3.12/
site-packages/sklearn/metrics/_classification.py:1517:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0
in labels with no predicted samples. Use `zero_division` parameter to
control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is",
len(result))
```

```
# Feature importance
feature_importance = pd.Series(model.feature_importances_,
index=X.columns)
feature_importance.nlargest(10).plot(kind='barh')
plt.title('Top 10 Feature Importances')
plt.show()
```



Summary of Analysis for Heart Disease Prediction

In this analysis, we aim to identify key factors that contribute to heart disease. Using the Heart Disease UCI dataset, we performed data cleaning, exploratory data analysis, and built a machine learning model to predict heart disease. The findings are summarized as follows:

- **Age and sex are significant predictors** of heart disease.
- **Chest pain type, resting blood pressure, and cholesterol levels** also play crucial roles.
- **Patients with higher oldpeak values (ST depression induced by exercise)** and abnormal exercise-induced ST segment slopes are more likely to have heart disease.
- The **Random Forest model** was used for prediction, and the top features influencing the model were identified.
- Model performance was evaluated using **classification metrics** and **ROC curve analysis**.

Key Findings:

1. **Age Distribution by Sex**: Males tend to have a higher prevalence of heart disease compared to females.
2. **Chest Pain Type Distribution**: Different types of chest pain are associated with varying risks of heart disease.
3. **Heart Rate Analysis**: Higher maximum heart rates are observed in patients with certain types of chest pain.
4. **Blood Pressure and Cholesterol**: There is a significant relationship between higher resting blood pressure, cholesterol levels, and heart disease.

5. **Heart Disease by Sex**: The prevalence of heart disease varies between males and females.
6. **Oldpeak Analysis**: Higher ST depression values are indicative of higher heart disease risk.

Recommendations:

1. **Targeted Health Programs**: Develop targeted health programs focusing on high-risk groups identified by age, sex, and other significant features.
2. **Preventive Measures**: Encourage regular health screenings for early detection and management of high blood pressure and cholesterol levels.
3. **Public Awareness**: Increase public awareness about the significance of recognizing and managing chest pain and related symptoms early.