# ai-ml-tasks

June 25, 2025

## 1 Zoya Hafeez

## 2 DHC-1549

https://colab.research.google.com/drive/1SR0yLmwUS68UMHsDGHSB1Ofzt_kNPzYY?usp=sharing

## 3 Github Repository

[1]:
```
%cd https://github.com/zoya4477/AI-Ml.git
!git clone
!git config --global user.email "zoyahafeez785@gmail.com"
!git config --global user.name "zoya4477"
```

```
[Errno 2] No such file or directory: 'https://github.com/zoya4477/AI-Ml.git'
/content
fatal: You must specify a repository to clone.

usage: git clone [<options>] [--] <repo> [<dir>]

    -v, --verbose         be more verbose
    -q, --quiet           be more quiet
    --progress            force progress reporting
    --reject-shallow      don't clone shallow repository
    -n, --no-checkout     don't create a checkout
    --bare                create a bare repository
    --mirror              create a mirror repository (implies bare)
    -l, --local           to clone from a local repository
    --no-hardlinks        don't use local hardlinks, always copy
    -s, --shared          setup as shared repository
    --recurse-submodules[=<pathspec>]
                          initialize submodules in the clone
    --recursive …       alias of --recurse-submodules
    -j, --jobs <n>        number of submodules cloned in parallel
    --template <template-directory>
                          directory from which templates will be used
    --reference <repo>    reference repository
    --reference-if-able <repo>
```

```
                          reference repository
    --dissociate          use --reference only while cloning
    -o, --origin <name>   use <name> instead of 'origin' to track upstream
    -b, --branch <branch>
                          checkout <branch> instead of the remote's HEAD
    -u, --upload-pack <path>
                          path to git-upload-pack on the remote
    --depth <depth>       create a shallow clone of that depth
    --shallow-since <time>
                          create a shallow clone since a specific time
    --shallow-exclude <revision>
                          deepen history of shallow clone, excluding rev
    --single-branch       clone only one branch, HEAD or --branch
    --no-tags             don't clone any tags, and make later fetches not to
follow them
    --shallow-submodules  any cloned submodules will be shallow
    --separate-git-dir <gitdir>
                          separate git dir from working tree
    -c, --config <key=value>
                          set config inside the new repository
    --server-option <server-specific>
                          option to transmit
    -4, --ipv4            use IPv4 addresses only
    -6, --ipv6            use IPv6 addresses only
    --filter <args>       object filtering
    --remote-submodules   any cloned submodules will use their remote-tracking
branch
    --sparse              initialize sparse-checkout file to include only files
at root
```

# 4 Task 1: Exploring and Visualizing the Iris Dataset

# 5 Load the Dataset

[2]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
#load Dataset
df = pd.read_csv('/content/Iris.csv')
```

## 6 Inspect the dataset

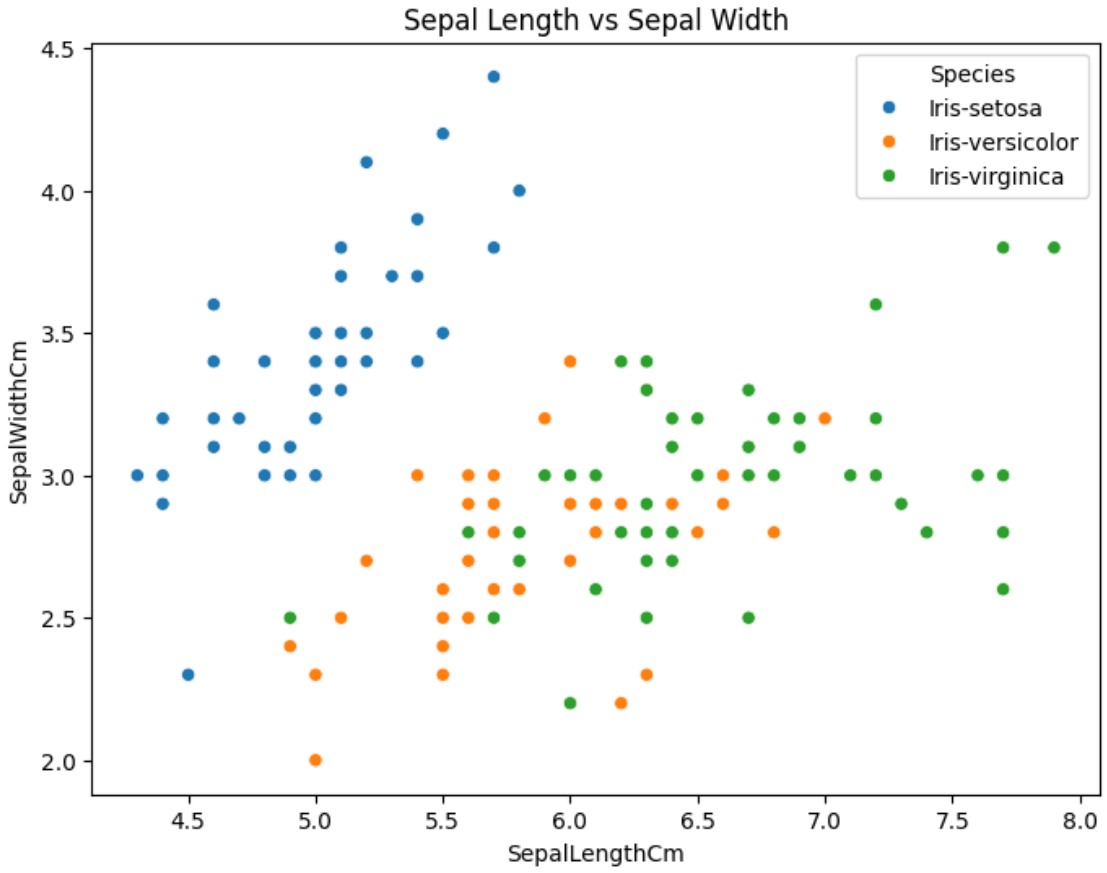
[3]:
```
# Shape of the dataset
print("Shape of the dataset:", df.shape)

# Column names
print("Column names:", df.columns.tolist())

# First 5 rows
print(df.head())

# Info summary
print("\nDataset Info:")
print(df.info())

# Descriptive statistics
print("\nDescriptive Statistics:")
print(df.describe())
```

```
Shape of the dataset: (152, 6)
Column names: ['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm',
'PetalWidthCm', 'Species']
   Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm      Species
0   1            5.1           3.5            NaN           0.2  Iris-setosa
1   2            4.9           3.0            1.4           0.2  Iris-setosa
2   3            NaN           3.2            1.3           0.2  Iris-setosa
3   4            4.6           3.1            1.5           NaN  Iris-setosa
4   5            5.0           NaN            1.4           0.2  Iris-setosa

Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 152 entries, 0 to 151
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             152 non-null    int64
 1   SepalLengthCm  151 non-null    float64
 2   SepalWidthCm   151 non-null    float64
 3   PetalLengthCm  151 non-null    float64
 4   PetalWidthCm   151 non-null    float64
 5   Species        152 non-null    object
dtypes: float64(4), int64(1), object(1)
memory usage: 7.3+ KB
None

Descriptive Statistics:
               Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
```

| | | | | | |
|---|---|---|---|---|---|
| count | 152.000000 | 151.000000 | 151.000000 | 151.000000 | 151.000000 |
| mean | 75.414474 | 5.849007 | 3.055629 | 3.770861 | 1.206623 |
| std | 43.866813 | 0.823073 | 0.432302 | 1.764902 | 0.766870 |
| min | 1.000000 | 4.300000 | 2.000000 | 1.000000 | 0.100000 |
| 25% | 37.750000 | 5.100000 | 2.800000 | 1.600000 | 0.300000 |
| 50% | 75.500000 | 5.800000 | 3.000000 | 4.400000 | 1.300000 |
| 75% | 113.250000 | 6.400000 | 3.300000 | 5.100000 | 1.800000 |
| max | 150.000000 | 7.900000 | 4.400000 | 6.900000 | 2.500000 |

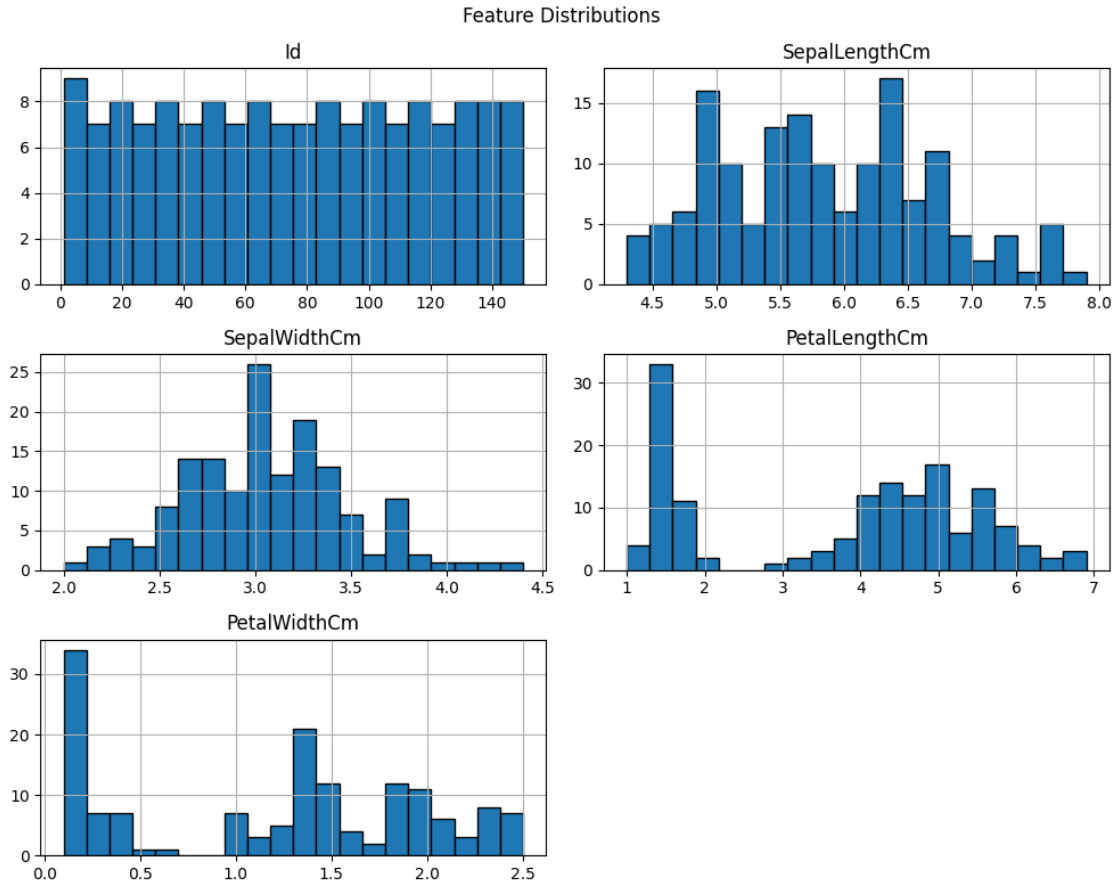
#Visualize the Dataset

[4]:
```
#Scatter Plot -- Sepal Length vs Sepal Width

plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='SepalLengthCm', y='SepalWidthCm', hue='Species')
plt.title('Sepal Length vs Sepal Width')
plt.show()
```

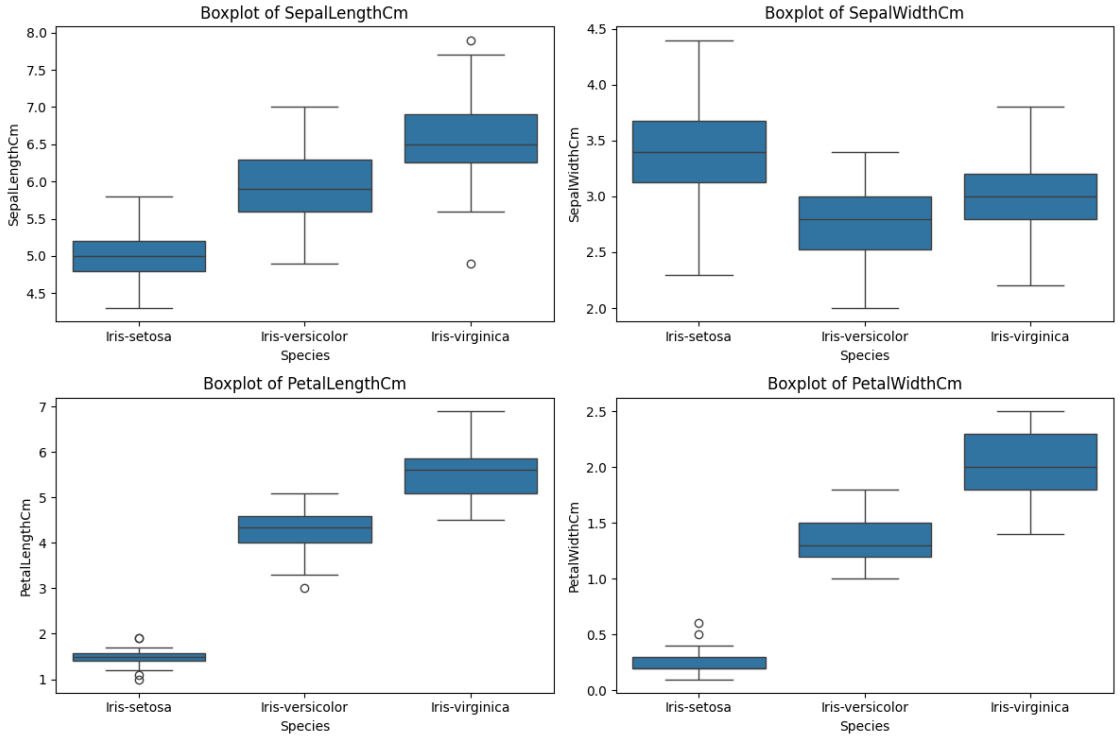



[5]:
```
#Histograms -- Distribution of Each Feature
df.hist(figsize=(10, 8), bins=20, edgecolor='black')
```

```
plt.suptitle('Feature Distributions')
plt.tight_layout()
plt.show()
```




[6]:
```
#Boxplot -- To Identify Outlier
plt.figure(figsize=(12, 8))
features = ['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']

for i, column in enumerate(features):
    plt.subplot(2, 2, i + 1)
    sns.boxplot(x='Species', y=column, data=df)
    plt.title(f'Boxplot of {column}')
plt.tight_layout()
plt.show()
```




[6]:

## 7 Task 2: Predict Future Stock Prices (Short-Term)

[7]:
```
pip install yfinance
```

```
Requirement already satisfied: yfinance in /usr/local/lib/python3.11/dist-
packages (0.2.63)
Requirement already satisfied: pandas>=1.3.0 in /usr/local/lib/python3.11/dist-
packages (from yfinance) (2.2.2)
Requirement already satisfied: numpy>=1.16.5 in /usr/local/lib/python3.11/dist-
packages (from yfinance) (2.0.2)
Requirement already satisfied: requests>=2.31 in /usr/local/lib/python3.11/dist-
packages (from yfinance) (2.32.3)
Requirement already satisfied: multitasking>=0.0.7 in
/usr/local/lib/python3.11/dist-packages (from yfinance) (0.0.11)
Requirement already satisfied: platformdirs>=2.0.0 in
/usr/local/lib/python3.11/dist-packages (from yfinance) (4.3.8)
Requirement already satisfied: pytz>=2022.5 in /usr/local/lib/python3.11/dist-
packages (from yfinance) (2025.2)
Requirement already satisfied: frozendict>=2.3.4 in
/usr/local/lib/python3.11/dist-packages (from yfinance) (2.4.6)
Requirement already satisfied: peewee>=3.16.2 in /usr/local/lib/python3.11/dist-
```

```
packages (from yfinance) (3.18.1)
Requirement already satisfied: beautifulsoup4>=4.11.1 in
/usr/local/lib/python3.11/dist-packages (from yfinance) (4.13.4)
Requirement already satisfied: curl_cffi>=0.7 in /usr/local/lib/python3.11/dist-
packages (from yfinance) (0.11.3)
Requirement already satisfied: protobuf>=3.19.0 in
/usr/local/lib/python3.11/dist-packages (from yfinance) (5.29.5)
Requirement already satisfied: websockets>=13.0 in
/usr/local/lib/python3.11/dist-packages (from yfinance) (15.0.1)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.11/dist-
packages (from beautifulsoup4>=4.11.1->yfinance) (2.7)
Requirement already satisfied: typing-extensions>=4.0.0 in
/usr/local/lib/python3.11/dist-packages (from beautifulsoup4>=4.11.1->yfinance)
(4.14.0)
Requirement already satisfied: cffi>=1.12.0 in /usr/local/lib/python3.11/dist-
packages (from curl_cffi>=0.7->yfinance) (1.17.1)
Requirement already satisfied: certifi>=2024.2.2 in
/usr/local/lib/python3.11/dist-packages (from curl_cffi>=0.7->yfinance)
(2025.6.15)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.11/dist-packages (from pandas>=1.3.0->yfinance)
(2.9.0.post0)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-
packages (from pandas>=1.3.0->yfinance) (2025.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.31->yfinance) (3.4.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-
packages (from requests>=2.31->yfinance) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.31->yfinance) (2.4.0)
Requirement already satisfied: pycparser in /usr/local/lib/python3.11/dist-
packages (from cffi>=1.12.0->curl_cffi>=0.7->yfinance) (2.22)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-
packages (from python-dateutil>=2.8.2->pandas>=1.3.0->yfinance) (1.17.0)
```

#Import libraries and load data

[8]:
```python
import yfinance as yf
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt

# Download historical data for Apple (AAPL)
ticker = 'AAPL'
```

```
df = yf.download(ticker, start='2020-01-01', end='2024-01-01')

# Display first rows
print(df.head())
```

```
/tmp/ipython-input-8-2732541920.py:12: FutureWarning: YF.download() has changed
argument auto_adjust default to True
  df = yf.download(ticker, start='2020-01-01', end='2024-01-01')
[*********************100%***********************]  1 of 1 completed
```

| Price | Close | High | Low | Open | Volume |
|---|---|---|---|---|---|
| Ticker | AAPL | AAPL | AAPL | AAPL | AAPL |
| Date | | | | | |
| 2020-01-02 | 72.620842 | 72.681289 | 71.373218 | 71.627092 | 135480400 |
| 2020-01-03 | 71.914825 | 72.676454 | 71.689965 | 71.847125 | 146322800 |
| 2020-01-06 | 72.487846 | 72.526533 | 70.783248 | 71.034709 | 118387200 |
| 2020-01-07 | 72.146927 | 72.753808 | 71.926900 | 72.497514 | 108872000 |
| 2020-01-08 | 73.307526 | 73.609760 | 71.849548 | 71.849548 | 132079200 |

## 8 Prepare features and target

[9]:
```
# Shift the Close column up by 1 to represent next day close price
df['Next_Close'] = df['Close'].shift(-1)

# Drop last row with NaN target
df = df[:-1]

# Features and target
features = ['Open', 'High', 'Low', 'Volume']
X = df[features]
y = df['Next_Close']
```

#Split data into train/test sets

[10]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, shuffle=False,␣
↪test_size=0.2)
```

## 9 Train the model

[11]:
```
#Linear Regression
model = LinearRegression()
model.fit(X_train, y_train)
```

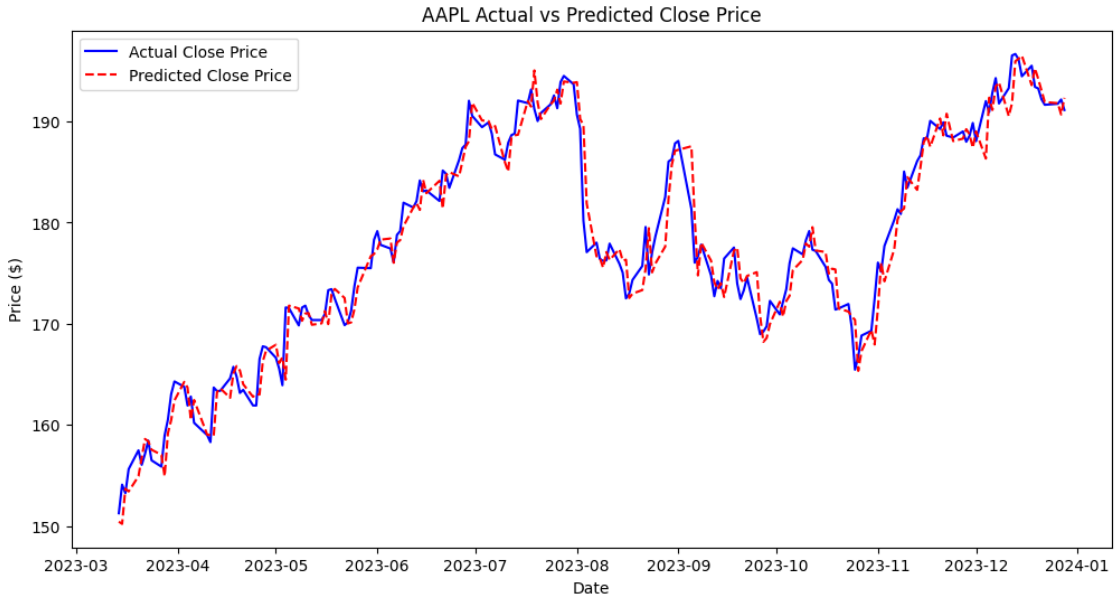
[11]:
```
LinearRegression()
```

[12]:
```
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error: {mse:.4f}")
```

```
Mean Squared Error: 4.9761
```

#Plot actual vs predicted closing prices

[13]:
```
plt.figure(figsize=(12, 6))
plt.plot(y_test.index, y_test, label='Actual Close Price', color='blue')
plt.plot(y_test.index, y_pred, label='Predicted Close Price', color='red',␣
↪linestyle='--')
plt.title(f'{ticker} Actual vs Predicted Close Price')
plt.xlabel('Date')
plt.ylabel('Price ($)')
plt.legend()
plt.show()
```




[13]:

# 10 Task 3: Heart Disease Prediction

# 11 Import Libraries

[14]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, roc_auc_score, roc_curve,␣
↪confusion_matrix, ConfusionMatrixDisplay
```

# 12 Load Dataset

[15]:
```
data = pd.read_csv('/content/archive.zip')
data.head()
```

[15]:

| | age | sex | cp | trestbps | chol | fbs | restecg | thalach | exang | oldpeak | slope | \ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 69 | 1 | 0 | 160 | 234 | 1 | 2 | 131 | 0 | 0.1 | 1 | |
| 1 | 69 | 0 | 0 | 140 | 239 | 0 | 0 | 151 | 0 | 1.8 | 0 | |
| 2 | 66 | 0 | 0 | 150 | 226 | 0 | 0 | 114 | 0 | 2.6 | 2 | |
| 3 | 65 | 1 | 0 | 138 | 282 | 1 | 2 | 174 | 0 | 1.4 | 1 | |
| 4 | 64 | 1 | 0 | 110 | 211 | 0 | 2 | 144 | 1 | 1.8 | 1 | |

| | ca | thal | condition |
|---|---|---|---|
| 0 | 1 | 0 | 0 |
| 1 | 2 | 0 | 0 |
| 2 | 0 | 0 | 0 |
| 3 | 1 | 0 | 1 |
| 4 | 0 | 0 | 0 |

## 12.1 Check Missing Values

[16]:
```
print("Missing values in each column:")
print(data.isnull().sum())
```

```
Missing values in each column:
age          0
sex          0
cp           0
trestbps     0
chol         0
fbs          0
restecg      0
```

```
thalach      0
exang        0
oldpeak      0
slope        0
ca           0
thal         0
condition    0
dtype: int64
```

## 13 Basic Info and Description

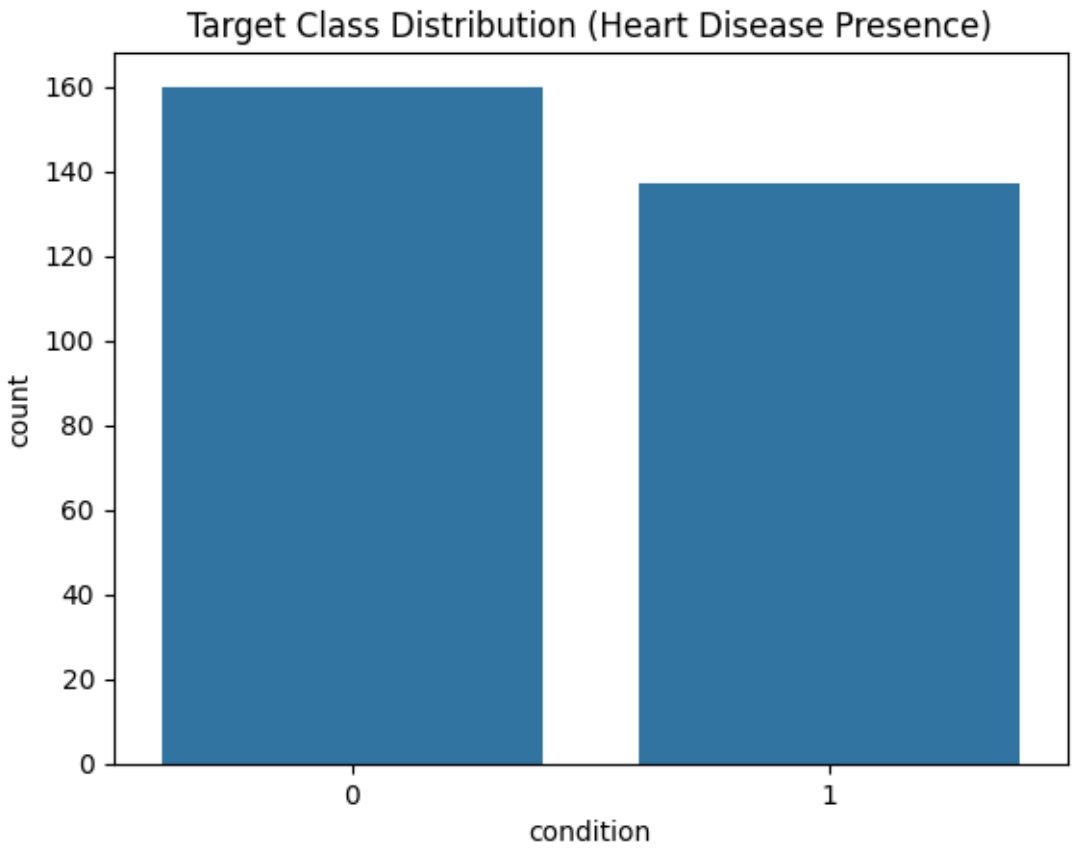
[17]:
```
print(data.info())
print(data.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 297 entries, 0 to 296
Data columns (total 14 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   age        297 non-null    int64
 1   sex        297 non-null    int64
 2   cp         297 non-null    int64
 3   trestbps   297 non-null    int64
 4   chol       297 non-null    int64
 5   fbs        297 non-null    int64
 6   restecg    297 non-null    int64
 7   thalach    297 non-null    int64
 8   exang      297 non-null    int64
 9   oldpeak    297 non-null    float64
 10  slope      297 non-null    int64
 11  ca         297 non-null    int64
 12  thal       297 non-null    int64
 13  condition  297 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 32.6 KB
None
              age         sex          cp    trestbps        chol         fbs  \
count  297.000000  297.000000  297.000000  297.000000  297.000000  297.000000
mean    54.542088    0.676768    2.158249  131.693603  247.350168    0.144781
std      9.049736    0.468500    0.964859   17.762806   51.997583    0.352474
min     29.000000    0.000000    0.000000   94.000000  126.000000    0.000000
25%     48.000000    0.000000    2.000000  120.000000  211.000000    0.000000
50%     56.000000    1.000000    2.000000  130.000000  243.000000    0.000000
75%     61.000000    1.000000    3.000000  140.000000  276.000000    0.000000
max     77.000000    1.000000    3.000000  200.000000  564.000000    1.000000

          restecg     thalach       exang     oldpeak       slope          ca  \
```

```
count  297.000000  297.000000  297.000000  297.000000  297.000000  297.000000
mean     0.996633  149.599327    0.326599    1.055556    0.602694    0.676768
std      0.994914   22.941562    0.469761    1.166123    0.618187    0.938965
min      0.000000   71.000000    0.000000    0.000000    0.000000    0.000000
25%      0.000000  133.000000    0.000000    0.000000    0.000000    0.000000
50%      1.000000  153.000000    0.000000    0.800000    1.000000    0.000000
75%      2.000000  166.000000    1.000000    1.600000    1.000000    1.000000
max      2.000000  202.000000    1.000000    6.200000    2.000000    3.000000

             thal   condition
count  297.000000  297.000000
mean     0.835017    0.461279
std      0.956690    0.499340
min      0.000000    0.000000
25%      0.000000    0.000000
50%      0.000000    0.000000
75%      2.000000    1.000000
max      2.000000    1.000000
```

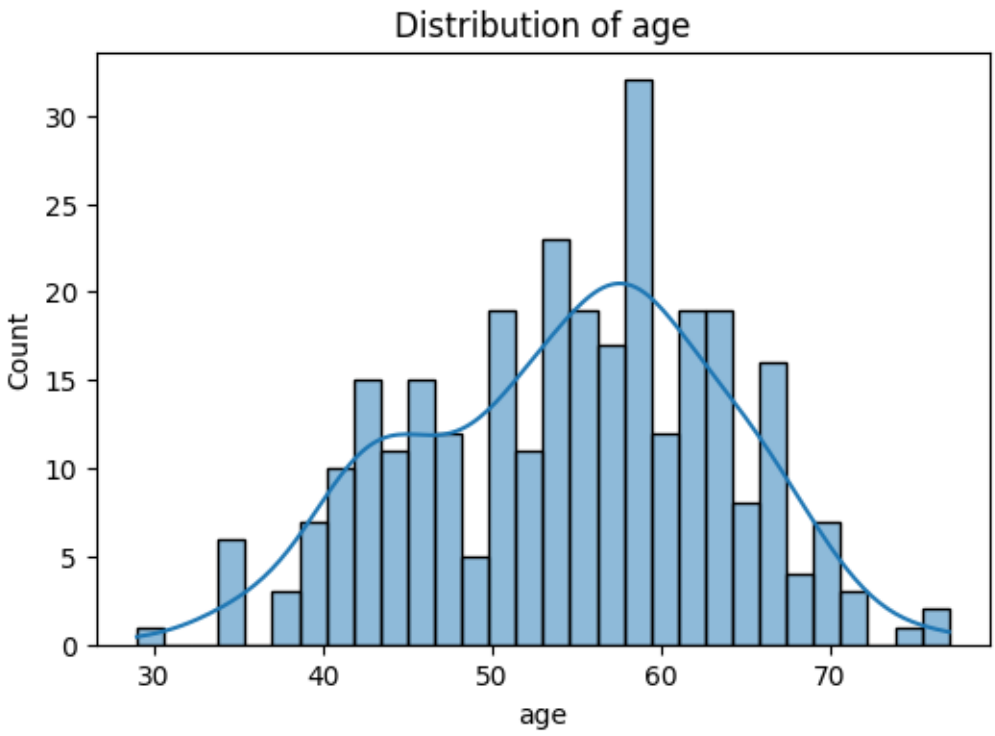
## 14 Visualize Target Distribution

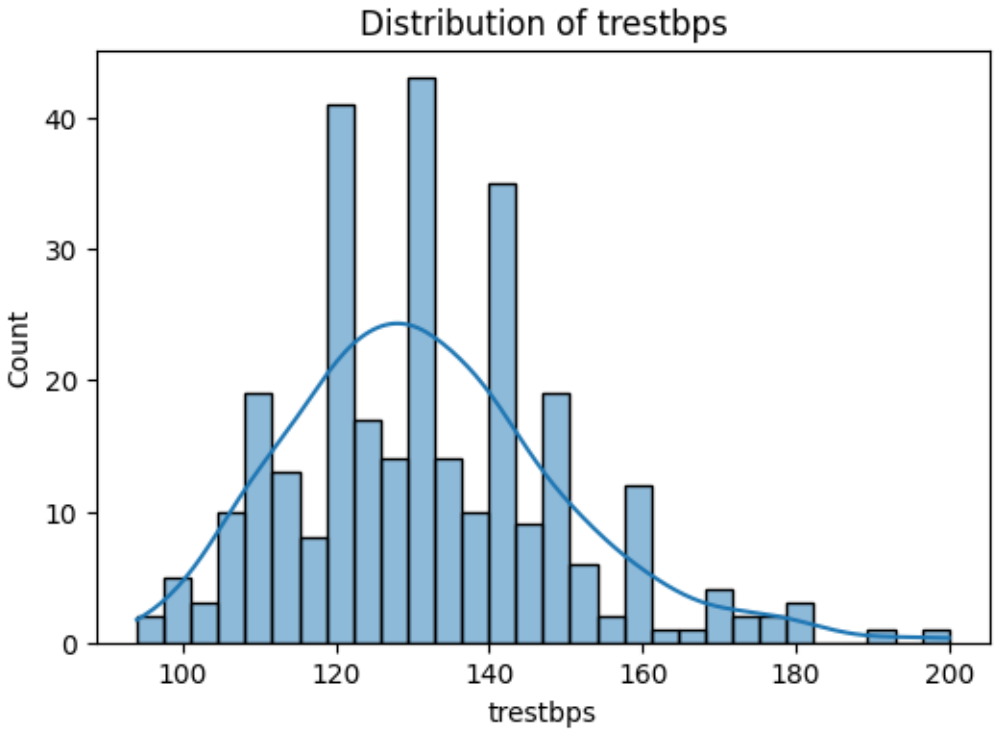
[18]:
```
sns.countplot(x='condition', data=data)
plt.title('Target Class Distribution (Heart Disease Presence)')
plt.show()
```

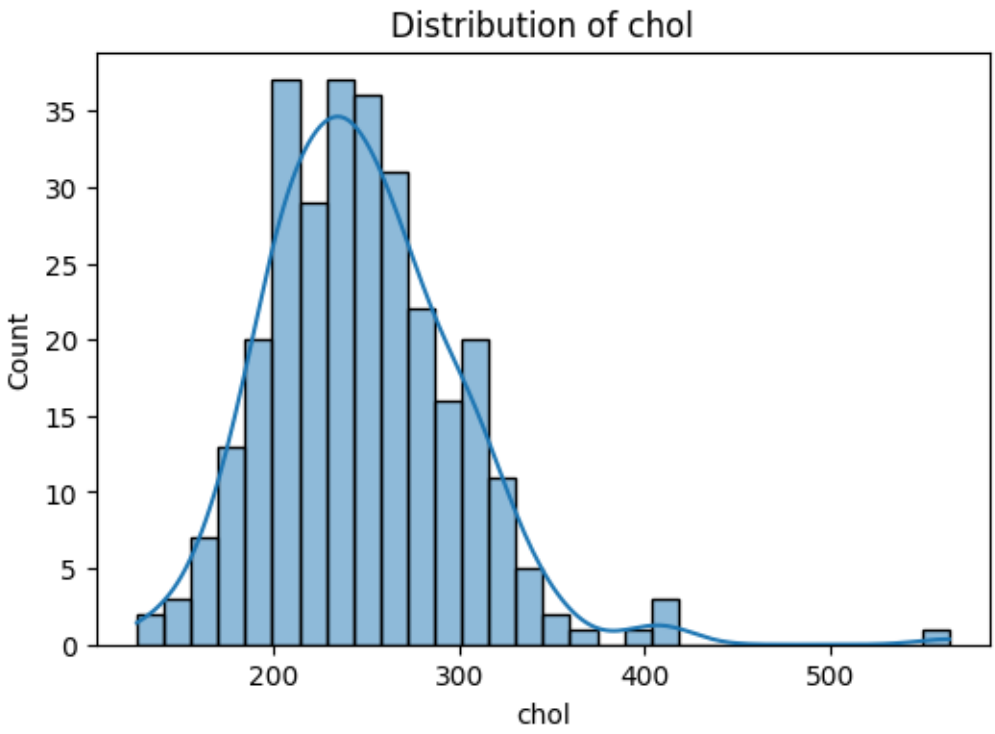


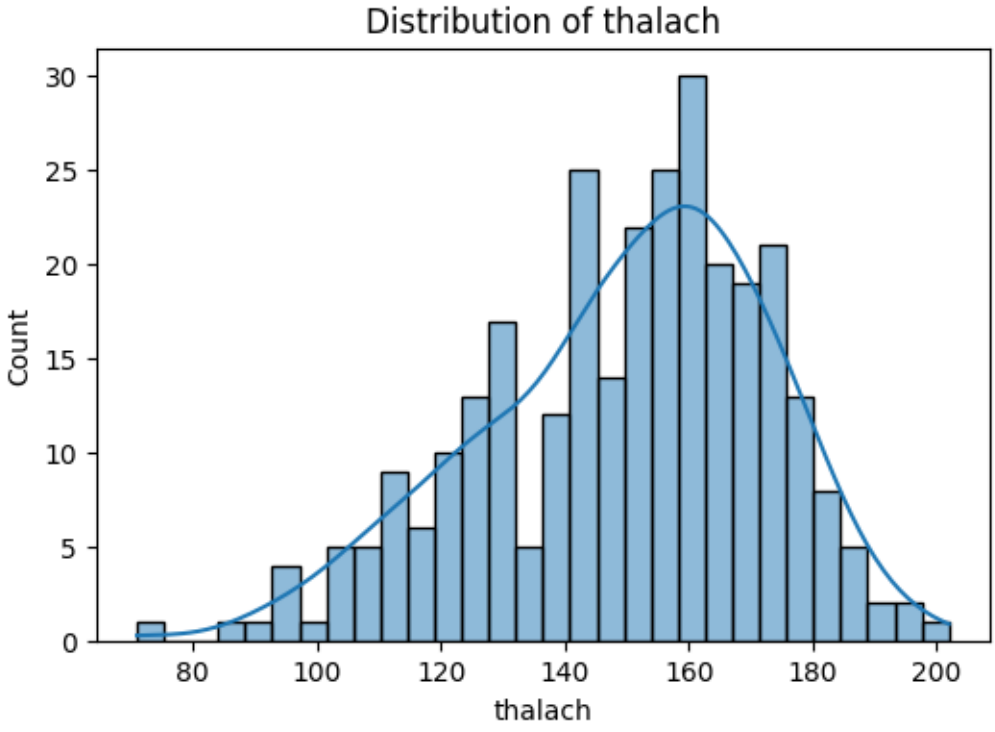

## 15 Distribution of Numerical Features

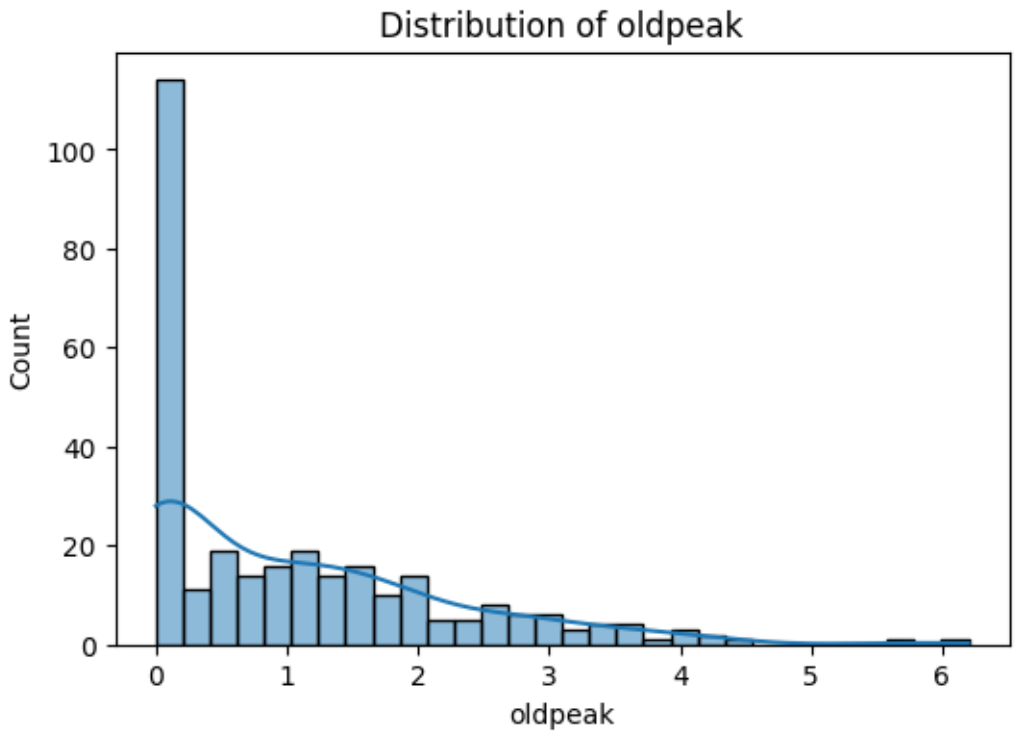
[19]:
```
numerical_cols = ['age', 'trestbps', 'chol', 'thalach', 'oldpeak']

for col in numerical_cols:
    plt.figure(figsize=(6, 4))
    sns.histplot(data[col], kde=True, bins=30)
    plt.title(f'Distribution of {col}')
    plt.show()
```


Distribution of age
Count
age
0
5
10
15
20
25
30
30
40
50
60
70


Distribution of trestbps
Count
trestbps
0
10
20
30
40
100
120
140
160
180
200

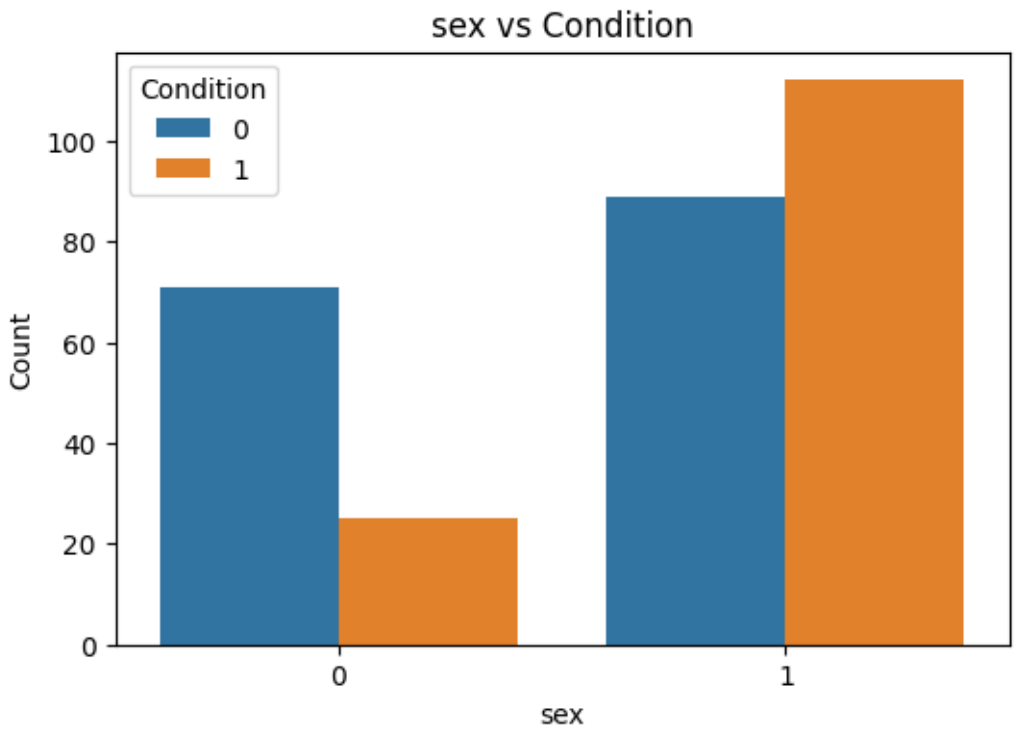

Distribution of chol
Count
chol
0
5
10
15
20
25
30
35
200
300
400
500

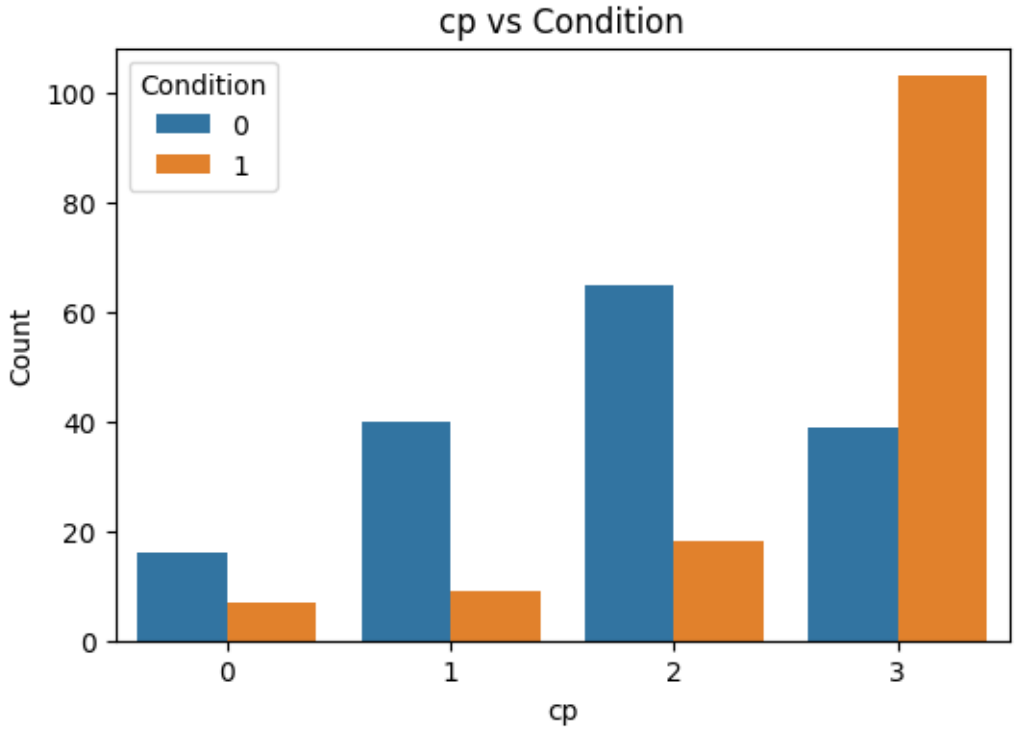

Distribution of thalach
Count
thalach
0
5
10
15
20
25
30
80
100
120
140
160
180
200

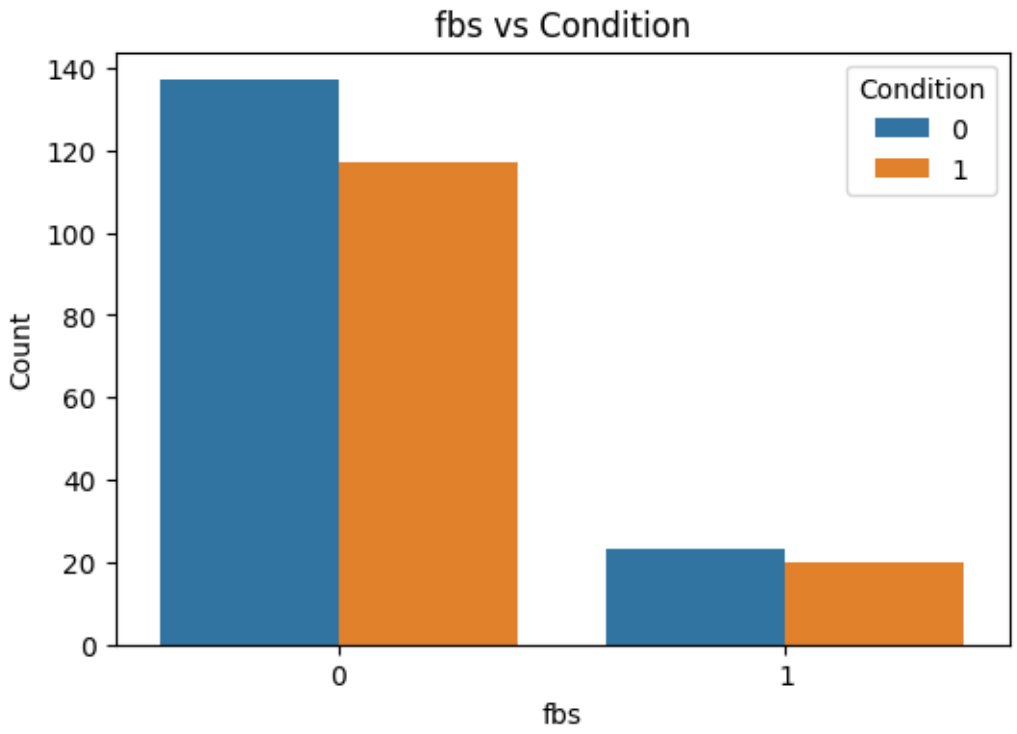


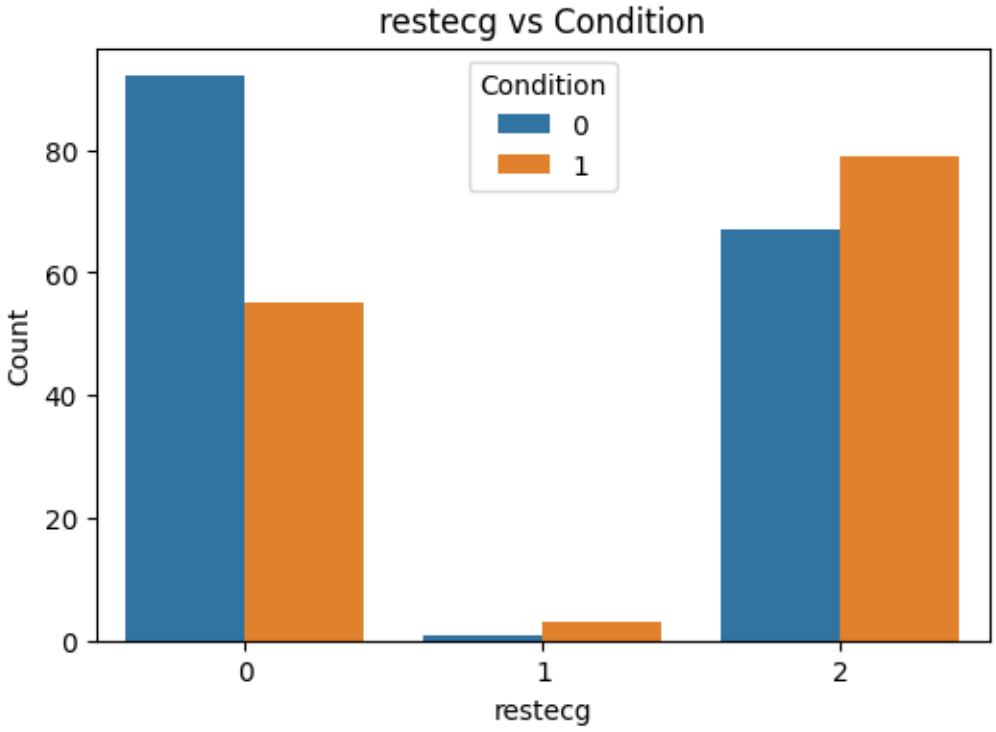

## 16 Categorical Features vs Target

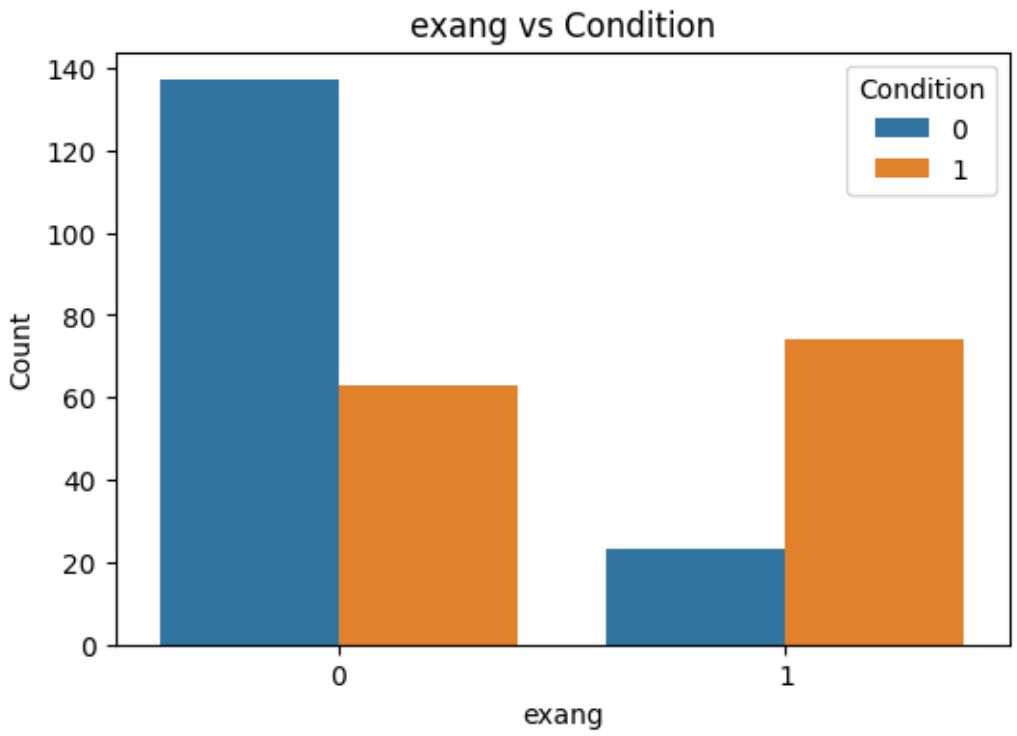
[20]:
```
categorical_cols = ['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca',␣
 ↪'thal']

for col in categorical_cols:
    plt.figure(figsize=(6, 4))
    sns.countplot(x=col, hue='condition', data=data)
    plt.title(f'{col} vs Condition')
    plt.xlabel(col)
    plt.ylabel('Count')
    plt.legend(title='Condition')
    plt.show()
```

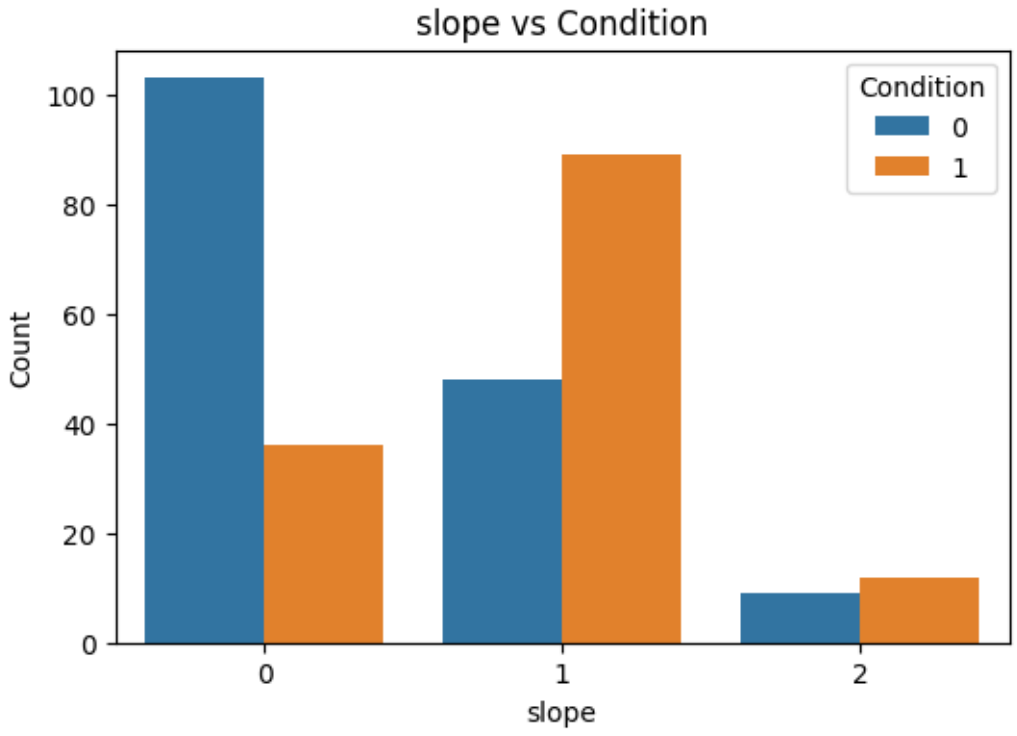

sex vs Condition
Condition
0
1
Count
0
20
40
60
80
100
0
1
sex

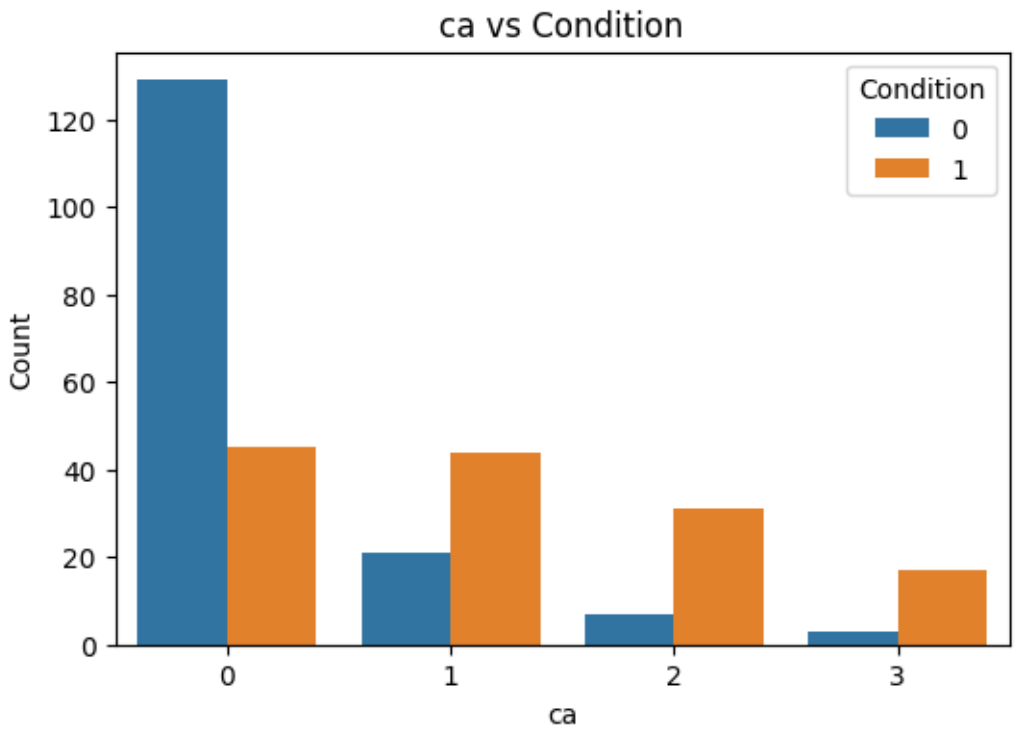

cp vs Condition
Condition
0
1
Count
0
20
40
60
80
100
0
1
2
3
cp

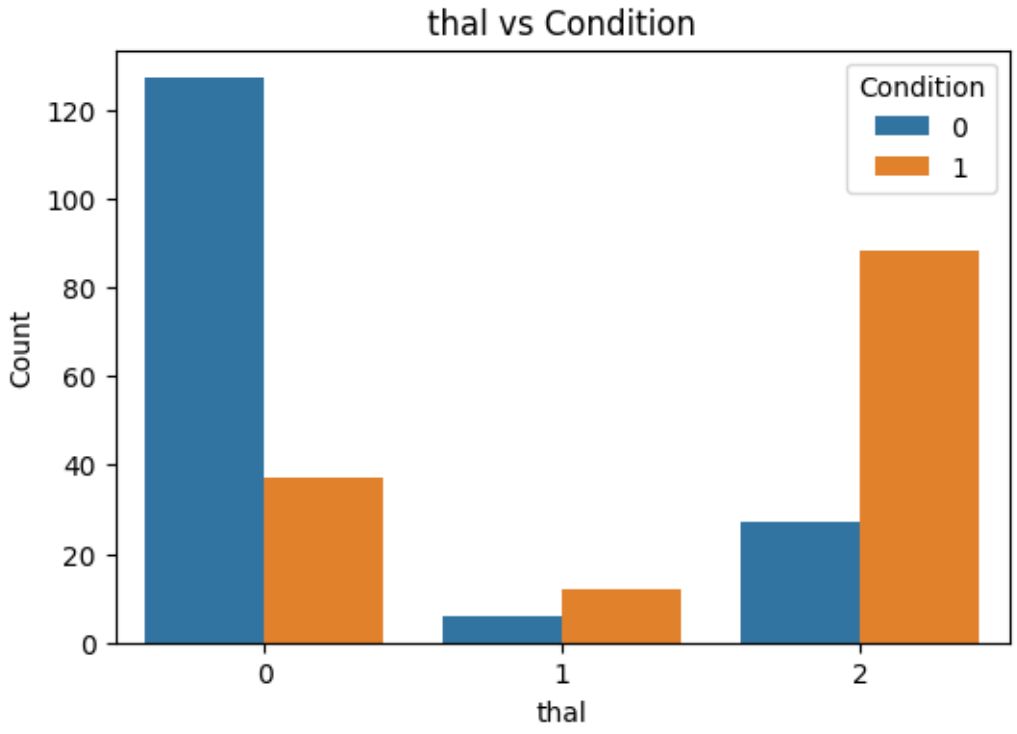

fbs vs Condition
Condition
0
1
Count
140
120
100
80
60
40
20
0
0
1
fbs


restecg vs Condition
Condition
0
1
Count
80
60
40
20
0
0
1
2
restecg


exang vs Condition
Condition
0
1
Count
140
120
100
80
60
40
20
0
0
1
exang

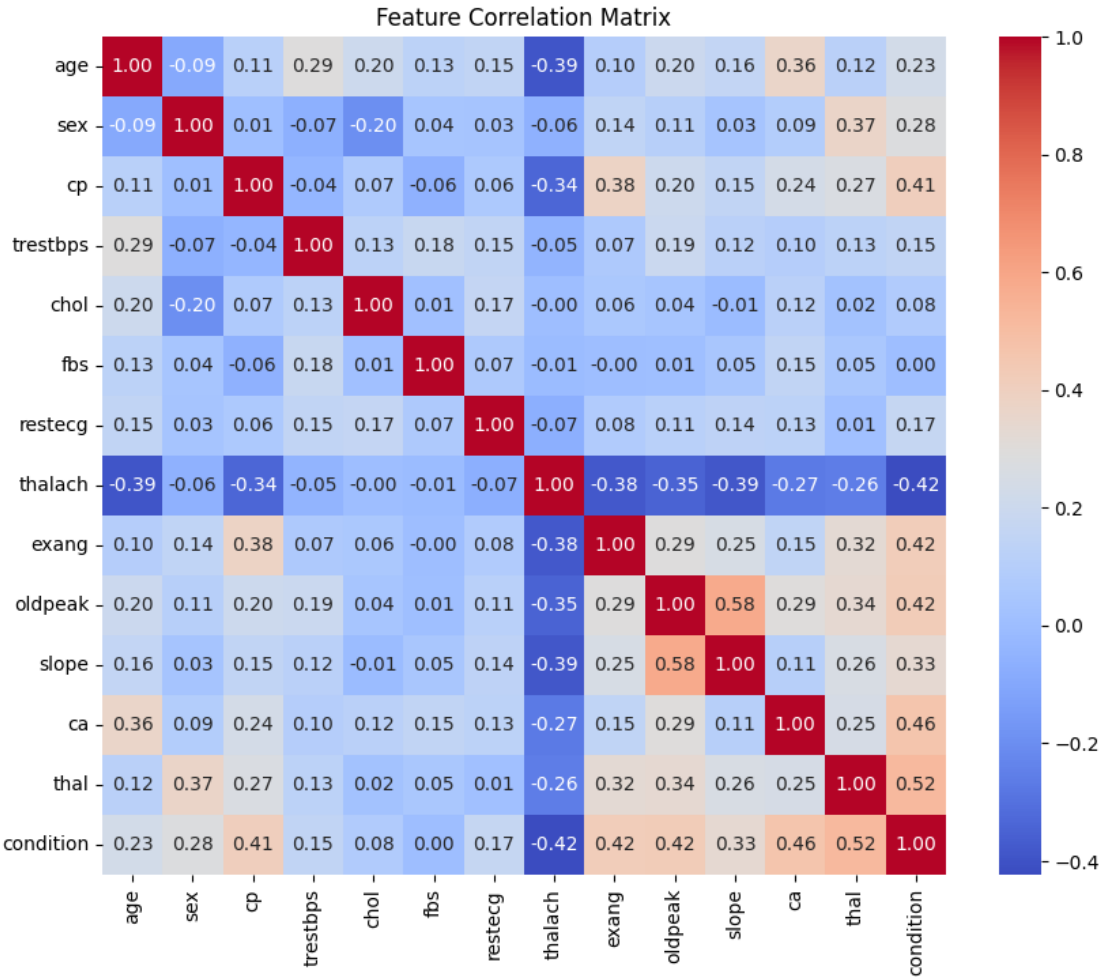

slope vs Condition
Condition
0
1
Count
100
80
60
40
20
0
0
1
2
slope


ca vs Condition
Condition
0
1
Count
120
100
80
60
40
20
0
0
1
2
3
ca


thal vs Condition
Condition
0
1
Count
120
100
80
60
40
20
0
0
1
2
thal

## 17 Correlation Heatmap

[21]:
```
plt.figure(figsize=(10,8))
sns.heatmap(data.corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Feature Correlation Matrix')
plt.show()
```




## 18 Prepare Data for Modeling

[22]:
```
X = data.drop('condition', axis=1)
y = data['condition']

# Train-test split (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=42, stratify=y)
```

## 19 Train Logistic Regression Model

[23]:
```
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)
```

[23]:
```
LogisticRegression(max_iter=1000)
```

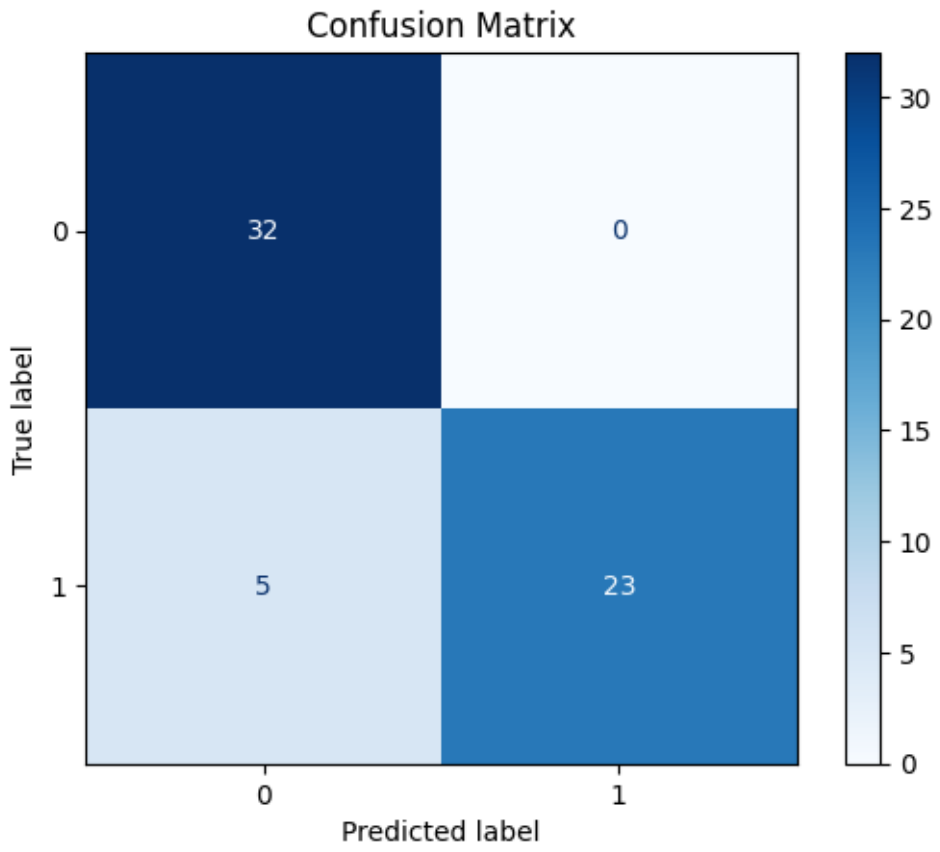
## 20 Make Predictions and Evaluate

[24]:
```
y_pred = model.predict(X_test)
y_prob = model.predict_proba(X_test)[:, 1]

accuracy = accuracy_score(y_test, y_pred)
roc_auc = roc_auc_score(y_test, y_prob)

print(f'Accuracy: {accuracy:.4f}')
print(f'ROC AUC: {roc_auc:.4f}')
```

```
Accuracy: 0.9167
ROC AUC: 0.9509
```

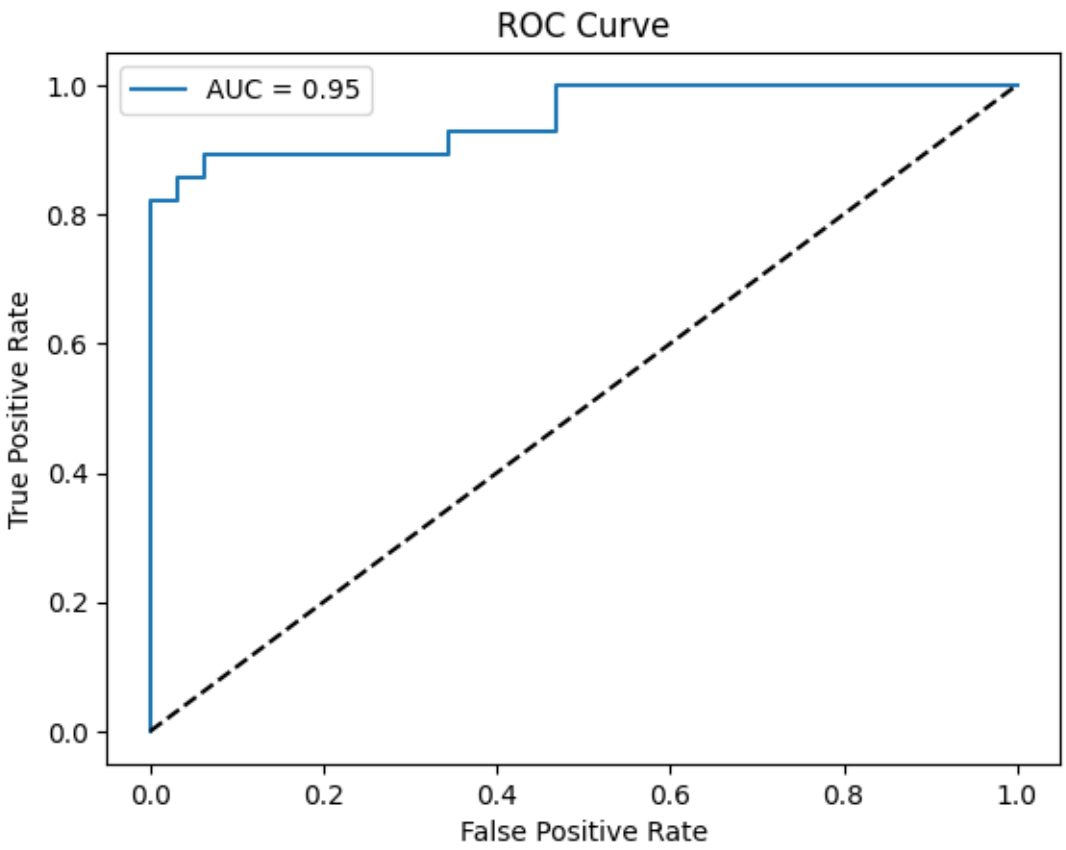
## 21 Plot Confusion Matrix

[25]:
```
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap='Blues')
plt.title('Confusion Matrix')
plt.show()
```




## 22 Plot ROC Curve

[26]:
```
fpr, tpr, thresholds = roc_curve(y_test, y_prob)

plt.plot(fpr, tpr, label=f'AUC = {roc_auc:.2f}')
plt.plot([0,1], [0,1], 'k--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve')
plt.legend()
plt.show()
```




## 23 Feature Importance (Logistic Regression Coefficients)

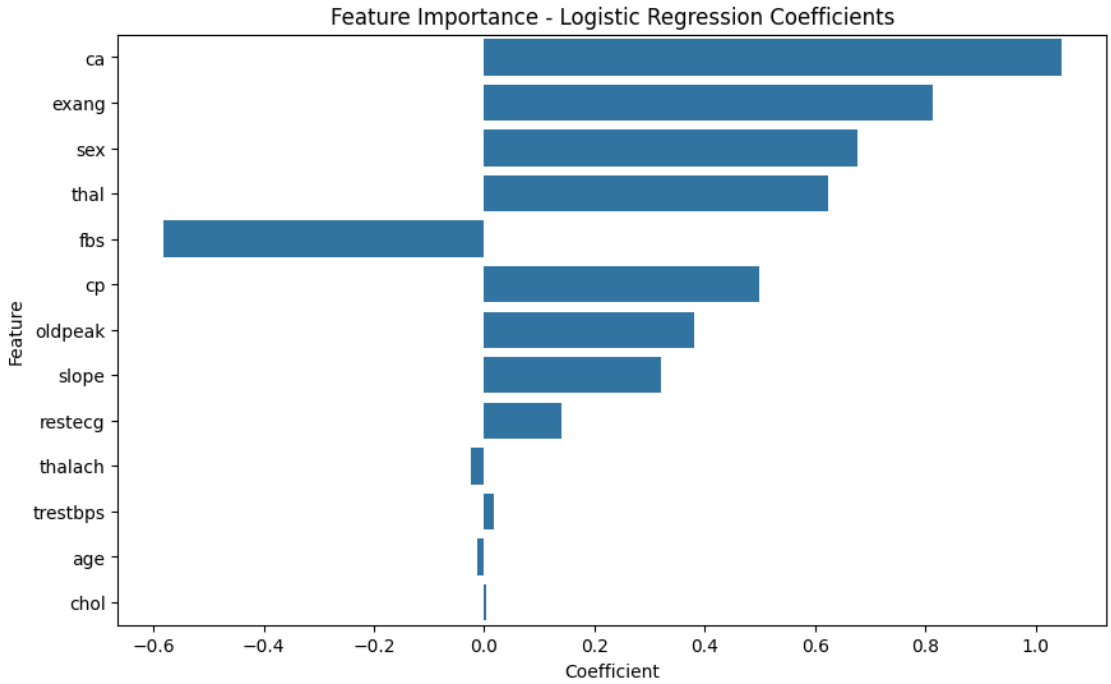
[27]:
```
features = X.columns
coefficients = model.coef_[0]

importance_df = pd.DataFrame({'Feature': features, 'Coefficient': coefficients})
importance_df['AbsCoefficient'] = importance_df['Coefficient'].abs()
importance_df = importance_df.sort_values(by='AbsCoefficient', ascending=False)

print("Important Features (based on logistic regression coefficients):")
print(importance_df[['Feature', 'Coefficient']])
```

```
Important Features (based on logistic regression coefficients):
     Feature  Coefficient
11        ca     1.046757
8      exang     0.812330
1        sex     0.675442
12      thal     0.623016
5        fbs    -0.582774
```

```
2          cp      0.498075
9     oldpeak      0.379411
10      slope      0.320660
6     restecg      0.139551
7     thalach     -0.023170
3    trestbps      0.017056
0         age     -0.013472
4        chol      0.002924
```

## 24 Visualize Feature Importance

[28]:
```
plt.figure(figsize=(10,6))
sns.barplot(x='Coefficient', y='Feature', data=importance_df)
plt.title('Feature Importance - Logistic Regression Coefficients')
plt.show()
```




[28]:
```
```

## 25 Task 5: Mental Health Support Chatbot (Fine-Tuned)

[29]:
```
!!wget https://dl.fbaipublicfiles.com/parlai/empatheticdialogues/
↪empatheticdialogues.tar.gz
```

[29]:
```
['--2025-06-25 07:19:55--  https://dl.fbaipublicfiles.com/parlai/empatheticdialo
gues/empatheticdialogues.tar.gz',
 'Resolving dl.fbaipublicfiles.com (dl.fbaipublicfiles.com)… 3.163.189.108,
3.163.189.96, 3.163.189.51, …',
 'Connecting to dl.fbaipublicfiles.com
(dl.fbaipublicfiles.com)|3.163.189.108|:443… connected.',
 'HTTP request sent, awaiting response… 200 OK',
 'Length: 28022709 (27M) [application/gzip]',
 'Saving to: ‘empatheticdialogues.tar.gz.1’',
 '',
 '',
 '           empatheti    0%[                    ]       0  --.-KB/s
',
 '          empathetic  32%[=====>               ]   8.80M  43.6MB/s
',
 'empatheticdialogues 100%[===================>]  26.72M  76.1MB/s    in 0.4s
',
 '',
 '2025-06-25 07:19:55 (76.1 MB/s) - ‘empatheticdialogues.tar.gz.1’ saved
[28022709/28022709]',
 '']
```

[30]:
```
!tar -xf empatheticdialogues.tar.gz
!ls empatheticdialogues
```

```
test.csv  train.csv  valid.csv
```

[31]:
```
!pip install -q transformers datasets accelerate
!pip install sentencepiece
```

```
Requirement already satisfied: sentencepiece in /usr/local/lib/python3.11/dist-
packages (0.2.0)
```

## 26 Load & Preprocess the Dataset

[32]:
```
import pandas as pd

# Load the dataset
df = pd.read_csv("/content/empatheticdialogues/train.csv", quoting=3,␣
 ↪on_bad_lines="skip")


df.head()
```

[32]:

| | conv_id | utterance_idx | context | \ |
|---|---|---|---|---|
| 0 | hit:0_conv:1 | 1 | sentimental | |
| 1 | hit:0_conv:1 | 2 | sentimental | |

```
2  hit:0_conv:1              3  sentimental
3  hit:0_conv:1              4  sentimental
4  hit:0_conv:1              5  sentimental

                                                prompt  speaker_idx  \
0  I remember going to the fireworks with my best…            1
1  I remember going to the fireworks with my best…            0
2  I remember going to the fireworks with my best…            1
3  I remember going to the fireworks with my best…            0
4  I remember going to the fireworks with my best…            1

                                           utterance     selfeval tags
0  I remember going to see the fireworks with my …  5|5|5_2|2|5  NaN
1  Was this a friend you were in love with_comma_…  5|5|5_2|2|5  NaN
2               This was a best friend. I miss her.  5|5|5_2|2|5  NaN
3                           Where has she gone?  5|5|5_2|2|5  NaN
4                            We no longer talk.  5|5|5_2|2|5  NaN
```

[33]:
```
# Combine context + situation as prompt; target is the utterance (empathetic response)
df['prompt'] = "Person: " + df['context'] + "\nYou: "
df['response'] = df['utterance']
```

## 27 Convert to Hugging Face Dataset Format

[34]:
```
from datasets import Dataset

# Create a smaller version for testing
df = df[['prompt', 'response']].dropna().sample(5000, random_state=42)

# Convert to Hugging Face Dataset
dataset = Dataset.from_pandas(df)
```

## 28 Tokenization

[35]:
```
# Install required libraries
!pip install -U transformers datasets
```

```
Requirement already satisfied: transformers in /usr/local/lib/python3.11/dist-
packages (4.52.4)
Requirement already satisfied: datasets in /usr/local/lib/python3.11/dist-
packages (3.6.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-
packages (from transformers) (3.18.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.30.0 in
```

```
/usr/local/lib/python3.11/dist-packages (from transformers) (0.33.0)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-
packages (from transformers) (2.0.2)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.11/dist-packages (from transformers) (24.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-
packages (from transformers) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.11/dist-packages (from transformers) (2024.11.6)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-
packages (from transformers) (2.32.3)
Requirement already satisfied: tokenizers<0.22,>=0.21 in
/usr/local/lib/python3.11/dist-packages (from transformers) (0.21.1)
Requirement already satisfied: safetensors>=0.4.3 in
/usr/local/lib/python3.11/dist-packages (from transformers) (0.5.3)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.11/dist-
packages (from transformers) (4.67.1)
Requirement already satisfied: pyarrow>=15.0.0 in
/usr/local/lib/python3.11/dist-packages (from datasets) (18.1.0)
Requirement already satisfied: dill<0.3.9,>=0.3.0 in
/usr/local/lib/python3.11/dist-packages (from datasets) (0.3.7)
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages
(from datasets) (2.2.2)
Requirement already satisfied: xxhash in /usr/local/lib/python3.11/dist-packages
(from datasets) (3.5.0)
Requirement already satisfied: multiprocess<0.70.17 in
/usr/local/lib/python3.11/dist-packages (from datasets) (0.70.15)
Requirement already satisfied: fsspec<=2025.3.0,>=2023.1.0 in
/usr/local/lib/python3.11/dist-packages (from
fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (2025.3.0)
Requirement already satisfied: aiohttp!=4.0.0a0,!=4.0.0a1 in
/usr/local/lib/python3.11/dist-packages (from
fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (3.11.15)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.11/dist-packages (from huggingface-
hub<1.0,>=0.30.0->transformers) (4.14.0)
Requirement already satisfied: hf-xet<2.0.0,>=1.1.2 in
/usr/local/lib/python3.11/dist-packages (from huggingface-
hub<1.0,>=0.30.0->transformers) (1.1.4)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.11/dist-packages (from requests->transformers) (3.4.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-
packages (from requests->transformers) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.11/dist-packages (from requests->transformers) (2.4.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.11/dist-packages (from requests->transformers)
(2025.6.15)
```

```
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.11/dist-packages (from pandas->datasets) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-
packages (from pandas->datasets) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-
packages (from pandas->datasets) (2025.2)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in
/usr/local/lib/python3.11/dist-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (2.6.1)
Requirement already satisfied: aiosignal>=1.1.2 in
/usr/local/lib/python3.11/dist-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (1.3.2)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.11/dist-
packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets)
(25.3.0)
Requirement already satisfied: frozenlist>=1.1.1 in
/usr/local/lib/python3.11/dist-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (1.7.0)
Requirement already satisfied: multidict<7.0,>=4.5 in
/usr/local/lib/python3.11/dist-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (6.4.4)
Requirement already satisfied: propcache>=0.2.0 in
/usr/local/lib/python3.11/dist-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (0.3.2)
Requirement already satisfied: yarl<2.0,>=1.17.0 in
/usr/local/lib/python3.11/dist-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets)
(1.20.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-
packages (from python-dateutil>=2.8.2->pandas->datasets) (1.17.0)
```

[36]:
```python
import os
from transformers import AutoTokenizer, AutoModelForCausalLM, pipeline
from datasets import load_dataset

# Set timeout to avoid connection issues
os.environ['HF_HUB_TIMEOUT'] = '60'

# Define model checkpoint
model_checkpoint = "distilgpt2"

# Load tokenizer and set pad_token to eos_token
tokenizer = AutoTokenizer.from_pretrained(model_checkpoint)
tokenizer.pad_token = tokenizer.eos_token

# Load model
```

```
model = AutoModelForCausalLM.from_pretrained(model_checkpoint)

# Define tokenization function
def tokenize(batch):
    texts = [f"{p} {r}" for p, r in zip(batch["prompt"], batch["response"])]
    tokenized = tokenizer(texts, padding="max_length", truncation=True,␣
 ↪max_length=128)
    tokenized["labels"] = tokenized["input_ids"].copy()
    return tokenized

# Tokenize the dataset
tokenized_dataset = dataset.map(tokenize, batched=True)
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94:
UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab
(https://huggingface.co/settings/tokens), set it as secret in your Google Colab
and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access
public models or datasets.
  warnings.warn(
```

```
Map:   0%|          | 0/5000 [00:00<?, ? examples/s]
```

## 29 Fine-Tune the Model with Trainer API

[37]:
```
from transformers import AutoModelForCausalLM, Trainer, TrainingArguments,␣
 ↪DataCollatorForLanguageModeling

# Load model
model = AutoModelForCausalLM.from_pretrained(model_checkpoint)

# Define training arguments
training_args = TrainingArguments(
    output_dir="./mental-health-chatbot",
    per_device_train_batch_size=4,
    num_train_epochs=3,
    save_strategy="epoch",
    logging_dir="./logs",
    report_to="none"
)

# Data collator
data_collator = DataCollatorForLanguageModeling(tokenizer=tokenizer, mlm=False)
```

```
# Trainer setup
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_dataset,
    tokenizer=tokenizer,
    data_collator=data_collator
)

# Train model
trainer.train()
```

```
/tmp/ipython-input-37-1698186836.py:20: FutureWarning: `tokenizer` is deprecated
and will be removed in version 5.0.0 for `Trainer.__init__`. Use
`processing_class` instead.
  trainer = Trainer(
`loss_type=None` was set in the config but it is unrecognised.Using the default
loss: `ForCausalLMLoss`.
```

```
<IPython.core.display.HTML object>
```

[37]:
```
TrainOutput(global_step=3750, training_loss=2.3889587727864585,
metrics={'train_runtime': 481.3998, 'train_samples_per_second': 31.159,
'train_steps_per_second': 7.79, 'total_flos': 489931407360000.0, 'train_loss':
2.3889587727864585, 'epoch': 3.0})
```

## 30 Save the Fine-Tuned Model

[38]:
```
trainer.save_model("empathetic-chatbot-model")
tokenizer.save_pretrained("empathetic-chatbot-model")
```

[38]:
```
('empathetic-chatbot-model/tokenizer_config.json',
 'empathetic-chatbot-model/special_tokens_map.json',
 'empathetic-chatbot-model/vocab.json',
 'empathetic-chatbot-model/merges.txt',
 'empathetic-chatbot-model/added_tokens.json',
 'empathetic-chatbot-model/tokenizer.json')
```

[39]:
```
!zip -r empathetic-chatbot-model.zip empathetic-chatbot-model/
```

```
  adding: empathetic-chatbot-model/ (stored 0%)
  adding: empathetic-chatbot-model/special_tokens_map.json (deflated 60%)
  adding: empathetic-chatbot-model/model.safetensors (deflated 7%)
  adding: empathetic-chatbot-model/config.json (deflated 52%)
  adding: empathetic-chatbot-model/tokenizer_config.json (deflated 54%)
  adding: empathetic-chatbot-model/generation_config.json (deflated 24%)
  adding: empathetic-chatbot-model/tokenizer.json (deflated 82%)
```

```
  adding: empathetic-chatbot-model/merges.txt (deflated 53%)
  adding: empathetic-chatbot-model/training_args.bin (deflated 52%)
  adding: empathetic-chatbot-model/vocab.json (deflated 59%)
```

[39]:

[40]:
```
import getpass
username = "zoya4477"
token = getpass.getpass("Enter your GitHub token: ")
```

```
Enter your GitHub token: ··········
```

[41]:
```
!git add .
!git commit -m "Trained empathetic-chatbot-model"
```

```
fatal: not a git repository (or any of the parent directories): .git
fatal: not a git repository (or any of the parent directories): .git
```

[42]:
```
!git pull https://github.com/zoya4477/AI-Ml.git
!git push https://{username}:{token}@github.com/zoya4477/AI-Ml.git
```

```
fatal: not a git repository (or any of the parent directories): .git
fatal: not a git repository (or any of the parent directories): .git
```

[42]:

## 31 Task 6: House Price Prediction

[43]:
```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_absolute_error, mean_squared_error
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import GradientBoostingRegressor
```

[44]:
```
# unzip the file

!unzip /content/house-prices-advanced-regression-techniques.zip
```

```
Archive:  /content/house-prices-advanced-regression-techniques.zip
  inflating: data_description.txt
  inflating: sample_submission.csv
```

```
  inflating: test.csv
  inflating: train.csv
```

[45]:
```
# Load the dataset
data = pd.read_csv("train.csv")
# Show basic info
print(data[['GrLivArea', 'BedroomAbvGr', 'Neighborhood', 'SalePrice']].
 ↪describe())
```

```
         GrLivArea  BedroomAbvGr      SalePrice
count  1460.000000   1460.000000    1460.000000
mean   1515.463699      2.866438  180921.195890
std     525.480383      0.815778   79442.502883
min     334.000000      0.000000   34900.000000
25%    1129.500000      2.000000  129975.000000
50%    1464.000000      3.000000  163000.000000
75%    1776.750000      3.000000  214000.000000
max    5642.000000      8.000000  755000.000000
```

[46]:
```
# Select relevant features
df = data[['GrLivArea', 'BedroomAbvGr', 'Neighborhood', 'SalePrice']]

# Encode categorical features (Neighborhood)
df = pd.get_dummies(df, columns=['Neighborhood'], drop_first=True)
```

[47]:
```
# Split features and target
X = df.drop('SalePrice', axis=1)
y = df['SalePrice']

# Train/test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)
```

[48]:
```
# Scale the numeric features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

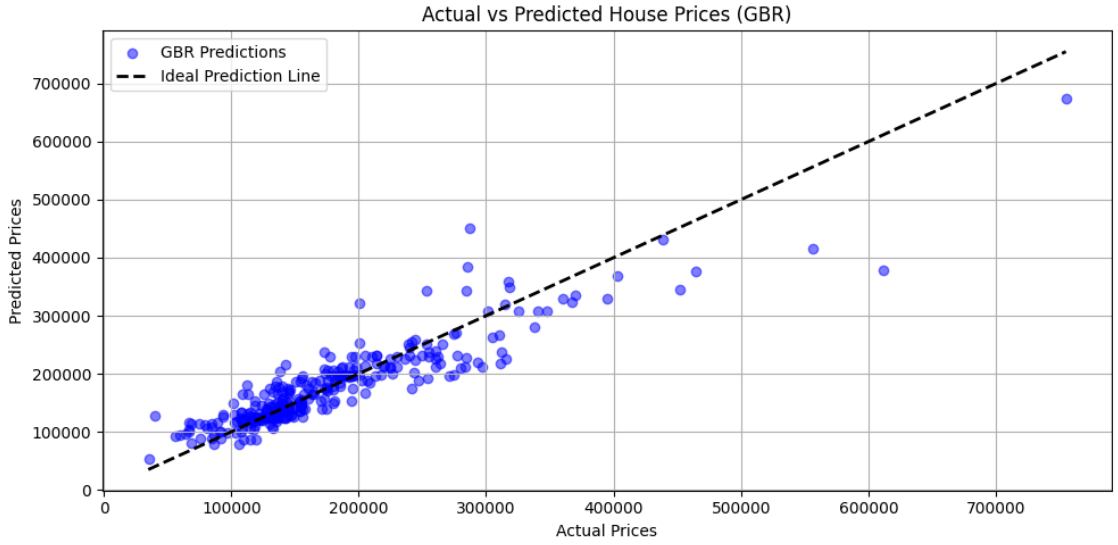
[49]:
```
# ----------- LINEAR REGRESSION -----------
lr_model = LinearRegression()
lr_model.fit(X_train_scaled, y_train)
lr_preds = lr_model.predict(X_test_scaled)
```

[50]:
```
# -------- GRADIENT BOOSTING REGRESSOR -------
gbr_model = GradientBoostingRegressor()
gbr_model.fit(X_train, y_train)  # No scaling needed for GBR
gbr_preds = gbr_model.predict(X_test)
```

[51]:
```
# Evaluation Function
def evaluate_model(name, y_true, y_pred):
    mae = mean_absolute_error(y_true, y_pred)
    rmse = np.sqrt(mean_squared_error(y_true, y_pred))
    print(f"{name} - MAE: {mae:.2f}, RMSE: {rmse:.2f}")

evaluate_model("Linear Regression", y_test, lr_preds)
evaluate_model("Gradient Boosting", y_test, gbr_preds)
```

```
Linear Regression - MAE: 27380.29, RMSE: 41835.27
Gradient Boosting - MAE: 24716.54, RMSE: 36822.92
```

[52]:
```
# ---------- Visualization ---------------
plt.figure(figsize=(10, 5))
plt.scatter(y_test, gbr_preds, alpha=0.5, label='GBR Predictions', color='blue')
plt.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=2, label='Ideal␣
 ↪Prediction Line')
plt.xlabel("Actual Prices")
plt.ylabel("Predicted Prices")
plt.title("Actual vs Predicted House Prices (GBR)")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```




[52]: