Full-name of dear student: Thai Thi Kim Yen

Dear student code: 47.01.104.250

Linear regression

About dataset

Independent fields: Age, RestBP.

About Dependent field, "con" choose Chol.

Import libraries

In [34]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
```

Loading data

In [35]:

```
data = pd.read_csv('Heart.csv', usecols = ['Age', 'RestBP', 'Chol'])
data.head()
```

Out[35]:

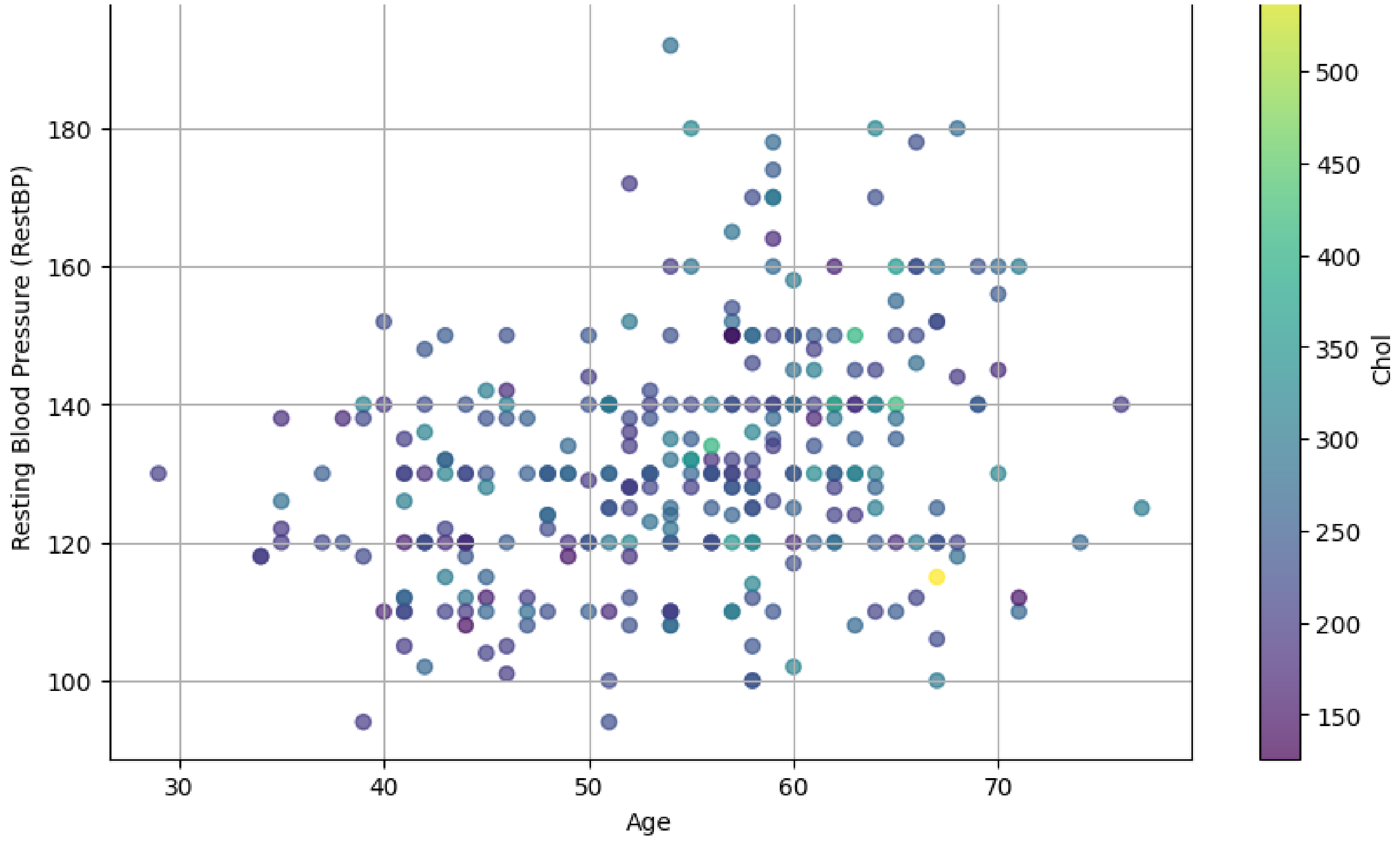
	Age	RestBP	Chol
0	63	145	233
1	67	160	286
2	67	120	229
3	37	130	250
4	41	130	204

Visualize the Data

In [36]:

```
plt.figure(figsize=(10, 6))
plt.scatter(data['Age'], data['RestBP'], c=data['Chol'], cmap='viridis', alpha=0.7)
plt.xlabel('Age')
plt.ylabel('Resting Blood Pressure (RestBP)')
plt.title('Scatter Plot of Age vs Resting Blood Pressure (RestBP) colored by Chol')
plt.colorbar(label='Chol')
plt.grid(True)
plt.show()
```

Scatter Plot of Age vs Resting Blood Pressure (RestBP) colored by Chol
200
550



Split data into train and test sets

In [37]:

```
X = data[['Age', 'RestBP']]
y = data['Chol']

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

Fitting and Evaluating the Model

In [38]:

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

print("Coefficients:", model.coef_)
print("Intercept:", model.intercept_)
```

```
Coefficients: [0.80766582 0.2036818 ]
Intercept: 174.41016111348628
```

Evaluate the model

In [39]:

```
y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse}")
print(f"R-squared: {r2}")
```

```
Mean Squared Error: 3793.5615470970715
R-squared: 0.06262322353974825
```

Conclusion

- The MSE value of approximately 3793.56 indicates the average squared difference between the actual cholesterol ('Chol') values and the predicted values from the linear regression model.
- The R2 value of approximately 0.0626 indicates that the linear regression model explains only a very small proportion of the variance in 'Chol' values. This suggests that the linear relationship between 'Age', 'RestBP', and 'Chol' is weak.

Logistic regression

About dataset

Independent fields: Age, RestBP, Chol.

Target field: AHD.

Import libraries

In [40]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
```

Loading data

In [41]:

```
data = pd.read_csv('Heart.csv', usecols = ['Age', 'RestBP', 'Chol', 'AHD'])
data.head()
```

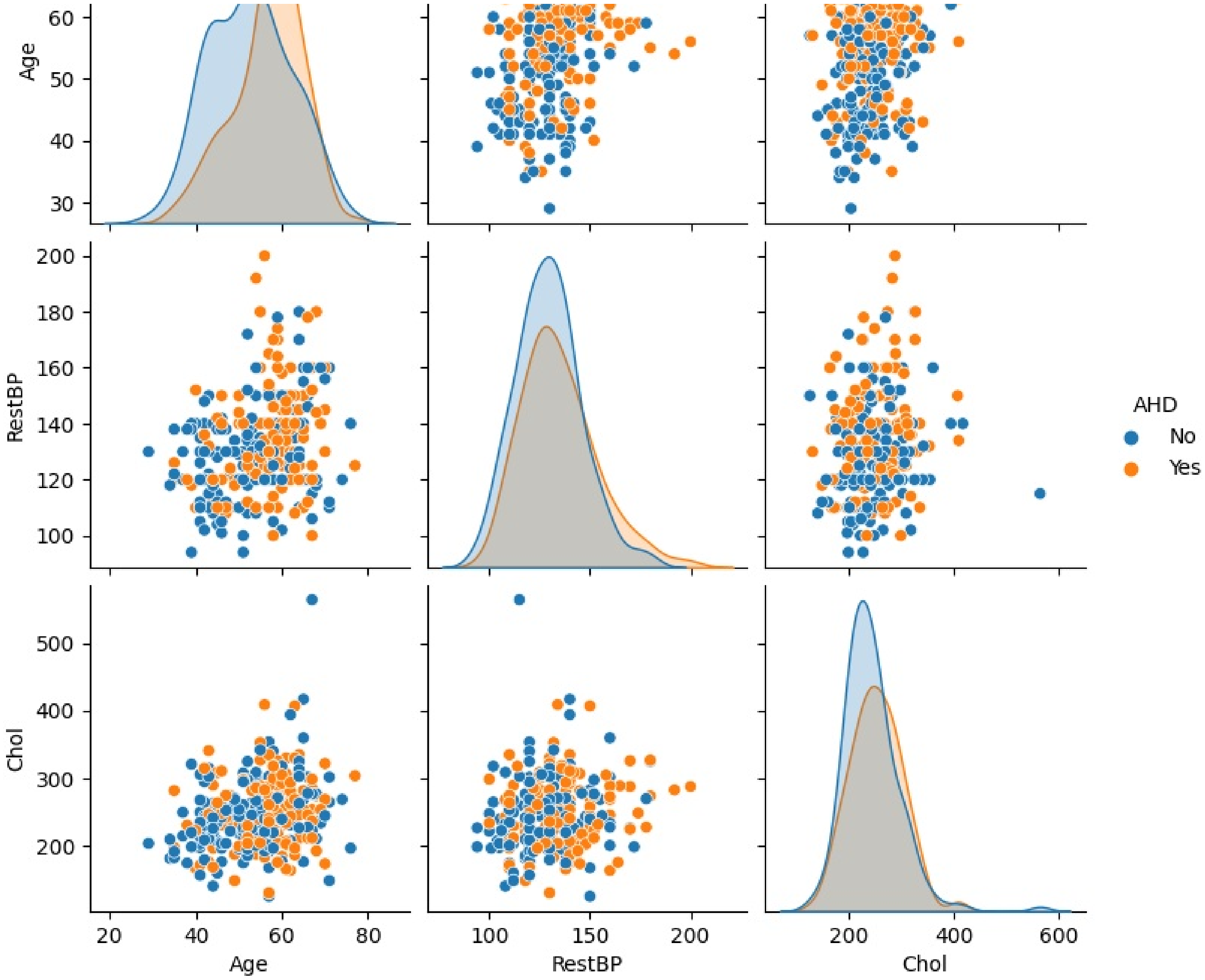
Out[41]:

	Age	RestBP	Chol	AHD
0	63	145	233	No
1	67	160	286	Yes
2	67	120	229	Yes
3	37	130	250	No
4	41	130	204	No

Visualize the Data

In [42]:

```
sns.pairplot(data, hue='AHD', diag_kind='kde')
plt.show()
```



Split data into train and test sets

In [43]:

```
X = data[['Age', 'RestBP', 'Chol']]
y = data['AHD']

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

Fitting and Evaluating the Model

In [44]:

```
clf = LogisticRegression()
clf.fit(X_train, y_train)
```

Out[44]:

```
▾ LogisticRegression
LogisticRegression()
```

Evaluate the model

In [45]:

```
y_pred = clf.predict(X_test)
```

```
print('Accuracy:', accuracy_score(y_test, y_pred))
print('Confusion matrix:')
print(confusion_matrix(y_test, y_pred))
```

```
Accuracy: 0.5573770491803278
Confusion matrix:
[[23  6]
 [21 11]]
```

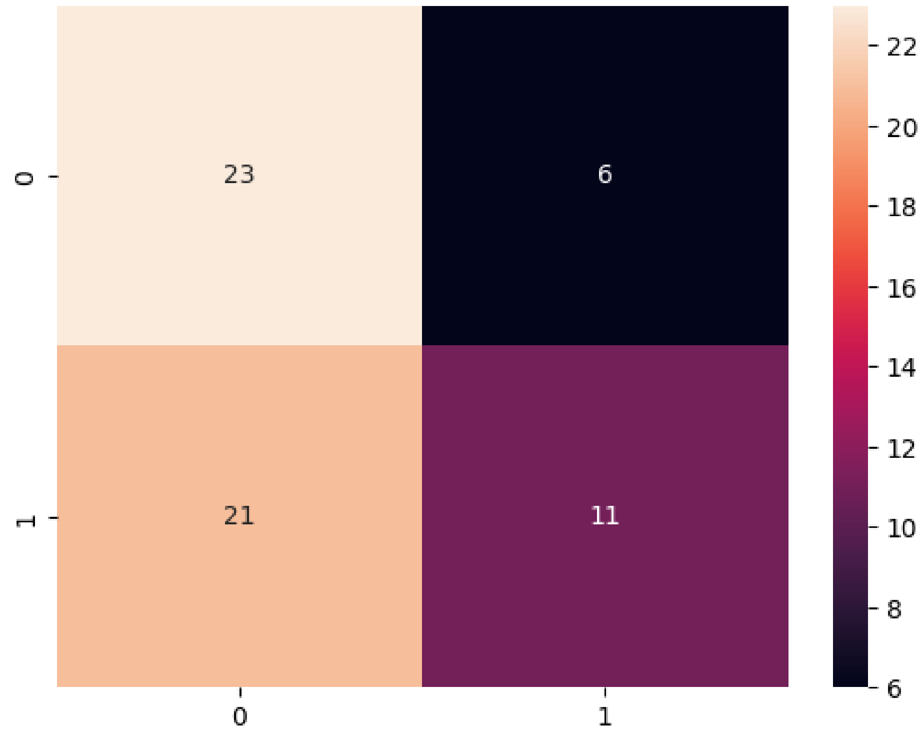
In [47]:

```
cm = confusion_matrix(y_test, y_pred)
TP = cm[0][0]
TN = cm[1][1]
FP = cm[0][1]
FN = cm[1][0]

accuracy = (TP + TN) / (TP + TN + FP + FN)
precision = TP / (TP + FP)
recall = TP / (TP + FN)
f1_score = 2 * (precision * recall) / (precision + recall)

print('Accuracy:', accuracy)
print('Precision:', precision)
print('Recall:', recall)
print('F1 Score:', f1_score)
```

```
Accuracy: 0.5573770491803278
Precision: 0.7931034482758621
Recall: 0.5227272727272727
F1 Score: 0.6301369863013699
```

In [48]:

```
sns.heatmap(cm, annot=True)
plt.show()
```



Reference

[1] Ngo Quoc Viet, Machine Learning lecture slides. (Bài giảng của thầy Việt dzễ thương)

[2] Understanding Logistic Regression in Python Tutorial, DataCamp.
https://www.datacamp.com/tutorial/understanding-logistic-regression-python