Supervised Learning: Regression

5 min

Machine learning can be branched out into the following categories:

- Supervised Learning
- Unsupervised Learning

Supervised Learning is where the data is labeled and the program learns to predict the output from the input data. For instance, a supervised learning algorithm for credit card fraud detection would take as input a set of recorded transactions. For each transaction, the program would predict if it is fraudulent or not.

Supervised learning problems can be further grouped into regression and classification problems.

Regression:

In regression problems, we are trying to predict a continuous-valued output. Examples are:

- What is the housing price in New York?
- What is the value of cryptocurrencies?

Classification:

In classification problems, we are trying to predict a discrete number of values. Examples are:

- Is this a picture of a human or a picture of a cyborg?
- Is this email spam?

For a quick preview, here's an example of a regression problem.

A real estate company wants to analyze housing costs in New York. They built a linear regression model to predict rent prices from two variables: the square footage of each apartment and the number of burglaries in the apartment's neighborhood during the past year.

Instructions

1. Checkpoint 1 Passed

1.

Run the code to train the model and predict the monthly rent for a 950-square-foot apartment in a neighborhood with 2 burglaries in the last year.

Stuck? Get extra guidance

2. Checkpoint 2 Passed

2.

Change the variables square_footage and number_of_burglaries to different values. Run the code again. How do those changes affect the predicted rent?

If you want to experiment more you can change the values to other numbers and see how that affects the predicted rent.

script.py

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression

# Load the data
housing_data = pd.read_csv('housing_data.csv')
X = housing_data[['Sq ft', 'Burglaries']]
y = housing_data['Rent']

# Create the model
reg = LinearRegression()

# Train the model
reg.fit(X, y)

square_footage = 850
number_of_burglaries = 44

y_pred = reg.predict(np.array([square_footage, number_of_burglaries]).reshape(1, 2))

print(y_pred)
```

housing_data.csv

```
Sq ft,Burglaries,Rent
916.0,15.0,2490.0
571.0,8.0,890.0
1019.0,7.0,2050.0
1277.0,1.0,3250.0
1221.0,16.0,2770.0
1825.0,8.0,4450.0
1682.0,16.0,3280.0
```

1306.0,16.0,2900.0

872.0,12.0,990.0

1477.0,10.0,2600.0

837.0,5.0,2230.0

821.0,7.0,1220.0

1062.0,9.0,2370.0

826.0,11.0,2170.0

926.0,0.0,2280.0

1912.0,10.0,4070.0

1280.0,3.0,2990.0

1714.0,19.0,3920.0

880.0,16.0,2270.0

1463.0,11.0,3550.0

1265.0,17.0,2130.0

1097.0,0.0,2450.0

1531.0,4.0,3610.0

1280.0,0.0,3240.0

1318.0,17.0,3560.0

721.0,16.0,1420.0

1570.0,3.0,3400.0

1289.0,14.0,1770.0

966.0,10.0,2790.0

1100.0,9.0,1980.0

944.0,6.0,1920.0

1329.0,2.0,3630.0

1241.0,3.0,1940.0

850.0,10.0,2050.0

808.0,1.0,2780.0

972.0,8.0,2270.0

624.0,15.0,1680.0

1367.0,4.0,3050.0

1162.0,10.0,3090.0

840.0,12.0,1150.0

1817.0,7.0,3930.0

1382.0,3.0,2860.0

1435.0,6.0,3760.0

912.0,7.0,2700.0

1891.0,7.0,3230.0

907.0,14.0,1850.0

1653.0,15.0,3320.0

904.0,3.0,2290.0

643.0,11.0,2400.0

1279.0,17.0,3250.0

1453.0,15.0,2600.0

780.0,5.0,2850.0

1570.0,1.0,3260.0

1246.0,17.0,3030.0

1548.0,4.0,4370.0

1156.0,7.0,2590.0

1253.0,8.0,2110.0

1766.0,16.0,3520.0

1646.0,11.0,3540.0

1538.0,14.0,3300.0

1314.0,3.0,2920.0

1150.0,11.0,2930.0

1850.0,15.0,3860.0

1727.0,10.0,4280.0

1146.0,19.0,2970.0

1077.0,13.0,2830.0

1444.0,12.0,2960.0

1146.0,11.0,2530.0

1551.0,14.0,3060.0

647.0,12.0,2360.0

996.0,4.0,2480.0

1067.0,9.0,3010.0

1414.0,15.0,3940.0

2061.0,13.0,4340.0

905.0,10.0,3160.0

1317.0,3.0,3020.0

1794.0,10.0,3890.0

1124.0,7.0,2720.0

1903.0,4.0,5780.0

1605.0,3.0,3710.0

905.0,9.0,2210.0

1435.0,12.0,3180.0

863.0,1.0,1690.0

1622.0,15.0,3160.0

1128.0,4.0,3000.0

1957.0,17.0,3140.0

1564.0,5.0,2870.0

1024.0,7.0,1520.0

2086.0,9.0,3710.0

683.0,14.0,1300.0

1217.0,14.0,3090.0

1317.0,9.0,3560.0

1365.0,17.0,2940.0

909.0,5.0,2320.0

1614.0,8.0,3720.0

899.0,14.0,2030.0

723.0,5.0,2400.0

340.0,15.0,1250.0

1512.0,19.0,3710.0

1458.0,15.0,2320.0