

??????? 017138 (8 ??????)

[Ask a question about this product](#)

????????????

```

1  # Import the necessary libraries
2  import pandas as pd
3  import numpy as np
4  import matplotlib.pyplot as plt
5  import seaborn as sns
6  from sklearn.preprocessing import StandardScaler
7  from sklearn.model_selection import train_test_split
8  from sklearn.metrics import mean_squared_error, r2_score
9  from sklearn.linear_model import LinearRegression
10 from sklearn.ensemble import RandomForestRegressor
11 from sklearn.svm import SVR
12 from sklearn.neighbors import KNeighborsRegressor
13 from sklearn.tree import DecisionTreeRegressor
14 from sklearn.metrics import mean_absolute_error
15
16 # Load the dataset
17 data = pd.read_csv('data.csv')
18
19 # Display the first few rows of the dataset
20 data.head()
21
22 # Check the data types of the columns
23 data.dtypes
24
25 # Check for missing values
26 data.isnull().sum()
27
28 # Drop rows with missing values
29 data = data.dropna()
30
31 # Split the data into features (X) and target variable (y)
32 X = data[['feature1', 'feature2', 'feature3']]
33 y = data['target']
34
35 # Split the data into training and testing sets
36 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
37
38 # Standardize the features
39 scaler = StandardScaler()
40 X_train = scaler.fit_transform(X_train)
41 X_test = scaler.transform(X_test)
42
43 # Train the Linear Regression model
44 lr = LinearRegression()
45 lr.fit(X_train, y_train)
46
47 # Predict the target variable using the Linear Regression model
48 y_pred_lr = lr.predict(X_test)
49
50 # Calculate the Mean Squared Error (MSE) for the Linear Regression model
51 mse_lr = mean_squared_error(y_test, y_pred_lr)
52
53 # Calculate the R-squared score for the Linear Regression model
54 r2_lr = r2_score(y_test, y_pred_lr)
55
56 # Train the Random Forest Regressor model
57 rf = RandomForestRegressor()
58 rf.fit(X_train, y_train)
59
60 # Predict the target variable using the Random Forest Regressor model
61 y_pred_rf = rf.predict(X_test)
62
63 # Calculate the Mean Squared Error (MSE) for the Random Forest Regressor model
64 mse_rf = mean_squared_error(y_test, y_pred_rf)
65
66 # Calculate the R-squared score for the Random Forest Regressor model
67 r2_rf = r2_score(y_test, y_pred_rf)
68
69 # Train the Support Vector Regression (SVR) model
70 svr = SVR()
71 svr.fit(X_train, y_train)
72
73 # Predict the target variable using the SVR model
74 y_pred_svr = svr.predict(X_test)
75
76 # Calculate the Mean Squared Error (MSE) for the SVR model
77 mse_svr = mean_squared_error(y_test, y_pred_svr)
78
79 # Calculate the R-squared score for the SVR model
80 r2_svr = r2_score(y_test, y_pred_svr)
81
82 # Train the K-Nearest Neighbors (KNN) model
83 knn = KNeighborsRegressor()
84 knn.fit(X_train, y_train)
85
86 # Predict the target variable using the KNN model
87 y_pred_knn = knn.predict(X_test)
88
89 # Calculate the Mean Squared Error (MSE) for the KNN model
90 mse_knn = mean_squared_error(y_test, y_pred_knn)
91
92 # Calculate the R-squared score for the KNN model
93 r2_knn = r2_score(y_test, y_pred_knn)
94
95 # Train the Decision Tree Regressor model
96 dt = DecisionTreeRegressor()
97 dt.fit(X_train, y_train)
98
99 # Predict the target variable using the Decision Tree Regressor model
100 y_pred_dt = dt.predict(X_test)
101
102 # Calculate the Mean Squared Error (MSE) for the Decision Tree Regressor model
103 mse_dt = mean_squared_error(y_test, y_pred_dt)
104
105 # Calculate the R-squared score for the Decision Tree Regressor model
106 r2_dt = r2_score(y_test, y_pred_dt)
107
108 # Print the results
109 print('Linear Regression MSE: ', mse_lr)
110 print('Linear Regression R-squared: ', r2_lr)
111 print('Random Forest Regressor MSE: ', mse_rf)
112 print('Random Forest Regressor R-squared: ', r2_rf)
113 print('SVR MSE: ', mse_svr)
114 print('SVR R-squared: ', r2_svr)
115 print('KNN MSE: ', mse_knn)
116 print('KNN R-squared: ', r2_knn)
117 print('Decision Tree Regressor MSE: ', mse_dt)
118 print('Decision Tree Regressor R-squared: ', r2_dt)

```

Reviews

There are yet no reviews for this product.