

```

1  # Import the necessary libraries
2  import pandas as pd
3  import numpy as np
4  from sklearn.preprocessing import StandardScaler
5  from sklearn.model_selection import train_test_split
6  from sklearn.metrics import mean_squared_error, r2_score
7  from sklearn.linear_model import LinearRegression
8  from sklearn.ensemble import RandomForestRegressor
9  from sklearn.svm import SVR
10 from sklearn.neighbors import KNeighborsRegressor
11 from sklearn.tree import DecisionTreeRegressor
12 from sklearn.metrics import mean_absolute_error
13
14 # Load the dataset
15 data = pd.read_csv('data.csv')
16
17 # Split the data into features and target variable
18 X = data[['feature1', 'feature2', 'feature3', 'feature4', 'feature5']]
19 y = data['target']
20
21 # Split the data into training and testing sets
22 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
23
24 # Standardize the features
25 scaler = StandardScaler()
26 X_train = scaler.fit_transform(X_train)
27 X_test = scaler.transform(X_test)
28
29 # Train the Linear Regression model
30 lr = LinearRegression()
31 lr.fit(X_train, y_train)
32
30 # Train the Random Forest model
31 rf = RandomForestRegressor()
32 rf.fit(X_train, y_train)
33
34 # Train the Support Vector Regression model
35 svm = SVR()
36 svm.fit(X_train, y_train)
37
38 # Train the K-Nearest Neighbors model
39 knn = KNeighborsRegressor()
40 knn.fit(X_train, y_train)
41
42 # Train the Decision Tree model
43 dt = DecisionTreeRegressor()
44 dt.fit(X_train, y_train)
45
46 # Predict the target variable for the test set using the Linear Regression model
47 lr_predictions = lr.predict(X_test)
48
49 # Predict the target variable for the test set using the Random Forest model
50 rf_predictions = rf.predict(X_test)
51
52 # Predict the target variable for the test set using the Support Vector Regression model
53 svm_predictions = svm.predict(X_test)
54
55 # Predict the target variable for the test set using the K-Nearest Neighbors model
56 knn_predictions = knn.predict(X_test)
57
58 # Predict the target variable for the test set using the Decision Tree model
59 dt_predictions = dt.predict(X_test)
60
61 # Calculate the Mean Squared Error (MSE) for the Linear Regression model
62 lr_mse = mean_squared_error(y_test, lr_predictions)
63
64 # Calculate the Mean Squared Error (MSE) for the Random Forest model
65 rf_mse = mean_squared_error(y_test, rf_predictions)
66
67 # Calculate the Mean Squared Error (MSE) for the Support Vector Regression model
68 svm_mse = mean_squared_error(y_test, svm_predictions)
69
70 # Calculate the Mean Squared Error (MSE) for the K-Nearest Neighbors model
71 knn_mse = mean_squared_error(y_test, knn_predictions)
72
73 # Calculate the Mean Squared Error (MSE) for the Decision Tree model
74 dt_mse = mean_squared_error(y_test, dt_predictions)
75
76 # Calculate the R-squared (R^2) score for the Linear Regression model
77 lr_r2 = r2_score(y_test, lr_predictions)
78
79 # Calculate the R-squared (R^2) score for the Random Forest model
80 rf_r2 = r2_score(y_test, rf_predictions)
81
82 # Calculate the R-squared (R^2) score for the Support Vector Regression model
83 svm_r2 = r2_score(y_test, svm_predictions)
84
85 # Calculate the R-squared (R^2) score for the K-Nearest Neighbors model
86 knn_r2 = r2_score(y_test, knn_predictions)
87
88 # Calculate the R-squared (R^2) score for the Decision Tree model
89 dt_r2 = r2_score(y_test, dt_predictions)
90
91 # Print the Mean Squared Error (MSE) and R-squared (R^2) score for each model
92 print('Linear Regression MSE: ', lr_mse, 'R^2: ', lr_r2)
93 print('Random Forest MSE: ', rf_mse, 'R^2: ', rf_r2)
94 print('Support Vector Regression MSE: ', svm_mse, 'R^2: ', svm_r2)
95 print('K-Nearest Neighbors MSE: ', knn_mse, 'R^2: ', knn_r2)
96 print('Decision Tree MSE: ', dt_mse, 'R^2: ', dt_r2)
97
98 # End of the script

```