

Our Solution(s)

Run Code

Solution 1Solution 2

```
1 # Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 # O(nm) time | O(nm) space
4 def levenshteinDistance(str1, str2):
5     edits = [[x for x in range(len(str1) + 1)] for y in range(le
6     for i in range(1, len(str2) + 1):
7         edits[i][0] = edits[i - 1][0] + 1
8     for i in range(1, len(str2) + 1):
9         for j in range(1, len(str1) + 1):
10             if str2[i - 1] == str1[j - 1]:
11                 edits[i][j] = edits[i - 1][j - 1]
12             else:
13                 edits[i][j] = 1 + min(edits[i - 1][j - 1], edits
14     return edits[-1][-1]
15
```

Your Solutions

Run Code

Solution 1Solution 2Solution 3

```
1 def levenshteinDistance(str1, str2):
2     # Write your code here.
3     pass
4
```

Our Tests

Custom Output

Submit Code

```

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
9  # Load the dataset
10 data = pd.read_csv('data.csv')
11
12 # Split the data into features (X) and target variable (y)
13 X = data[['feature1', 'feature2', 'feature3']]
14 y = data['target']
15
16 # Standardize the features
17 scaler = StandardScaler()
18 X_scaled = scaler.fit_transform(X)
19
20 # Split the data into training and testing sets
21 X_train, X_test, y_train, y_test = train_test_split(X_scaled, y,
22                                                    test_size=0.2,
23                                                    random_state=42)
24
25 # Create a linear regression model
26 model = LinearRegression()
27
28 # Train the model on the training data
29 model.fit(X_train, y_train)
30
31 # Predict the target variable for the test data
32 y_pred = model.predict(X_test)
33
34 # Calculate the Mean Squared Error (MSE) and R-squared (R2) score
35 mse = mean_squared_error(y_test, y_pred)
36 r2 = r2_score(y_test, y_pred)
37
38 # Print the results
39 print("Mean Squared Error: {}".format(mse))
40 print("R-squared Score: {}".format(r2))

```

Run or submit code when you're ready.