

C:\VS Code\Spyder\Car_Sales_KNN_Regression _code.py

```
1  # ===== KNN Regression Model ===== #
2
3  # ----- importing libraries ----- #
4
5  import numpy as np
6  import matplotlib.pyplot as plt
7  import pandas as pd
8
9  # ----- loading dataset ----- #
10
11 df= pd.read_csv(r"C:\Users\Jan Saida\OneDrive\Documents\Desktop\Excel sheets\logit classification.csv")
12
13 # ----- independent variable ----- #
14
15 x=df.iloc[:, [2,3]].values
16
17 # ----- dependent variables ----- #
18
19 y=df.iloc[:, -1].values
20
21 # ----- training,testing and splitting the data ----- #
22
23 from sklearn.model_selection import train_test_split
24 x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
25
26 # ----- feature scaling ----- #
27
28 from sklearn.preprocessing import StandardScaler
29 sc=StandardScaler()
30 x_train=sc.fit_transform(x_train)
31 x_test=sc.transform(x_test)
32
33 # ----- building K Nearest Neighbor model ----- #
34
```

```
35 from sklearn.neighbors import KNeighborsClassifier
36 classifier=KNeighborsClassifier()
37 classifier.fit(x_train,y_train)
38
39 # ----- making predictions ----- #
40
41 y_pred=classifier.predict(x_test)
42
43 # ----- confusion matrix ----- #
44
45 from sklearn.metrics import confusion_matrix
46 cm=confusion_matrix(y_test,y_pred)
47 print(cm)
48
49 # ----- Accuracy Score ----- #
50
51 from sklearn.metrics import accuracy_score
52 ac = accuracy_score(y_test, y_pred)
53 print(ac)
54
55 # ----- Classification Report ----- #
56
57 from sklearn.metrics import classification_report
58 cr = classification_report(y_test, y_pred)
59 cr
60
61 # ----- Bias - Training Accuracy ----- #
62
63 bias = classifier.score(x_train, y_train)
64 print(bias)
65
66 # ----- Variance - Testing Accuracy ----- #
67
68 variance = classifier.score(x_test, y_test)
69 print(variance)
70
71 # ----- Future Prediction ----- #
```

```
72
73 d1 = pd.read_csv(r"C:\Users\Jan Saida\OneDrive\Documents\Desktop\Excel sheets\logit classification.csv")
74 d2 = d1.copy()
75
76 # ----- Extracts the relevant columns for prediction ----- #
77
78 d1 = d1.iloc[:, [2,3]].values
79
80 # ----- Feature Scaling ----- #
81
82 from sklearn.preprocessing import StandardScaler
83 sc = StandardScaler()
84 m = sc.fit_transform(d1)
85
86 # ----- Initializing an empty DataFrame for storing predictions ----- #
87
88 y_pred1 = pd.DataFrame()
89
90 # ----- Column for the Future Predictions ----- #
91
92 d2['y_pred1'] = classifier.predict(m)
93 d2
94
95 # ----- Saving the code ----- #
96
97 d2.to_csv('KNN_Prediction_Car_Sales.csv')
98
99 # ----- pickle file ----- #
100
101 import pickle
102
103 with open('knn_model.pkl','wb') as file:
104     pickle.dump(classifier,file)
105 print('Model Saved!')
106
107 with open('knn_scaler.pkl','wb') as file:
108     pickle.dump(sc,file)
```

```
109 print('Scaler Saved!')
110
111 # ----- getting file directory path location ----- #
112
113 import os
114 os.getcwd()
115
```

Streamlit\KNN_regression_car_sale.py

```
1 import streamlit as st
2 import pickle
3 import numpy as np
4
5 with open('C:\\Users\\Jan Saida\\knn_model.pkl','rb') as file:
6     model=pickle.load(file)
7
8 with open('C:\\Users\\Jan Saida\\knn_scaler.pkl','rb') as file:
9     scaler=pickle.load(file)
10
11 st.title('Car Purchase Prediction Using K-Nearest-Neighbor')
12 st.write('Predict Whether a Customer will purchase a car or not:')
13
14 age=st.number_input('Enter Age of the customer:',min_value=20,max_value=100,step=1)
15 estimated_salary=st.number_input('Enter Estimated Salary of the Customer:',min_value=0,step=1000)
16
17 if st.button('Predict'):
18     input_data=np.array([[age,estimated_salary]])
19     input_data_scaled=scaler.transform(input_data)
20
21     predict=model.predict(input_data_scaled)
22
23     if predict[0] == 1:
24         st.success(f"The customer is likely to purchase the car")
25     else:
26         st.error(f"The customer is unlikely to purchase the car")
```

Car Purchase Prediction Using K-Nearest-Neighbor

Predict Whether a Customer will purchase a car or not:

Enter Age of the customer:

28

-

+

Enter Estimated Salary of the Customer:

59000

-

+

Predict

The customer is unlikely to purchase the car

Car Purchase Prediction Using K-Nearest-Neighbor

Predict Whether a Customer will purchase a car or not:

Enter Age of the customer:

42

-

+

Enter Estimated Salary of the Customer:

149000

-

+

Predict

The customer is likely to purchase the car