

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

```
1  #----- Logistic Regression Model ----- #
2
3  # importng 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 logistic regression model
34
```

```
35 from sklearn.linear_model import LogisticRegression
36 classifier=LogisticRegression(penalty='l1',solver='liblinear',C=100.0)
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 print(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 predictions
```

```
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 # extracting the relevant columns for predictions
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 preedictions
91
92 d2['y_pred1']=classifier.predict(m)
93 d2
94
95 # saving the data
96
97 d2.to_csv('Logistic_Regression_Prediction_Car_Sales.csv')
98
99 import os
100 os.getcwd()
101
102 # pickle file
103
104 import pickle
105
106 with open('logistic_model.pkl','wb') as file:
107     pickle.dump(classifier,file)
108 print('Model Saved!')
```

```
109
110 with open('logistic_scaler.pkl','wb') as file:
111     pickle.dump(sc,file)
112     print('Scaler Saved!')
113
114 # getting file directory path location
115
116 import os
117 os.getcwd()
118
```