

```

# ----- Importing Libraries ----- #

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# ----- Loading the dataset ----- #

df = pd.read_csv(r"C:\Users\AKSHAY\Downloads\logit_classification.csv")

# We are splitting the dataset, we are splitting the Dependent columns and the Independent columns

# ----- Independent Variable ----- #

x = df.iloc[:, [2,3]].values

# ----- Dependent Variable ----- #

y = df.iloc[:, -1].values

# ----- We are Training, Testing and Splitting the data ----- #

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.20, random_state = 0)

# ----- Feature Scaling ----- #

from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train) # Fits the scaler to the training data and transforms it
x_test = sc.transform(x_test) # Applies the scaling parameters from training data to test data

# ----- Building Logistic Regression Model ----- #

from sklearn.linear_model import LogisticRegression # --> Algorithm
classifier = LogisticRegression(penalty = 'l1', solver = 'liblinear', C = 100.0) # --> Model
classifier.fit(x_train, y_train) # Fits the logistic regression model to the training data

# ----- Making Prediccctions ----- #

y_pred = classifier.predict(x_test)

# ----- Confusion Matrix ----- #

from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)

# ----- Accuracy Score ----- #

from sklearn.metrics import accuracy_score
ac = accuracy_score(y_test, y_pred)
print(ac)

# ----- Classification Report ----- #

from sklearn.metrics import classification_report

```

```

cr = classification_report(y_test, y_pred)
cr

# ----- Bias - Training Accuracy ----- #

bias = classifier.score(x_train, y_train)
print(bias)

# ----- Variance - Testing Accuracy ----- #

variance = classifier.score(x_test, y_test)
print(variance)

# ----- Future Prediction ----- #

d1 = pd.read_csv(r"C:\Users\AKSHAY\Downloads\logit_classification.csv")
d2 = d1.copy()

# ----- Extracts the relevant columns for prediction ----- #

d1 = d1.iloc[:, [2,3]].values

# ----- Feature Scaling ----- #

from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
m = sc.fit_transform(d1)

# ----- Initializing an empty DataFrame for storing predictions ----- #

y_pred1 = pd.DataFrame()

# ----- Column for the Future Predictions ----- #

d2['y_pred1'] = classifier.predict(m)
d2

# ----- Saving the code ----- #

d2.to_csv('LogisticRegressionPredictionCarSales.csv')

import os
os.getcwd()

# ----- Pickle File ----- #

import pickle

with open('logistic_model.pkl', 'wb') as file:
    pickle.dump(classifier, file)
print("Model Saved!")

with open('logistic_scaler.pkl', 'wb') as file:
    pickle.dump(sc, file)
print('Scaler Saved!')

```

```
# ----- Getting the File Directory ----- #  
  
import os  
os.getcwd()
```