

```

1 #Linear Regression
2 import pandas as pd
3 import numpy as np
4 import pandas_profiling as pp
5 df=pd.read_csv("/storage/emulated/0/jhansi/admissionl.
  csv")
6 print(df.head())
7 print(df.tail())
8 print(df.dtypes)
9 print(df.shape)
10 print(df.info)
11 print(df.describe())
12 Profile = pp.ProfileReport(df)
13 Profile.to_file("/storage/emulated/0/jhansi/report1.html")
14 df=df.drop("Research",axis=1)
15 print(df.shape)
16 print(df.head())
17 mean_G=df.GRE_Score.mean()
18 df.GRE_Score=df.GRE_Score.replace({0:mean_G})
19 mean_T=df.TOEFL_Score.mean()
20 df.TOEFL_Score=df.TOEFL_Score.replace({0:mean_T})
21 mean_U=df.University_Rating.mean()
22 df.University_Rating=df.University_Rating.
  replace({0:mean_T})
23 mean_S=df.SOP.mean()
24 df.SOP=df.SOP.replace({0:mean_S})
25 mean_L=df.LOR.mean()
26 df.LOR=df.LOR.replace({0:mean_L})
27 mean_C=df.CGPA.mean()
28 df.CGPA=df.CGPA.replace({0:mean_C})r|
29 mean_A=df.Chance_of_Admit.mean()
30 df.Chance_of_Admit=df.Chance_of_Admit.
  replace({0:mean_A})
31 y=df[['Chance_of_Admit']]
32 X=df.drop('Chance_of_Admit',axis=1)
33 from sklearn.model_selection import train_test

```



```

33 from sklearn.model_selection import train_test_split
34 X_train,X_test,y_train,y_test=train_test_split(X,y,
    test_size=0.25,random_state=1)
35 #model building
36 from sklearn.linear_model import LinearRegression
37 regression_model=LinearRegression()
38 print(regression_model.fit(X_train,y_train))
39 intercept=regression_model.intercept_[0]
40 print(intercept)
41 for idx,col_name in enumerate(X_train.columns):
42     print("The co-efficient for {} is {}".format(col_name,
    regression_model.coef_[0][idx]))
43 #Evaluation metrics
44 from sklearn.metrics import mean_squared_error
45 y_pred=regression_model.predict(X_test)
46 print(y_pred)
47 regression_model_mse=mean_squared_error(y_pred,
    y_test)
48 print(regression_model_mse)
49 import math
50 mae=math.sqrt(regression_model_mse)
51 print(mae)
52 accuracy=regression_model.score(X_test,y_test)
53 print(accuracy)
54 # pre_deployment test
55 new_parameters=[[340,120,4.5,5.0,5.0,9.8,1]]
56 Chance_of_Admit_predicted=regression_model.
    predict(new_parameters)
57 print(Chance_of_Admit_predicted)

```



```

1 import pandas as pd
2 import numpy as np
3 import pandas_profiling as pp
4 df=pd.read_csv("/storage/emulated/0/jhansi/admission1.
  csv")
5 print(df.head())
6 print(df.tail())
7 print(df.dtypes)
8 print(df.shape)
9 print(df.info)
10 print(df.describe())
11 Profile = pp.ProfileReport(df)
12 Profile.to_file("/storage/emulated/0/jhansi/report1.html")
13 df=df.drop("Research",axis=1)
14 print(df.shape)
15 print(df.head())
16 from sklearn.model_selection import train_test_split
17 y=df["Chance_of_Admit"]
18 X=df.drop('Chance_of_Admit',axis=1)
19 X_train,X_test,y_train,y_test = train_test_split(X,y,
  test_size = 0.25)
20 from sklearn.neighbors import KNeighborsClassifier
21 knn = KNeighborsClassifier(n_neighbors=1)
22 knn.fit(X_train,y_train)
23 y_pred = knn.predict(X_test)
24 from sklearn.metrics import confusion_matrix,
  accuracy_score
25 conf_mat = confusion_matrix(y_test,y_pred)
26 print(conf_mat)
27 acc_score = accuracy_score(y_test,y_pred)
28 print(acc_score)
29 new_parameters=[[340,120,4.5,5.0,5.0,9.8,1]]
30 Chance_of_Admit_predicted=knn.predict(new_

```




```

1 import pandas as pd
2 import numpy as np
3 import pandas_profiling as pp
4 df=pd.read_csv("/storage/emulated/0/jhansi/admission1.
  csv")
5 print(df.head())
6 print(df.tail())
7 print(df.dtypes)
8 print(df.shape)
9 print(df.info)
10 print(df.describe())
11 Profile = pp.ProfileReport(df)
12 Profile.to_file("/storage/emulated/0/jhansi/report1.html")
13 df=df.drop("Research",axis=1)
14 print(df.shape)
15 print(df.head())
16 y=df[['Chance_of_Admit']]
17 X=df.drop('Chance_of_Admit',axis=1)
18 from sklearn.model_selection import train_test_split
19 X_train,X_test,y_train,y_test = train_test_split(X,y,
  test_size=0.25)
20 from sklearn.tree import DecisionTreeClassifier
21 model = DecisionTreeClassifier()
22 print(model.fit(X_train,y_train))
23 y_pred = model.predict(X_test)
24 print(y_pred)
25 print(y_test)
26 from sklearn.metrics import accuracy_score,
  confusion_matrix
27 conf_mat = confusion_matrix(y_pred,y_test)
28 acc_score = accuracy_score(y_pred,y_test)
29 print(conf_mat)
30 print(acc_score)
31 new_parameters=[[340,120,4.5,5.0,5.0,9.8,1]]

```

