IBM Advance Data Science

Daoud Tebbakh

OUTLINES

- Data Set
- Data Exploration
- Model Definition / Training
- Model Selaction

Data Set

Dataset name: Forest Cover Type Dataset

CSV file :

covtype.csv

581,012 instances

55 features

11.02 MB

I had to use Git LFS (Large File Storage) to upload csv file to GitHub

Data Set

- Labels : Cover_Type

  Integer value between 1 and 7, with the following key:
(Spruce/Fir ,Lodgepole Pine ,Ponderosa Pine ,Cottonwood/Willow ,Aspen ,Douglas-fir ,Krummholz)

- Features :there are 54

  (Elevation,Aspect,Slope,Horizontal_Distance_To_Hydrology,Vertical_D,stance_To_Hydrology,Horizontal_Distance_To_Roadways,Hillshade_9am,Hillshade_Noon,Hillshade_3pm,Horizontal_Distance_To_Fire_Points,Wilderness_Area1,Wilderness_Area2,Wilderness_Area3,Wilderness_rea4,Soil_Type1.....,Soil_Type40)

Data describe

```
In [32]: data.describe()
```

Out[32]:

	Elevation	Aspect	Slope	Horizontal_Distance_To_Hydrology	Vertical_Distance_To_Hydrology	Horizontal_Distance_To_Roadways	Hillshade_9am	Hillshade_Noon	Hillshade_3pm	Horizontal_[
count	581012.000000	581012.000000	581012.000000	581012.000000	581012.000000	581012.000000	581012.000000	581012.000000	581012.000000	
mean	2959.365301	155.656807	14.103704	269.428217	46.418855	2350.146611	212.146049	223.318716	142.528263	
std	279.984734	111.913721	7.488242	212.549356	58.295232	1559.254870	26.769889	19.768697	38.274529	
min	1859.000000	0.000000	0.000000	0.000000	-173.000000	0.000000	0.000000	0.000000	0.000000	
25%	2809.000000	58.000000	9.000000	108.000000	7.000000	1106.000000	198.000000	213.000000	119.000000	
50%	2996.000000	127.000000	13.000000	218.000000	30.000000	1997.000000	218.000000	226.000000	143.000000	
75%	3163.000000	260.000000	18.000000	384.000000	69.000000	3328.000000	231.000000	237.000000	168.000000	
max	3858.000000	360.000000	66.000000	1397.000000	601.000000	7117.000000	254.000000	254.000000	254.000000	

8 rows × 55 columns

Data Dimension

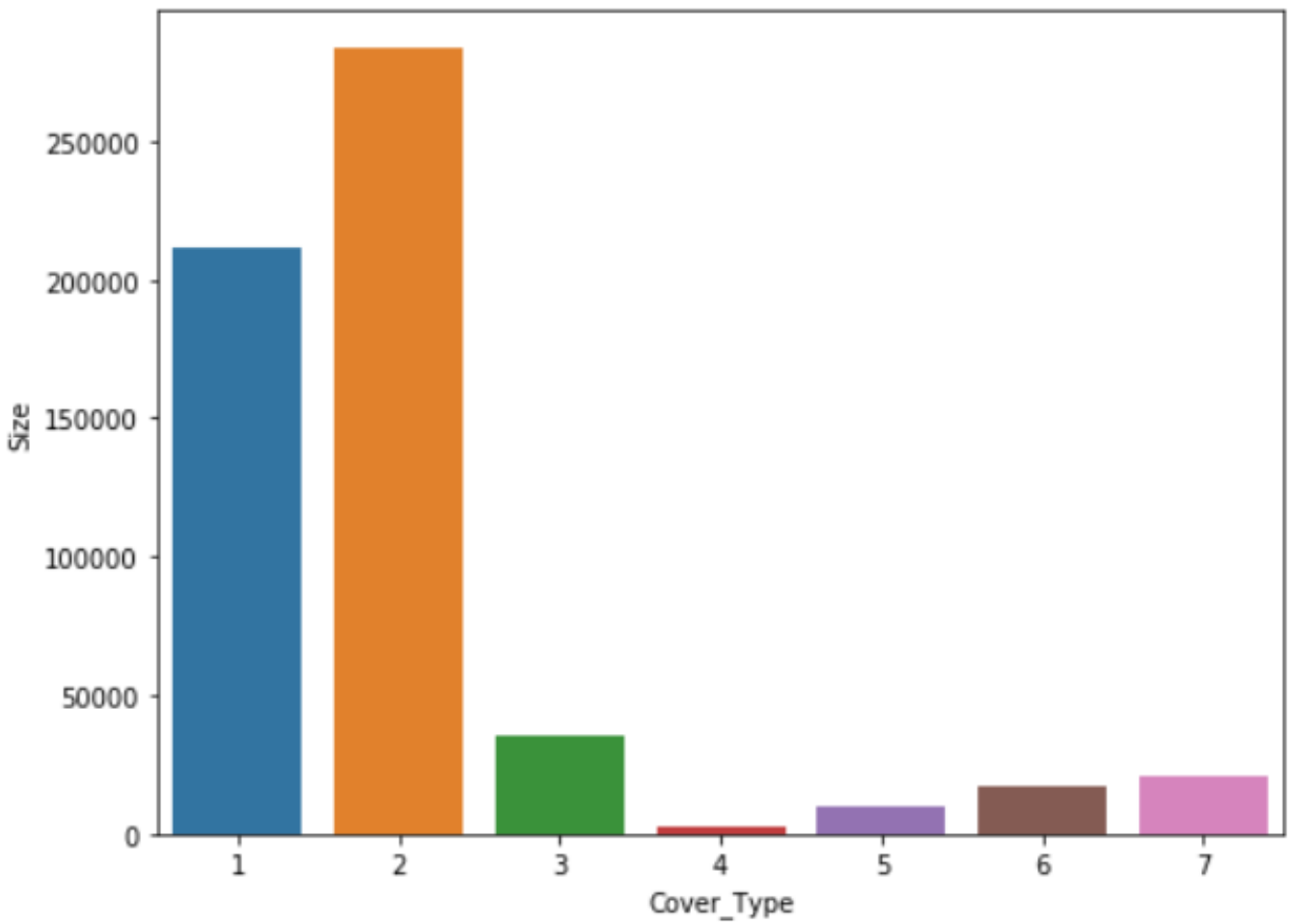
```
Data Dimension:
Number of Records: 581012
Number of Features: 55
```

Data Split:

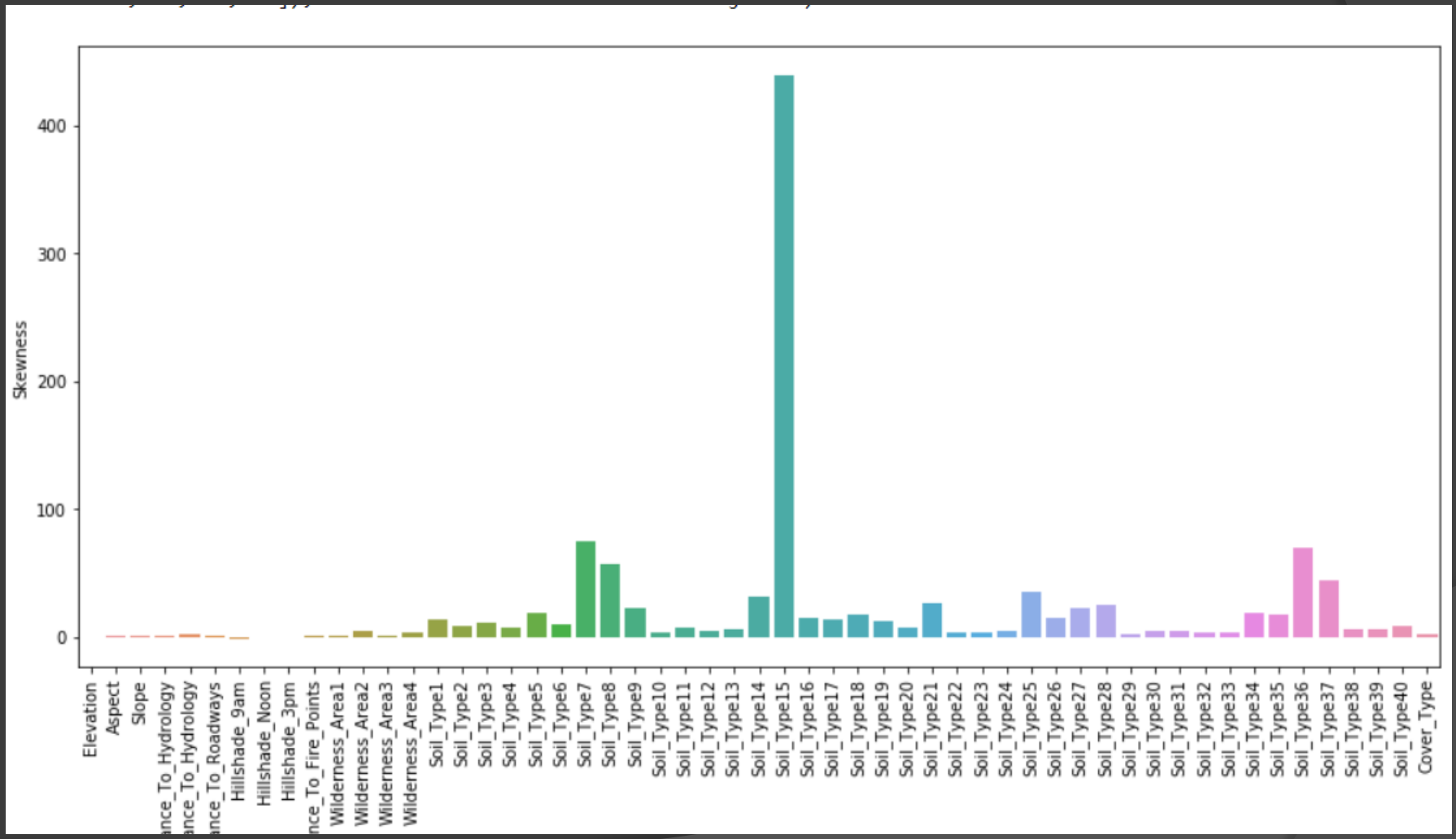
```
data split:

((406708, 54), (174304, 54), (406708,), (174304,))
```

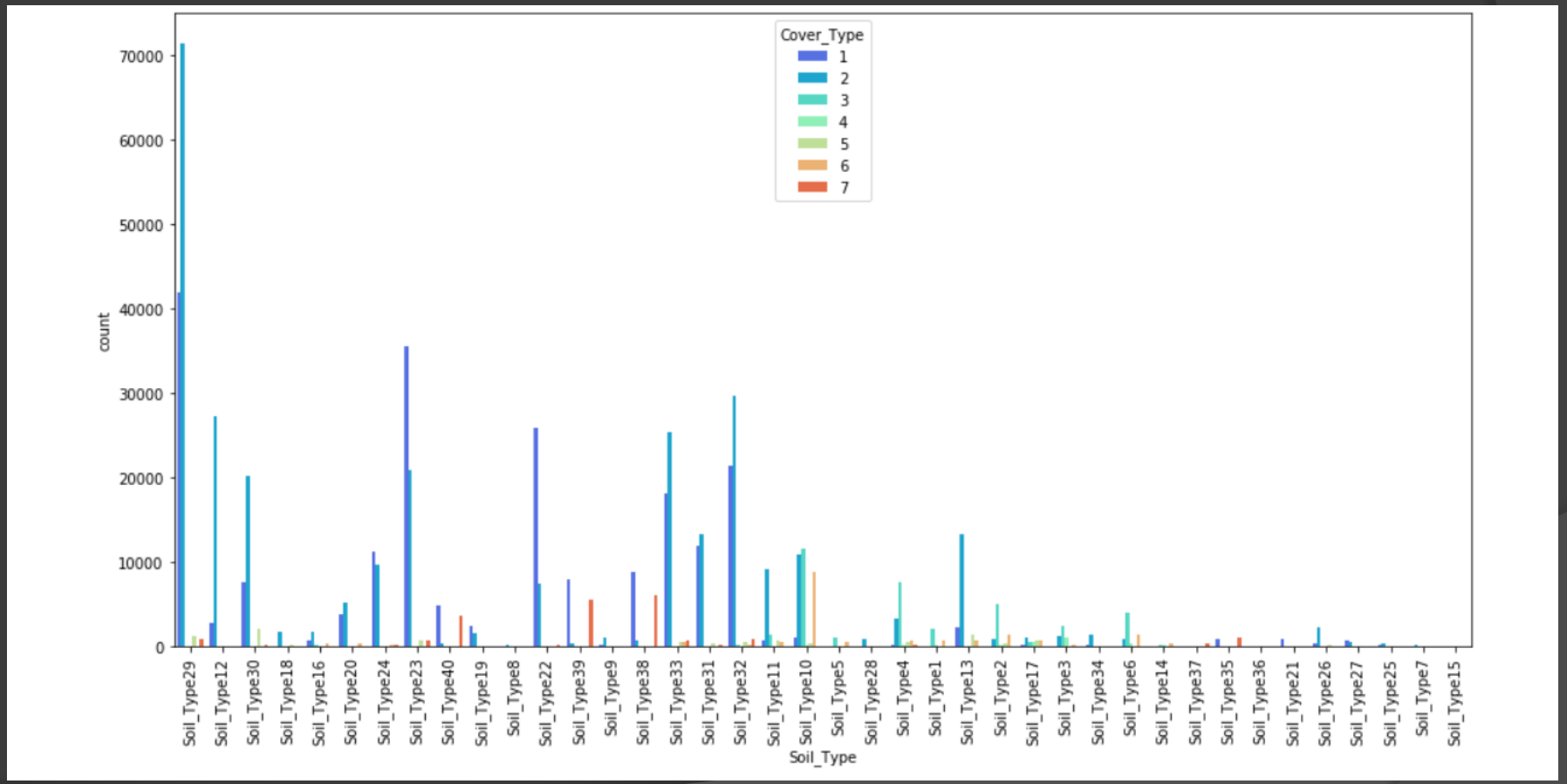
We notes that 1 and 2 cover approx 85.2% of all data



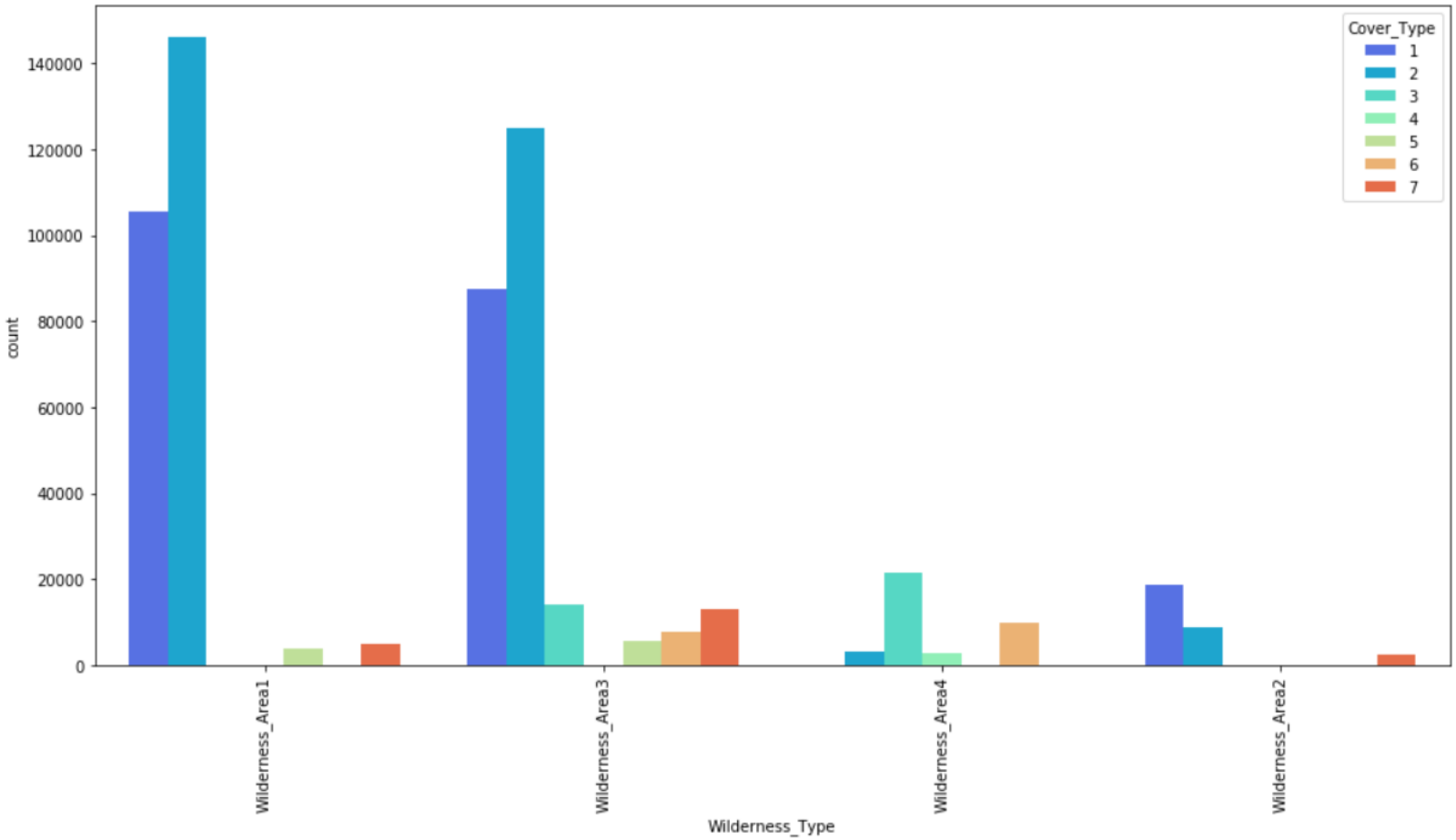
skew


Skewness
400
300
200
100
0
Elevation
Aspect
Slope
Hillshade_9am
Hillshade_Noon
Hillshade_3pm
Wilderness_Area1
Wilderness_Area2
Wilderness_Area3
Wilderness_Area4
Soil_Type1
Soil_Type2
Soil_Type3
Soil_Type4
Soil_Type5
Soil_Type6
Soil_Type7
Soil_Type8
Soil_Type9
Soil_Type10
Soil_Type11
Soil_Type12
Soil_Type13
Soil_Type14
Soil_Type15
Soil_Type16
Soil_Type17
Soil_Type18
Soil_Type19
Soil_Type20
Soil_Type21
Soil_Type22
Soil_Type23
Soil_Type24
Soil_Type25
Soil_Type26
Soil_Type27
Soil_Type28
Soil_Type29
Soil_Type30
Soil_Type31
Soil_Type32
Soil_Type33
Soil_Type34
Soil_Type35
Soil_Type36
Soil_Type37
Soil_Type38
Soil_Type39
Soil_Type40
Cover_Type

Soil_Type quantity



Wilderness_Type quantity



Model : Decision Tree

Accuracy:

```
In [18]: accuracy_score(Y_test, Y_pred)
Out[18]: 0.9357444464843033
```

Confusion Matrix:

```
In [21]: confusion_matrix(Y_test, Y_pred)
Out[21]: array([[59238,  3778,     4,     0,    50,     8,   321],
                [ 3642, 80491,   235,     0,   403,   182,    40],
                [    2,   220, 10067,   100,    29,   404,     0],
                [    0,     2,   102,   685,     0,    40,     0],
                [   69,   438,    29,     0,  2320,    12,     2],
                [    8,   182,   479,    36,    18,  4526,     0],
                [  320,    45,     0,     0,     0,     0,  5777]])
```

Classification report:

```
In [22]: print(classification_report(Y_test, Y_pred))
```

	precision	recall	f1-score	support
1	0.94	0.93	0.94	63399
2	0.95	0.95	0.95	84993
3	0.92	0.93	0.93	10822
4	0.83	0.83	0.83	829
5	0.82	0.81	0.82	2870
6	0.88	0.86	0.87	5249
7	0.94	0.94	0.94	6142
micro avg	0.94	0.94	0.94	174304
macro avg	0.90	0.89	0.89	174304
weighted avg	0.94	0.94	0.94	174304

Model : Random Forest

Accuracy:

```
In [25]: accuracy_score(Y_test, Y_pred)

Out[25]: 0.9536327336148338
```

Confusion Matrix:

```
In [26]: confusion_matrix(Y_test, Y_pred)

Out[26]: array([[59764,  3457,     2,     0,    15,     9,   152],
                [ 1873, 82719,   167,     5,   119,    94,    16],
                [    0,   163, 10406,    45,     7,   201,     0],
                [    0,     0,    91,   703,     0,    35,     0],
                [   37,   632,    39,     0,  2152,    10,     0],
                [    9,   162,   385,    15,     3,  4675,     0],
                [  298,    41,     0,     0,     0,     0,  5803]])
```

Classification report:

```
In [27]: print(classification_report(Y_test, Y_pred))
```

	precision	recall	f1-score	support
1	0.96	0.94	0.95	63399
2	0.95	0.97	0.96	84993
3	0.94	0.96	0.95	10822
4	0.92	0.85	0.88	829
5	0.94	0.75	0.83	2870
6	0.93	0.89	0.91	5249
7	0.97	0.94	0.96	6142
micro avg	0.95	0.95	0.95	174304
macro avg	0.94	0.90	0.92	174304
weighted avg	0.95	0.95	0.95	174304

Model : Logistic Regresion

Accuracy:

```
In [9]: accuracy_score(Y_test, Y_pred)
Out[9]: 0.7118081053791078
```

Confusion Matrix:

```
In [10]: confusion_matrix(Y_test, Y_pred)
Out[10]: array([[43437, 18683,    48,     0,     0,     0,  1231],
                [15225, 67611,  1942,     7,     0,   108,   100],
                [    7,  1139,  9297,   109,     0,   269,     1],
                [    0,     0,   524,   217,     0,    88,     0],
                [   67,  2535,   267,     0,     0,     0,     1],
                [    5,  1949,  2978,    18,     0,   299,     0],
                [ 2857,    48,    27,     0,     0,     0,  3210]])
```

Classification report:

```
In [11]: print(classification_report(Y_test, Y_pred))
```

	precision	recall	f1-score	support
1	0.71	0.69	0.70	63399
2	0.74	0.80	0.76	84993
3	0.62	0.86	0.72	10822
4	0.62	0.26	0.37	829
5	0.00	0.00	0.00	2870
6	0.39	0.06	0.10	5249
7	0.71	0.52	0.60	6142
micro avg	0.71	0.71	0.71	174304
macro avg	0.54	0.45	0.46	174304
weighted avg	0.69	0.71	0.70	174304

Model : XGBoost

Accuracy:

```
In [29]: accuracy_score(Y_test, Y_pred)
Out[29]: 0.7861724343675418
```

Confusion Matrix:

```
In [30]: confusion_matrix(Y_test, Y_pred)
Out[30]: array([[48306, 14508,     5,     0,    13,     8,   559],
                [12347, 71566,   599,     6,    88,   370,    17],
                [    0,   964,  9313,    86,     0,   459,     0],
                [    0,     0,   153,   652,     0,    24,     0],
                [    9,  2136,    63,     0,   659,     3,     0],
                [    1,  1109,  2099,    19,     0,  2021,     0],
                [ 1608,    18,     0,     0,     0,     0,  4516]])
```

Classification report:

```
In [31]: print(classification_report(Y_test, Y_pred))
```

	precision	recall	f1-score	support
1	0.78	0.76	0.77	63399
2	0.79	0.84	0.82	84993
3	0.76	0.86	0.81	10822
4	0.85	0.79	0.82	829
5	0.87	0.23	0.36	2870
6	0.70	0.39	0.50	5249
7	0.89	0.74	0.80	6142
micro avg	0.79	0.79	0.79	174304
macro avg	0.81	0.66	0.70	174304
weighted avg	0.79	0.79	0.78	174304

Model : KNeighbors

Accuracy:

```
In [34]: accuracy_score(Y_test, Y_pred)
Out[34]: 0.7861724343675418
```

Confusion Matrix:

```
In [35]: confusion_matrix(Y_test, Y_pred)
Out[35]: array([[48306, 14508,     5,     0,    13,     8,   559],
                [12347, 71566,   599,     6,    88,   370,    17],
                [    0,   964,  9313,    86,     0,   459,     0],
                [    0,     0,   153,   652,     0,    24,     0],
                [    9,  2136,    63,     0,   659,     3,     0],
                [    1,  1109,  2099,    19,     0,  2021,     0],
                [ 1608,    18,     0,     0,     0,     0,  4516]])
```

Classification report:

```
In [36]: print(classification_report(Y_test, Y_pred))
```

	precision	recall	f1-score	support
1	0.78	0.76	0.77	63399
2	0.79	0.84	0.82	84993
3	0.76	0.86	0.81	10822
4	0.85	0.79	0.82	829
5	0.87	0.23	0.36	2870
6	0.70	0.39	0.50	5249
7	0.89	0.74	0.80	6142
micro avg	0.79	0.79	0.79	174304
macro avg	0.81	0.66	0.70	174304
weighted avg	0.79	0.79	0.78	174304

Model Summary

Model	Accuracy
Decision Tree	0.93
Random Forest	0.95
Logistic Regresion	0.71
XGBoost	0.78
KNeighbors	0.78

The highest Accuracy is : 0.95

model : Random Forest