PROBLEM STATEMENT:- The model which is best for the flight price

IMPORT LIBRARY

In [3]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from sklearn.model_selection import train_test_split
```

READ THE DATA SET

In [4]:
```
train_df=pd.read_csv(r"C:\Users\pucha\Downloads\Data_Train.csv")
train_df
```

Out[4]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR ? DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	Air India	1/05/2019	Kolkata	Banglore	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL ? LKO ? BOM ? COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU ? NAG ? BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR ? NAG ? DEL	16:50	21:35	4h 45m	1 stop	No info	13302
...	...	...	...	...	...	...	...	...	...	...	...
10678	Air Asia	9/04/2019	Kolkata	Banglore	CCU ? BLR	19:55	22:25	2h 30m	non-stop	No info	4107
10679	Air India	27/04/2019	Kolkata	Banglore	CCU ? BLR	20:45	23:20	2h 35m	non-stop	No info	4145
10680	Jet Airways	27/04/2019	Banglore	Delhi	BLR ? DEL	08:20	11:20	3h	non-stop	No info	7229
10681	Vistara	01/03/2019	Banglore	New Delhi	BLR ? DEL	11:30	14:10	2h 40m	non-stop	No info	12648
10682	Air India	9/05/2019	Delhi	Cochin	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m	2 stops	No info	11753

10683 rows × 11 columns

In [5]:
```
test_df=pd.read_csv(r"C:\Users\pucha\Downloads\Test_set.csv")
test_df
```

Out[5]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info
0	Jet Airways	6/06/2019	Delhi	Cochin	DEL ? BOM ? COK	17:30	04:25 07 Jun	10h 55m	1 stop	No info
1	IndiGo	12/05/2019	Kolkata	Banglore	CCU ? MAA ? BLR	06:20	10:20	4h	1 stop	No info
2	Jet Airways	21/05/2019	Delhi	Cochin	DEL ? BOM ? COK	19:15	19:00 22 May	23h 45m	1 stop	In-flight meal not included
3	Multiple carriers	21/05/2019	Delhi	Cochin	DEL ? BOM ? COK	08:00	21:00	13h	1 stop	No info
4	Air Asia	24/06/2019	Banglore	Delhi	BLR ? DEL	23:55	02:45 25 Jun	2h 50m	non-stop	No info
...	...	...	...	...	...	...	...	...	...	...
2666	Air India	6/06/2019	Kolkata	Banglore	CCU ? DEL ? BLR	20:30	20:25 07 Jun	23h 55m	1 stop	No info
2667	IndiGo	27/03/2019	Kolkata	Banglore	CCU ? BLR	14:20	16:55	2h 35m	non-stop	No info
2668	Jet Airways	6/03/2019	Delhi	Cochin	DEL ? BOM ? COK	21:50	04:25 07 Mar	6h 35m	1 stop	No info
2669	Air India	6/03/2019	Delhi	Cochin	DEL ? BOM ? COK	04:00	19:15	15h 15m	1 stop	No info
2670	Multiple carriers	15/06/2019	Delhi	Cochin	DEL ? BOM ? COK	04:55	19:15	14h 20m	1 stop	No info

2671 rows × 10 columns

DATA CLEANING AND PREPROCESSING

In [6]:
```
train_df.head()
```

Out[6]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR ? DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	Air India	1/05/2019	Kolkata	Banglore	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL ? LKO ? BOM ? COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU ? NAG ? BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR ? NAG ? DEL	16:50	21:35	4h 45m	1 stop	No info	13302

In [7]:
```
test_df.head()
```

Out[7]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info
0	Jet Airways	6/06/2019	Delhi	Cochin	DEL ? BOM ? COK	17:30	04:25 07 Jun	10h 55m	1 stop	No info
1	IndiGo	12/05/2019	Kolkata	Banglore	CCU ? MAA ? BLR	06:20	10:20	4h	1 stop	No info
2	Jet Airways	21/05/2019	Delhi	Cochin	DEL ? BOM ? COK	19:15	19:00 22 May	23h 45m	1 stop	In-flight meal not included
3	Multiple carriers	21/05/2019	Delhi	Cochin	DEL ? BOM ? COK	08:00	21:00	13h	1 stop	No info
4	Air Asia	24/06/2019	Banglore	Delhi	BLR ? DEL	23:55	02:45 25 Jun	2h 50m	non-stop	No info

In [8]:
```
train_df.tail()
```

Out[8]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
10678	Air Asia	9/04/2019	Kolkata	Banglore	CCU ? BLR	19:55	22:25	2h 30m	non-stop	No info	4107
10679	Air India	27/04/2019	Kolkata	Banglore	CCU ? BLR	20:45	23:20	2h 35m	non-stop	No info	4145
10680	Jet Airways	27/04/2019	Banglore	Delhi	BLR ? DEL	08:20	11:20	3h	non-stop	No info	7229
10681	Vistara	01/03/2019	Banglore	New Delhi	BLR ? DEL	11:30	14:10	2h 40m	non-stop	No info	12648
10682	Air India	9/05/2019	Delhi	Cochin	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m	2 stops	No info	11753

In [9]:
```
test_df.tail()
```

Out[9]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info
2666	Air India	6/06/2019	Kolkata	Banglore	CCU ? DEL ? BLR	20:30	20:25 07 Jun	23h 55m	1 stop	No info
2667	IndiGo	27/03/2019	Kolkata	Banglore	CCU ? BLR	14:20	16:55	2h 35m	non-stop	No info
2668	Jet Airways	6/03/2019	Delhi	Cochin	DEL ? BOM ? COK	21:50	04:25 07 Mar	6h 35m	1 stop	No info
2669	Air India	6/03/2019	Delhi	Cochin	DEL ? BOM ? COK	04:00	19:15	15h 15m	1 stop	No info
2670	Multiple carriers	15/06/2019	Delhi	Cochin	DEL ? BOM ? COK	04:55	19:15	14h 20m	1 stop	No info

REPLACE THE CHARACTERS AS STRING

In [13]:
```
city={"Source":{"Kolkata":1,"Banglore":2,"Delhi":3,"Mumbai":4,"Chennai":5}}
train_df=train_df.replace(city)
train_df
```

Out[13]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	24/03/2019	2	New Delhi	BLR ? DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	Air India	1/05/2019	1	Banglore	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	9/06/2019	3	Cochin	DEL ? LKO ? BOM ? COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882
3	IndiGo	12/05/2019	1	Banglore	CCU ? NAG ? BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	IndiGo	01/03/2019	2	New Delhi	BLR ? NAG ? DEL	16:50	21:35	4h 45m	1 stop	No info	13302
...	...	...	...	...	...	...	...	...	...	...	...
10678	Air Asia	9/04/2019	1	Banglore	CCU ? BLR	19:55	22:25	2h 30m	non-stop	No info	4107
10679	Air India	27/04/2019	1	Banglore	CCU ? BLR	20:45	23:20	2h 35m	non-stop	No info	4145
10680	Jet Airways	27/04/2019	2	Delhi	BLR ? DEL	08:20	11:20	3h	non-stop	No info	7229
10681	Vistara	01/03/2019	2	New Delhi	BLR ? DEL	11:30	14:10	2h 40m	non-stop	No info	12648
10682	Air India	9/05/2019	3	Cochin	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m	2 stops	No info	11753

10683 rows × 11 columns

In [14]:
```
destination={"Destination":{"Cochin":1,"Banglore":2,"New Delhi":3,"Delhi":4,"Hyderabad":5,"Kolkata":6}}
train_df=train_df.replace(destination)
train_df
```

Out[14]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	24/03/2019	2	3	BLR ? DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	Air India	1/05/2019	1	2	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	9/06/2019	3	1	DEL ? LKO ? BOM ? COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882
3	IndiGo	12/05/2019	1	2	CCU ? NAG ? BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	IndiGo	01/03/2019	2	3	BLR ? NAG ? DEL	16:50	21:35	4h 45m	1 stop	No info	13302
...	...	...	...	...	...	...	...	...	...	...	...
10678	Air Asia	9/04/2019	1	2	CCU ? BLR	19:55	22:25	2h 30m	non-stop	No info	4107
10679	Air India	27/04/2019	1	2	CCU ? BLR	20:45	23:20	2h 35m	non-stop	No info	4145
10680	Jet Airways	27/04/2019	2	4	BLR ? DEL	08:20	11:20	3h	non-stop	No info	7229
10681	Vistara	01/03/2019	2	3	BLR ? DEL	11:30	14:10	2h 40m	non-stop	No info	12648
10682	Air India	9/05/2019	3	1	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m	2 stops	No info	11753

10683 rows × 11 columns

In [15]:
```
total={"Total_Stops":{"non-stop":1,"2 stops":3,"1 stop":2,"3 stops":4,"4 stops":5}}
train_df=train_df.replace(total)
train_df
```

Out[15]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	24/03/2019	2	3	BLR ? DEL	22:20	01:10 22 Mar	2h 50m	1.0	No info	3897
1	Air India	1/05/2019	1	2	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m	3.0	No info	7662
2	Jet Airways	9/06/2019	3	1	DEL ? LKO ? BOM ? COK	09:25	04:25 10 Jun	19h	3.0	No info	13882
3	IndiGo	12/05/2019	1	2	CCU ? NAG ? BLR	18:05	23:30	5h 25m	2.0	No info	6218
4	IndiGo	01/03/2019	2	3	BLR ? NAG ? DEL	16:50	21:35	4h 45m	2.0	No info	13302
...	...	...	...	...	...	...	...	...	...	...	...
10678	Air Asia	9/04/2019	1	2	CCU ? BLR	19:55	22:25	2h 30m	1.0	No info	4107
10679	Air India	27/04/2019	1	2	CCU ? BLR	20:45	23:20	2h 35m	1.0	No info	4145
10680	Jet Airways	27/04/2019	2	4	BLR ? DEL	08:20	11:20	3h	1.0	No info	7229
10681	Vistara	01/03/2019	2	3	BLR ? DEL	11:30	14:10	2h 40m	1.0	No info	12648
10682	Air India	9/05/2019	3	1	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m	3.0	No info	11753

10683 rows × 11 columns

In [56]:
```
airline={"Airline":{"Jet Airways":0,"IndiGo":1,"Air India":2,"Multiple carriers":3,
"SpiceJet":4,"Vistara":5,"Air Asia":6,"GoAir":7,
"Multiple carriers Premium economy":8,
"Jet Airways Business":9,"Vistara Premium economy":10,"Trujet":11}}
train_df=train_df.replace(airline)
train_df
```

Out[56]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	1	24/03/2019	2	3	BLR ? DEL	22:20	01:10 22 Mar	2h 50m	1.0	No info	3897
1	2	1/05/2019	1	2	CCU ? IXR ? BBI ? BLR	05:50	13:15	7h 25m	3.0	No info	7662
2	0	9/06/2019	3	1	DEL ? LKO ? BOM ? COK	09:25	04:25 10 Jun	19h	3.0	No info	13882
3	1	12/05/2019	1	2	CCU ? NAG ? BLR	18:05	23:30	5h 25m	2.0	No info	6218
4	1	01/03/2019	2	3	BLR ? NAG ? DEL	16:50	21:35	4h 45m	2.0	No info	13302
...	...	...	...	...	...	...	...	...	...	...	...
10678	6	9/04/2019	1	2	CCU ? BLR	19:55	22:25	2h 30m	1.0	No info	4107
10679	2	27/04/2019	1	2	CCU ? BLR	20:45	23:20	2h 35m	1.0	No info	4145
10680	0	27/04/2019	2	4	BLR ? DEL	08:20	11:20	3h	1.0	No info	7229
10681	5	01/03/2019	2	3	BLR ? DEL	11:30	14:10	2h 40m	1.0	No info	12648
10682	2	9/05/2019	3	1	DEL ? GOI ? BOM ? COK	10:55	19:15	8h 20m	3.0	No info	11753

10682 rows × 11 columns

In [17]:
```
train_df.describe()
```

Out[17]:

	Airline	Source	Destination	Total_Stops	Price
count	10683.000000	10683.000000	10683.000000	10682.000000	10683.000000
mean	1.711411	2.393429	2.237761	1.824190	9087.064121
std	1.844124	1.058636	1.444670	0.675229	4611.359167
min	0.000000	1.000000	1.000000	1.000000	1759.000000
25%	0.000000	1.000000	1.000000	1.000000	5277.000000
50%	1.000000	3.000000	2.000000	2.000000	8372.000000
75%	3.000000	3.000000	3.000000	2.000000	12373.000000
max	11.000000	5.000000	6.000000	5.000000	79512.000000

TO COUNT THE VALUES

In [18]:
```
train_df['Source'].value_counts()
```

Out[18]:
```
Source
3    4537
1    2871
2    2197
4     697
5     381
Name: count, dtype: int64
```

In [19]:
```
train_df['Destination'].value_counts()
```

Out[19]:
```
Destination
1    4537
2    2871
4    1265
3     932
5     697
6     381
Name: count, dtype: int64
```

In [20]:
```
train_df['Total_Stops'].value_counts()
```

Out[20]:
```
Total_Stops
2.0    5625
1.0    3491
3.0    1520
4.0      45
5.0       1
Name: count, dtype: int64
```

In [21]:
```
train_df['Price'].value_counts()
```

Out[21]:
```
Price
10262    258
10844    212
7229     162
4804     160
4823     131
        ...
14153      1
8488       1
7826       1
6315       1
12648      1
Name: count, Length: 1870, dtype: int64
```

In [22]:
```
train_df['Airline'].value_counts()
```

Out[22]:
```
Airline
0     3849
1     2053
2     1752
3     1196
4      818
5      479
6      319
7      194
8       13
9        6
10       3
11       1
Name: count, dtype: int64
```

In [23]:
```
train_df.shape
```

Out[23]:
```
(10683, 11)
```

In [24]:
```
test_df.shape
```

Out[24]:
```
(2671, 10)
```

TO FIND THE DUPLICATE VALUES

In [25]:
```
train_df.duplicated().sum()
```

Out[25]:
```
220
```

In [26]:
```
test_df.duplicated().sum()
```

Out[26]:
```
26
```

TO FIND THE NULL VALUES

In [27]:
```
train_df.isnull().sum()
```

Out[27]:
```
Airline            0
Date_of_Journey    0
Source             0
Destination        0
Route              1
Dep_Time           0
Arrival_Time       0
Duration           0
Total_Stops        1
Additional_Info    0
Price              0
dtype: int64
```

In [28]:
```
test_df.isnull().sum()
```

Out[28]:
```
Airline            0
Date_of_Journey    0
Source             0
Destination        0
Route              0
Dep_Time           0
Arrival_Time       0
Duration           0
Total_Stops        0
Additional_Info    0
dtype: int64
```

FEATURE SCALLING:-Split the dataset into independent and dependent variables

Split your dataset in two catagories 1.Train data 2.Test data

In [29]:
```
x=train_df[['Airline','Source','Destination']]
y=train_df['Price']
```

In [30]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=100)
```

LINEAR REGRESSION

In [31]:
```
from sklearn.linear_model import LinearRegression
regr=LinearRegression()
regr.fit(x_train,y_train)
```

Out[31]:
```
LinearRegression
LinearRegression()
```

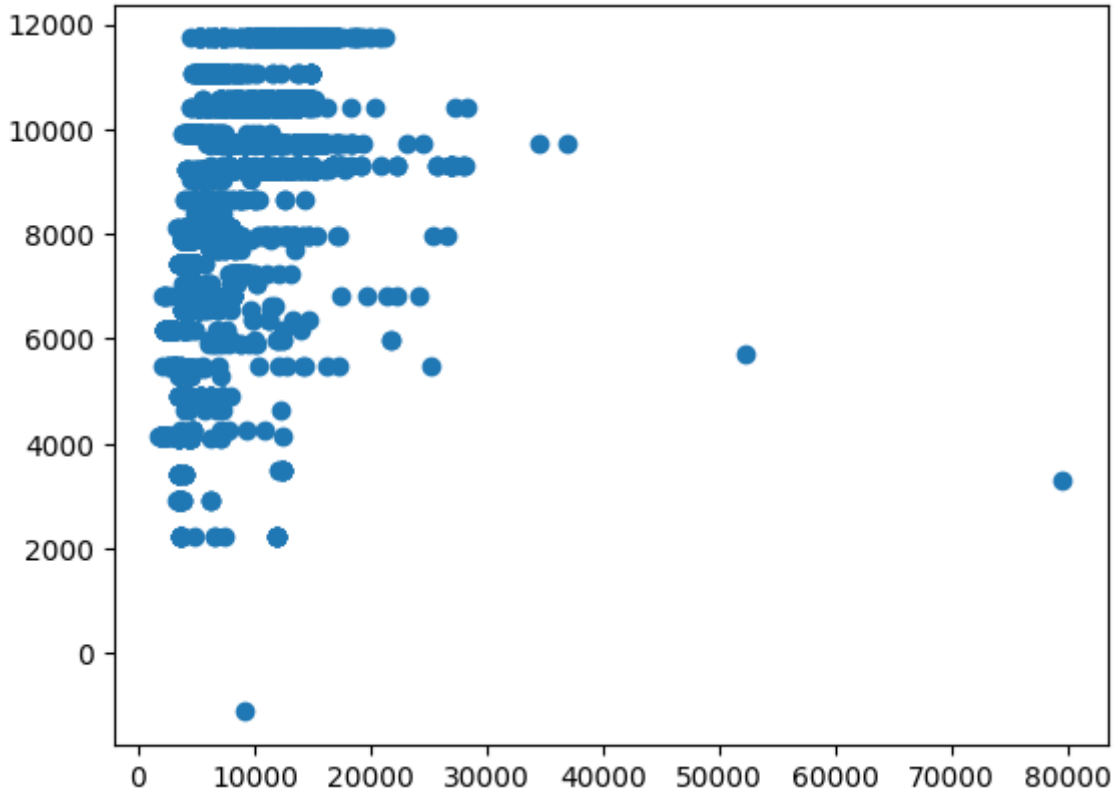
In [32]:
```
score=regr.score(x_test,y_test)
print(score)
```

```
0.2209255246334636
```

In [33]:
```
predictions=regr.predict(x_test)
```

In [34]:
```
plt.scatter(y_test,predictions)
```

Out[34]: <matplotlib.collections.PathCollection at 0x135df496080>



In [35]:
```
x=np.array(train_df['Price']).reshape(-1,1)
y=np.array(train_df['Airline']).reshape(-1,1)
train_df.dropna(inplace=True)
```

In [36]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
regr.fit(x_train,y_train)
regr.fit(x_train,y_train)
```

Out[36]:
```
LinearRegression
LinearRegression()
```

In [37]:
```
x=np.array(train_df['Price']).reshape(-1,1)
y=np.array(train_df['Airline']).reshape(-1,1)
train_df.dropna(inplace=True)
```

In [38]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=1)
```

LOGISTIC REGRESSION

In [58]:
```
from sklearn.linear_model import LogisticRegression
a=LogisticRegression(max_iter=1000)
```

In [59]:
```
a.fit(x_train,y_train)
```

```
C:\Users\pucha\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\utils\validation.py:1143: DataC
onversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_
samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

Out[59]:
```
LogisticRegression
LogisticRegression(max_iter=1000)
```

In [60]:
```
score=a.score(x_test,y_test)
print(score)
```

```
0.37160686427457096
```

DECISION TREE CLASSIFIER

In [61]:
```
from sklearn.tree import DecisionTreeClassifier
df=DecisionTreeClassifier(random_state=2)
df.fit(x_train,y_train)
```

Out[61]:
```
DecisionTreeClassifier(random_state=2)
```

In [62]:
```
score=df.score(x_test,y_test)
print(score)
```

```
0.8911076443057723
```

RANDOM FOREST CLASSIFIER

In [44]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

```
C:\Users\pucha\AppData\Local\Temp\ipykernel_15636\2210184639.py:3: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
  rfc.fit(x_train,y_train)
```

Out[44]:
```
RandomForestClassifier()
```

In [45]:
```
params={'max_depth':[2,3,5,10,20],
'min_samples_leaf':[5,10,20,50,100,200],
'n_estimators':[10,25,30,50,100,200]}
```

In [46]:
```
from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rfc,param_grid=params,cv=2,scoring="accuracy")
```

In [47]:
```
grid_search.fit(x_train,y_train)
```

```
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\pucha\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\model_selection\_validation.
py:686: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the s
hape of y to (n_samples,), for example using ravel().
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\pucha\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\model_selection\_validation.
py:686: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the s
hape of y to (n_samples,), for example using ravel().
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\pucha\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\model_selection\_validation.
py:686: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the s
hape of y to (n_samples,), for example using ravel().
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\pucha\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\model_selection\_validation.
py:686: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the s
hape of y to (n_samples,), for example using ravel().
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\pucha\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\model_selection\_validation.
py:686: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the s
hape of y to (n_samples,), for example using ravel().
  estimator.fit(X_train, y_train, **fit_params)
```

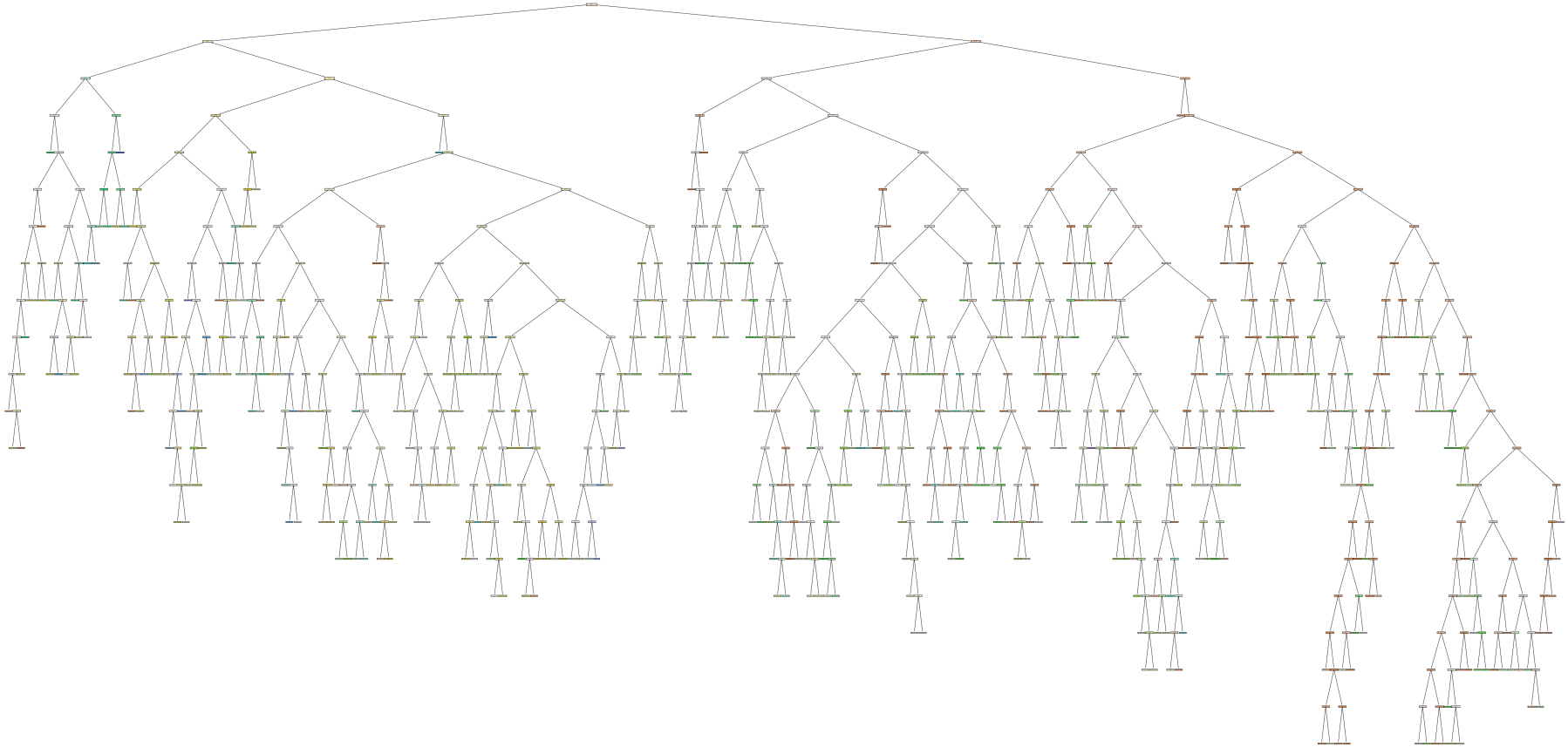
In [48]:
```
grid_search.best_score_
```

Out[48]:
```
0.7456215421129732
```

In [49]:
```
rf_best=grid_search.best_estimator_
rf_best
```

Out[49]:
```
RandomForestClassifier(max_depth=20, min_samples_leaf=5, n_estimators=50)
```

In [54]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[4],filled=True);
```



In [63]:
```
score=rfc.score(x_test,y_test)
print(score)
```

```
0.8911076443057723
```

CONCLUSION:-

```
      FROM THE ABOVE DATASET I HAVE TO DEFINE THAT WHICH MODEL HAS THE BEST ACCURACY FROM THE AMONG MODEL
   S
        STEP:-1. IMPORT THE LIBRARY
        STEP:-2. I HAVE TO READ THE DATASET FOR
        STEP:-3. TAKE DATA CLEANING AND PREPROCESSING
        STEP:-4. TO FIND COUNT,DUPLICATE AND NULL VALUES
        STEP:-5. SPLITING THE DATASET INTO TRAIN DATA AND TEST DATA
        STEP:-6.  IMPORTING THE LINEAR REGREESION
        STEP:-7.  IMPORTING THE LOGISTIC REGRESSION
        STEP:-8. IMPORTING THE DECISIONTREECLASSIFIER
        STEP:-9. IMPORTING THE RANDOMFORESTCLASSIFIER
    FROM THE DATASET LINEAR REGRESSION I GOT THE ACCURACY     : 0.2209255246334636
                     LOGISTIC REGRESSION I GOT THE ACCURACY   : 0.37160686427457096
                     DECISIONTREECLASSIFIER I GOT THE ACCURACY: 0.8911076443057723
                     RANDOMFORESTCLASSIFIER I GOT THE ACCURACY: 0.8911076443057723
```

FINALLY I CONCLUDED THAT BOTH RANDOMFORESTCLASSIFIER AND DECISIONTREECLASSIFIER HAS GOT SAME AND HIGH ACCURACY COMPARE TO LOGISTIC REGRESSION AND LINEAR REGRESSION

SO THE RANDOMFORESTCLASSIFIER AND DECISIONTREECLASSIFIER THE BEST FIT MODEL FOR THE FLIGHT PRICE DATASET

In [ ]:
```

```