

```
[32] from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()

data.Airline=le.fit_transform(data.Airline)
data.Source=le.fit_transform(data.Source)
data.Destination=le.fit_transform(data.Destination)
data.Total_Stops=le.fit_transform(data.Total_Stops)
data.City1 =le.fit_transform(data.City1 )
data.City2=le.fit_transform(data.City2)
data.City3=le.fit_transform(data.City3)
data.Additional_Info=le.fit_transform(data.Additional_Info)
data.head()
```

	Airline	Source	Destination	Total_Stops	Additional_Info	Price	Date	Month	Year	City1	City2	City3	Dep_Time_Hour	Dep_Time_Mins	Arrival_date	Arrival_Ti
0	3	0	5	4	7	3897	2	4	/	0	0	3	22	20	0	
1	1	3	0	1	7	7662	1	/	0	0	1	1	05	50	1	
2	4	2	1	1	7	13882	9	/	0	0	2	2	09	25	0	
3	3	3	0	0	7	6218	1	2	/	0	1	1	18	05	2	
4	3	0	5	0	7	13302	0	1	/	0	0	3	16	50	2	

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

```
[68] from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor, AdaBoostRegressor
rfr=RandomForestRegressor()
gb=GradientBoostingRegressor()
ad=AdaBoostRegressor()
```

```
[69] from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
for i in [rfr, gb, ad]:
    i.fit(x_train, y_train)
    y_pred=i.predict(x_test)
    test_score=r2_score(y_test, y_pred)
    train_score=r2_score(y_train, i.predict(x_train))
    if abs(train_score-test_score)<=0.2:
        print(i)
        print("R2 score is", r2_score(y_test, y_pred))
        print("R2 for train data", r2_score(y_train, i.predict(x_train)))
        print("Mean Absolute Error is", mean_absolute_error(y_pred, y_test))
        print("Mean Squared Error is", mean_squared_error(y_pred, y_test))
        print("Root Mean Squared Error is", (mean_squared_error(y_pred, y_test, squared=False)))
```

```
from sklearn.neighbors import KNeighborsRegressor
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
```

+ Code + Text

RAM
Disk

```
[69] print("Root Mean Squared Error is",(mean_squared_error(y_pred,y_test,squared=False)))
```

```
from sklearn.neighbors import KNeighborsRegressor
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import r2_score,mean_absolute_error,mean_squared_error
Knn=KNeighborsRegressor()
svr=SVR()
dt=DecisionTreeRegressor()
for i in [Knn,svr,dt]:
    i.fit(x_train,y_train)
    y_pred=i.predict(x_test)
    test_score=r2_score(y_test,y_pred)
    train_score=r2_score(y_train,i.predict(x_train))
    if abs(train_score-test_score)<=0.1:
        print(i)
        print("R2 score is",r2_score(y_test,y_pred))
        print("R2 for train data",r2_score(y_train,i.predit(x_train)))
        print("Mean Absolute Error is",mean_absolute_error(y_test,y_pred))
        print("Mean Squared Error is",mean_squared_error(y_test,y_pred))
        print("Root Mean Squared Error is",(mean_squared_error(y_test,y_pred,squared=False)))
```

NameError Traceback (most recent call last)

0s completed at 1:38 PM

```

[71] from sklearn.preprocessing import StandardScaler
      ss=StandardScaler()

x_Scaled=ss.fit_transform(x)

[73] x_scaled=pd.DataFrame(x_scaled,columns=x.columns)
      x_scaled.head()

[ ] from sklearn.model_selection import train_test_split
     x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)

x_train.head()

[68] from sklearn.ensemble import RandomForestRegressor,GradientBoostingRegressor,AdaBoostRegressor
      rfr=RandomForestRegressor()
      gb=GradientBoostingRegressor()
      ad=AdaBoostRegressor()
    
```