```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier,GradientBoostingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import f1_score
from sklearn.metrics import classification_report,confusion_matrix
import warnings
import pickle
from scipy import stats
warnings.filterwarnings('ignore')
plt.style.use('fivethirtyeight')
```

```
data=pd.read_csv('/content/Data_Train.csv')
```

```
data.head()
```

	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

```
for i in data:
  print(i,data[i].unique())
```

```
       '11h 25m' '14h 20m' '12h 5m' '24h 5m' '28h 15m' '17h 50m' '20h 20m'
       '28h 5m' '10h 20m' '14h 15m' '35h 15m' '35h 35m' '26h 40m' '28h'
       '14h 25m' '13h 5m' '37h 20m' '36h 10m' '25h 55m' '35h 5m' '19h 45m'
       '27h 55m' '47h' '10h 35m' '1h 35m' '16h 10m' '38h 20m' '6h' '16h 50m'
       '14h 10m' '23h 20m' '17h 40m' '11h 35m' '18h 20m' '6h 40m' '30h 55m'
       '24h 40m' '29h 50m' '28h 25m' '17h 15m' '22h 45m' '25h 25m' '21h 50m'
       '33h 15m' '30h 15m' '3h 35m' '27h 40m' '30h 25m' '18h 50m' '27h 45m'
       '15h 15m' '10h 40m' '26h 15m' '36h 25m' '26h 50m' '15h 45m' '19h 40m'
       '22h 25m' '19h 35m' '25h' '26h 45m' '38h' '4h 15m' '25h 10m' '18h 15m'
       '6h 50m' '23h 55m' '17h 55m' '23h 25m' '17h 10m' '24h 20m' '28h 30m'
       '27h 10m' '19h 20m' '15h 35m' '9h 25m' '21h 30m' '34h 25m' '18h 35m'
       '29h 40m' '26h 5m' '29h 5m' '27h 25m' '16h 30m' '11h 10m' '28h 55m'
       '29h 10m' '34h' '30h 40m' '30h 45m' '32h 55m' '10h 5m' '35h 20m' '32h 5m'
       '31h 40m' '19h 50m' '33h 45m' '30h 10m' '13h 40m' '19h 30m' '31h 30m'
       '24h 20m' '27h 50m' '28h 35m' '42h 5m' '4h 10m' '20h 5m' '2h 50m' '5m'
```

```
data.Date_of_Journey=data.Date_of_Journey.str.split('/')
```

```
data.Date_of_Journey
```

```
0        [24, 03, 2019]
1         [1, 05, 2019]
2         [9, 06, 2019]
3        [12, 05, 2019]
4        [01, 03, 2019]
               ...
10678     [9, 04, 2019]
10679    [27, 04, 2019]
10680    [27, 04, 2019]
10681    [01, 03, 2019]
10682     [9, 05, 2019]
Name: Date_of_Journey, Length: 10683, dtype: object
```

```
data['Date']=data.Date_of_Journey.str[0]
data['Month']=data.Date_of_Journey.str[1]
data['Year']=data.Date_of_Journey.str[2]
```

```
data.Total_Stops.unique()
```

```
array(['non-stop', '2 stops', '1 stop', '3 stops', nan, '4 stops'],
      dtype=object)
```

```
data.Route=data.Route.str.split(' ')
data.Route
```

```
0                        [BLR, ?, DEL]
1        [CCU, ?, IXR, ?, BBI, ?, BLR]
2        [DEL, ?, LKO, ?, BOM, ?, COK]
3                [CCU, ?, NAG, ?, BLR]
4                [BLR, ?, NAG, ?, DEL]
                     ...
10678                    [CCU, ?, BLR]
10679                    [CCU, ?, BLR]
10680                    [BLR, ?, DEL]
10681                    [BLR, ?, DEL]
10682    [DEL, ?, GOI, ?, BOM, ?, COK]
Name: Route, Length: 10683, dtype: object
```

```
data['City1']=data.Route.str[0]
data['City2']=data.Route.str[1]
data['City3']=data.Route.str[2]
data['City4']=data.Route.str[3]
data['City5']=data.Route.str[4]
data['City6']=data.Route.str[5]
```

```
data.Dep_Time=data.Dep_Time.str.split(':')
```

```
data['Dep_Time_hour']=data.Dep_Time.str[0]
data['Dep_Time_Mins']=data.Dep_Time.str[1]
```

```
data.Arrival_Time=data.Arrival_Time.str.split(' ')
```

```
data['Arrival_Date']=data.Arrival_Time.str[1]
data['Time_of_Arrival']=data.Arrival_Time.str[0]


data['Time_of_Arrival']=data.Time_of_Arrival.str.split(':')


data['Arrival_Time_Hours']=data.Time_of_Arrival.str[0]
data['Arrival_Time_Mins']=data.Time_of_Arrival.str[1]


data.Duration=data.Duration.str.split(' ')



data['Travel_Hours']=data.Duration.str[0]
data['Travel_Hours']=data['Travel_Hours'].str.split('h')
data['Travel_Hours']=data['Travel_Hours'].str[0]
data.Travel_Hours=data.Travel_Hours
data['Travel_Mins']=data.Duration.str[1]
data.Travel_Mins=data.Travel_Mins.str.split('m')
data.Travel_Mins=data.Travel_Mins.str[0]



data.Total_Stops.replace('non_stops',0,inplace=True)
data.Total_Stops=data.Total_Stops.str.split(' ')
data.Total_Stops=data.Total_Stops.str[0]



data.Total_Stops.replace('non_stop',0,inplace=True)
data.Total_Stops=data.Total_Stops.str.split(' ')
data.Total_Stops=data.Total_Stops.str[0]




data.Additional_Info.unique()
```

```
array(['No info', 'In-flight meal not included',
       'No check-in baggage included', '1 Short layover', 'No Info',
       '1 Long layover', 'Change airports', 'Business class',
       'Red-eye flight', '2 Long layover'], dtype=object)
```

```
data.Additional_Info.replace('No Info','No info',inplace=True)


data.isnull().sum()
```

```
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
Date                   0
Month                  0
Year                   0
City1                  1
City2                  1
City3                  1
City4               3492
City5               3492
City6               9117
Dep_Time_hour          0
Dep_Time_Mins          0
Arrival_Date        6348
Time_of_Arrival        0
Arrival_Time_Hours     0
Arrival_Time_Mins      0
Travel_Hours           0
Travel_Mins         1032
dtype: int64
```

```
data.drop(['City4','City5','City6'],axis=1,inplace=True)
```

```
data.drop(['Date_of_Journey','Route','Dep_Time','Arrival_Time','Duration'],axis=1,inplace=True)
data.drop(['Time_of_Arrival'],axis=1,inplace=True)
```

```
data.isnull().sum()
```

```
Airline                 0
Source                  0
Destination             0
Total_Stops             1
Additional_Info         0
Price                   0
Date                    0
Month                   0
Year                    0
City1                   1
City2                   1
City3                   1
Dep_Time_hour           0
Dep_Time_Mins           0
Arrival_Date         6348
Arrival_Time_Hours      0
Arrival_Time_Mins       0
Travel_Hours            0
Travel_Mins          1032
dtype: int64
```

```
data['City3'].fillna('None',inplace=True)
```

```
data['Arrival_Date'].fillna(data['Date'],inplace=True)
```

```
data['Travel_Mins'].fillna(0,inplace=True)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 20 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   Airline             10683 non-null  object
 1   Source              10683 non-null  object
 2   Destination         10683 non-null  object
 3   Total_Stops         10682 non-null  object
 4   Additional_Info     10683 non-null  object
 5   Price               10683 non-null  int64 
 6   Date                10683 non-null  object
 7   Month               10683 non-null  object
 8   Year                10683 non-null  object
 9   City1               10682 non-null  object
 10  City2               10682 non-null  object
 11  City3               10683 non-null  object
 12  Dep_Time_hour       10683 non-null  int64 
 13  Dep_Time_Mins       10683 non-null  int64 
 14  Arrival_Date        10683 non-null  int64 
 15  Arrival_Time_Hours  10683 non-null  int64 
 16  Arrival_Time_Mins   10683 non-null  int64 
 17  Travel_Hours        10683 non-null  object
 18  Travel_Mins         10683 non-null  int64 
 19  date                10683 non-null  int64 
dtypes: int64(8), object(12)
memory usage: 1.6+ MB
```

```
data['date']=data.Date.astype('int64')
data['Month'].astype(str).astype(int, errors='ignore')
data['Year'].astype(str).astype(int, errors='ignore')

data['Dep_Time_hour']=data.Dep_Time_hour.astype('int64')
data['Dep_Time_hour']=data.Dep_Time_hour.astype('int64')
data['Dep_Time_Mins']=data.Dep_Time_Mins.astype('int64')
data['Arrival_Date']=data.Arrival_Date.astype('int64')
data['Arrival_Time_Hours']=data.Arrival_Time_Hours.astype('int64')
```

```
data['Arrival_Time_Mins']=data.Arrival_Time_Mins.astype('int64')
data.Travel_Mins=data.Travel_Mins.astype('int64')
```

```
data[data['Travel_Hours']=='5m']
```

	Airline	Source	Destination	Total_Stops	Additional_Info	Price	Date	Month	Year	City1	City2	City3	Dep_Time_hour	Dep_Time_
6474	Air India	Mumbai	Hyderabad	2	No info	17327	6	03	2019	BOM	?	GOI	16	

```
categorical=['Airline','Source','Destination','Additional_Info','City1','month','year']
numerical=['Total_Stops','Date','Dep_Time_Hour','Dep_Time_Mins','Arrival_Date','Arrival_Time_Hours','Arrival_Time_Mins','Travel_Hours','Trave
```

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

```
data.Airli=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_Min
0	IndiGo	0	5	4	7	3897	24	03	2019	0	0	10	22	2
1	Air India	3	0	1	7	7662	1	05	2019	2	0	20	5	5
2	Jet Airways	2	1	1	7	13882	9	06	2019	3	0	27	9	2
3	IndiGo	3	0	0	7	6218	12	05	2019	2	0	29	18	
4	IndiGo	0	5	0	7	13302	01	03	2019	0	0	29	16	5

```
data.head()
```

	Airline	Source	Destination	Total_Stops	Additional_Info	Price	Date	Month	Year	City1	City2	City3	Dep_Time_hour	Dep_Time_Min
0	IndiGo	0	5	4	7	3897	24	03	2019	0	0	10	22	2
1	Air India	3	0	1	7	7662	1	05	2019	2	0	20	5	5
2	Jet Airways	2	1	1	7	13882	9	06	2019	3	0	27	9	2
3	IndiGo	3	0	0	7	6218	12	05	2019	2	0	29	18	
4	IndiGo	0	5	0	7	13302	01	03	2019	0	0	29	16	5

```
data.head()
```

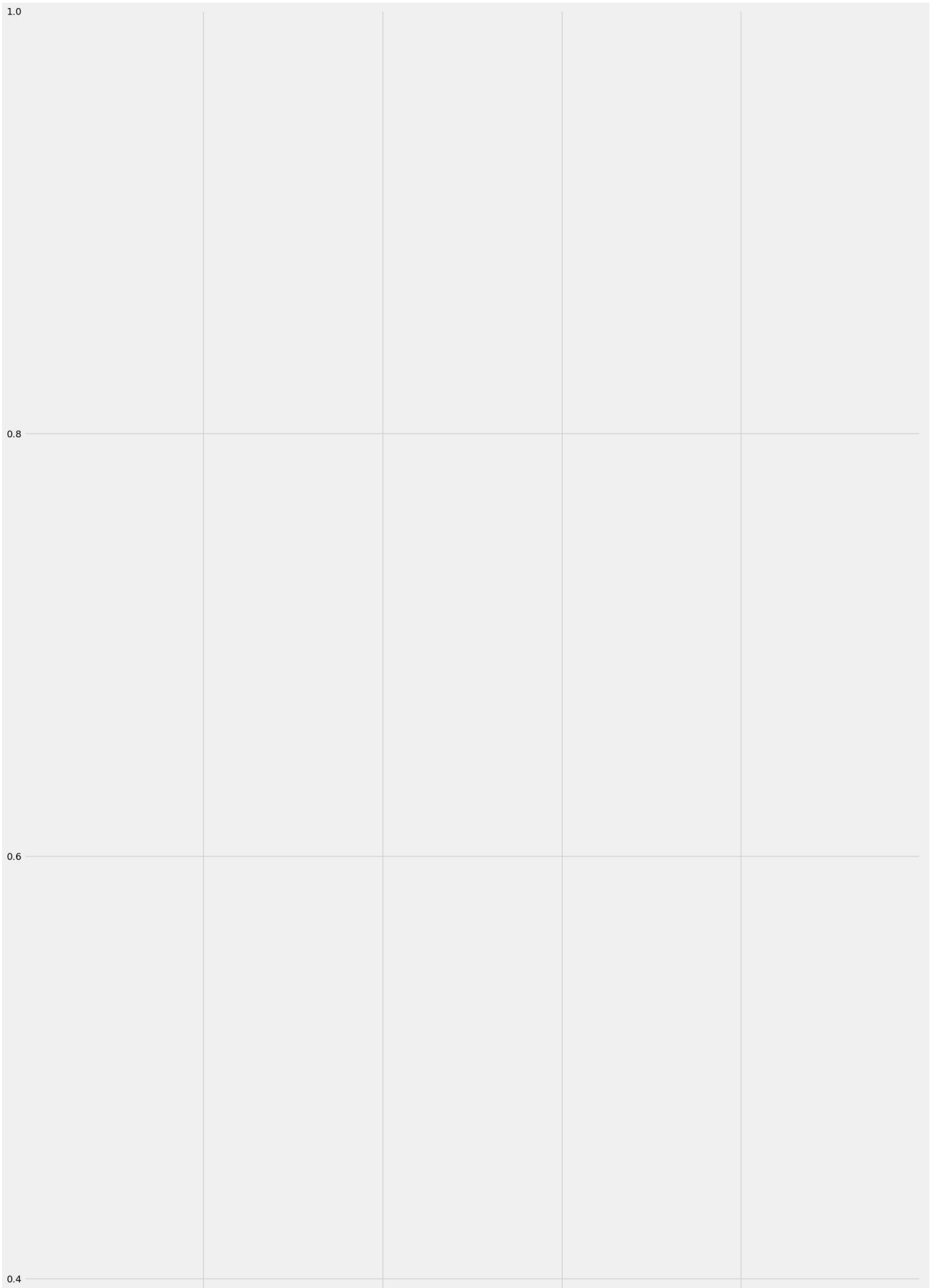
	Airline	Source	Destination	Total_Stops	Additional_Info	Price	Date	Month	Year	City1	City2	City3	Dep_Time_hour	Dep_Time_Min
0	IndiGo	0	5	4	7	3897	24	03	2019	0	0	10	22	2
1	Air India	3	0	1	7	7662	1	05	2019	2	0	20	5	5

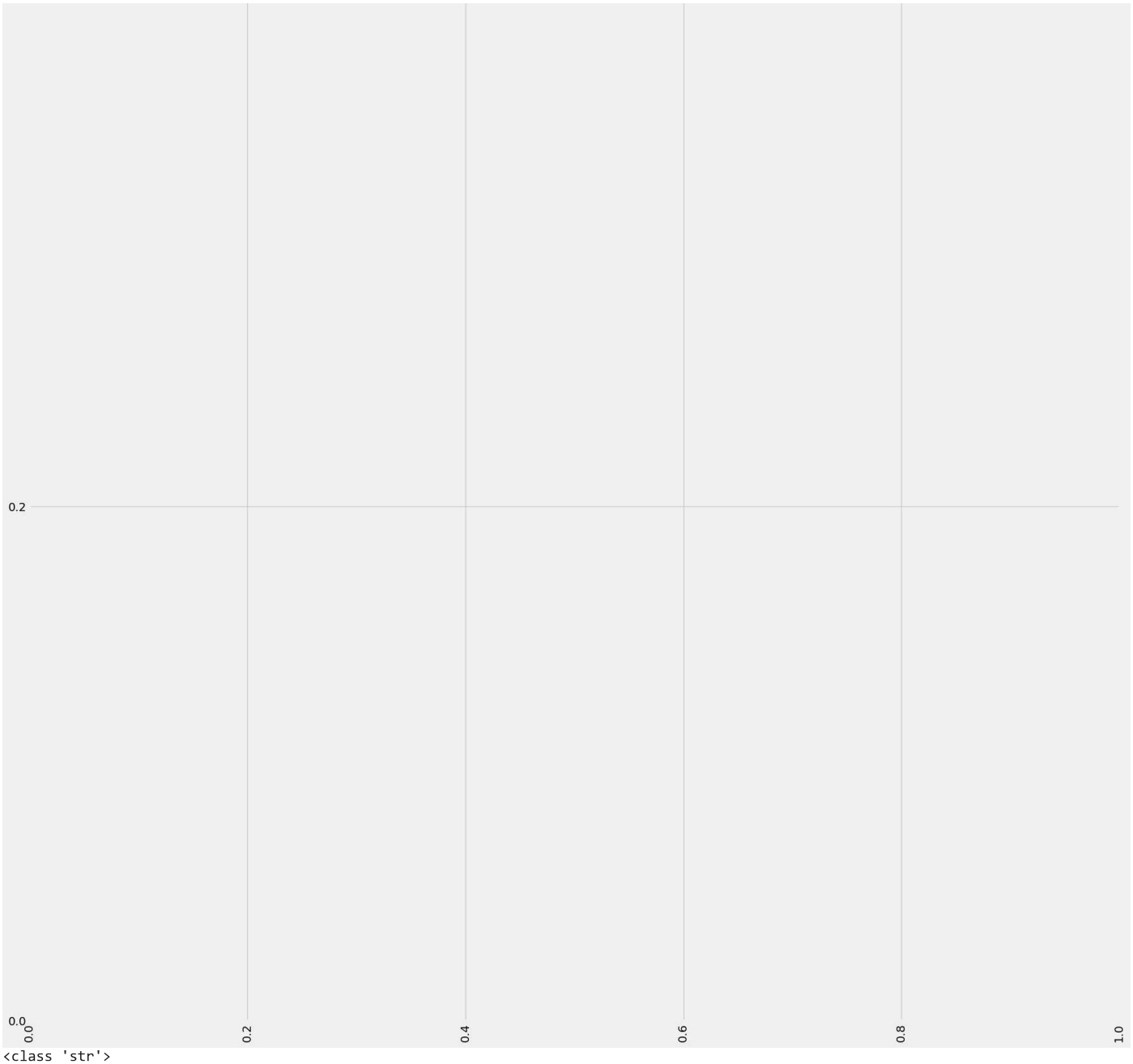
```
data.describe()
```

	Source	Destination	Total_Stops	Additional_Info	Price	City1	City2	City3	Dep_Time_hour	De
count	10683.000000	10683.000000	10683.000000	10683.000000	10683.000000	10683.000000	10683.000000	10683.000000	10683.000000	1
mean	1.952261	1.436113	1.458579	6.582140	9087.064121	2.019657	0.000094	9.683890	12.490686	
std	1.177221	1.474782	1.806560	0.838073	4611.359167	1.206527	0.009675	6.567734	5.748650	
min	0.000000	0.000000	0.000000	0.000000	1759.000000	0.000000	0.000000	0.000000	0.000000	
25%	2.000000	0.000000	0.000000	7.000000	5277.000000	1.000000	0.000000	6.000000	8.000000	
50%	2.000000	1.000000	0.000000	7.000000	8372.000000	2.000000	0.000000	7.000000	11.000000	
75%	3.000000	2.000000	4.000000	7.000000	12373.000000	3.000000	0.000000	10.000000	18.000000	
max	4.000000	5.000000	5.000000	8.000000	79512.000000	5.000000	1.000000	40.000000	23.000000	

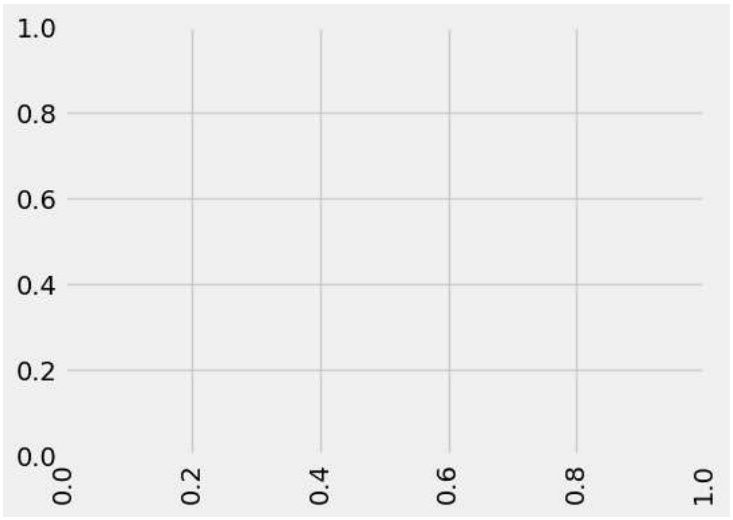
```
import seaborn as sns
c=1
plt.figure(figsize=(20,45))

for i in data:

  #sns.countplot(data(i))
  plt.xticks(rotation=90)
  plt.tight_layout(pad=3.0)
  c=c+1
  print(type (i))
  plt.show()
```

```
<class 'str'>
```

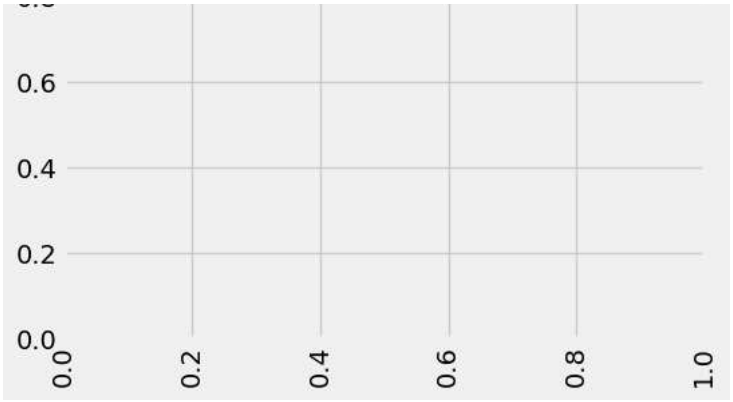




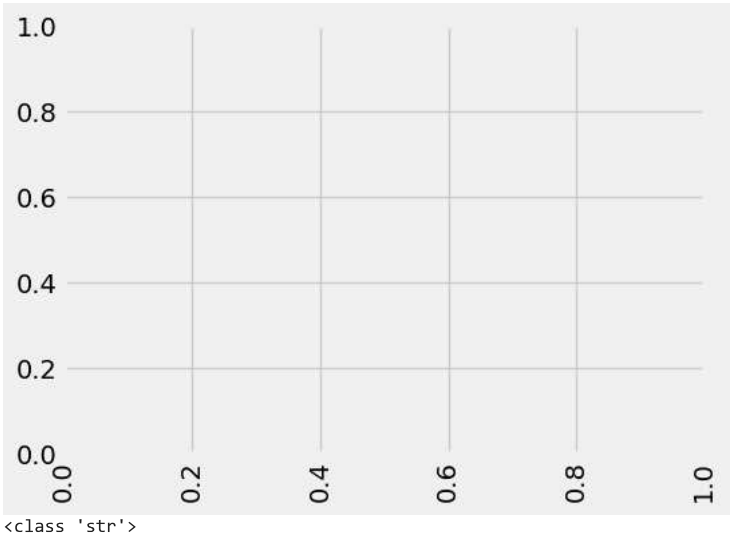

```
<class 'str'>
```



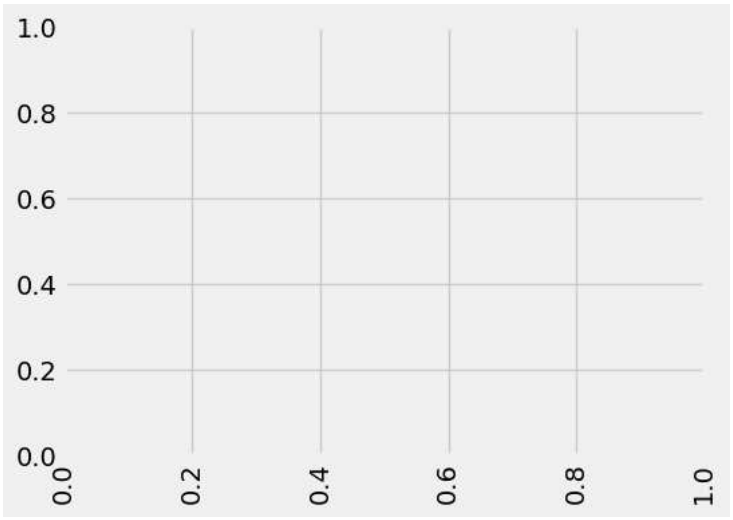

```
<class 'str'>
```



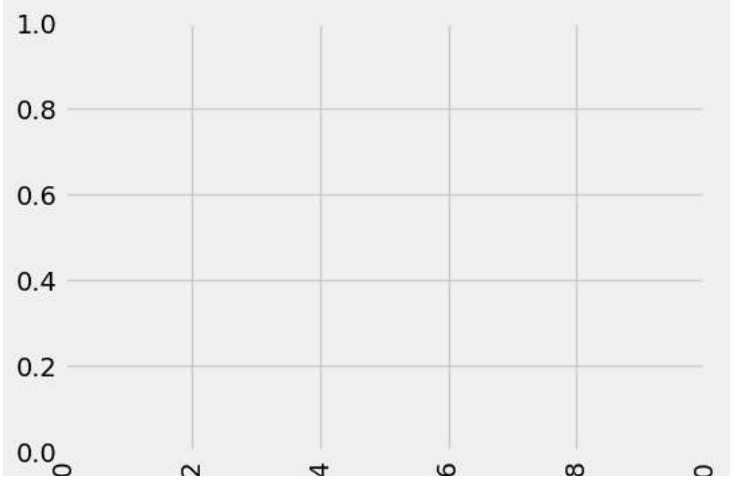

```
<class 'str'>
```

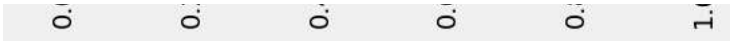



```
<class 'str'>
```

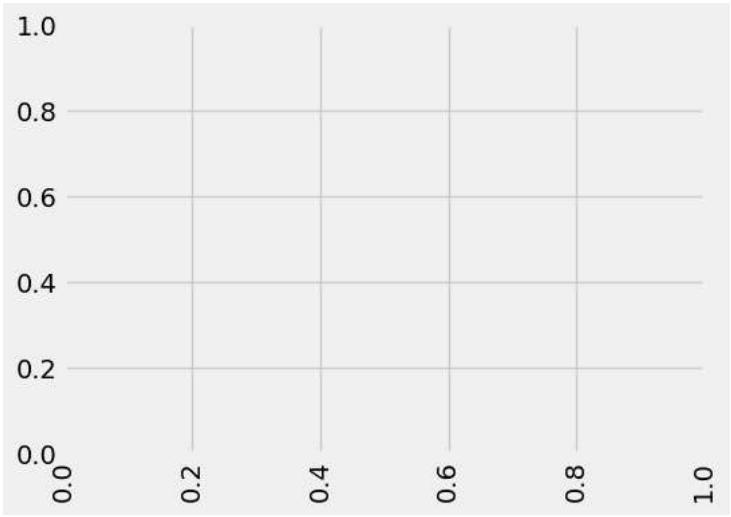



```
<class 'str'>
```

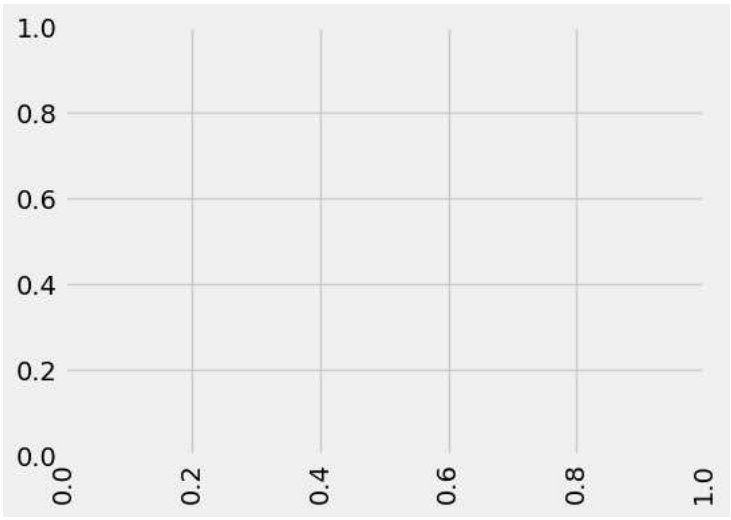




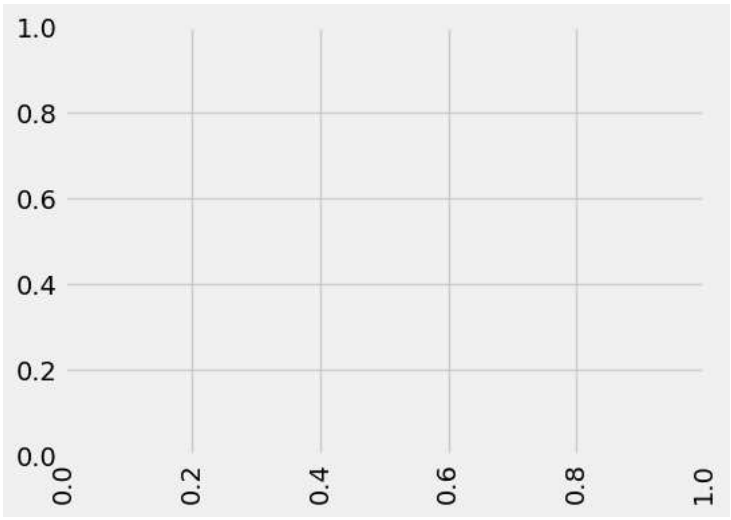
```
<class 'str'>
```



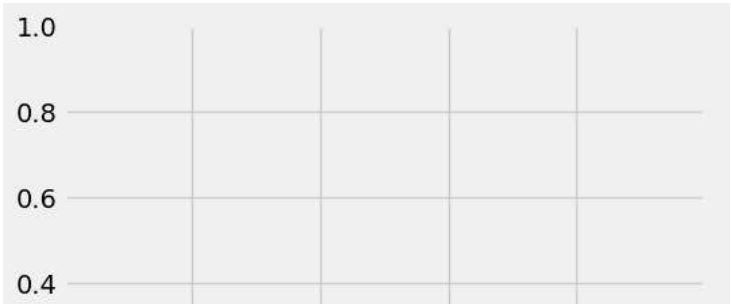

```
<class 'str'>
```

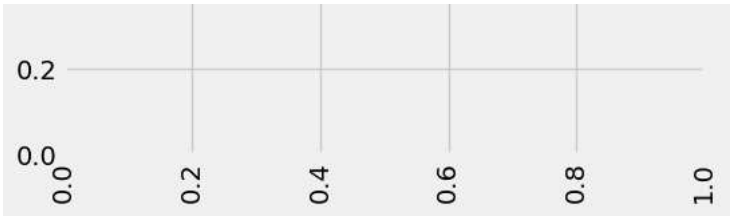



```
<class 'str'>
```

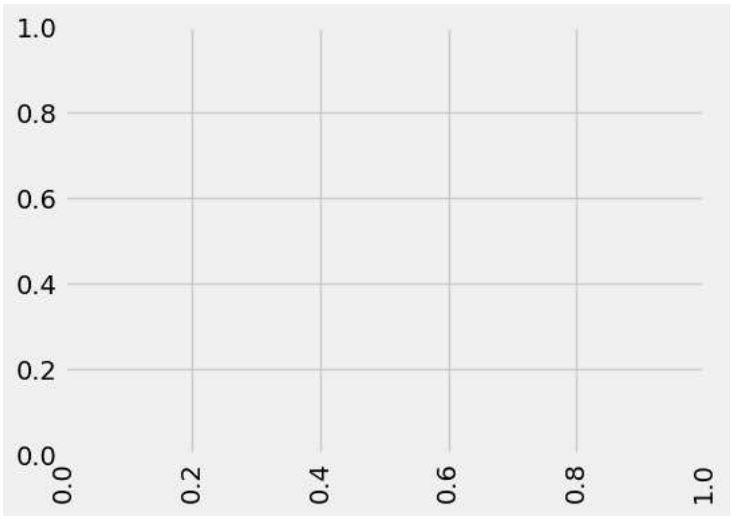



```
<class 'str'>
```

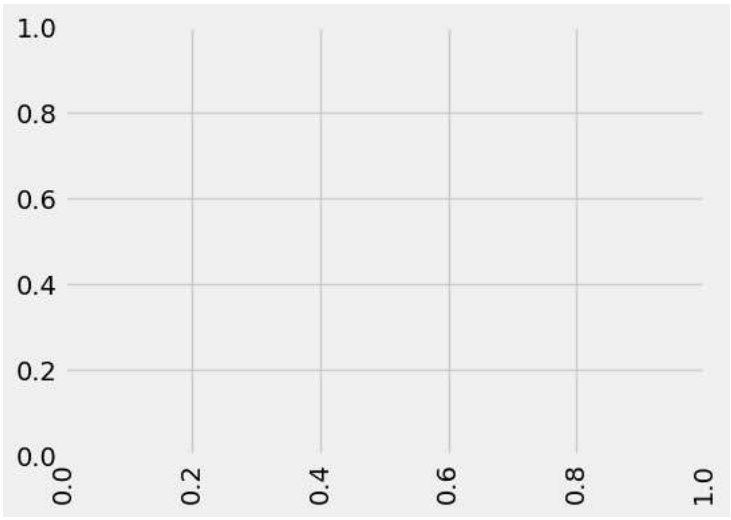




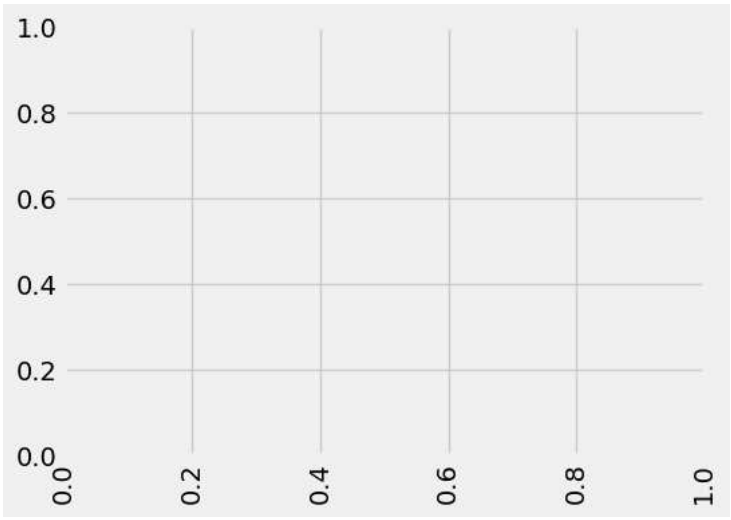

```
<class 'str'>
```



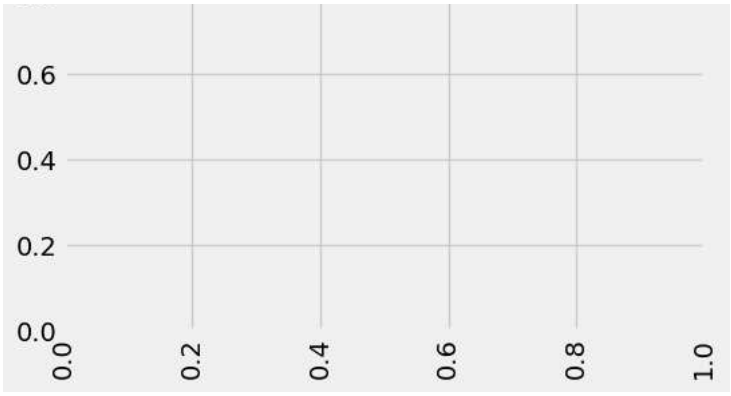

```
<class 'str'>
```



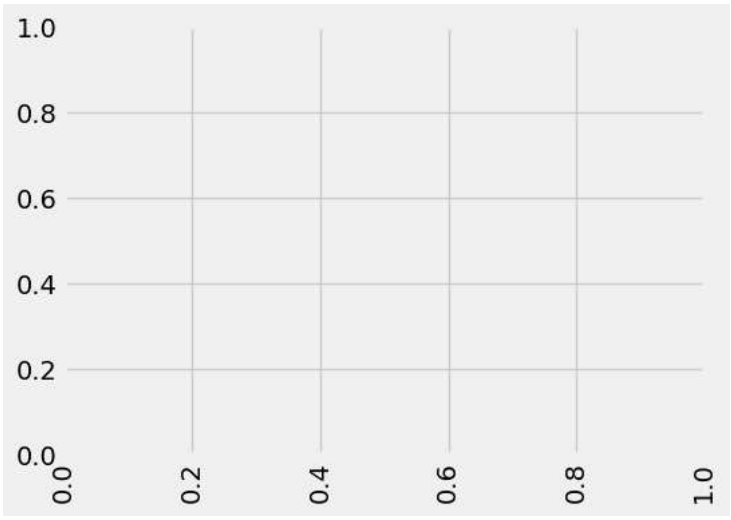

```
<class 'str'>
```



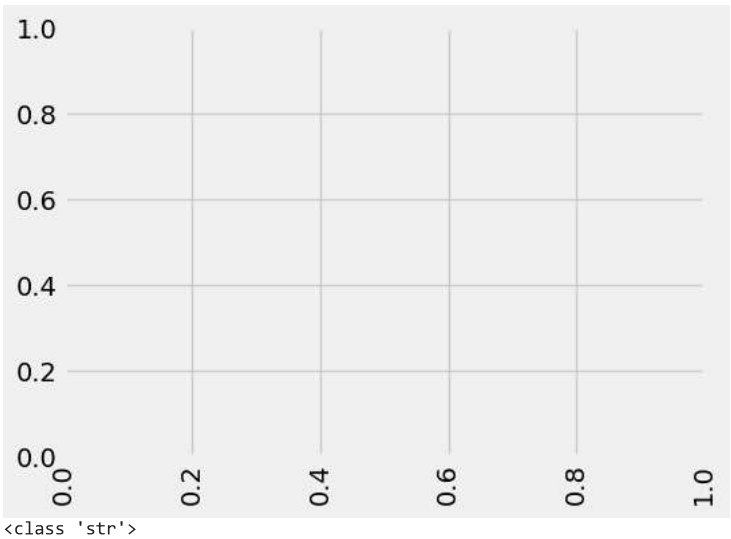

```
<class 'str'>
```



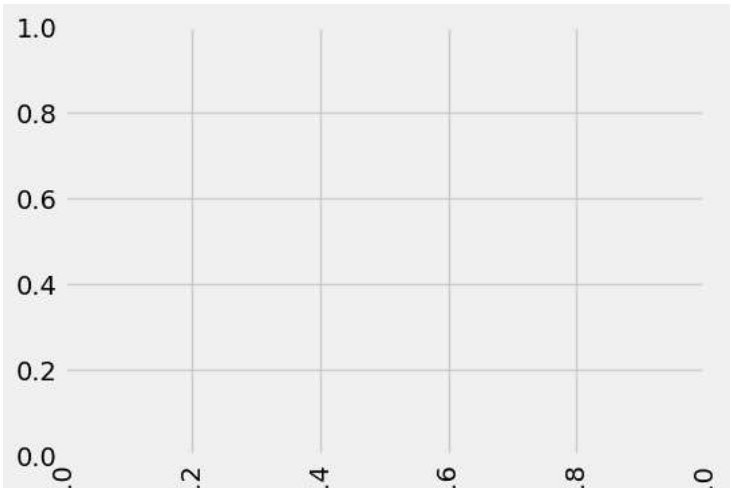

```
<class 'str'>
```

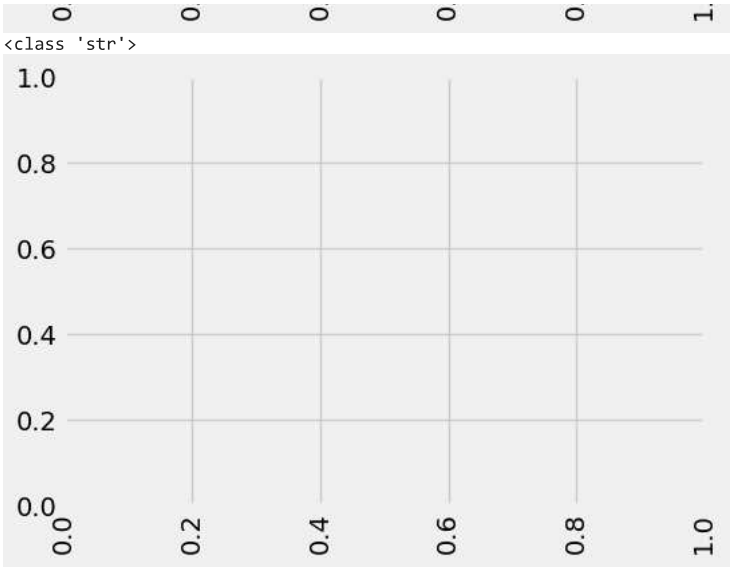



```
<class 'str'>
```

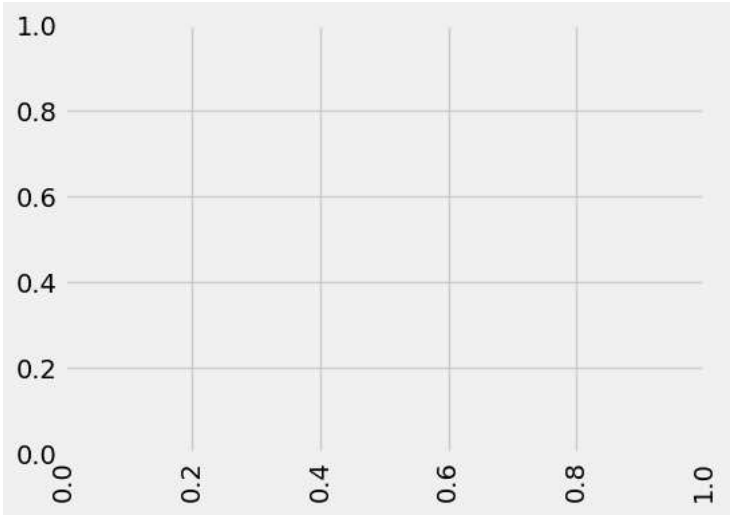



```
<class 'str'>
```

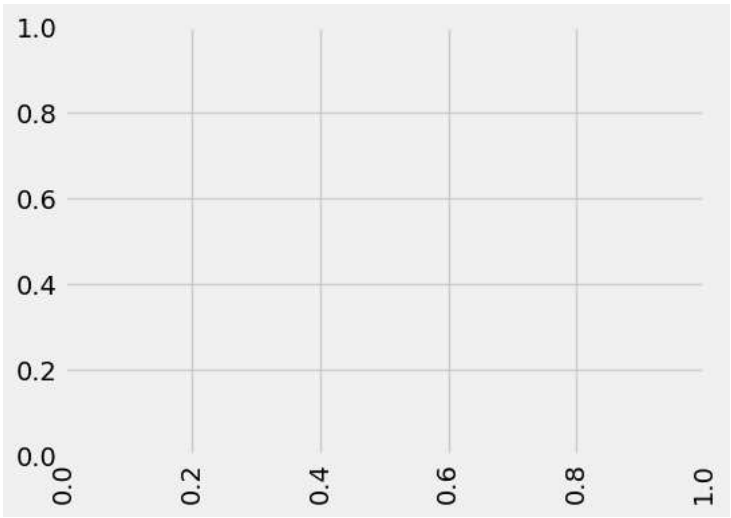





```
<class 'str'>
```




```
<class 'str'>
```

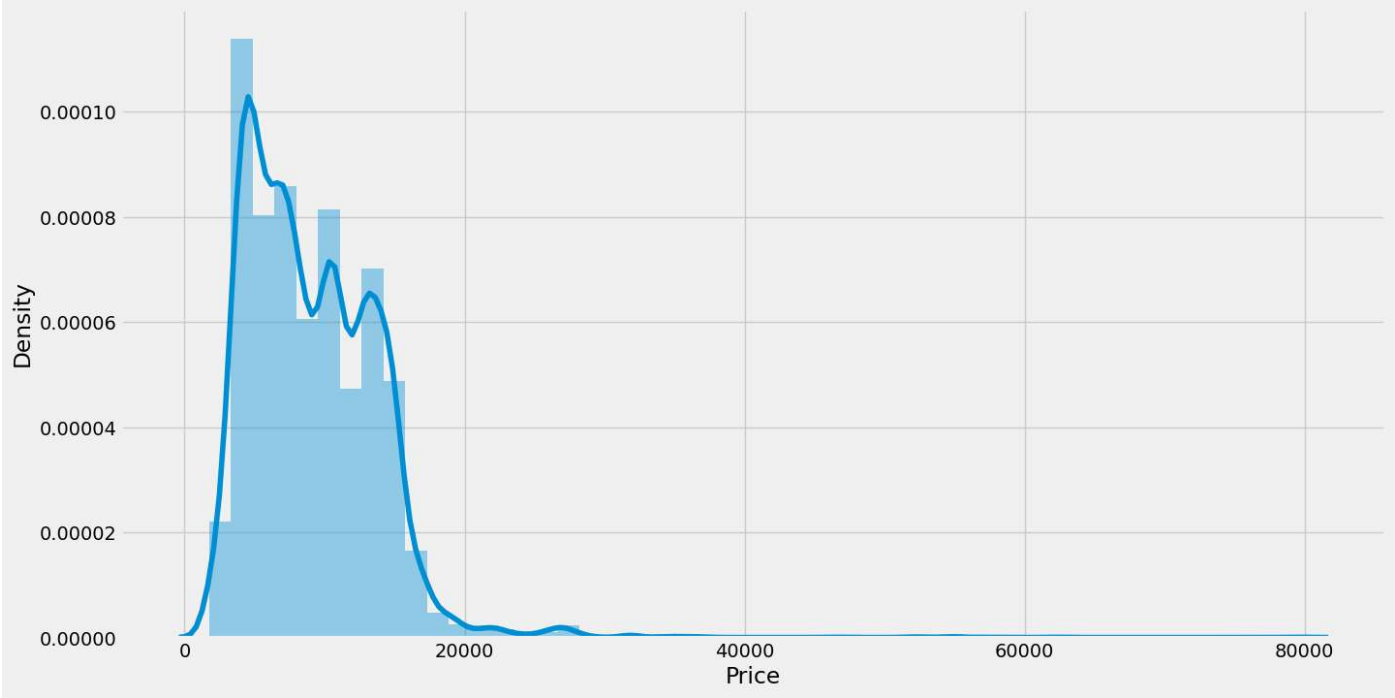



```
<class 'str'>
```



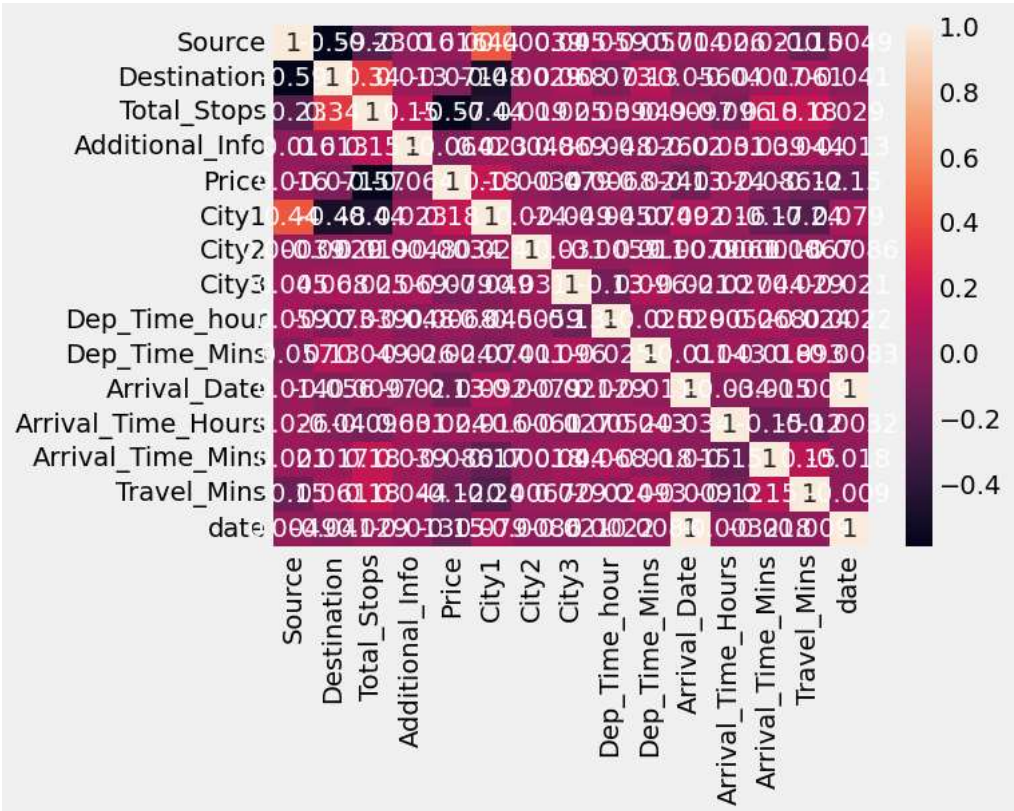
```
plt.figure(figsize=(15,8))
sns.distplot(data.Price)
```

```
<Axes: xlabel='Price', ylabel='Density'>
```



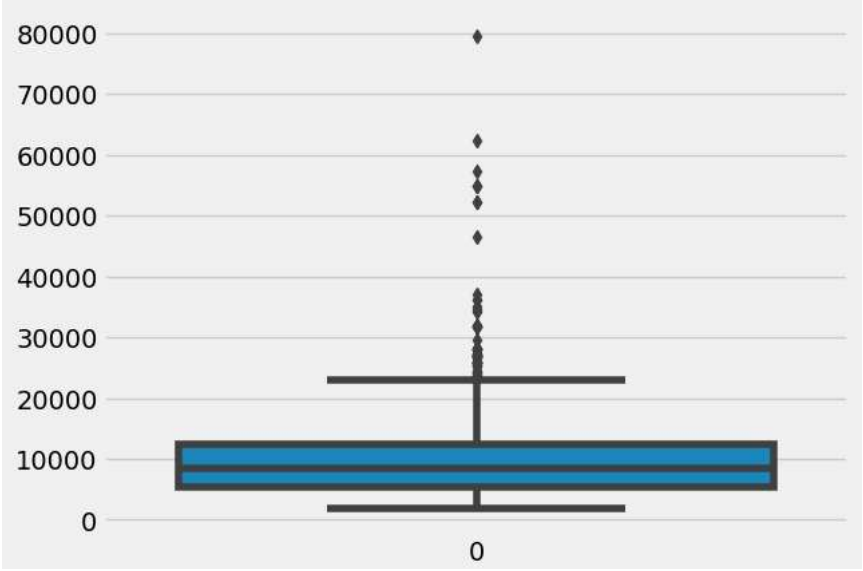

```
sns.heatmap(data.corr(),annot=True)
```

```
<Axes: >
```




```
import seaborn as sns
sns.boxplot(data['Price'])
```

```
<Axes: >
```




```
y=data['Price']
x=data.drop(columns=['Price'],axis=1)


from sklearn.preprocessing import StandardScaler
ss=StandardScaler()


x_scaled = ss.fit_transform(x)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-69-ca9912d0bdd8> in <cell line: 1>()
----> 1 x_scaled = ss.fit_transform(x)

 ──────────────────────────────── 7 frames ────────────────────────────────
/usr/local/lib/python3.9/dist-packages/pandas/core/generic.py in __array__(self, dtype)
   2062
   2063     def __array__(self, dtype: npt.DTypeLike | None = None) -> np.ndarray:
-> 2064         return np.asarray(self._values, dtype=dtype)
   2065
   2066     def __array_wrap__(

ValueError: could not convert string to float: 'IndiGo'
```

SEARCH STACK OVERFLOW

```
x_scaled = pd.DataFrame(x_scaled,columns=x.columns)
x_scaled.head()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-70-d1fc7ecb22a9> in <cell line: 1>()
----> 1 x_scaled = pd.DataFrame(x_scaled,columns=x.columns)
      2 x_scaled.head()

NameError: name 'x_scaled' is not defined
```

SEARCH STACK OVERFLOW

```
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
```

```
<bound method NDFrame.head of                Airline  Source  Destination  Total_Stops  Additional_Info  \
8990        Jet Airways       4            3            1               7
3684        Jet Airways       2            1            0               5
1034          SpiceJet        2            1            0               7
3909  Multiple carriers       2            1            0               7
3088          Air India       2            1            1               7
...                 ...     ...          ...          ...             ...
```