Project 2- Group 6

In [1]:
```
import numpy as np
import pandas as pd
import math
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import accuracy_score, roc_curve, auc
import matplotlib.pylab as plt
from dmba import regressionSummary, classificationSummary
from dmba import liftChart, gainsChart
import seaborn as sns
#importing libraries
```

In [2]:
```
# reading the main dataset

df1= pd.read_excel('flightdelay1.xlsx')
```

In [3]:
```
df1.head(20)
```

Out[3]:

	CRS_DEP_TIME	CARRIER	DEP_TIME	DEST	DISTANCE	FL_DATE	FL_NUM	ORIGIN	W
0	1455	DL	1458	JFK	213	2004-01-01	746	DCA	
1	1455	DL	1458	JFK	213	2004-01-02	746	DCA	
2	1455	DL	1505	JFK	213	2004-01-03	746	DCA	
3	1455	DL	1500	JFK	213	2004-01-04	746	DCA	
4	1455	DL	1459	JFK	213	2004-01-05	746	DCA	
5	1455	DL	1457	JFK	213	2004-01-06	746	DCA	
6	1455	DL	1501	JFK	213	2004-01-07	746	DCA	
7	1455	DL	1601	JFK	213	2004-01-08	746	DCA	
8	1455	DL	1506	JFK	213	2004-01-09	746	DCA	
9	1455	DL	1505	JFK	213	2004-01-10	746	DCA	
10	1455	DL	1456	JFK	213	2004-01-11	746	DCA	
11	1455	DL	1451	JFK	213	2004-01-12	746	DCA	
12	1455	DL	1453	JFK	213	2004-01-13	746	DCA	
13	1455	DL	1454	JFK	213	2004-01-14	746	DCA	
14	1455	DL	1501	JFK	213	2004-01-15	746	DCA	
15	1455	DL	1500	JFK	213	2004-01-16	746	DCA	
16	1455	DL	1509	JFK	213	2004-01-17	746	DCA	
17	1455	DL	1555	JFK	213	2004-01-18	746	DCA	
18	1455	DL	1506	JFK	213	2004-01-19	746	DCA	
19	1455	DL	1514	JFK	213	2004-01-20	746	DCA	

In [4]:
```
#converting categorical to numeric

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

In [5]:
```
#selecting the data for encoding

df2 = df1.iloc[1: , :]
```

In [6]:
```
#making a copy for the numeric version of dataset

df1_enc=df2
```

In [7]:
```
for i in df1_enc:
        df1_enc[i]=lab_enc.fit_transform(df1_enc[i])
```

```
C:\Users\Aditya Dabrase\AppData\Local\Temp\ipykernel_17372\3801913590.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://p
andas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-vi
ew-versus-a-copy)
  df1_enc[i]=lab_enc.fit_transform(df1_enc[i])
```

In [8]:
```
df1_enc
```

Out[8]:

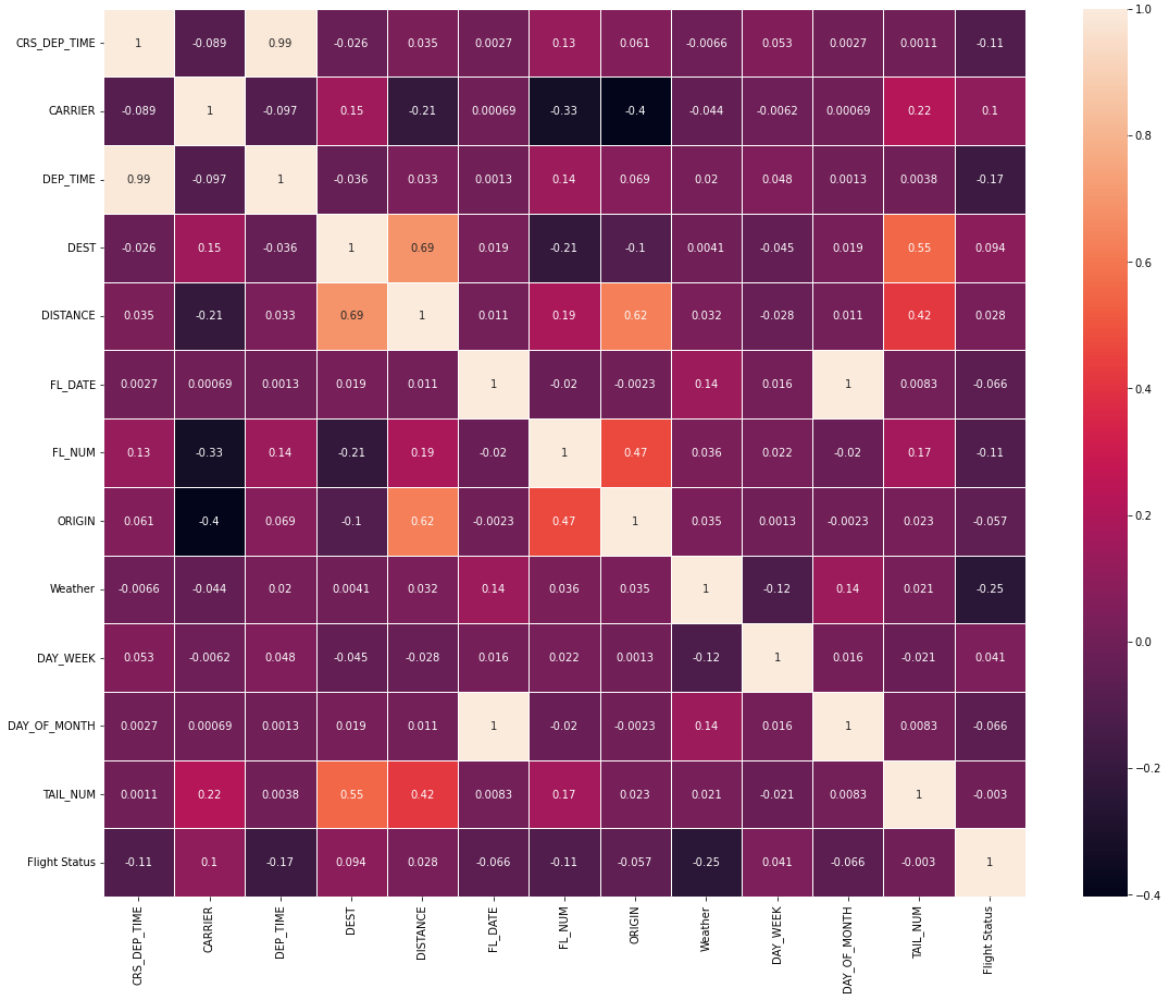
	CRS_DEP_TIME	CARRIER	DEP_TIME	DEST	DISTANCE	FL_DATE	FL_NUM	ORIGIN
1	32	2	316	1	3	1	0	1
2	32	2	323	1	3	2	0	1
3	32	2	318	1	3	3	0	1
4	32	2	317	1	3	4	0	1
5	32	2	315	1	3	5	0	1
...	...	...	...	...	...	...	...	...
2196	57	1	595	2	6	1	102	2
2197	57	1	610	2	6	2	102	2
2198	57	1	593	2	6	3	102	2
2199	57	1	601	2	6	4	102	2
2200	57	1	595	2	6	5	102	2

2200 rows × 13 columns

In [9]:
```
#corelation matrix using the numeric dataframe

plt.figure(figsize = (19,15))
sns.heatmap(df1_enc.corr(),annot=True,linewidths=1)
```

Out[9]: <AxesSubplot:>



In [10]:
```
#from the corelation chart above we can eliminate the crs departure time ,car
#we can see that distance and origin are corelated
#origin and flight number are corelated
#Distance, destination and origin are corelated
#Destination and tail number are corelated
```

In [11]:
```
#deleting the extra columns which will not be used for our analysis

df1=df1.drop(["CRS_DEP_TIME","CARRIER"],axis = 1)
```

In [12]:
```
#making a copy of dataframe for further data exploration

df3=df1
```

In [13]:
```
df1.to_excel(r'C:\Users\Aditya Dabrase\Desktop\FlightDelaysTrainingData.xlsx'
df3.to_excel(r'C:\Users\Aditya Dabrase\Desktop\FlightDelaysDataExploration.xl
```

In [14]:
```
df3.head()
#final df with reduced columns
```

Out[14]:

	DEP_TIME	DEST	DISTANCE	FL_DATE	FL_NUM	ORIGIN	Weather	DAY_WEEK	DAY_OF_
0	1458	JFK	213	2004-01-01	746	DCA	0	4	
1	1458	JFK	213	2004-01-02	746	DCA	0	5	
2	1505	JFK	213	2004-01-03	746	DCA	0	6	
3	1500	JFK	213	2004-01-04	746	DCA	0	7	
4	1459	JFK	213	2004-01-05	746	DCA	0	1	

In [15]:
```
df1_enc.head()
```

Out[15]:

	CRS_DEP_TIME	CARRIER	DEP_TIME	DEST	DISTANCE	FL_DATE	FL_NUM	ORIGIN	Wea
1	32	2	316	1	3	1	0	1	
2	32	2	323	1	3	2	0	1	
3	32	2	318	1	3	3	0	1	
4	32	2	317	1	3	4	0	1	
5	32	2	315	1	3	5	0	1	

In [16]:
```
df1_enc=df1_enc.drop(["CRS_DEP_TIME","CARRIER"],axis = 1)
```

In [17]:
```
df1_enc.head()
#final encoded dataframe with reduced columns
```

Out[17]:

	DEP_TIME	DEST	DISTANCE	FL_DATE	FL_NUM	ORIGIN	Weather	DAY_WEEK	DAY_OF_
1	316	1	3	1	0	1	0	4	
2	323	1	3	2	0	1	0	5	
3	318	1	3	3	0	1	0	6	
4	317	1	3	4	0	1	0	0	
5	315	1	3	5	0	1	0	1	

Data Exploration and Pivot tables

In [96]:
```
df1_enc.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2200 entries, 1 to 2200
Data columns (total 11 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   DEP_TIME       2200 non-null   int64
 1   DEST           2200 non-null   int32
 2   DISTANCE       2200 non-null   int64
 3   FL_DATE        2200 non-null   int64
 4   FL_NUM         2200 non-null   int64
 5   ORIGIN         2200 non-null   int32
 6   Weather        2200 non-null   int64
 7   DAY_WEEK       2200 non-null   int64
 8   DAY_OF_MONTH   2200 non-null   int64
 9   TAIL_NUM       2200 non-null   int32
 10  Flight Status  2200 non-null   int32
dtypes: int32(4), int64(7)
memory usage: 154.8 KB
```

In [97]:
```
#this pivot table gives us information about the distance from origin to dest

pivot1 = df3.pivot_table(index=['ORIGIN','DEST'],
                         values=['DISTANCE'], aggfunc={'mean'})
pivot1
```

Out[97]:

		DISTANCE
		mean
ORIGIN	DEST	
BWI	EWR	169.0
	JFK	184.0
DCA	EWR	199.0
	JFK	213.0
	LGA	214.0
IAD	EWR	213.0
	JFK	228.0
	LGA	229.0

In [94]:
```
# table 2 counts for number of flights that were delayed (0)vs number of flig

pivot2 = df3.pivot_table(index=['Flight Status'],
                         values=['DEST'], aggfunc={'count'})
pivot2
```

Out[94]:

	DEST
	count
Flight Status	
delayed	428
ontime	1773

In [93]:
```
# table 3 shows the effect of weather on the flight status

pivot3 = df3.pivot_table(index=['Weather','Flight Status'],
                         values=['DISTANCE'], aggfunc={'count'})
pivot3
```

Out[93]:

		DISTANCE
		count
Weather	Flight Status	
0	delayed	396
	ontime	1773
1	delayed	32

In [91]:
```
# Pivot4 shows us the flight status depending on Origin


pivot4=df3.pivot_table(index = ['ORIGIN','Flight Status'], values = "DISTANCE
aggfunc = [ len],
margins=True,
margins_name='Grand Totals')
pivot4
```

Out[91]:

		len
		DISTANCE
ORIGIN	Flight Status	
BWI	delayed	37
	ontime	108
DCA	delayed	221
	ontime	1149
IAD	delayed	170
	ontime	516
Grand Totals		2201

In [98]:
```
# Pivot 5 shows us the flight status depending on Destination

pivot5= df3.pivot_table(index = ['DEST','Flight Status'], values = "DISTANCE"
aggfunc = [ len],
margins=True,
margins_name='Grand Totals')
pivot5
```

Out[98]:

		len
		DISTANCE
DEST	Flight Status	
EWR	delayed	161
	ontime	504
JFK	delayed	84
	ontime	302
LGA	delayed	183
	ontime	967
Grand Totals		2201

In [ ]:
```

```