flight-delay-analysis

November 17, 2024

#data collection

[94]:
```
%matplotlib inline
import datetime, warnings, scipy
warnings.filterwarnings("ignore")
import numpy as np
import pandas as pd
pd.set_option('display.max_columns', None)
import matplotlib.pyplot as plt
import seaborn as sns
```

[95]:
```
flight_data = pd.read_csv('2018.csv')

# Checking first 2 instances and last 2 instances
pd.concat([flight_data.head(2), flight_data.tail(2)])
```

[95]:

	FL_DATE	OP_CARRIER	OP_CARRIER_FL_NUM	ORIGIN	DEST	CRS_DEP_TIME	DEP_TIME	DEP_DELAY	TAXI_OUT	WHEELS_OFF	WHEELS_ON	TAXI_IN	CRS_ARR_TIME	ARR_TIME	ARR_DELAY	CANCELLED	CANCELLATION_CODE	DIVERTED	CRS_ELAPSED_TIME	ACTUAL_ELAPSED_TIME	AIR_TIME	DISTANCE
0	2018-01-01	UA	2429	EWR	DEN	1517	1512.0	-5.0	15.0	1527.0	1712.0	10.0	1745	1722.0	-23.0	0.0	NaN	0.0	268.0	250.0	225.0	1605.0
1	2018-01-01	UA	2427	LAS	SFO	1115	1107.0	-8.0	11.0	1118.0	1223.0	7.0	1254	1230.0	-24.0	0.0	NaN	0.0	99.0	83.0	65.0	414.0
7213444	2018-12-31	AA	1818	CLT	RDU	1300	1323.0	23.0	11.0	1334.0	1400.0	4.0	1350	1404.0	14.0	0.0	NaN	0.0	50.0	41.0	26.0	130.0
7213445	2018-12-31	AA	1818	RDU	CLT	1435	1443.0	8.0	8.0	1451.0	1535.0	7.0	1546	1542.0	-4.0	0.0	NaN	0.0	71.0	59.0	43.0	231.0

```
7213445       0.0              71.0                  59.0       44.0      130.0

         CARRIER_DELAY  WEATHER_DELAY  NAS_DELAY  SECURITY_DELAY  \
0                  NaN            NaN        NaN             NaN
1                  NaN            NaN        NaN             NaN
7213444            NaN            NaN        NaN             NaN
7213445            NaN            NaN        NaN             NaN

         LATE_AIRCRAFT_DELAY  Unnamed: 27
0                        NaN          NaN
1                        NaN          NaN
7213444                  NaN          NaN
7213445                  NaN          NaN
```

[96]:
```
flight_data.shape
print("There are "+ str(flight_data.shape[0]) +" rows and "+ str(flight_data.
 ↪shape[1]) +" columns from the flight dataset.")
```

```
There are 7213446 rows and 28 columns from the flight dataset.
```

[97]:
```
flight_data['FL_DATE'] =  pd.to_datetime(flight_data['FL_DATE'],␣
 ↪format='%Y-%m-%d')
```

[98]:
```
flight_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7213446 entries, 0 to 7213445
Data columns (total 28 columns):
 #   Column               Dtype
---  ------               -----
 0   FL_DATE              datetime64[ns]
 1   OP_CARRIER           object
 2   OP_CARRIER_FL_NUM    int64
 3   ORIGIN               object
 4   DEST                 object
 5   CRS_DEP_TIME         int64
 6   DEP_TIME             float64
 7   DEP_DELAY            float64
 8   TAXI_OUT             float64
 9   WHEELS_OFF           float64
 10  WHEELS_ON            float64
 11  TAXI_IN              float64
 12  CRS_ARR_TIME         int64
 13  ARR_TIME             float64
 14  ARR_DELAY            float64
 15  CANCELLED            float64
 16  CANCELLATION_CODE    object
```

```
 17  DIVERTED              float64
 18  CRS_ELAPSED_TIME      float64
 19  ACTUAL_ELAPSED_TIME   float64
 20  AIR_TIME              float64
 21  DISTANCE              float64
 22  CARRIER_DELAY         float64
 23  WEATHER_DELAY         float64
 24  NAS_DELAY             float64
 25  SECURITY_DELAY        float64
 26  LATE_AIRCRAFT_DELAY   float64
 27  Unnamed: 27           float64
dtypes: datetime64[ns](1), float64(20), int64(3), object(4)
memory usage: 1.5+ GB
```

#Data Preprocessing

[99]:
```
def checkMissing(data,perc=0):
    missing_values = [(i, data[i].isna().mean()*100) for i in data]
    missing_values = pd.DataFrame(missing_values, columns=["column_name",␣
 ↪"percentage"])
    missing_values = missing_values[missing_values.percentage > perc]
    print(missing_values.sort_values("percentage", ascending=False).
 ↪reset_index(drop=True))

print("Proportion of missing data in columns")
checkMissing(flight_data)
```

```
Proportion of missing data in columns
            column_name  percentage
0           Unnamed: 27  100.000000
1     CANCELLATION_CODE   98.383796
2   LATE_AIRCRAFT_DELAY   81.247382
3         CARRIER_DELAY   81.247382
4         WEATHER_DELAY   81.247382
5             NAS_DELAY   81.247382
6        SECURITY_DELAY   81.247382
7             ARR_DELAY    1.899785
8   ACTUAL_ELAPSED_TIME    1.863769
9              AIR_TIME    1.863769
10            WHEELS_ON    1.653107
11              TAXI_IN    1.653107
12             ARR_TIME    1.653093
13            DEP_DELAY    1.625215
14             TAXI_OUT    1.605751
15           WHEELS_OFF    1.605737
16             DEP_TIME    1.557051
17     CRS_ELAPSED_TIME    0.000139
```

[100]:
```
flight_data['LATE_AIRCRAFT_DELAY']=flight_data['LATE_AIRCRAFT_DELAY'].fillna(0)
flight_data['CARRIER_DELAY']=flight_data['CARRIER_DELAY'].fillna(0)
flight_data['WEATHER_DELAY']=flight_data['WEATHER_DELAY'].fillna(0)
flight_data['NAS_DELAY']=flight_data['NAS_DELAY'].fillna(0)
flight_data['SECURITY_DELAY']=flight_data['SECURITY_DELAY'].fillna(0)
```

[101]:
```
def format_data(x):
    if pd.isnull(x):
        return np.nan
    else:
        if x == 2400: x = 0
        x = "{0:04d}".format(int(x))
        res = datetime.time(int(x[0:2]), int(x[2:4]))
        return res
```

[102]:
```
flight_data['DEP_TIME'] = flight_data['DEP_TIME'].apply(format_data)
flight_data['CRS_DEP_TIME'] = flight_data['CRS_DEP_TIME'].apply(format_data)

flight_data['ARR_TIME'] = flight_data['ARR_TIME'].apply(format_data)
flight_data['CRS_ARR_TIME'] = flight_data['CRS_ARR_TIME'].apply(format_data)

flight_data['WHEELS_OFF'] = flight_data['WHEELS_OFF'].apply(format_data)
flight_data['WHEELS_ON'] = flight_data['WHEELS_ON'].apply(format_data)
```

[103]:
```
def time_difference(actual,plan):
    actual_time = pd.to_timedelta(actual.astype(str))
    plan_time = pd.to_timedelta(plan.astype(str))
    return actual_time.sub(plan_time).dt.total_seconds().div(60)

flight_data['WHEELS_OFF_elapse'] = ␣
 ↪time_difference(flight_data['WHEELS_OFF'],flight_data['DEP_TIME'])
flight_data['WHEELS_ON_elapse'] =  time_difference(flight_data['ARR_TIME']␣
 ↪,flight_data['WHEELS_ON'])
flight_data=flight_data[flight_data['WHEELS_OFF_elapse']>0]
flight_data=flight_data[flight_data['WHEELS_ON_elapse']>0]
```

[104]:
```
flight_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 7051259 entries, 0 to 7213445
Data columns (total 30 columns):
 #   Column               Dtype
---  ------               -----
 0   FL_DATE              datetime64[ns]
 1   OP_CARRIER           object
 2   OP_CARRIER_FL_NUM    int64
 3   ORIGIN               object
```

```
 4   DEST                 object
 5   CRS_DEP_TIME         object
 6   DEP_TIME             object
 7   DEP_DELAY            float64
 8   TAXI_OUT             float64
 9   WHEELS_OFF           object
 10  WHEELS_ON            object
 11  TAXI_IN              float64
 12  CRS_ARR_TIME         object
 13  ARR_TIME             object
 14  ARR_DELAY            float64
 15  CANCELLED            float64
 16  CANCELLATION_CODE    object
 17  DIVERTED             float64
 18  CRS_ELAPSED_TIME     float64
 19  ACTUAL_ELAPSED_TIME  float64
 20  AIR_TIME             float64
 21  DISTANCE             float64
 22  CARRIER_DELAY        float64
 23  WEATHER_DELAY        float64
 24  NAS_DELAY            float64
 25  SECURITY_DELAY       float64
 26  LATE_AIRCRAFT_DELAY  float64
 27  Unnamed: 27          float64
 28  WHEELS_OFF_elapse    float64
 29  WHEELS_ON_elapse     float64
dtypes: datetime64[ns](1), float64(18), int64(1), object(10)
memory usage: 1.6+ GB
```

[105]:
```
flight_data.drop(['Unnamed: 27',
                  'CANCELLATION_CODE',
                  'CANCELLED',
                  'OP_CARRIER_FL_NUM',
                  'CRS_DEP_TIME',
                  'DEP_TIME',
                  'CRS_ARR_TIME',
                  'ARR_TIME',
                  'WHEELS_ON',
                  'WHEELS_OFF'
                 ],
                 axis = 1, inplace = True)
```

[106]:
```
flight_data.isna().sum()
```

[106]:
```
FL_DATE                    0
OP_CARRIER                 0
ORIGIN                     0
```

```
DEST                      0
DEP_DELAY              4735
TAXI_OUT                  0
TAXI_IN                   0
ARR_DELAY             17560
DIVERTED                  0
CRS_ELAPSED_TIME          7
ACTUAL_ELAPSED_TIME   14962
AIR_TIME              14962
DISTANCE                  0
CARRIER_DELAY             0
WEATHER_DELAY             0
NAS_DELAY                 0
SECURITY_DELAY            0
LATE_AIRCRAFT_DELAY       0
WHEELS_OFF_elapse         0
WHEELS_ON_elapse          0
dtype: int64
```

[107]:
```
flight_data = flight_data.dropna()
flight_data.isna().sum()
```

[107]:
```
FL_DATE               0
OP_CARRIER            0
ORIGIN                0
DEST                  0
DEP_DELAY             0
TAXI_OUT              0
TAXI_IN               0
ARR_DELAY             0
DIVERTED              0
CRS_ELAPSED_TIME      0
ACTUAL_ELAPSED_TIME   0
AIR_TIME              0
DISTANCE              0
CARRIER_DELAY         0
WEATHER_DELAY         0
NAS_DELAY             0
SECURITY_DELAY        0
LATE_AIRCRAFT_DELAY   0
WHEELS_OFF_elapse     0
WHEELS_ON_elapse      0
dtype: int64
```

[108]:
```
flight_data.OP_CARRIER.unique()
```

[108]:
```
array(['UA', 'AS', '9E', 'B6', 'EV', 'F9', 'G4', 'HA', 'MQ', 'NK', 'OH',
       'OO', 'VX', 'WN', 'YV', 'YX', 'AA', 'DL'], dtype=object)
```

[109]:
```
flight_data['OP_CARRIER'].replace({
    'UA':'United Airlines',
    'AS':'Alaska Airlines',
    '9E':'Endeavor Air',
    'B6':'JetBlue Airways',
    'EV':'ExpressJet',
    'F9':'Frontier Airlines',
    'G4':'Allegiant Air',
    'HA':'Hawaiian Airlines',
    'MQ':'Envoy Air',
    'NK':'Spirit Airlines',
    'OH':'PSA Airlines',
    'OO':'SkyWest Airlines',
    'VX':'Virgin America',
    'WN':'Southwest Airlines',
    'YV':'Mesa Airline',
    'YX':'Republic Airways',
    'AA':'American Airlines',
    'DL':'Delta Airlines'
}, inplace=True)
```

[110]:
```
flight_data.OP_CARRIER.nunique()
```

[110]:
```
18
```

[111]:
```
flight_data.OP_CARRIER.value_counts()
```

[111]:
```
OP_CARRIER
Southwest Airlines    1326376
Delta Airlines         938464
American Airlines      892021
SkyWest Airlines       758717
United Airlines        609226
Republic Airways       303927
JetBlue Airways        293075
Envoy Air              283788
PSA Airlines           264929
Alaska Airlines        240352
Endeavor Air           231211
Mesa Airline           208382
ExpressJet             196072
Spirit Airlines        171359
Frontier Airlines      116058
Allegiant Air           94982
```

```
Hawaiian Airlines       83161
Virgin America          17012
Name: count, dtype: int64
```

[112]:
```
flight_data.DEST.value_counts().iloc[:20]
```

[112]:
```
DEST
ATL    384813
ORD    322119
DFW    271096
DEN    232583
CLT    225450
LAX    217452
SFO    171824
PHX    171433
IAH    170255
LGA    162095
LAS    158721
MSP    156532
DTW    154229
BOS    142239
SEA    137829
EWR    137238
MCO    135272
DCA    127517
JFK    122315
PHL    112410
Name: count, dtype: int64
```

[113]:
```
top_cities = flight_data.DEST.value_counts().iloc[0:1].rename_axis('DEST').
 ↪reset_index(name='TOTAL_FLIGHTS')
top_cities.head()
```

[113]:
```
  DEST  TOTAL_FLIGHTS
0  ATL         384813
```

[114]:
```
top_cities.DEST.unique()
city_list = top_cities['DEST'].tolist()
```

[115]:
```
boolean_check_dest =  flight_data.DEST.isin(city_list)
flight_data =  flight_data[boolean_check_dest]
flight_data.head()
```

[115]:
```
        FL_DATE       OP_CARRIER ORIGIN DEST  DEP_DELAY  TAXI_OUT  TAXI_IN  \
13   2018-01-01  United Airlines    EWR  ATL       11.0      11.0      5.0
241  2018-01-01  United Airlines    EWR  ATL       20.0      13.0      9.0
349  2018-01-01  United Airlines    EWR  ATL        0.0      14.0      5.0
```

```
517 2018-01-01  United Airlines    IAH  ATL      201.0      12.0       7.0
686 2018-01-01  United Airlines    EWR  ATL        9.0      26.0       5.0

     ARR_DELAY  DIVERTED  CRS_ELAPSED_TIME  ACTUAL_ELAPSED_TIME  AIR_TIME  \
13        -3.0       0.0             154.0                140.0     124.0
241       12.0       0.0             154.0                146.0     124.0
349      -17.0       0.0             154.0                137.0     118.0
517      184.0       0.0             121.0                104.0      85.0
686        5.0       0.0             154.0                150.0     119.0

     DISTANCE  CARRIER_DELAY  WEATHER_DELAY  NAS_DELAY  SECURITY_DELAY  \
13      746.0            0.0            0.0        0.0             0.0
241     746.0            0.0            0.0        0.0             0.0
349     746.0            0.0            0.0        0.0             0.0
517     689.0            0.0            0.0      132.0             0.0
686     746.0            0.0            0.0        0.0             0.0

     LATE_AIRCRAFT_DELAY  WHEELS_OFF_elapse  WHEELS_ON_elapse
13                   0.0               11.0               5.0
241                  0.0               13.0               9.0
349                  0.0               14.0               5.0
517                 52.0               12.0               7.0
686                  0.0               26.0               5.0
```

[116]:
```
flight_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 384813 entries, 13 to 7213438
Data columns (total 20 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   FL_DATE              384813 non-null  datetime64[ns]
 1   OP_CARRIER           384813 non-null  object
 2   ORIGIN               384813 non-null  object
 3   DEST                 384813 non-null  object
 4   DEP_DELAY            384813 non-null  float64
 5   TAXI_OUT             384813 non-null  float64
 6   TAXI_IN              384813 non-null  float64
 7   ARR_DELAY            384813 non-null  float64
 8   DIVERTED             384813 non-null  float64
 9   CRS_ELAPSED_TIME     384813 non-null  float64
 10  ACTUAL_ELAPSED_TIME  384813 non-null  float64
 11  AIR_TIME             384813 non-null  float64
 12  DISTANCE             384813 non-null  float64
 13  CARRIER_DELAY        384813 non-null  float64
 14  WEATHER_DELAY        384813 non-null  float64
 15  NAS_DELAY            384813 non-null  float64
```

```
 16  SECURITY_DELAY       384813 non-null  float64
 17  LATE_AIRCRAFT_DELAY  384813 non-null  float64
 18  WHEELS_OFF_elapse    384813 non-null  float64
 19  WHEELS_ON_elapse     384813 non-null  float64
dtypes: datetime64[ns](1), float64(16), object(3)
memory usage: 61.7+ MB
```

[117]:
```
import calendar
flight_date=pd.DatetimeIndex(flight_data['FL_DATE'])
flight_data['DAY'] = flight_date.day
flight_data['MONTH'] = flight_date.month
flight_data['MONTH_AB'] = flight_data['MONTH'].apply(lambda x: calendar.
 ↪month_abbr[x])
flight_data['DAYOFWEEK'] = flight_date.dayofweek
flight_data['DAYNAME'] = flight_data['DAYOFWEEK'].apply(lambda x: calendar.
 ↪day_name[x])
daytype = []
for value in flight_data['DAYOFWEEK']:
    if value in (0,1,2,3,4):
        daytype.append(1)
    else:
        daytype.append(0)
flight_data['WEEKDAY'] = daytype
flight_data.head(2)
```

[117]:

	FL_DATE	OP_CARRIER	ORIGIN	DEST	DEP_DELAY	TAXI_OUT	TAXI_IN	\
13	2018-01-01	United Airlines	EWR	ATL	11.0	11.0	5.0	
241	2018-01-01	United Airlines	EWR	ATL	20.0	13.0	9.0	

	ARR_DELAY	DIVERTED	CRS_ELAPSED_TIME	ACTUAL_ELAPSED_TIME	AIR_TIME	\
13	-3.0	0.0	154.0	140.0	124.0	
241	12.0	0.0	154.0	146.0	124.0	

	DISTANCE	CARRIER_DELAY	WEATHER_DELAY	NAS_DELAY	SECURITY_DELAY	\
13	746.0	0.0	0.0	0.0	0.0	
241	746.0	0.0	0.0	0.0	0.0	

	LATE_AIRCRAFT_DELAY	WHEELS_OFF_elapse	WHEELS_ON_elapse	DAY	MONTH	\
13	0.0	11.0	5.0	1	1	
241	0.0	13.0	9.0	1	1	

	MONTH_AB	DAYOFWEEK	DAYNAME	WEEKDAY
13	Jan	0	Monday	1
241	Jan	0	Monday	1

[118]:
```
airports_data = pd.read_csv('airports.csv')
airports_data.head(10)
```

[118]:

	IATA_CODE	AIRPORT	CITY
0	AZA	Phoenix-Mesa Gateway Airport	NaN
1	BKG	Branson Airport	NaN
2	ABE	Lehigh Valley International Airport	Allentown
3	ABI	Abilene Regional Airport	Abilene
4	ABQ	Albuquerque International Sunport	Albuquerque
5	ABR	Aberdeen Regional Airport	Aberdeen
6	ABY	Southwest Georgia Regional Airport	Albany
7	ACK	Nantucket Memorial Airport	Nantucket
8	ACT	Waco Regional Airport	Waco
9	ACV	Arcata Airport	Arcata/Eureka

[119]:
```
airport_IATA_CODE = list(airports_data['IATA_CODE'])
```

[120]:
```
flight_data.ORIGIN.unique()
```

[120]:
```
array(['EWR', 'IAH', 'SFO', 'ORD', 'SEA', 'FSD', 'DSM', 'ILM', 'JAN',
       'OAJ', 'TLH', 'CHS', 'CID', 'BMI', 'ABY', 'DHN', 'MDT', 'FSM',
       'HSV', 'TYS', 'BQK', 'CHA', 'GSP', 'AGS', 'MOB', 'MGM', 'TRI',
       'GSO', 'AVL', 'CSG', 'VLD', 'PIA', 'LFT', 'GNV', 'FAY', 'BTV',
       'AEX', 'FAR', 'EWN', 'CAE', 'BOS', 'MYR', 'BTR', 'TUL', 'ECP',
       'LEX', 'ELM', 'VPS', 'PHF', 'RST', 'CRW', 'SDF', 'XNA', 'LNK',
       'HPN', 'BHM', 'GRK', 'EYW', 'SGF', 'GPT', 'ABE', 'MLI', 'EVV',
       'DEN', 'MCO', 'AUS', 'MIA', 'LGA', 'SLC', 'BWI', 'DTW', 'FLL',
       'LAS', 'PHL', 'DFW', 'CLE', 'LAX', 'MSY', 'MSP', 'TPA', 'CLT',
       'FWA', 'SHV', 'ASE', 'SBN', 'MLU', 'GTR', 'ROA', 'CMH', 'DAL',
       'DCA', 'HOU', 'IAD', 'IND', 'JAX', 'MCI', 'MDW', 'MKE', 'OAK',
       'PBI', 'PHX', 'PIT', 'RDU', 'RIC', 'RSW', 'SAN', 'SAT', 'STL',
       'JFK', 'STT', 'DAB', 'SJU', 'OMA', 'CAK', 'STX', 'PDX', 'BDL',
       'HNL', 'ELP', 'CVG', 'GRR', 'FNT', 'MEM', 'SJC', 'BNA', 'ORF',
       'OKC', 'BUF', 'ABQ', 'BZN', 'MLB', 'PNS', 'DAY', 'TUS', 'PWM',
       'PVD', 'ROC', 'LIT', 'ATW', 'SRQ', 'HDN', 'MTJ', 'CHO', 'JAC',
       'SAV', 'SNA', 'RNO', 'EGE', 'MSN', 'SYR', 'COS', 'MHT', 'SMF',
       'ALB', 'ICT', 'AVP', 'TTN', 'GRB', 'GJT', 'ISP', 'ACY', 'ANC',
       'FCA', 'RAP', 'MSO', 'TVC', 'PSP'], dtype=object)
```

[121]:
```
flight_ORIGIN = flight_data.ORIGIN.unique().tolist()
flight_DEST = flight_data.DEST.unique().tolist()

print("Type:")
print(type(flight_ORIGIN))
print(type(flight_DEST))
print()
print("Length:")
print("Origin: "+str(len(flight_ORIGIN)))
print("Destination: "+str(len(flight_DEST)))
```

```
Type:
<class 'list'>
<class 'list'>

Length:
Origin: 167
Destination: 1
```

[122]:
```
Origin_code = [item for item in flight_ORIGIN if item not in airport_IATA_CODE]
print("IATA Code (Origin) that is not found from the airport data:")
print(Origin_code)
print()
print("There are "+str(len(Origin_code)))
```

```
IATA Code (Origin) that is not found from the airport data:
[]

There are 0
```

[123]:
```
dest_code = [item for item in flight_DEST if item not in airport_IATA_CODE]
print("IATA Code (Destination) that is not found from airport data:")
print(dest_code)
print()
print("There are "+str(len(dest_code)))
```

```
IATA Code (Destination) that is not found from airport data:
[]

There are 0
```

[124]:
```
airports_dict = pd.Series(airports_data.AIRPORT.values, index=airports_data.
  ↪IATA_CODE).to_dict()
print(type(airports_dict))
```

```
<class 'dict'>
```

[125]:
```
flight_data['ORIGIN'].replace(airports_dict, inplace=True)
flight_data['DEST'].replace(airports_dict, inplace=True)
flight_data.head()
```

[125]:

	FL_DATE	OP_CARRIER	ORIGIN	\
13	2018-01-01	United Airlines	Newark Liberty International Airport	
241	2018-01-01	United Airlines	Newark Liberty International Airport	
349	2018-01-01	United Airlines	Newark Liberty International Airport	
517	2018-01-01	United Airlines	George Bush Intercontinental Airport	
686	2018-01-01	United Airlines	Newark Liberty International Airport	

	DEST	DEP_DELAY	TAXI_OUT	\

```
13   Hartsfield-Jackson Atlanta International Airport       11.0      11.0
241  Hartsfield-Jackson Atlanta International Airport       20.0      13.0
349  Hartsfield-Jackson Atlanta International Airport        0.0      14.0
517  Hartsfield-Jackson Atlanta International Airport      201.0      12.0
686  Hartsfield-Jackson Atlanta International Airport        9.0      26.0

     TAXI_IN  ARR_DELAY  DIVERTED  CRS_ELAPSED_TIME  ACTUAL_ELAPSED_TIME  \
13       5.0       -3.0       0.0             154.0                140.0
241      9.0       12.0       0.0             154.0                146.0
349      5.0      -17.0       0.0             154.0                137.0
517      7.0      184.0       0.0             121.0                104.0
686      5.0        5.0       0.0             154.0                150.0

     AIR_TIME  DISTANCE  CARRIER_DELAY  WEATHER_DELAY  NAS_DELAY  \
13      124.0     746.0            0.0            0.0        0.0
241     124.0     746.0            0.0            0.0        0.0
349     118.0     746.0            0.0            0.0        0.0
517      85.0     689.0            0.0            0.0      132.0
686     119.0     746.0            0.0            0.0        0.0

     SECURITY_DELAY  LATE_AIRCRAFT_DELAY  WHEELS_OFF_elapse  WHEELS_ON_elapse  \
13              0.0                  0.0               11.0               5.0
241             0.0                  0.0               13.0               9.0
349             0.0                  0.0               14.0               5.0
517             0.0                 52.0               12.0               7.0
686             0.0                  0.0               26.0               5.0

     DAY  MONTH MONTH_AB  DAYOFWEEK DAYNAME  WEEKDAY
13     1      1      Jan          0  Monday        1
241    1      1      Jan          0  Monday        1
349    1      1      Jan          0  Monday        1
517    1      1      Jan          0  Monday        1
686    1      1      Jan          0  Monday        1
```

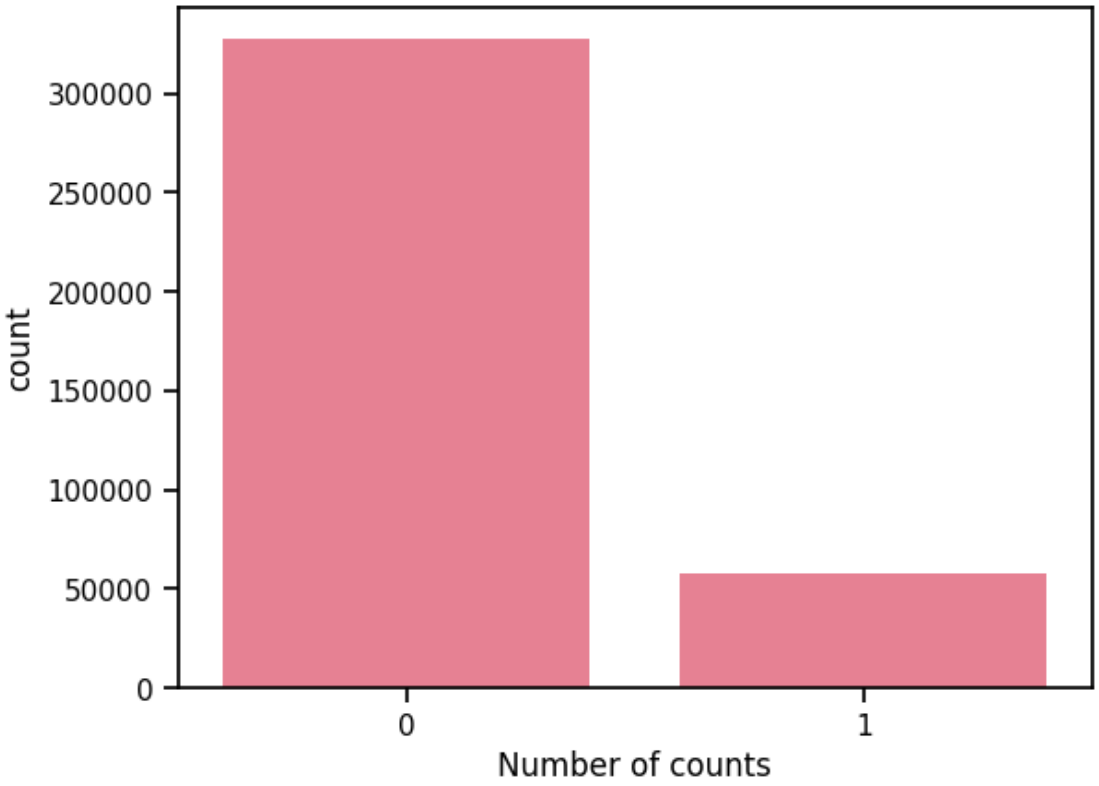
[126]:
```
flight_status = []

for value in flight_data['ARR_DELAY']:
    if value <= 15:
        flight_status.append(0)
    else:
        flight_status.append(1)
flight_data['FLIGHT_STATUS'] = flight_status
```

EDA

[127]:
```
flight_category = flight_data.select_dtypes(include=['object','category'])
flight_numerical = flight_data.select_dtypes(exclude=['object'])
```

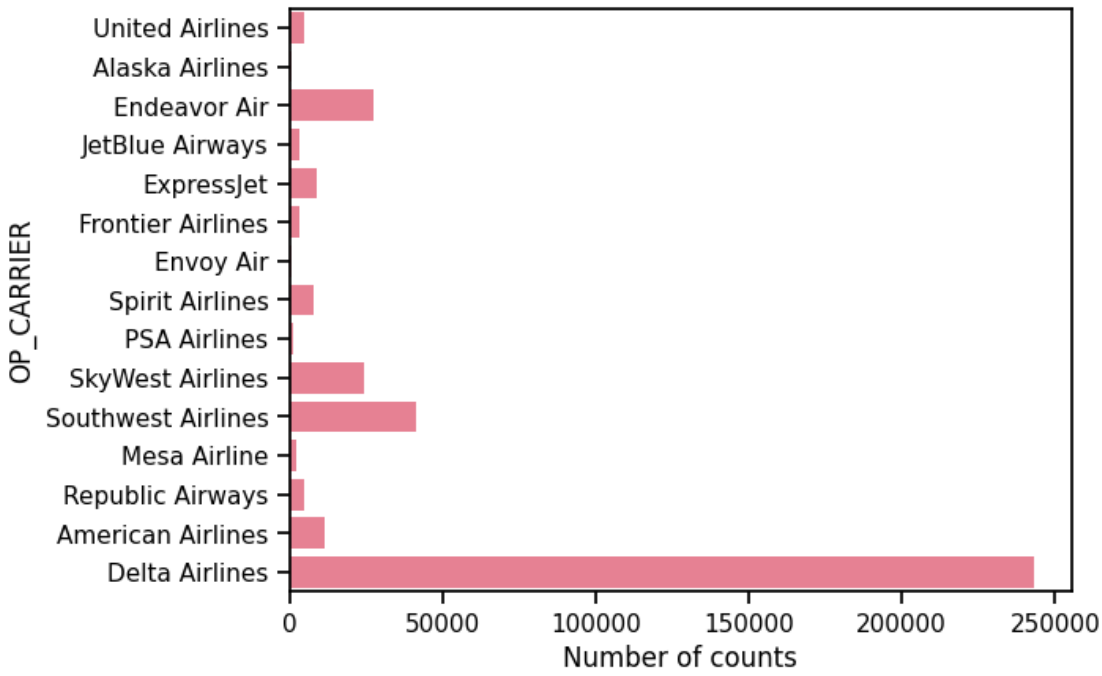
[128]:
```
plot=sns.countplot(x="FLIGHT_STATUS",data=flight_data)
plot.set(xlabel="Number of counts")
```

[128]: [Text(0.5, 0, 'Number of counts')]



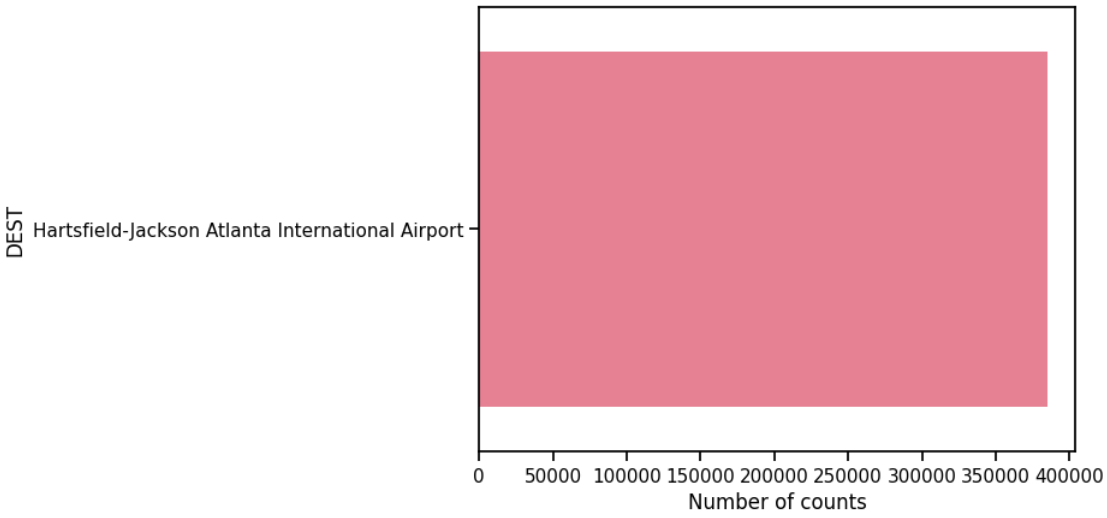
[129]:
```
plot=sns.countplot(y="OP_CARRIER",data=flight_category)
plot.set(xlabel="Number of counts")
```

[129]: [Text(0.5, 0, 'Number of counts')]

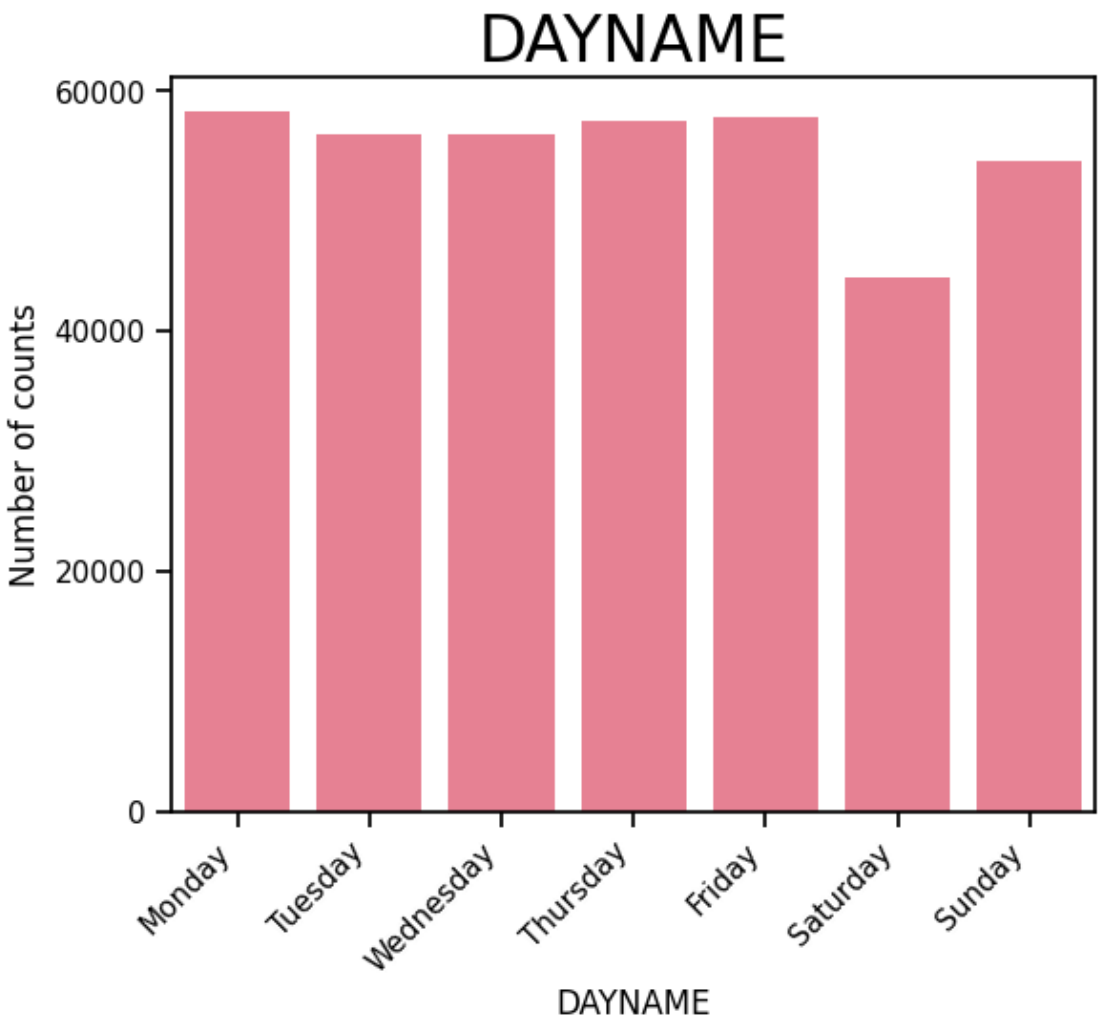


[130]:
```
sns.set_context("notebook")
plot=sns.countplot(y="DEST",data=flight_category)
plot.set(xlabel="Number of counts")
```

[130]: [Text(0.5, 0, 'Number of counts')]



[131]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Filter out specific columns to plot, excluding "MONTH_AB" explicitly
category = [i for i in flight_category if i not in ["ORIGIN", "DEST",␣
 ↪"OP_CARRIER", "FLIGHT_STATUS", "MONTH", "MONTH_AB"]]

# Set up the figure with a grid of subplots
num_features = len(category)
ncols = 2
nrows = (num_features + ncols - 1) // ncols  # Calculate rows needed for 2␣
 ↪columns

fig, axes = plt.subplots(nrows=nrows, ncols=ncols, figsize=(15, nrows * 5))
fig.subplots_adjust(hspace=0.4, wspace=0.4)  # Adjust space between plots

sns.set_palette("husl")
sns.set_context("poster")

# Flatten axes array to iterate easily
axes = axes.flatten()

for i, feature in enumerate(category):
    ax = axes[i]
    sns.countplot(x=feature, data=flight_category, ax=ax)
    ax.set_ylabel("Number of counts")
    ax.set_title(feature)
    ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha="right")

# Remove any unused axes
for j in range(len(category), len(axes)):
    fig.delaxes(axes[j])

plt.show()
```



[132]:
```
Carrier = pd.
  ↪get_dummies(flight_data['OP_CARRIER'],prefix="OP_CARRIER",drop_first=False)
month= pd.get_dummies(flight_data['MONTH_AB'],prefix="MONTH",drop_first=False)
day= pd.get_dummies(flight_data['DAYNAME'],prefix="DAY",drop_first=False)
flight_data_new=pd.concat([flight_data,Carrier,month,day],axis=1)
```

[133]:
```
flight_data_new.head(2)
```

[133]:

	FL_DATE	OP_CARRIER	ORIGIN	DEST	DEP_DELAY	TAXI_OUT
13	2018-01-01	United Airlines	Newark Liberty International Airport	Hartsfield-Jackson Atlanta International Airport	11.0	11.0
241	2018-01-01	United Airlines	Newark Liberty International Airport	Hartsfield-Jackson Atlanta International Airport	20.0	13.0

```
     TAXI_IN  ARR_DELAY  DIVERTED  CRS_ELAPSED_TIME  ACTUAL_ELAPSED_TIME  \
13       5.0       -3.0       0.0             154.0                140.0
241      9.0       12.0       0.0             154.0                146.0

     AIR_TIME  DISTANCE  CARRIER_DELAY  WEATHER_DELAY  NAS_DELAY  \
13      124.0     746.0            0.0            0.0        0.0
241     124.0     746.0            0.0            0.0        0.0

     SECURITY_DELAY  LATE_AIRCRAFT_DELAY  WHEELS_OFF_elapse  WHEELS_ON_elapse  \
13              0.0                  0.0               11.0               5.0
241             0.0                  0.0               13.0               9.0

     DAY  MONTH MONTH_AB  DAYOFWEEK DAYNAME  WEEKDAY  FLIGHT_STATUS  \
13     1      1      Jan          0  Monday        1              0
241    1      1      Jan          0  Monday        1              0

     OP_CARRIER_Alaska Airlines  OP_CARRIER_American Airlines  \
13                        False                         False
241                       False                         False

     OP_CARRIER_Delta Airlines  OP_CARRIER_Endeavor Air  OP_CARRIER_Envoy Air  \
13                       False                   False                 False
241                      False                   False                 False

     OP_CARRIER_ExpressJet  OP_CARRIER_Frontier Airlines  \
13                   False                         False
241                  False                         False

     OP_CARRIER_JetBlue Airways  OP_CARRIER_Mesa Airline  \
13                        False                   False
241                       False                   False

     OP_CARRIER_PSA Airlines  OP_CARRIER_Republic Airways  \
13                     False                        False
241                    False                        False

     OP_CARRIER_SkyWest Airlines  OP_CARRIER_Southwest Airlines  \
13                         False                          False
241                        False                          False

     OP_CARRIER_Spirit Airlines  OP_CARRIER_United Airlines  MONTH_Apr  \
13                        False                        True      False
241                       False                        True      False

     MONTH_Aug  MONTH_Dec  MONTH_Feb  MONTH_Jan  MONTH_Jul  MONTH_Jun  \
13       False      False      False       True      False      False
```

```
241      False      False      False       True      False      False

     MONTH_Mar  MONTH_May  MONTH_Nov  MONTH_Oct  MONTH_Sep  DAY_Friday  \
13       False      False      False      False      False       False
241      False      False      False      False      False       False

     DAY_Monday  DAY_Saturday  DAY_Sunday  DAY_Thursday  DAY_Tuesday  \
13         True         False       False         False        False
241        True         False       False         False        False

     DAY_Wednesday
13           False
241          False
```

[134]:
```
flight_data_new.drop(['FL_DATE',
                      'OP_CARRIER',
                      'ORIGIN',
                      'DEST',
                      'DAYOFWEEK',
                      'MONTH',
                      'MONTH_AB',
                      'DAY',
                      'DAYNAME',
                      'WEEKDAY',
                      'ARR_DELAY',
                      'CARRIER_DELAY',
                      'WEATHER_DELAY',
                      'NAS_DELAY',
                      'SECURITY_DELAY',
                      'LATE_AIRCRAFT_DELAY'
                  ],
                  axis = 1, inplace = True)
```

[135]:
```
flight_data_new.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 384813 entries, 13 to 7213438
Data columns (total 45 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   DEP_DELAY                       384813 non-null  float64
 1   TAXI_OUT                        384813 non-null  float64
 2   TAXI_IN                         384813 non-null  float64
 3   DIVERTED                        384813 non-null  float64
 4   CRS_ELAPSED_TIME                384813 non-null  float64
 5   ACTUAL_ELAPSED_TIME             384813 non-null  float64
 6   AIR_TIME                        384813 non-null  float64
```

```
 7   DISTANCE                         384813 non-null  float64
 8   WHEELS_OFF_elapse                384813 non-null  float64
 9   WHEELS_ON_elapse                 384813 non-null  float64
 10  FLIGHT_STATUS                    384813 non-null  int64
 11  OP_CARRIER_Alaska Airlines       384813 non-null  bool
 12  OP_CARRIER_American Airlines     384813 non-null  bool
 13  OP_CARRIER_Delta Airlines        384813 non-null  bool
 14  OP_CARRIER_Endeavor Air          384813 non-null  bool
 15  OP_CARRIER_Envoy Air             384813 non-null  bool
 16  OP_CARRIER_ExpressJet            384813 non-null  bool
 17  OP_CARRIER_Frontier Airlines     384813 non-null  bool
 18  OP_CARRIER_JetBlue Airways       384813 non-null  bool
 19  OP_CARRIER_Mesa Airline          384813 non-null  bool
 20  OP_CARRIER_PSA Airlines          384813 non-null  bool
 21  OP_CARRIER_Republic Airways      384813 non-null  bool
 22  OP_CARRIER_SkyWest Airlines      384813 non-null  bool
 23  OP_CARRIER_Southwest Airlines    384813 non-null  bool
 24  OP_CARRIER_Spirit Airlines       384813 non-null  bool
 25  OP_CARRIER_United Airlines       384813 non-null  bool
 26  MONTH_Apr                        384813 non-null  bool
 27  MONTH_Aug                        384813 non-null  bool
 28  MONTH_Dec                        384813 non-null  bool
 29  MONTH_Feb                        384813 non-null  bool
 30  MONTH_Jan                        384813 non-null  bool
 31  MONTH_Jul                        384813 non-null  bool
 32  MONTH_Jun                        384813 non-null  bool
 33  MONTH_Mar                        384813 non-null  bool
 34  MONTH_May                        384813 non-null  bool
 35  MONTH_Nov                        384813 non-null  bool
 36  MONTH_Oct                        384813 non-null  bool
 37  MONTH_Sep                        384813 non-null  bool
 38  DAY_Friday                       384813 non-null  bool
 39  DAY_Monday                       384813 non-null  bool
 40  DAY_Saturday                     384813 non-null  bool
 41  DAY_Sunday                       384813 non-null  bool
 42  DAY_Thursday                     384813 non-null  bool
 43  DAY_Tuesday                      384813 non-null  bool
 44  DAY_Wednesday                    384813 non-null  bool
dtypes: bool(34), float64(10), int64(1)
memory usage: 47.7 MB
```

[136]:
```
!pip install feature_engine
```

```
Requirement already satisfied: feature_engine in /usr/local/lib/python3.10/dist-
packages (1.8.2)
Requirement already satisfied: numpy>=1.18.2 in /usr/local/lib/python3.10/dist-
packages (from feature_engine) (1.26.4)
Requirement already satisfied: pandas>=2.2.0 in /usr/local/lib/python3.10/dist-
```

```
packages (from feature_engine) (2.2.2)
Requirement already satisfied: scikit-learn>=1.4.0 in
/usr/local/lib/python3.10/dist-packages (from feature_engine) (1.5.2)
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.10/dist-
packages (from feature_engine) (1.13.1)
Requirement already satisfied: statsmodels>=0.11.1 in
/usr/local/lib/python3.10/dist-packages (from feature_engine) (0.14.4)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas>=2.2.0->feature_engine)
(2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
packages (from pandas>=2.2.0->feature_engine) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-
packages (from pandas>=2.2.0->feature_engine) (2024.2)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-
packages (from scikit-learn>=1.4.0->feature_engine) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-
learn>=1.4.0->feature_engine) (3.5.0)
Requirement already satisfied: patsy>=0.5.6 in /usr/local/lib/python3.10/dist-
packages (from statsmodels>=0.11.1->feature_engine) (1.0.1)
Requirement already satisfied: packaging>=21.3 in
/usr/local/lib/python3.10/dist-packages (from
statsmodels>=0.11.1->feature_engine) (24.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-
packages (from python-dateutil>=2.8.2->pandas>=2.2.0->feature_engine) (1.16.0)
```

[137]:
```
from sklearn.model_selection import train_test_split
from feature_engine.selection import DropConstantFeatures
```

[138]:
```
X_train, X_test, y_train, y_test = train_test_split(
    flight_data_new.drop(labels=['FLIGHT_STATUS'], axis=1),
    flight_data_new['FLIGHT_STATUS'],
    test_size=0.25,
    random_state=100)

X_train.shape, X_test.shape
```

[138]:
```
((288609, 44), (96204, 44))
```

[139]:
```
sel = DropConstantFeatures(tol=1, variables=None, missing_values='raise')
sel.fit(X_train)
```

[139]:
```
DropConstantFeatures()
```

[140]:
```
len(sel.features_to_drop_)
```

[140]:
```
1
```

[141]:
```
sel.features_to_drop_
```

[141]:
```
['DIVERTED']
```

[142]:
```
flight_data_new.drop(['DIVERTED'], axis = 1, inplace = True)
```

[143]:
```
X_train = sel.transform(X_train)
X_test = sel.transform(X_test)

X_train.shape, X_test.shape
```

[143]:
```
((288609, 43), (96204, 43))
```

[144]:
```
corr_mat = X_train.corr()
corr_mat = corr_mat.abs().unstack()
corr_mat = corr_mat.sort_values(ascending=False)
corr_mat = corr_mat[corr_mat >= 0.8]
corr_mat = corr_mat[corr_mat < 1]
corr_mat = pd.DataFrame(corr_mat).reset_index()
corr_mat.columns = ['feature1', 'feature2', 'corr']
```

[145]:
```
corr_mat
```

[145]:

	feature1	feature2	corr
0	AIR_TIME	CRS_ELAPSED_TIME	0.983998
1	CRS_ELAPSED_TIME	AIR_TIME	0.983998
2	DISTANCE	AIR_TIME	0.982972
3	AIR_TIME	DISTANCE	0.982972
4	DISTANCE	CRS_ELAPSED_TIME	0.981438
5	CRS_ELAPSED_TIME	DISTANCE	0.981438
6	AIR_TIME	ACTUAL_ELAPSED_TIME	0.978477
7	ACTUAL_ELAPSED_TIME	AIR_TIME	0.978477
8	CRS_ELAPSED_TIME	ACTUAL_ELAPSED_TIME	0.968028
9	ACTUAL_ELAPSED_TIME	CRS_ELAPSED_TIME	0.968028
10	DISTANCE	ACTUAL_ELAPSED_TIME	0.957654
11	ACTUAL_ELAPSED_TIME	DISTANCE	0.957654

[146]:
```
corr_mat.feature1.unique()
```

[146]:
```
array(['AIR_TIME', 'CRS_ELAPSED_TIME', 'DISTANCE', 'ACTUAL_ELAPSED_TIME'],
      dtype=object)
```

[147]:
```
grouped_feature_ls = []
correlated_groups = []
```

```
for feature in corr_mat.feature1.unique():

    if feature not in grouped_feature_ls:
        correlated_block = corr_mat[corr_mat.feature1 == feature]
        grouped_feature_ls = grouped_feature_ls + list(
            correlated_block.feature2.unique()) + [feature]
        correlated_groups.append(correlated_block)

print('found {} correlated groups'.format(len(correlated_groups)))
print('out of {} total features'.format(X_train.shape[1]))
```

```
found 1 correlated groups
out of 43 total features
```

[148]:
```
for group in correlated_groups:
    print(group)
    print()
```

```
   feature1             feature2      corr
0  AIR_TIME     CRS_ELAPSED_TIME  0.983998
3  AIR_TIME             DISTANCE  0.982972
6  AIR_TIME  ACTUAL_ELAPSED_TIME  0.978477
```

[149]:
```
group = correlated_groups[0]
group
```

[149]:
```
   feature1             feature2      corr
0  AIR_TIME     CRS_ELAPSED_TIME  0.983998
3  AIR_TIME             DISTANCE  0.982972
6  AIR_TIME  ACTUAL_ELAPSED_TIME  0.978477
```

[150]:
```
from sklearn.ensemble import RandomForestClassifier
features = list(group['feature2'].unique())+['CRS_ELAPSED_TIME']
rf = RandomForestClassifier(n_estimators=100, random_state=100, max_depth=4)
rf.fit(X_train[features].fillna(0), y_train)
```

[150]:
```
RandomForestClassifier(max_depth=4, random_state=100)
```

[151]:
```
importance = pd.concat(
    [pd.Series(features),
     pd.Series(rf.feature_importances_)], axis=1)
importance.columns = ['feature', 'importance']
importance.sort_values(by='importance', ascending=False)
```

[151]:
```
               feature  importance
2  ACTUAL_ELAPSED_TIME    0.502578
```

```
3     CRS_ELAPSED_TIME    0.246256
0     CRS_ELAPSED_TIME    0.156903
1             DISTANCE    0.094263
```

[152]:
```
flight_data_new.drop(['CRS_ELAPSED_TIME','AIR_TIME'], axis = 1, inplace = True)
```

[153]:
```
corr_mat = X_train.corr()
corr_mat = corr_mat.abs().unstack()
corr_mat = corr_mat.sort_values(ascending=False)
corr_mat = corr_mat[corr_mat <= -0.8]
corr_mat = corr_mat[corr_mat > -1]
corr_mat = pd.DataFrame(corr_mat).reset_index()
corr_mat.columns = ['feature1', 'feature2', 'corr']
```

[154]:
```
grouped_feature_ls = []
correlated_groups = []
for feature in corr_mat.feature1.unique():
    if feature not in grouped_feature_ls:
        correlated_block = corr_mat[corr_mat.feature1 == feature]
        grouped_feature_ls = grouped_feature_ls + list(
            correlated_block.feature2.unique()) + [feature]
        correlated_groups.append(correlated_block)
print('found {} correlated groups'.format(len(correlated_groups)))
print('out of {} total features'.format(X_train.shape[1]))
```

```
found 0 correlated groups
out of 43 total features
```

[155]:
```
X_train, X_test, y_train, y_test = train_test_split(
    flight_data_new.drop(labels=['FLIGHT_STATUS'], axis=1),
    flight_data_new['FLIGHT_STATUS'],
    test_size=0.25,
    random_state=100)

X_train.shape, X_test.shape
```

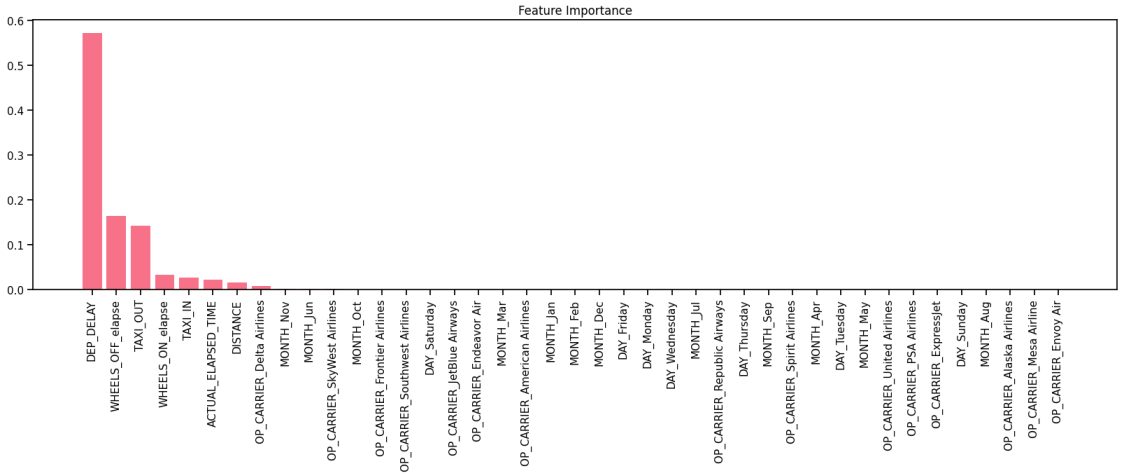
[155]:
```
((288609, 41), (96204, 41))
```

[156]:
```
rf = RandomForestClassifier(n_estimators=100, random_state=100, max_depth=5)
rf.fit(X_train, y_train)
rf.feature_importances_
```

[156]:
```
array([5.71747377e-01, 1.43078297e-01, 2.65832165e-02, 2.26953123e-02,
       1.57224396e-02, 1.64326828e-01, 3.41609818e-02, 1.25502144e-05,
       2.49100335e-04, 9.07634387e-03, 3.07907710e-04, 5.25082441e-06,
       3.50888948e-05, 7.64430380e-04, 3.56512218e-04, 1.06139520e-05,
       4.42739558e-05, 1.24468662e-04, 1.79931201e-03, 7.41044306e-04,
```

```
9.20104654e-05, 4.43660916e-05, 9.18431731e-05, 2.42056131e-05,
2.13943783e-04, 2.23361159e-04, 2.47074546e-04, 1.27206313e-04,
2.13591874e-03, 2.56883834e-04, 6.88833057e-05, 2.30826195e-03,
1.09648657e-03, 1.01059905e-04, 2.02686414e-04, 1.51035649e-04,
4.12357311e-04, 3.46575792e-05, 1.23885051e-04, 7.45662953e-05,
1.27956749e-04])
```

[157]:
```
importance = rf.feature_importances_
indices = np.argsort(importance)[::-1]
names = [X_train.columns[i] for i in indices]
sns.set_context("notebook")
plt.figure(figsize=(20, 5))
plt.title("Feature Importance")
plt.bar(range(X_train.shape[1]), importance[indices])
plt.xticks(range(X_train.shape[1]), names, rotation = 90)

plt.show()
```



[158]:
```
features=pd.DataFrame(names)
scale = pd.DataFrame(importance[indices])
keep_features = pd.concat([features, scale],axis=1)
keep_features
```

[158]:

	0	0
0	DEP_DELAY	0.571747
1	WHEELS_OFF_elapse	0.164327
2	TAXI_OUT	0.143078
3	WHEELS_ON_elapse	0.034161
4	TAXI_IN	0.026583
5	ACTUAL_ELAPSED_TIME	0.022695
6	DISTANCE	0.015722

```
7       OP_CARRIER_Delta Airlines  0.009076
8                     MONTH_Nov  0.002308
9                     MONTH_Jun  0.002136
10    OP_CARRIER_SkyWest Airlines  0.001799
11                    MONTH_Oct  0.001096
12   OP_CARRIER_Frontier Airlines  0.000764
13  OP_CARRIER_Southwest Airlines  0.000741
14                  DAY_Saturday  0.000412
15     OP_CARRIER_JetBlue Airways  0.000357
16        OP_CARRIER_Endeavor Air  0.000308
17                    MONTH_Mar  0.000257
18   OP_CARRIER_American Airlines  0.000249
19                    MONTH_Jan  0.000247
20                    MONTH_Feb  0.000223
21                    MONTH_Dec  0.000214
22                   DAY_Friday  0.000203
23                   DAY_Monday  0.000151
24                DAY_Wednesday  0.000128
25                    MONTH_Jul  0.000127
26    OP_CARRIER_Republic Airways  0.000124
27                  DAY_Thursday  0.000124
28                    MONTH_Sep  0.000101
29     OP_CARRIER_Spirit Airlines  0.000092
30                    MONTH_Apr  0.000092
31                  DAY_Tuesday  0.000075
32                    MONTH_May  0.000069
33     OP_CARRIER_United Airlines  0.000044
34        OP_CARRIER_PSA Airlines  0.000044
35          OP_CARRIER_ExpressJet  0.000035
36                    DAY_Sunday  0.000035
37                    MONTH_Aug  0.000024
38     OP_CARRIER_Alaska Airlines  0.000013
39        OP_CARRIER_Mesa Airline  0.000011
40           OP_CARRIER_Envoy Air  0.000005
```

[159]:
```
from sklearn.metrics import (
    precision_score,
    recall_score,
    f1_score,
    roc_auc_score,
    accuracy_score,
    confusion_matrix,
    classification_report
)
from sklearn.metrics import ConfusionMatrixDisplay
from sklearn.metrics import PrecisionRecallDisplay
```

[160]:
```
flight_data_new.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 384813 entries, 13 to 7213438
Data columns (total 42 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   DEP_DELAY                       384813 non-null  float64
 1   TAXI_OUT                        384813 non-null  float64
 2   TAXI_IN                         384813 non-null  float64
 3   ACTUAL_ELAPSED_TIME             384813 non-null  float64
 4   DISTANCE                        384813 non-null  float64
 5   WHEELS_OFF_elapse               384813 non-null  float64
 6   WHEELS_ON_elapse                384813 non-null  float64
 7   FLIGHT_STATUS                   384813 non-null  int64
 8   OP_CARRIER_Alaska Airlines      384813 non-null  bool
 9   OP_CARRIER_American Airlines    384813 non-null  bool
 10  OP_CARRIER_Delta Airlines       384813 non-null  bool
 11  OP_CARRIER_Endeavor Air         384813 non-null  bool
 12  OP_CARRIER_Envoy Air            384813 non-null  bool
 13  OP_CARRIER_ExpressJet           384813 non-null  bool
 14  OP_CARRIER_Frontier Airlines    384813 non-null  bool
 15  OP_CARRIER_JetBlue Airways      384813 non-null  bool
 16  OP_CARRIER_Mesa Airline         384813 non-null  bool
 17  OP_CARRIER_PSA Airlines         384813 non-null  bool
 18  OP_CARRIER_Republic Airways     384813 non-null  bool
 19  OP_CARRIER_SkyWest Airlines     384813 non-null  bool
 20  OP_CARRIER_Southwest Airlines   384813 non-null  bool
 21  OP_CARRIER_Spirit Airlines      384813 non-null  bool
 22  OP_CARRIER_United Airlines      384813 non-null  bool
 23  MONTH_Apr                       384813 non-null  bool
 24  MONTH_Aug                       384813 non-null  bool
 25  MONTH_Dec                       384813 non-null  bool
 26  MONTH_Feb                       384813 non-null  bool
 27  MONTH_Jan                       384813 non-null  bool
 28  MONTH_Jul                       384813 non-null  bool
 29  MONTH_Jun                       384813 non-null  bool
 30  MONTH_Mar                       384813 non-null  bool
 31  MONTH_May                       384813 non-null  bool
 32  MONTH_Nov                       384813 non-null  bool
 33  MONTH_Oct                       384813 non-null  bool
 34  MONTH_Sep                       384813 non-null  bool
 35  DAY_Friday                      384813 non-null  bool
 36  DAY_Monday                      384813 non-null  bool
 37  DAY_Saturday                    384813 non-null  bool
 38  DAY_Sunday                      384813 non-null  bool
 39  DAY_Thursday                    384813 non-null  bool
```

```
 40  DAY_Tuesday                    384813 non-null  bool
 41  DAY_Wednesday                  384813 non-null  bool
dtypes: bool(34), float64(7), int64(1)
memory usage: 38.9 MB
```

[161]:
```
flight_data_new["FLIGHT_STATUS"] = flight_data_new["FLIGHT_STATUS"].
 ↪astype('category')
flight_data_new.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 384813 entries, 13 to 7213438
Data columns (total 42 columns):
 #   Column                         Non-Null Count   Dtype
---  ------                         --------------   -----
 0   DEP_DELAY                      384813 non-null  float64
 1   TAXI_OUT                       384813 non-null  float64
 2   TAXI_IN                        384813 non-null  float64
 3   ACTUAL_ELAPSED_TIME            384813 non-null  float64
 4   DISTANCE                       384813 non-null  float64
 5   WHEELS_OFF_elapse              384813 non-null  float64
 6   WHEELS_ON_elapse               384813 non-null  float64
 7   FLIGHT_STATUS                  384813 non-null  category
 8   OP_CARRIER_Alaska Airlines     384813 non-null  bool
 9   OP_CARRIER_American Airlines   384813 non-null  bool
 10  OP_CARRIER_Delta Airlines      384813 non-null  bool
 11  OP_CARRIER_Endeavor Air        384813 non-null  bool
 12  OP_CARRIER_Envoy Air           384813 non-null  bool
 13  OP_CARRIER_ExpressJet          384813 non-null  bool
 14  OP_CARRIER_Frontier Airlines   384813 non-null  bool
 15  OP_CARRIER_JetBlue Airways     384813 non-null  bool
 16  OP_CARRIER_Mesa Airline        384813 non-null  bool
 17  OP_CARRIER_PSA Airlines        384813 non-null  bool
 18  OP_CARRIER_Republic Airways    384813 non-null  bool
 19  OP_CARRIER_SkyWest Airlines    384813 non-null  bool
 20  OP_CARRIER_Southwest Airlines  384813 non-null  bool
 21  OP_CARRIER_Spirit Airlines     384813 non-null  bool
 22  OP_CARRIER_United Airlines     384813 non-null  bool
 23  MONTH_Apr                      384813 non-null  bool
 24  MONTH_Aug                      384813 non-null  bool
 25  MONTH_Dec                      384813 non-null  bool
 26  MONTH_Feb                      384813 non-null  bool
 27  MONTH_Jan                      384813 non-null  bool
 28  MONTH_Jul                      384813 non-null  bool
 29  MONTH_Jun                      384813 non-null  bool
 30  MONTH_Mar                      384813 non-null  bool
 31  MONTH_May                      384813 non-null  bool
 32  MONTH_Nov                      384813 non-null  bool
 33  MONTH_Oct                      384813 non-null  bool
```

```
 34  MONTH_Sep                    384813 non-null  bool
 35  DAY_Friday                   384813 non-null  bool
 36  DAY_Monday                   384813 non-null  bool
 37  DAY_Saturday                 384813 non-null  bool
 38  DAY_Sunday                   384813 non-null  bool
 39  DAY_Thursday                 384813 non-null  bool
 40  DAY_Tuesday                  384813 non-null  bool
 41  DAY_Wednesday                384813 non-null  bool
dtypes: bool(34), category(1), float64(7)
memory usage: 36.3 MB
```

[162]:
```
flight_data_new.to_csv('flight_data_new.csv',index=False)
```

split data into 75:25 for training and testing

[163]:
```
X_train, X_test, y_train, y_test = train_test_split(
    flight_data_new.drop(labels=['FLIGHT_STATUS'], axis=1),
    flight_data_new['FLIGHT_STATUS'],
    test_size=0.2,
    random_state=100)

X_train.shape, X_test.shape
```

[163]:
```
((307850, 41), (76963, 41))
```

[164]:
```
def run_randomForests(X_train, X_test, y_train, y_test):
    rf = RandomForestClassifier(n_estimators=100, random_state=100, max_depth=4)
    rf.fit(X_train, y_train)

    print('Test set')
    pred = rf.predict_proba(X_test)
    print('Roc-auc Random Forests roc-auc: {}'.format(roc_auc_score(y_test,␣
 ↪pred[:,1])))
    print()
    print('Accuracy Random Forest test:', accuracy_score(y_test, rf.
 ↪predict(X_test)))
    print()
    print('Precision Random Forest test:', precision_score(y_test, rf.
 ↪predict(X_test),pos_label=1))
    print()
    print('Recall Random Forest test:', recall_score(y_test, rf.
 ↪predict(X_test),pos_label=1))
    print()
    print('F-measure Random Forest test:', f1_score(y_test, rf.
 ↪predict(X_test),pos_label=1))
    print()
    print('Summary Report:')
```

```
    print(classification_report(y_test, rf.predict(X_test)))
```

[165]:
```
run_randomForests(X_train, X_test, y_train, y_test)
```

```
Test set
Roc-auc Random Forests roc-auc: 0.967824946807227

Accuracy Random Forest test: 0.8826449072931148

Precision Random Forest test: 0.9463941380640185

Recall Random Forest test: 0.2162686172556623

F-measure Random Forest test: 0.35208034433285507

Summary Report:
              precision    recall  f1-score   support

           0       0.88      1.00      0.94     65616
           1       0.95      0.22      0.35     11347

    accuracy                           0.88     76963
   macro avg       0.91      0.61      0.64     76963
weighted avg       0.89      0.88      0.85     76963
```

[166]:
```
from sklearn.tree import DecisionTreeClassifier
```

[167]:
```
def run_DecisionTree(X_train, X_test, y_train, y_test):
    dt = DecisionTreeClassifier(random_state=100)
    dt = dt.fit(X_train,y_train)

    print('Test set')
    pred = dt.predict_proba(X_test)
    print('Roc-auc Decision Tree roc-auc: {}'.format(roc_auc_score(y_test,␣
 ↪pred[:,1])))
    print()
    print('Accuracy Decision Tree:', accuracy_score(y_test, dt.predict(X_test)))
    print()
    print('Precision Decision Tree:', precision_score(y_test, dt.
 ↪predict(X_test),pos_label=1))
    print()
    print('Recall Decision Tree:', recall_score(y_test, dt.
 ↪predict(X_test),pos_label=1))
    print()
    print('F-measure Decision Tree:', f1_score(y_test, dt.
 ↪predict(X_test),pos_label=1))
```

```
    print()
    print('Summary Report:')
    print(classification_report(y_test, dt.predict(X_test)))
```

[168]:
```
run_DecisionTree(X_train, X_test, y_train, y_test)
```

```
Test set
Roc-auc Decision Tree roc-auc: 0.911810166784978

Accuracy Decision Tree: 0.9553291841534244

Precision Decision Tree: 0.8474040235438812

Recall Decision Tree: 0.8500925354719309

F-measure Decision Tree: 0.8487461504619446

Summary Report:
              precision    recall  f1-score   support

           0       0.97      0.97      0.97     65616
           1       0.85      0.85      0.85     11347

    accuracy                           0.96     76963
   macro avg       0.91      0.91      0.91     76963
weighted avg       0.96      0.96      0.96     76963
```

[169]:
```
from collections import Counter
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import roc_auc_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler

from imblearn.datasets import fetch_datasets

from imblearn.under_sampling import (
    RandomUnderSampler,
    TomekLinks
)
```

[170]:
```
def run_all(X_train, X_test, y_train, y_test):

```

```
    # Decision Tree
    dt = DecisionTreeClassifier(random_state=100)
    dt.fit(X_train,y_train)

    print('Test set')
    pred = dt.predict_proba(X_test)
    print('Roc-auc Decision Tree roc-auc: {}'.format(roc_auc_score(y_test,␣
↪pred[:,1])))
    print()
    print('Accuracy Decision Tree:', accuracy_score(y_test, dt.predict(X_test)))
    print()
    print('Precision Decision Tree:', precision_score(y_test, dt.
↪predict(X_test),pos_label=1))
    print()
    print('Recall Decision Tree:', recall_score(y_test, dt.
↪predict(X_test),pos_label=1))
    print()
    print('F-measure Decision Tree:', f1_score(y_test, dt.
↪predict(X_test),pos_label=1))
    print()
    print('Summary Report:')
    print(classification_report(y_test, dt.predict(X_test)))

    rf = RandomForestClassifier(n_estimators=100, random_state=100, max_depth=4)
    rf.fit(X_train, y_train)

    print('Random Forest')
    pred = rf.predict_proba(X_test)
    print('Roc-auc Random Forests roc-auc: {}'.format(roc_auc_score(y_test,␣
↪pred[:,1])))
    print()
    print('Accuracy Random Forest test:', accuracy_score(y_test, rf.
↪predict(X_test)))
    print()
    print('Precision Random Forest test:', precision_score(y_test, rf.
↪predict(X_test),pos_label=1))
    print()
    print('Recall Random Forest test:', recall_score(y_test, rf.
↪predict(X_test),pos_label=1))
    print()
    print('F-measure Random Forest test:', f1_score(y_test, rf.
↪predict(X_test),pos_label=1))
    print()
    print('Summary Report:')
    print(classification_report(y_test, rf.predict(X_test)))
```

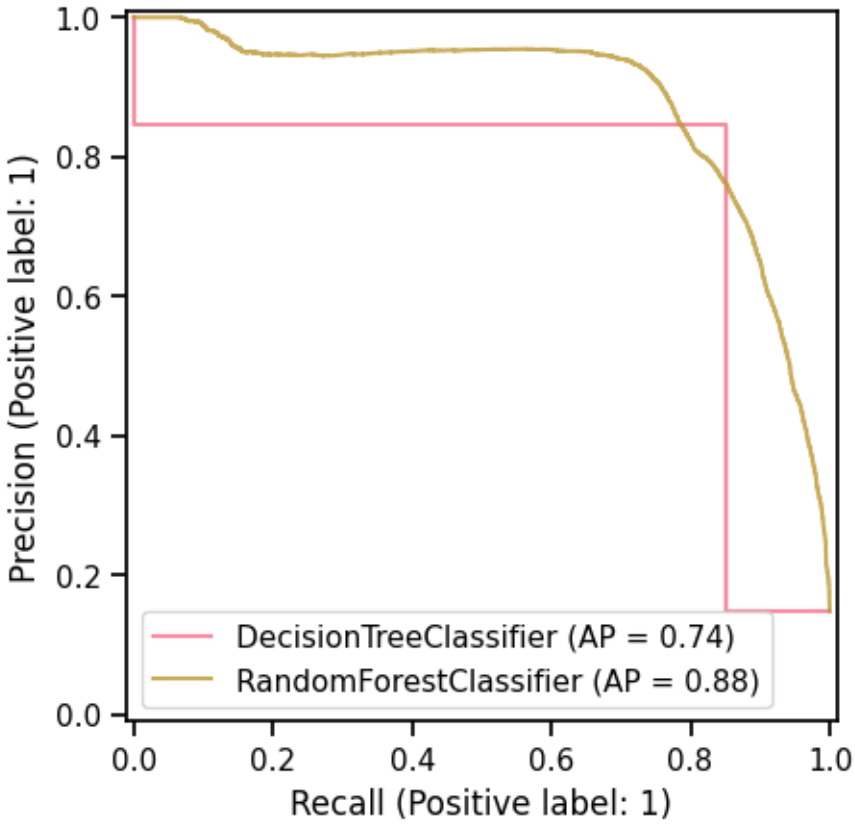
```
    print()
    print()
    print('Precision Recall Curve')
    ax = plt.gca()
    PrecisionRecallDisplay.from_estimator(dt, X_test, y_test, ax=ax, alpha=0.8)
    PrecisionRecallDisplay.from_estimator(rf, X_test, y_test, ax=ax, alpha=0.8)
    plt.show()
```

[171]:
```
run_all(X_train, X_test, y_train, y_test)
```

```
Test set
Roc-auc Decision Tree roc-auc: 0.911810166784978

Accuracy Decision Tree: 0.9553291841534244

Precision Decision Tree: 0.8474040235438812

Recall Decision Tree: 0.8500925354719309

F-measure Decision Tree: 0.8487461504619446

Summary Report:
              precision    recall  f1-score   support

           0       0.97      0.97      0.97     65616
           1       0.85      0.85      0.85     11347

    accuracy                           0.96     76963
   macro avg       0.91      0.91      0.91     76963
weighted avg       0.96      0.96      0.96     76963

Random Forest
Roc-auc Random Forests roc-auc: 0.967824946807227

Accuracy Random Forest test: 0.8826449072931148

Precision Random Forest test: 0.9463941380640185

Recall Random Forest test: 0.2162686172556623

F-measure Random Forest test: 0.35208034433285507

Summary Report:
              precision    recall  f1-score   support

           0       0.88      1.00      0.94     65616
           1       0.95      0.22      0.35     11347
```

```
              accuracy                           0.88     76963
             macro avg       0.91      0.61      0.64     76963
          weighted avg       0.89      0.88      0.85     76963
```

```
Precision Recall Curve
```



[172]:
```
import pickle
from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier(random_state=100)
dt = dt.fit(X_train, y_train)

filename = 'decision_tree_model.pkl'
pickle.dump(dt, open(filename, 'wb'))
```

[173]:
```
rf = RandomForestClassifier(n_estimators=100, random_state=100, max_depth=5)
rf.fit(X_train, y_train)

filename = 'random_forest_model.pkl'
```

```
pickle.dump(rf, open(filename, 'wb'))
```

[174]:
```
print(flight_data_new.columns)
```

```
Index(['DEP_DELAY', 'TAXI_OUT', 'TAXI_IN', 'ACTUAL_ELAPSED_TIME', 'DISTANCE',
       'WHEELS_OFF_elapse', 'WHEELS_ON_elapse', 'FLIGHT_STATUS',
       'OP_CARRIER_Alaska Airlines', 'OP_CARRIER_American Airlines',
       'OP_CARRIER_Delta Airlines', 'OP_CARRIER_Endeavor Air',
       'OP_CARRIER_Envoy Air', 'OP_CARRIER_ExpressJet',
       'OP_CARRIER_Frontier Airlines', 'OP_CARRIER_JetBlue Airways',
       'OP_CARRIER_Mesa Airline', 'OP_CARRIER_PSA Airlines',
       'OP_CARRIER_Republic Airways', 'OP_CARRIER_SkyWest Airlines',
       'OP_CARRIER_Southwest Airlines', 'OP_CARRIER_Spirit Airlines',
       'OP_CARRIER_United Airlines', 'MONTH_Apr', 'MONTH_Aug', 'MONTH_Dec',
       'MONTH_Feb', 'MONTH_Jan', 'MONTH_Jul', 'MONTH_Jun', 'MONTH_Mar',
       'MONTH_May', 'MONTH_Nov', 'MONTH_Oct', 'MONTH_Sep', 'DAY_Friday',
       'DAY_Monday', 'DAY_Saturday', 'DAY_Sunday', 'DAY_Thursday',
       'DAY_Tuesday', 'DAY_Wednesday'],
      dtype='object')
```