DATA COLLECTION AND PREPROCESSING

Team ID PNT2022TMID16204

TEAM MEMBERS:

TAMIL MANI P

SURIYA RAAJ P

SARAVANA KUMAR P B

SRIKANTH MU

```
!curl https://topcs.blob.core.windows.net/public/FlightData.csv -o
flightdata.csv
```

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time
Current
                                 Dload  Upload   Total   Spent    Left
Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:--
--:--:--     0curl: (6) Could not resolve host:
topcs.blob.core.windows.net
```

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage.
It includes your credentials.
# You might want to remove those credentials before you share the
notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='qbgeU05njYh_u7o7DjiZtO-jZaiGeNf8OWmacgANzHjR',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-
storage.appdomain.cloud')

bucket = 'flightdelay-donotdelete-pr-ti12fkh98hxjhh'
object_key = 'flightdata.csv'

body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
# add missing_iter method, so pandas accepts body as file-likeobject
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType(
__iter__, body )
```

```
df= pd.read_csv(body)
df.head()

   YEAR  QUARTER  MONTH  DAY_OF_MONTH  DAY_OF_WEEK UNIQUE_CARRIER
TAIL_NUM  \
0  2016        1      1             1            5             DL
N836DN
1  2016        1      1             1            5             DL
N964DN
2  2016        1      1             1            5             DL
```

```
N813DN
3  2016        1      1             1            5             DL
N587NW
4  2016        1      1             1            5             DL
N836DN

   FL_NUM ORIGIN_AIRPORT_ID ORIGIN ... CRS_ARR_TIME ARR_TIME
ARR_DELAY \
0    1399              10397    ATL  ...          2143   2102.0      -
41.0
1    1476              11433    DTW  ...          1435   1439.0
4.0
2    1597              10397    ATL  ...          1215   1142.0      -
33.0
3    1768              14747    SEA  ...          1335   1345.0
10.0
4    1823              14747    SEA  ...           607    615.0
8.0

   ARR_DEL15 CANCELLED DIVERTED CRS_ELAPSED_TIME
ACTUAL_ELAPSED_TIME \
0        0.0        0.0       0.0             338.0
295.0
1        0.0        0.0       0.0             110.0
115.0
2        0.0        0.0       0.0             335.0
300.0
3        0.0        0.0       0.0             196.0
205.0
4        0.0        0.0       0.0             247.0
259.0

   DISTANCE  Unnamed: 25
0    2182.0          NaN
1     528.0          NaN
2    2182.0          NaN
3    1399.0          NaN
4    1927.0          NaN

[5 rows x 26 columns]
```

```
df.shape
```

```
(11231, 26)
```

```
df.isnull().values.any()
```

```
True
```

```
df.isnull().sum()
```

```
YEAR                       0
QUARTER                    0
MONTH                      0
DAY_OF_MONTH               0
DAY_OF_WEEK                0
UNIQUE_CARRIER             0
TAIL_NUM                   0
FL_NUM                     0
ORIGIN_AIRPORT_ID          0
ORIGIN                     0
DEST_AIRPORT_ID            0
DEST                       0
CRS_DEP_TIME               0
DEP_TIME                 107
DEP_DELAY                107
DEP_DEL15                107
CRS_ARR_TIME               0
ARR_TIME                 115
ARR_DELAY                188
ARR_DEL15                188
CANCELLED                  0
DIVERTED                   0
CRS_ELAPSED_TIME           0
ACTUAL_ELAPSED_TIME      188
DISTANCE                   0
Unnamed: 25            11231
dtype: int64
```

```
df = df.drop('Unnamed: 25', axis=1)
df.isnull().sum()
```

```
YEAR                     0
QUARTER                  0
MONTH                    0
DAY_OF_MONTH             0
DAY_OF_WEEK              0
UNIQUE_CARRIER           0
TAIL_NUM                 0
FL_NUM                   0
ORIGIN_AIRPORT_ID        0
ORIGIN                   0
DEST_AIRPORT_ID          0
DEST                     0
CRS_DEP_TIME             0
DEP_TIME               107
DEP_DELAY              107
DEP_DEL15              107
CRS_ARR_TIME             0
ARR_TIME               115
```

```
ARR_DELAY              188
ARR_DEL15              188
CANCELLED                0
DIVERTED                 0
CRS_ELAPSED_TIME         0
ACTUAL_ELAPSED_TIME    188
DISTANCE                 0
dtype: int64
```

```
df = df[["MONTH", "DAY_OF_MONTH", "DAY_OF_WEEK", "ORIGIN", "DEST",
"CRS_DEP_TIME", "ARR_DEL15"]]
df.isnull().sum()
```

```
MONTH             0
DAY_OF_MONTH      0
DAY_OF_WEEK       0
ORIGIN            0
DEST              0
CRS_DEP_TIME      0
ARR_DEL15       188
dtype: int64
```

```
df[df.isnull().values.any(axis=1)].head()
```

```
     MONTH  DAY_OF_MONTH  DAY_OF_WEEK ORIGIN DEST  CRS_DEP_TIME
ARR_DEL15
177      1             9            6    MSP  SEA           701
NaN
179      1            10            7    MSP  DTW          1348
NaN
184      1            10            7    MSP  DTW           625
NaN
210      1            10            7    DTW  MSP          1200
NaN
478      1            22            5    SEA  JFK          2305
NaN
```

```
df = df.fillna({'ARR_DEL15': 1})
df.iloc[177:185]
```

```
     MONTH  DAY_OF_MONTH  DAY_OF_WEEK ORIGIN DEST  CRS_DEP_TIME
ARR_DEL15
177      1             9            6    MSP  SEA           701
1.0
178      1             9            6    DTW  JFK          1527
0.0
179      1            10            7    MSP  DTW          1348
1.0
180      1            10            7    DTW  MSP          1540
0.0
181      1            10            7    JFK  ATL          1325
```

```
0.0
182      1            10             7    JFK  ATL           610
0.0
183      1            10             7    JFK  SEA          1615
0.0
184      1            10             7    MSP  DTW           625
1.0
```

```
df.head()
```

```
   MONTH  DAY_OF_MONTH  DAY_OF_WEEK ORIGIN DEST  CRS_DEP_TIME
ARR_DEL15
0      1             1            5    ATL  SEA          1905
0.0
1      1             1            5    DTW  MSP          1345
0.0
2      1             1            5    ATL  SEA           940
0.0
3      1             1            5    SEA  MSP           819
0.0
4      1             1            5    SEA  DTW          2300
0.0
```

```
import math

for index, row in df.iterrows():
    df.loc[index, 'CRS_DEP_TIME'] = math.floor(row['CRS_DEP_TIME'] /
100)
df.head()
```

```
   MONTH DAY_OF_MONTH DAY_OF_WEEK ORIGIN DEST CRS_DEP_TIME
ARR_DEL15
0      1             1            5    ATL  SEA            19
0.0
1      1             1            5    DTW  MSP            13
0.0
2      1             1            5    ATL  SEA             9
0.0
3      1             1            5    SEA  MSP             8
0.0
4      1             1            5    SEA  DTW            23
0.0
```

```
df = pd.get_dummies(df, columns=['ORIGIN', 'DEST'])
df.head()
```

```
   MONTH DAY_OF_MONTH DAY_OF_WEEK CRS_DEP_TIME ARR_DEL15
ORIGIN_ATL \
0      1             1            5            19         0.0
1
1      1             1            5            13         0.0
```

```
0
2      1             1            5             9         0.0
1
3      1             1            5             8         0.0
0
4      1             1            5            23         0.0
0

   ORIGIN_DTW  ORIGIN_JFK  ORIGIN_MSP  ORIGIN_SEA  DEST_ATL  DEST_DTW
\
0           0           0           0           0         0         0

1           1           0           0           0         0         0

2           0           0           0           0         0         0

3           0           0           0           1         0         0

4           0           0           0           1         0         1


   DEST_JFK  DEST_MSP  DEST_SEA
0         0         0         1
1         0         1         0
2         0         0         1
3         0         1         0
4         0         0         0
```

```
from sklearn.model_selection import train_test_split
train_x, test_x, train_y, test_y =
train_test_split(df.drop('ARR_DEL15', axis=1), df['ARR_DEL15'],
test_size=0.2, random_state=42)
```

```
train_x.shape
```

```
(8984, 14)
```

```
test_x.shape
```

```
(2247, 14)
```

```
test_x.shape
```

```
(2247, 14)
```