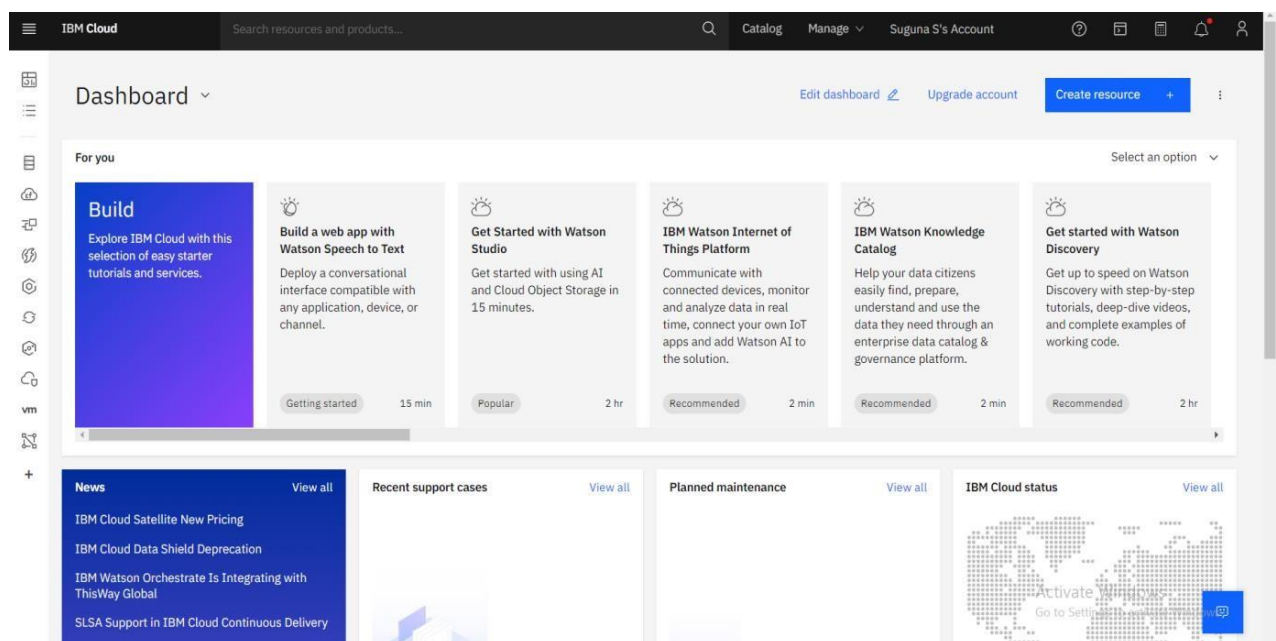| Date	17 November 22
Team ID	PNT2022TMID46686
Project Tittle	Developing A Flight Delay Prediction Using Machine Learning

ML Model on IBM

- Creating IBM cloud account and Required Resource
- Deploy our model in IBM Watson
- Creating Dashboard using HTML/ CSS
- Create web app and Hosting in Flask

Creating IBM cloud account &Required Resources:
Creating IBM cloud account:

First need to create IBM cloud account by using SI mail id and SI password which is provided by IBM in profile.

Preview asset   Visualization

Profile size: 31 Columns

The preview includes only a limited set of columns and rows. ⓘ   Last refresh: 11 seconds ago ↻   Prepare data

YEAR String	MONTH String	DAY String	DAY_OF_... String	AIRLINE String	FLIGHT_NU... String	TAIL_NU... String	ORIGIN_AIR... String	DESTINATION_AIR... String	SCHEDULED_DEPAR... String	DEPARTURE String
2015	1	1	4	AS	98	N407AS	ANC	SEA	0005	2354
2015	1	1	4	AA	2336	N3KUAA	LAX	PBI	0010	0002
2015	1	1	4	US	840	N171US	SFO	CLT	0020	0018
2015	1	1	4	AA	258	N3HYAA	LAX	MIA	0020	0015
2015	1	1	4	AS	135	N527AS	SEA	ANC	0025	0024
2015	1	1	4	DL	806	N3730B	SFO	MSP	0025	0020
2015	1	1	4	NK	612	N635NK	LAS	MSP	0025	0019
2015	1	1	4	US	2013	N584UW	LAX	CLT	0030	0044
2015	1	1	4	AA	1112	N3LAAA	SFO	DFW	0030	0019
2015	1	1	4	DL	1173	N826DN	LAS	ATL	0030	0033
2015	1	1	4	DL	2336	N958DN	DEN	ATL	0030	0024
2015	1	1	4	AA	1674	N853AA	LAS	MIA	0035	0027
2015	1	1	4	DL	1434	N547US	LAX	MSP	0035	0035

Preview asset   Visualization

Profile size: 7 Columns

The preview includes only a limited set of columns and rows. ⓘ   Last refresh: just now ↻   Prepare data

IATA_CODE String	AIRPORT String	CITY String	STATE String	COUNTRY String	LATITUDE String	LONGITUDE String
ABE	Lehigh Valley International Airp	Allentown	PA	USA	40.65236	-75.44040
ABI	Abilene Regional Airport	Abilene	TX	USA	32.41132	-99.68190
ABQ	Albuquerque International Sunp	Albuquerque	NM	USA	35.04022	-106.60919
ABR	Aberdeen Regional Airport	Aberdeen	SD	USA	45.44906	-98.42183
ABY	Southwest Georgia Regional Air	Albany	GA	USA	31.53552	-84.19447
ACK	Nantucket Memorial Airport	Nantucket	MA	USA	41.25305	-70.06018
ACT	Waco Regional Airport	Waco	TX	USA	31.61129	-97.23052
ACV	Arcata Airport	Arcata/Eureka	CA	USA	40.97812	-124.10862
ACY	Atlantic City International Airpo	Atlantic City	NJ	USA	39.45758	-74.57717
ADK	Adak Airport	Adak	AK	USA	51.87796	-176.64603
ADQ	Kodiak Airport	Kodiak	AK	USA	57.74997	-152.49386
AEX	Alexandria International Airport	Alexandria	LA	USA	31.32737	-92.54856
AGS	Augusta Regional Airport (Bush	Augusta	GA	USA	33.36996	-81.96450
AKN	King Salmon Airport	King Salmon	AK	USA	58.67680	-156.64922

Out[125]: True

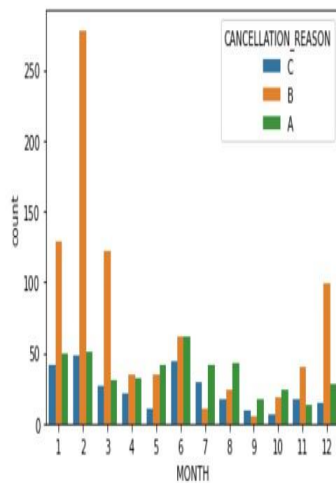In [126... `#Checking how many null values are there in each columns`
`flights.isnull().sum()`

Out[126]:
```
YEAR                    0
MONTH                   0
DAY                     0
DAY_OF_WEEK             0
AIRLINE                 0
FLIGHT_NUMBER           0
TAIL_NUMBER            257
ORIGIN_AIRPORT          0
DESTINATION_AIRPORT     0
SCHEDULED_DEPARTURE     0
DEPARTURE_TIME       1491
DEPARTURE_DELAY      1491
TAXI_OUT             1552
WHEELS_OFF           1552
SCHEDULED_TIME          0
ELAPSED_TIME         1816
AIR_TIME             1816
DISTANCE                0
WHEELS_ON            1615
TAXI_IN              1615
SCHEDULED_ARRIVAL       0
ARRIVAL_TIME         1615
ARRIVAL_DELAY        1816
DIVERTED                0
CANCELLED               0
CANCELLATION_REASON 98427
AIR_SYSTEM_DELAY    81706
SECURITY_DELAY      81706
AIRLINE_DELAY       81706
LATE_AIRCRAFT_DELAY 81706
WEATHER_DELAY       81706
dtype: int64
```

In [128... `sns.countplot(x="MONTH",hue="CANCELLATION_REASON",data=flights)`

Out[128]: <AxesSubplot:xlabel='MONTH', ylabel='count'>

In [129...
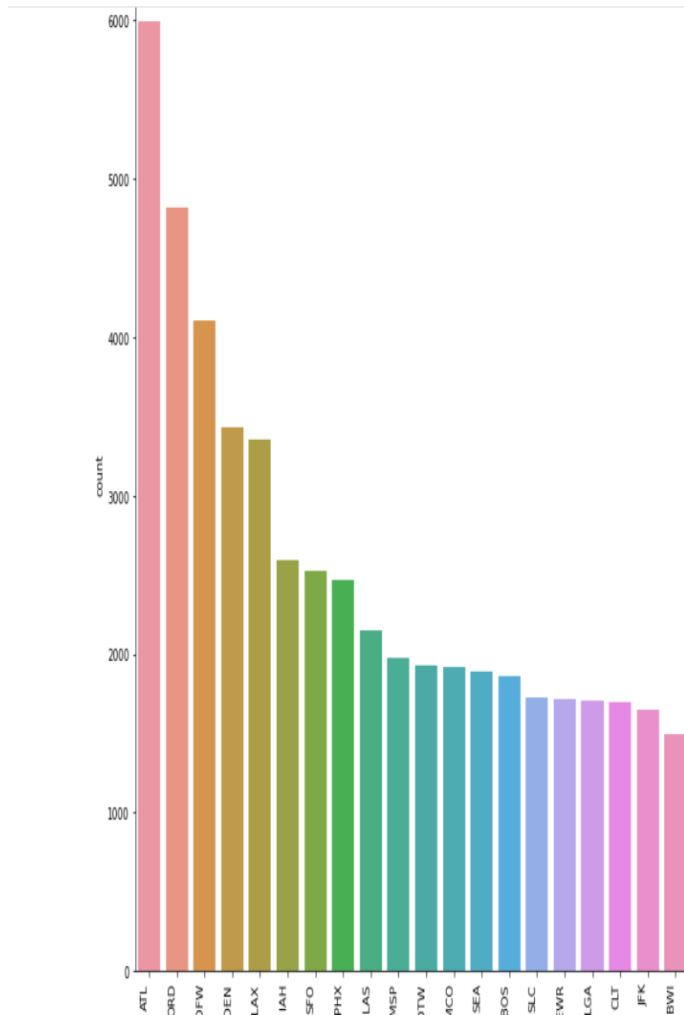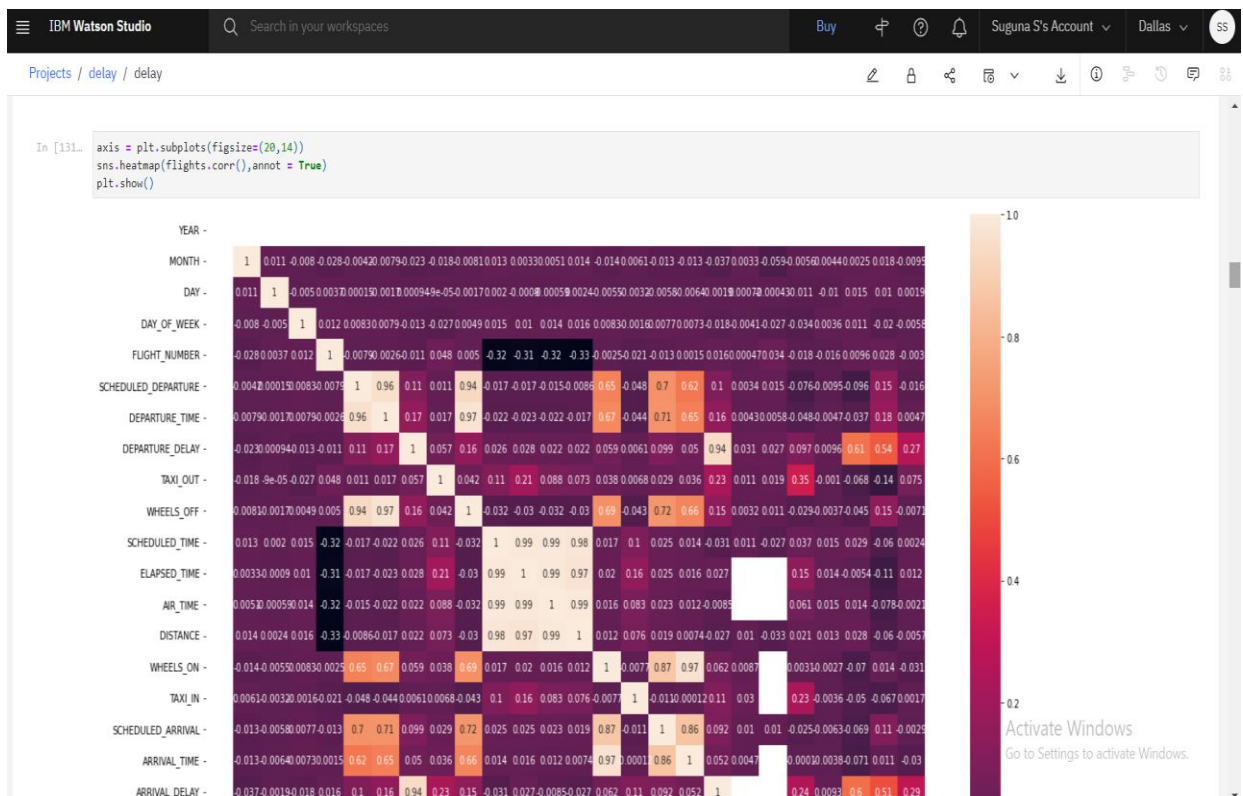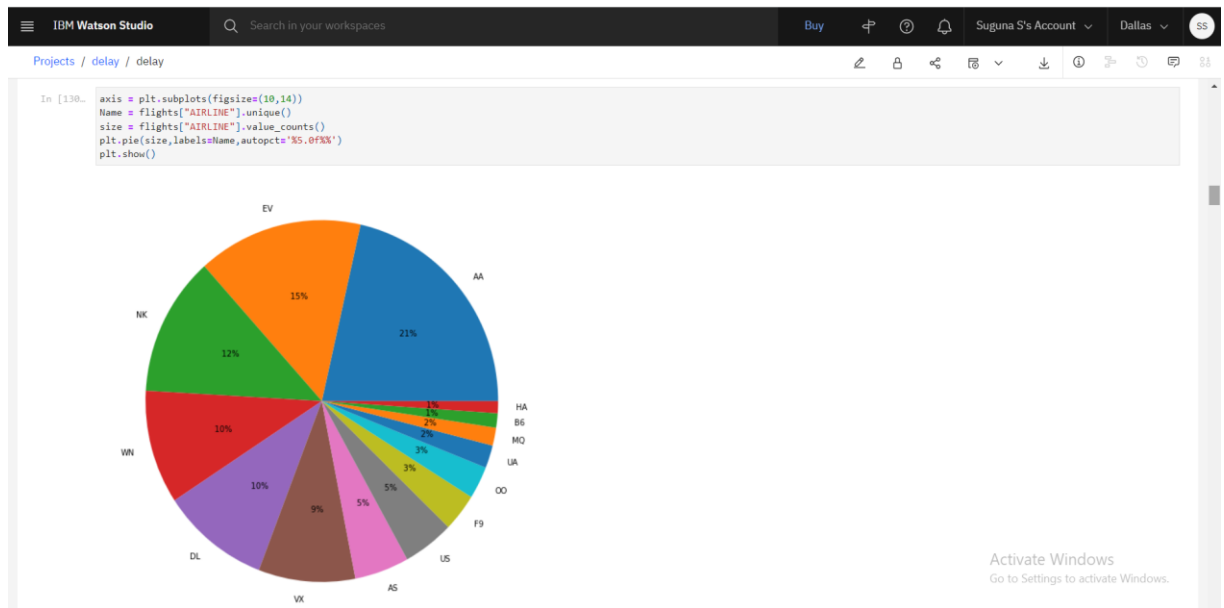```
plt.figure(figsize=(10, 10))
axis = sns.countplot(x=flights['ORIGIN_AIRPORT'], data =flights, order=flights['ORIGIN_AIRPORT'].value_counts().iloc[:20].index)
axis.set_xticklabels(axis.get_xticklabels(), rotation=90, ha="right")
plt.tight_layout()
plt.show()
```

Projects / delay / delay



Activate Windows
Go to Settings to activate Windows.

```
In [130]   axis = plt.subplots(figsize=(10,14))
           Name = flights["AIRLINE"].unique()
           size = flights["AIRLINE"].value_counts()
           plt.pie(size,labels=Name,autopct='%5.0f%%')
           plt.show()
```

```
In [131]   axis = plt.subplots(figsize=(20,14))
           sns.heatmap(flights.corr(),annot = True)
           plt.show()
```

Very High Correlation Between Arrival Delay and Departure Delay

It shows that maximum of the Arrival Delays are due to the Departure Delays.

```
In [132]...  corr=flights.corr()
             corr
```

Out[132]:

	YEAR	MONTH	DAY	DAY_OF_WEEK	FLIGHT_NUMBER	SCHEDULED_DEPARTURE	DEPARTURE_TIME	DEPARTURE_DELAY	TAXI_OUT	WHEELS_OFF	...	SCHEDULED_ARRIVAL	ARRIVAL_TIME	ARRIVA
YEAR	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	
MONTH	NaN	1.000000	0.011153	-0.008015	-0.027701	-0.004210	-0.007888	-0.022869	-0.017812	-0.008080	...	-0.012722	-0.013057	
DAY	NaN	0.011153	1.000000	-0.005012	0.003735	0.000148	-0.001734	0.000936	-0.000090	-0.001730	...	-0.005766	-0.006379	
DAY_OF_WEEK	NaN	-0.008015	-0.005012	1.000000	0.011809	0.008337	0.007895	-0.012553	-0.026787	0.004926	...	0.007653	0.007332	
FLIGHT_NUMBER	NaN	-0.027701	0.003735	0.011809	1.000000	-0.007933	-0.002590	-0.010903	0.048186	0.005024	...	-0.013216	0.001463	
SCHEDULED_DEPARTURE	NaN	-0.004210	0.000148	0.008337	-0.007933	1.000000	0.962237	0.109404	0.010858	0.935609	...	0.701890	0.624584	
DEPARTURE_TIME	NaN	-0.007888	-0.001734	0.007895	-0.002590	0.962237	1.000000	0.167134	0.017172	0.970906	...	0.706025	0.645159	
DEPARTURE_DELAY	NaN	-0.022869	0.000936	-0.012553	-0.010903	0.109404	0.167134	1.000000	0.057351	0.155539	...	0.099325	0.049568	
TAXI_OUT	NaN	-0.017812	-0.000090	-0.026787	0.048186	0.010858	0.017172	0.057351	1.000000	0.041900	...	0.028552	0.036181	
WHEELS_OFF	NaN	-0.008080	-0.001730	0.004926	0.005024	0.935609	0.970906	0.155539	0.041900	1.000000	...	0.716370	0.664687	
SCHEDULED_TIME	NaN	0.012815	0.001993	0.014531	-0.316624	-0.016681	-0.022267	0.025591	0.112607	-0.031522	...	0.024796	0.013658	
ELAPSED_TIME	NaN	0.003279	-0.000895	0.010041	-0.306548	-0.017331	-0.022737	0.028421	0.205129	-0.029646	...	0.024945	0.016438	
AIR_TIME	NaN	0.005071	-0.000590	0.013793	-0.319166	-0.015389	-0.021991	0.021704	0.087903	-0.032190	...	0.023060	0.012488	
DISTANCE	NaN	0.013987	0.002362	0.016323	-0.330516	-0.008556	-0.016964	0.022051	0.072947	-0.029986	...	0.019124	0.007391	
WHEELS_ON	NaN	-0.014054	-0.005518	0.008267	-0.002455	0.652256	0.673558	0.059385	0.038460	0.693404	...	0.872758	0.966288	
TAXI_IN	NaN	0.006086	-0.003178	-0.001606	-0.021153	-0.048041	-0.044294	0.006104	0.006773	-0.043245	...	0.011151	0.000115	
SCHEDULED_ARRIVAL	NaN	-0.012722	-0.005766	0.007653	-0.013216	0.701890	0.706025	0.099325	0.028552	0.716370	...	1.000000	0.862004	
ARRIVAL_TIME	NaN	-0.013057	-0.006379	0.007332	0.001463	0.624584	0.645159	0.049568	0.036181	0.664687	...	0.862004	1.000000	

```
In [138]...  flights.shape
```

Out[138]: (98184, 10)

```
In [139]...  df=pd.DataFrame(flights)
             df['DAY_OF_WEEK']= df['DAY_OF_WEEK'].apply(str)
             df["DAY_OF_WEEK"].replace({"1":"SUNDAY", "2": "MONDAY", "3": "TUESDAY", "4":"WEDNESDAY", "5":"THURSDAY", "6":"FRIDAY", "7":"SATURDAY"},inplace=True)
             flights
```

Out[139]:

	MONTH	DAY	DAY_OF_WEEK	AIRLINE	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	DEPARTURE_DELAY	DISTANCE	ARRIVAL_DELAY
4844280	10	30	THURSDAY	AA	OTHER	OTHER	530	-1.0	257	-7.0
5468262	12	9	TUESDAY	EV	EVV	DTW	1045	7.0	363	13.0
2472487	6	6	FRIDAY	EV	IAH	CLE	720	-7.0	1091	-21.0
2403482	6	2	MONDAY	NK	IAH	ORD	605	9.0	925	72.0
2590983	6	13	FRIDAY	WN	DEN	LAX	825	-2.0	862	-9.0
...	...	...	...	...	...	...	...	...	...	...
821745	2	24	MONDAY	WN	DAL	BWI	600	0.0	1209	-8.0
3686686	8	17	SUNDAY	DL	MSP	MEM	1358	-5.0	700	-4.0
530070	2	5	WEDNESDAY	OO	BJI	MSP	505	-1.0	199	-19.0
474911	2	1	SATURDAY	AA	LAX	DFW	1145	3.0	1235	8.0
5176221	11	20	THURSDAY	NK	DTW	IAH	710	23.0	1075	23.0

98184 rows × 10 columns

```
In [140... dums = ['AIRLINE','ORIGIN_AIRPORT','DESTINATION_AIRPORT','DAY_OF_WEEK']
         df_cat=pd.get_dummies(df[dums],drop_first=True)
         df_cat
```

Out[140]:

	AIRLINE_AS	AIRLINE_B6	AIRLINE_DL	AIRLINE_EV	AIRLINE_F9	AIRLINE_HA	AIRLINE_MQ	AIRLINE_NK	AIRLINE_OO	AIRLINE_UA	...	DESTINATION_AIRPORT_WYS	DESTINATION_AIRPORT_XNA	DESTINATION_AIRPORT
4844280	0	0	0	0	0	0	0	0	0	0	...	0	0	
5468262	0	0	0	1	0	0	0	0	0	0	...	0	0	
2472487	0	0	0	1	0	0	0	0	0	0	...	0	0	
2403482	0	0	0	0	0	0	0	1	0	0	...	0	0	
2590983	0	0	0	0	0	0	0	0	0	0	...	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
821745	0	0	0	0	0	0	0	0	0	0	...	0	0	
3686686	0	0	1	0	0	0	0	0	0	0	...	0	0	
530070	0	0	0	0	0	0	0	0	1	0	...	0	0	
474911	0	0	0	0	0	0	0	0	0	0	...	0	0	
5176221	0	0	0	0	0	0	0	1	0	0	...	0	0	

98184 rows × 661 columns

```
In [141... df_cat.columns
```

```
Out[141]: Index(['AIRLINE_AS', 'AIRLINE_B6', 'AIRLINE_DL', 'AIRLINE_EV', 'AIRLINE_F9',
               'AIRLINE_HA', 'AIRLINE_MQ', 'AIRLINE_NK', 'AIRLINE_OO', 'AIRLINE_UA',
               ...
               'DESTINATION_AIRPORT_WYS', 'DESTINATION_AIRPORT_XNA',
               'DESTINATION_AIRPORT_YAK', 'DESTINATION_AIRPORT_YUM',
               'DAY_OF_WEEK_MONDAY', 'DAY_OF_WEEK_SATURDAY', 'DAY_OF_WEEK_SUNDAY',
               'DAY_OF_WEEK_THURSDAY', 'DAY_OF_WEEK_TUESDAY', 'DAY_OF_WEEK_WEDNESDAY'],
              dtype='object', length=661)
```

```
In [144... var_to_remove=["DAY_OF_WEEK","AIRLINE","ORIGIN_AIRPORT","DESTINATION_AIRPORT"]
         df.drop(var_to_remove,axis=1,inplace=True)
         df
```

Out[144]:

	MONTH	DAY	SCHEDULED_DEPARTURE	DEPARTURE_DELAY	DISTANCE	ARRIVAL_DELAY
4844280	10	30	530	-1.0	257	-7.0
5468262	12	9	1045	7.0	363	13.0
2472487	6	6	720	-7.0	1091	-21.0
2403482	6	2	605	9.0	925	72.0
2590983	6	13	825	-2.0	862	-9.0
...	...	...	...	...	...	...
821745	2	24	600	0.0	1209	-8.0
3686686	8	17	1358	-5.0	700	-4.0
530070	2	5	505	-1.0	199	-19.0
474911	2	1	1145	3.0	1235	8.0
5176221	11	20	710	23.0	1075	23.0

98184 rows × 6 columns

```
In [145...  data=pd.concat([df,df_cat],axis=1)
           data
```

Out[145]:

	MONTH	DAY	SCHEDULED_DEPARTURE	DEPARTURE_DELAY	DISTANCE	ARRIVAL_DELAY	AIRLINE_AS	AIRLINE_B6	AIRLINE_DL	AIRLINE_EV	...	DESTINATION_AIRPORT_WYS	DESTINATION_AIRPORT_XNA	DESTINATION
4844280	10	30	530	-1.0	257	-7.0	0	0	0	0	...	0	0	
5468262	12	9	1045	7.0	363	13.0	0	0	0	1	...	0	0	
2472487	6	6	720	-7.0	1091	-21.0	0	0	0	1	...	0	0	
2403482	6	2	605	9.0	925	72.0	0	0	0	0	...	0	0	
2590983	6	13	825	-2.0	862	-9.0	0	0	0	0	...	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
821745	2	24	600	0.0	1209	-8.0	0	0	0	0	...	0	0	
3686686	8	17	1358	-5.0	700	-4.0	0	0	1	0	...	0	0	
530070	2	5	505	-1.0	199	-19.0	0	0	0	0	...	0	0	
474911	2	1	1145	3.0	1235	8.0	0	0	0	0	...	0	0	
5176221	11	20	710	23.0	1075	23.0	0	0	0	0	...	0	0	

98184 rows × 667 columns

```
In [146...  data.shape
```

Out[146]: (98184, 667)

```
In [136...  print(flights.ORIGIN_AIRPORT.nunique())
           print(flights.DESTINATION_AIRPORT.nunique())
           print(flights.AIRLINE.nunique())

           322
           322
           14
```

```
In [137...  flights=flights.dropna()
           flights
```

Out[137]:

	MONTH	DAY	DAY_OF_WEEK	AIRLINE	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	DEPARTURE_DELAY	DISTANCE	ARRIVAL_DELAY
4844280	10	30	5	AA	OTHER	OTHER	530	-1.0	257	-7.0
5468262	12	9	3	EV	EVV	DTW	1045	7.0	363	13.0
2472487	6	6	6	EV	IAH	CLE	720	-7.0	1091	-21.0
2403482	6	2	2	NK	IAH	ORD	605	9.0	925	72.0
2590983	6	13	6	WN	DEN	LAX	825	-2.0	862	-9.0
...	...	...	...	...	...	...	...	...	...	...
821745	2	24	2	WN	DAL	BWI	600	0.0	1209	-8.0
3686686	8	17	1	DL	MSP	MEM	1358	-5.0	700	-4.0
530070	2	5	4	OO	BJI	MSP	505	-1.0	199	-19.0
474911	2	1	7	AA	LAX	DFW	1145	3.0	1235	8.0
5176221	11	20	5	NK	DTW	IAH	710	23.0	1075	23.0

98184 rows × 10 columns

```
In [138...  flights.shape
```

Out[138]: (98184, 10)

```
In [133... variables_to_remove=["YEAR","FLIGHT_NUMBER","TAIL_NUMBER","DEPARTURE_TIME","TAXI_OUT","WHEELS_OFF","ELAPSED_TIME","AIR_TIME","WHEELS_ON","TAXI_IN","ARRIVAL_TIME","DIVERTED","CANCELLED","CANCE
           flights.drop(variables_to_remove,axis=1,inplace= True)
           flights.columns
```

```
Out[133]: Index(['MONTH', 'DAY', 'DAY_OF_WEEK', 'AIRLINE', 'ORIGIN_AIRPORT',
                  'DESTINATION_AIRPORT', 'SCHEDULED_DEPARTURE', 'DEPARTURE_DELAY',
                  'DISTANCE', 'ARRIVAL_DELAY'],
                 dtype='object')
```

```
In [134... 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='bMhy8NPFEgbvGvS4iBZAy9k6uvwCxNmaZGfGX2TsL6VH',
              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 = 'delay-donotdelete-pr-ahndfuqxfwvkgn'
          object_key = 'airports.csv'

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

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

```
Out[134]:    IATA CODE          AIRPORT          CITY  STATE  COUNTRY  LATITUDE  LONGITUDE
```

```
In [152... Y
```

```
Out[152]: 5760772    21.0
          2384896    -1.0
          4294434    -7.0
          4838951    10.0
          1081494    43.0
                     ...
          1745015     0.0
          5150614    26.0
          3740656    14.0
          5421824    -5.0
          5692067    -1.0
          Name: DEPARTURE_DELAY, Length: 60000, dtype: float64
```

```
In [153... from sklearn.model_selection import train_test_split, cross_val_score, cross_val_predict
          X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=0)
```

```
In [154... from sklearn.ensemble import RandomForestRegressor
          reg_rf = RandomForestRegressor()
          reg_rf.fit(X_train,y_train)
```

```
Out[154]: RandomForestRegressor()
```

```
In [155... y_pred = reg_rf.predict(X_test)
```

```
In [156... reg_rf.score(X_train,y_train)
```

```
Out[156]: 0.9895460147562944
```

```
In [157... reg_rf.score(X_test,y_test)
```

```
Out[157]: 0.9251558566324755
```

Activate Windows
Go to Settings to activate Windows.

```
In [158]   metrics.r2_score(y_test,y_pred)
```

Out[158]: 0.9251558566324755

```
In [159]   print('MAE:', metrics.mean_absolute_error(y_test,y_pred))
           print('MSE:', metrics.mean_squared_error(y_test,y_pred))
           print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test,y_pred)))
```

```
MAE: 6.002905833333333
MSE: 96.55449255833332
RMSE: 9.826214558940453
```

```
In [160]   pp=pd.DataFrame({'Actual':y_test,'Predicted':y_pred})
           pp
```

Out[160]:

	Actual	Predicted
272846	-5.0	-4.04
2840575	-10.0	-3.03
2900672	-3.0	2.89
5609530	-4.0	6.26
4954342	42.0	25.10
...	...	...
707976	23.0	5.30
3318061	12.0	4.17
2493555	20.0	15.86
2376690	0.0	0.42
5676318	57.0	49.48

12000 rows × 2 columns

```
In [178]   rf_random.best_params_
```

Out[178]: {'n_estimators': 61,
            'min_samples_split': 5,
            'min_samples_leaf': 5,
            'max_features': 'auto',
            'max_depth': 15}

```
In [166]   p=rf_random.predict(X_test)
```

```
In [167]   metrics.r2_score(y_test,p)
```

Out[167]: 0.9261197901555904

```
In [181]   print('MAE:', metrics.mean_absolute_error(y_test,p))
           print('MSE:', metrics.mean_squared_error(y_test,p))
           print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test,p)))
```

```
MAE: 6.0156529151472204
MSE: 95.31094686462018
RMSE: 9.762732551116013
```

```
In [182]   zz=pd.DataFrame({'Actual':y_test,'Predicted':p})
           zz
```

Out[182]:

	Actual	Predicted
272846	-5.0	-3.671236
2840575	-10.0	-2.779516
2900672	-3.0	2.766024
5609530	-4.0	5.657182
4954342	42.0	24.716256
...	...	...
707976	23.0	5.857987

```
In [170]_  from sklearn.ensemble import GradientBoostingRegressor
           gbr=GradientBoostingRegressor(random_state=0)
```

```
In [171]_  GBR=gbr.fit(X_train,y_train)
           pre=GBR.predict(X_test)
```

```
In [172]_  print('MAE:', metrics.mean_absolute_error(y_test,pre))
           print('MSE:', metrics.mean_squared_error(y_test,pre))
           print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test,pre)))

           MAE: 6.003761560382866
           MSE: 92.92993777268747
           RMSE: 9.640017519314343
```

```
In [173]_  metrics.r2_score(y_test,pre)
```

```
Out[173]:  0.927965427589067
```

```
In [174]_  gg=pd.DataFrame({'Actual':y_test,'Predicted':pre})
           gg
```

Out[174]:

	Actual	Predicted
272846	-5.0	-1.737581
2840575	-10.0	-4.177277
2900672	-3.0	1.024260
5609530	-4.0	6.056414
4954342	42.0	28.400018
...	...	...
707976	23.0	6.421029
3318061	12.0	4.003290

```
In [3]:  pip install ibm_watson_machine_learning

Requirement already satisfied: ibm_watson_machine_learning in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.257)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2.26.0)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (0.3.3)
Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (4.8.2)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2022.9.24)
Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (1.3.4)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (1.26.7)
Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (21.3)
Requirement already satisfied: ibm-cos-sdk==2.11.* in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2.11.0)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (0.8.9)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (0.10.0)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (2.11.0)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (2.11.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (1.20.3)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (1.15.0)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm_watson_machine_learning) (3.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm_watson_machine_learning) (2.0.4)
Requirement already satisfied: zipp>=0.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from importlib-metadata->ibm_watson_machine_learning) (3.6.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from packaging->ibm_watson_machine_learning) (3.0.4)
Note: you may need to restart the kernel to use updated packages.
```

```
In [ ]:  from ibm_watson_machine_learning import APIClient
         wml_credentials={
             "url":"https://us-south.ml.cloud.ibm.com"
             "API":"98jRIthntK-CL9W6TI5C0mblRv-scfWQQo9tao4YlIPP"
         }
         client=APIClient(wml_credentials)
```

```
In [ ]:  def guide_from_space_name(client, space_name):
             space=client.spaces.get_details()
             return(next(item for item in space['resources'] if item['entity']["name"]==space_name)['metadata']['id'])
```

```
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2.26.0)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (0.3.3)
Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (4.8.2)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2022.9.24)
Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (1.3.4)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (1.26.7)
Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (21.3)
Requirement already satisfied: ibm-cos-sdk==2.11.* in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2.11.0)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (0.8.9)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (0.10.0)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (2.11.0)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (2.11.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (1.20.3)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (1.15.0)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm_watson_machine_learning) (3.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm_watson_machine_learning) (2.0.4)
Requirement already satisfied: zipp>=0.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from importlib-metadata->ibm_watson_machine_learning) (3.6.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from packaging->ibm_watson_machine_learning) (3.0.4)
Note: you may need to restart the kernel to use updated packages.
```

```
In [ ]: from ibm_watson_machine_learning import APIClient
        wml_credentials={
            "url":"https://us-south.ml.cloud.ibm.com"
            "API":"98jRIthntK-CL9W6TI5C0mblRv-scfWQQo9tao4YlIPP"
        }
        client=APIClient(wml_credentials)
```

```
In [ ]: def guide_from_space_name(client, space_name):
            space=client.spaces.get_details()
            return(next(item for item in space['resources'] if item['entity']["name"]==space_name)['metadata']['id'])
```

```
In [5]: space_uid= guid_from_space_name(client, 'Newspace')
        print("Space UID = " + space_uid)
```

Activate Windows
Go to Settings to activate Windows.