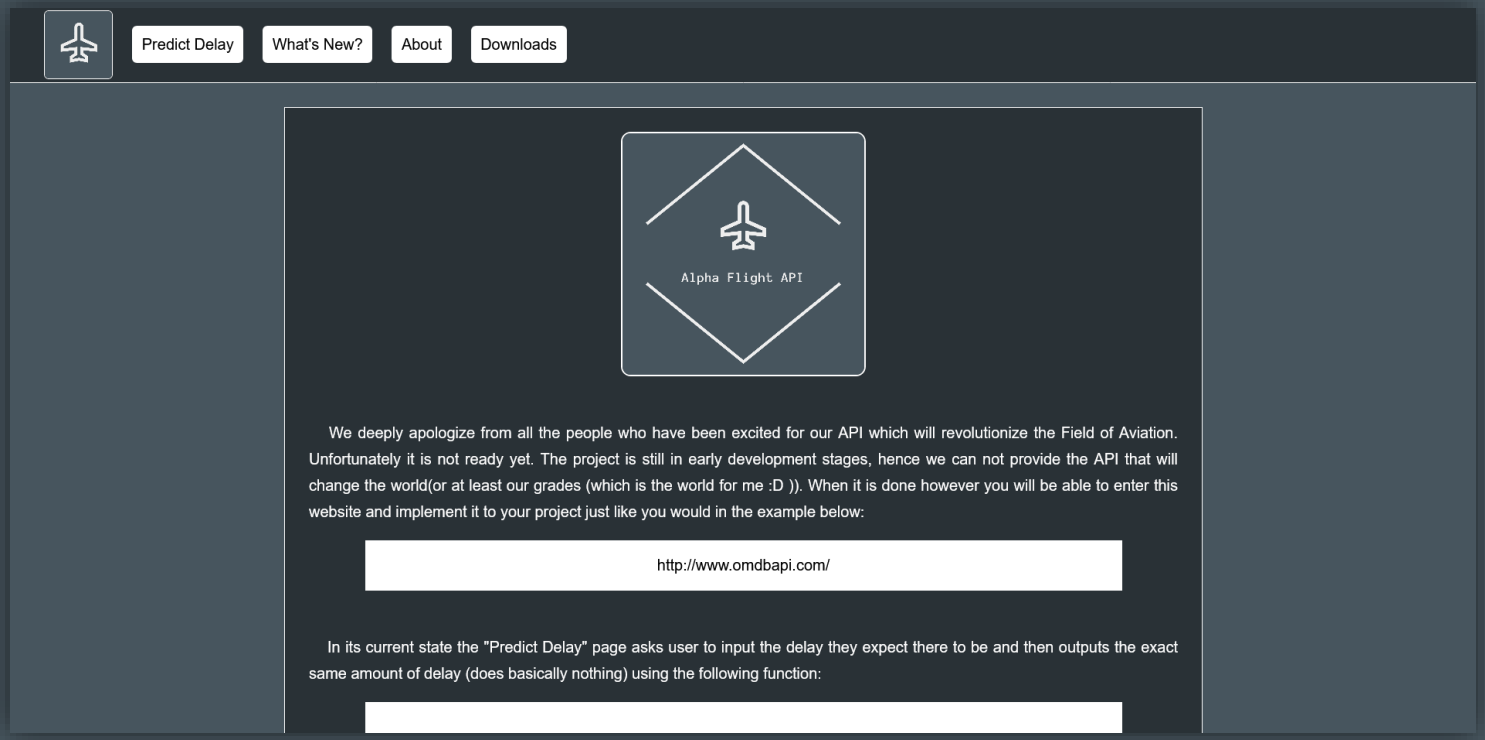
Teoman Berkay AYAZ
1800004169

Dennis BREZINA
1700004948

Zeynep Simge SEDEF
1700003227

Kerem Safa DIRICAN
1800002205



We have developed and set up a webpage to showcase our work.

https://keremec.github.io/alphaflight/

Our Homepage

We deeply apologize from all the people who have been excited for our API which will revolutionize the Field of Aviation. Unfortunately it is not ready yet. The project is still in early development stages, hence we can not provide the API that will change the world(or at least our grades (which is the world for me :D)). When it is done however you will be able to enter this website and implement it to your project just like you would in the example below:

http://www.omdbapi.com/

In its current state the "Predict Delay" page asks user to input the delay they expect there to be and then outputs the exact same amount of delay (does basically nothing) using the following function:

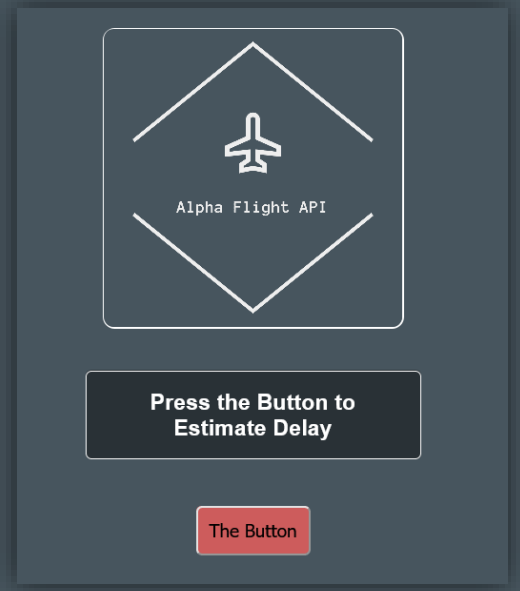
$$f(x) = x$$

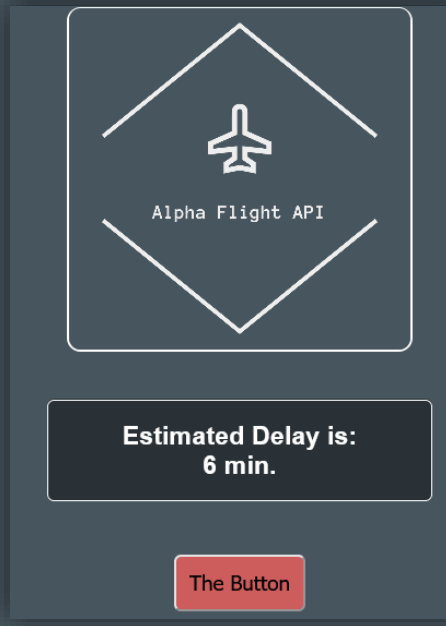
In the later stages we wish to implement a machine learning algorithm using a function similar to below with the features similar to below:

$$h\theta(x)=\theta 0 + \theta 1*x1 + \theta 2*x2 + \theta 3*x3 + \theta 4*x4 + \ldots + \theta n*xn$$

Feature Order	Feature Name
x1	Air Traffic
x2	Weather
x3	Crew Related
x4	Cargo Related
. . .	. . .
xn	Feature n

Predict Delay





What's New

V 0.0.0

Welcome to "What's New" page of our site. Here we will be updating on the progress of our project on a weekly basis. Even though we currently do not possess the technical knowledge to call it a machine learning project, we have developed an algorithm which later on will be used in our machine learning project. And since our project is an API that can be implemented across various projects we decided to develop a website to demo our API(even though it is not ready). If you wish to access additional resources such as; diagrams,code, reports etc. there will be links below to access them.

Resources:	Presentation of Week 5	Presentation of Week 4	Presentation of Week 3

Downloads

Our Python Code

Last Updated : 25/03/2021

Last Updated : 25/03/2021

Documentation

Last Updated : 25/03/2021

Additional Resources

Last Updated : 25/03/2021

Our Code

Author : Dennis Brezina

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
import seaborn as sns
from IPython.display import display
import os, gc, datetime
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
warnings.filterwarnings("ignore", category=FutureWarning)

flight = pd.read_csv("2018.csv")

#verisetini tanımak
flight.info()

#var olan uçak şirketleri ve uçuş sayılarını görmek
#flight["OP_CARRIER"].value_counts()

#kategorileri görmek
#flight.keys()
pd.set_option("display.max.columns", None)
pd.set_option("display.precision", 2)

#verisetinde eksik değişken görmek
flight.isnull().sum()

#betimsel istatistikleri incelemk
flight.describe().T

kmeans = KMeans(n_clusters = 4)
kmeans
```

[4]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
warnings.filterwarnings("ignore", category=FutureWarning)
```

[2]:
```
df = pd.read_csv('./2018.csv', index_col = 0)
df.head()
```

[3]:

FL_DATE	OP_CARRIER	OP_CARRIER_FL_NUM	ORIGIN	DEST	CRS_DEP_TIME	DEP_TIME	DEP_DELAY	TAXI_OUT	WHEELS_OFF	WHEELS_ON	...	CRS_ELAPSED_TIME	ACTUAL_ELAPSED_TIME	AIR_TIME	DISTANCE	CARRIER_DELAY	WEATHER_DELAY	…
2018-01-01	9A	2429	EWR	DEN	1517	1512.0	-5.0	15.0	1527.0	1712.0	...	268.0	260.0	225.0	1605.0	NaN	NaN	
2018-01-01	9A	2427	LAS	SFO	1115	1107.0	-8.0	11.0	1118.0	1223.0	...	80.0	85.0	65.0	414.0	NaN	NaN	
2018-01-01	9A	2428	SNA	DEN	1035	1030.0	-5.0	15.0	1045.0	1431.0	...	154.0	126.0	106.0	846.0	NaN	NaN	
2018-01-01	9A	2425	MSN	ORD	1546	1552.0	6.0	19.0	1611.0	1748.0	...	166.0	182.0	157.0	1129.0	NaN	NaN	
2018-01-01	9A	2424	ORD	ASE	630	650.0	20.0	15.0	705.0	935.0	...	112.0	106.0	85.0	723.0	NaN	NaN	

5 rows × 27 columns

```
[3]: df.isnull().sum()

[3]: OP_CARRIER                 0
     OP_CARRIER_FL_NUM          0
     ORIGIN                     0
     DEST                       0
     CRS_DEP_TIME               0
     DEP_TIME                 112317
     DEP_DELAY                117234
     TAXI_OUT                 115830
     WHEELS_OFF               115829
     WHEELS_ON                119246
     TAXI_IN                  119246
     CRS_ARR_TIME               0
     ARR_TIME                 119245
     ARR_DELAY                137040
     CANCELLED                  0
     CANCELLATION_CODE       7096862
     DIVERTED                   0
     CRS_ELAPSED_TIME           10
     ACTUAL_ELAPSED_TIME      134442
     AIR_TIME                 134442
     DISTANCE                   0
     CARRIER_DELAY           5860736
     WEATHER_DELAY           5860736
     NAS_DELAY               5860736
     SECURITY_DELAY          5860736
     LATE_AIRCRAFT_DELAY     5860736
     Unnamed: 27             7213446
     dtype: int64
```

```
]: df.info()

<class 'pandas.core.frame.DataFrame'>
Index: 7213446 entries, 2018-01-01 to 2018-12-31
Data columns (total 27 columns):
 #   Column               Dtype
---  ------               -----
 0   OP_CARRIER           object
 1   OP_CARRIER_FL_NUM    int64
 2   ORIGIN               object
 3   DEST                 object
 4   CRS_DEP_TIME         int64
 5   DEP_TIME             float64
 6   DEP_DELAY            float64
 7   TAXI_OUT             float64
 8   WHEELS_OFF           float64
 9   WHEELS_ON            float64
 10  TAXI_IN              float64
 11  CRS_ARR_TIME         int64
 12  ARR_TIME             float64
 13  ARR_DELAY            float64
 14  CANCELLED            float64
 15  CANCELLATION_CODE    object
 16  DIVERTED             float64
 17  CRS_ELAPSED_TIME     float64
 18  ACTUAL_ELAPSED_TIME  float64
 19  AIR_TIME             float64
 20  DISTANCE             float64
 21  CARRIER_DELAY        float64
 22  WEATHER_DELAY        float64
 23  NAS_DELAY            float64
 24  SECURITY_DELAY       float64
 25  LATE_AIRCRAFT_DELAY  float64
 26  Unnamed: 27          float64
dtypes: float64(20), int64(3), object(4)
memory usage: 1.5+ GB
```

```
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
OP_CARRIER_FL_NUM	7213446.0	2607.531335	1860.122265	1.0	1029.0	2131.0	4074.0	7909.0
CRS_DEP_TIME	7213446.0	1329.687018	490.931982	1.0	915.0	1320.0	1735.0	2359.0
DEP_TIME	7101129.0	1333.853806	504.505548	1.0	916.0	1326.0	1744.0	2400.0
DEP_DELAY	7096212.0	9.969858	44.829641	-122.0	-5.0	-2.0	7.0	2710.0
TAXI_OUT	7097616.0	17.410614	9.920409	1.0	11.0	15.0	20.0	196.0
WHEELS_OFF	7097617.0	1357.798878	505.972136	1.0	932.0	1340.0	1759.0	2400.0
WHEELS_ON	7094200.0	1462.162009	533.467516	1.0	1044.0	1502.0	1911.0	2400.0
TAXI_IN	7094200.0	7.601246	6.064797	1.0	4.0	6.0	9.0	259.0
CRS_ARR_TIME	7213446.0	1486.341099	518.312428	1.0	1100.0	1515.0	1919.0	2400.0
ARR_TIME	7094201.0	1466.784165	537.708924	1.0	1049.0	1506.0	1916.0	2400.0
ARR_DELAY	7076406.0	5.048581	46.926637	-120.0	-14.0	-6.0	8.0	2692.0
CANCELLED	7213446.0	0.016162	0.126098	0.0	0.0	0.0	0.0	1.0
DIVERTED	7213446.0	0.002476	0.049696	0.0	0.0	0.0	0.0	1.0
CRS_ELAPSED_TIME	7213436.0	141.135648	73.344332	-99.0	88.0	122.0	171.0	704.0
ACTUAL_ELAPSED_TIME	7079004.0	136.499938	73.137578	14.0	83.0	118.0	167.0	757.0
AIR_TIME	7079004.0	111.502048	71.112927	7.0	60.0	92.0	141.0	696.0
DISTANCE	7213446.0	799.989490	598.178288	31.0	363.0	632.0	1034.0	4983.0
CARRIER_DELAY	1352710.0	19.455006	58.908119	0.0	0.0	0.0	17.0	2109.0
WEATHER_DELAY	1352710.0	3.636459	29.996006	0.0	0.0	0.0	0.0	2692.0
NAS_DELAY	1352710.0	15.885471	35.893497	0.0	0.0	3.0	20.0	1848.0
SECURITY_DELAY	1352710.0	0.093539	3.174306	0.0	0.0	0.0	0.0	987.0
LATE_AIRCRAFT_DELAY	1352710.0	25.644120	49.787761	0.0	0.0	3.0	31.0	2454.0
Unnamed: 27	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
kmeans = KMeans(n_clusters = 4)
```

```
kmeans
```

```
KMeans(n_clusters=4)
```

```
k_fit = kmeans.fit(df)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-17-69f3e0231e45> in <module>
----> 1 k_fit = kmeans.fit(df)

D:\anaconda\lib\site-packages\sklearn\cluster\_kmeans.py in fit(self, X, y, sample_weight)
    977             Fitted estimator.
    978         """
--> 979         X = self._validate_data(X, accept_sparse='csr',
    980                                 dtype=[np.float64, np.float32],
    981                                 order='C', copy=self.copy_x,

D:\anaconda\lib\site-packages\sklearn\base.py in _validate_data(self, X, y, reset, validate_separately, **check_params)
    419             out = X
    420         elif isinstance(y, str) and y == 'no_validation':
--> 421             X = check_array(X, **check_params)
    422             out = X
    423         else:

D:\anaconda\lib\site-packages\sklearn\utils\validation.py in inner_f(*args, **kwargs)
     61             extra_args = len(args) - len(all_args)
     62             if extra_args <= 0:
---> 63                 return f(*args, **kwargs)
     64
     65             # extra_args > 0

D:\anaconda\lib\site-packages\sklearn\utils\validation.py in check_array(array, accept_sparse, accept_large_sparse, dtype, order, copy, force_all_finite, ensure_2d, allow_nd, ensure_min_samples, ensure_min_features, estimator)
    614                     array = array.astype(dtype, casting="unsafe", copy=False)
    615                 else:
--> 616                     array = np.asarray(array, order=order, dtype=dtype)
    617             except ComplexWarning as complex_warning:
    618                 raise ValueError("Complex data not supported\n"

D:\anaconda\lib\site-packages\numpy\core\_asarray.py in asarray(a, dtype, order)
     81
     82     """
---> 83     return array(a, dtype, copy=False, order=order)
     84
     85

D:\anaconda\lib\site-packages\pandas\core\generic.py in __array__(self, dtype)
   1779
   1780     def __array__(self, dtype=None) -> np.ndarray:
-> 1781         return np.asarray(self._values, dtype=dtype)
   1782
   1783     def __array_wrap__(self, result, context=None):

D:\anaconda\lib\site-packages\numpy\core\_asarray.py in asarray(a, dtype, order)
     81
     82     """
---> 83     return array(a, dtype, copy=False, order=order)
     84
     85

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

```
In [9]: flight["OP_CARRIER"].value_counts()
Out[9]:
WN    1352552
DL     949283
AA     916818
OO     774137
UA     621565
YX     316090
B6     305010
MQ     296001
OH     278457
9E     245917
AS     245761
YV     215138
EV     202890
NK     176178
F9     120035
G4      96221
HA      83723
VX      17670
Name: OP_CARRIER, dtype: int64
```