21.12.2020

Istanbul Water Crisis Detection Project

A Multi-Directional Data Science Project

Melih Kuru
Applied Data Science Capstone - IBM

A. Introduction

A.1. Description & Disscusion of the Background

Istanbul formerly Byzantium and Constantinople, is the largest city in Turkey and the country's economic, cultural and historic center. The city straddles the Bosporus strait, and lies in both Europe and Asia, with a population of over 15 million residents. Istanbul is the largest city in Europe, and the world's fifteenth-largest city.

However, the water problem has always been a problem for Istanbul since the middle ages. Since the Byzantine period, the city's rapid growth has consumed its water resources, and the city's rulers built water channels to carry the surrounding waters to the city.

In 2020, decreasing rainfall and global warming brought the city's water stocks back to 21.66%. In this study, I will first create the fresh water resources of the city according to the information on the official website and check their coordinates with the Foursquare Api.

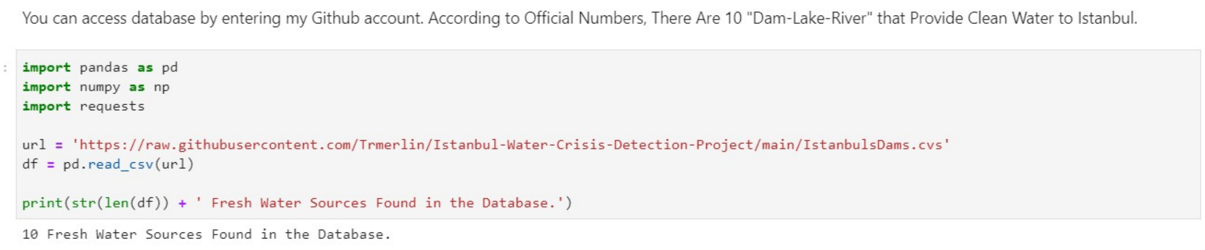
Then, I will calculate the water balance required for the city not to have water crisis, and the day when the city will be drought, with the multilayer perceptron, which is a neural network.

A.2. Data Description

To consider the problem we can list the datas as belows

- I will get the clean water resources information of the city from the official website of Istanbul, Official Website
- However, only the names and occupancy rates of water resources can be seen on this site. Therefore, I will manually determine the coordinates of the relevant water resources on "Google Maps" and check them with foursquare.
- I used Forsquare API to check if the coordinates of the Dams are correct.
- I received meteorological data such as the city's water consumption and rainfall from the open data portal of the metropolitan municipality. Istanbul Municipality Open Data Portal

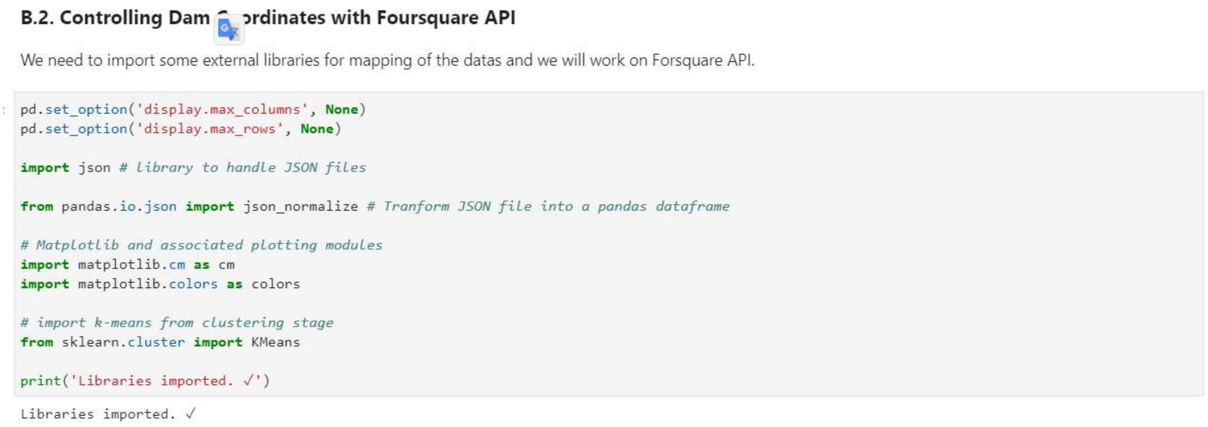
B. Methodology



You can access database by entering my Github account. According to Official Numbers, There Are 10 "Dam-Lake-River" that Provide Clean Water to Istanbul.

```
import pandas as pd
import numpy as np
import requests

url = 'https://raw.githubusercontent.com/Trmerlin/Istanbul-Water-Crisis-Detection-Project/main/IstanbulsDams.cvs'
df = pd.read_csv(url)

print(str(len(df)) + ' Fresh Water Sources Found in the Database.')
```

```
10 Fresh Water Sources Found in the Database.
```

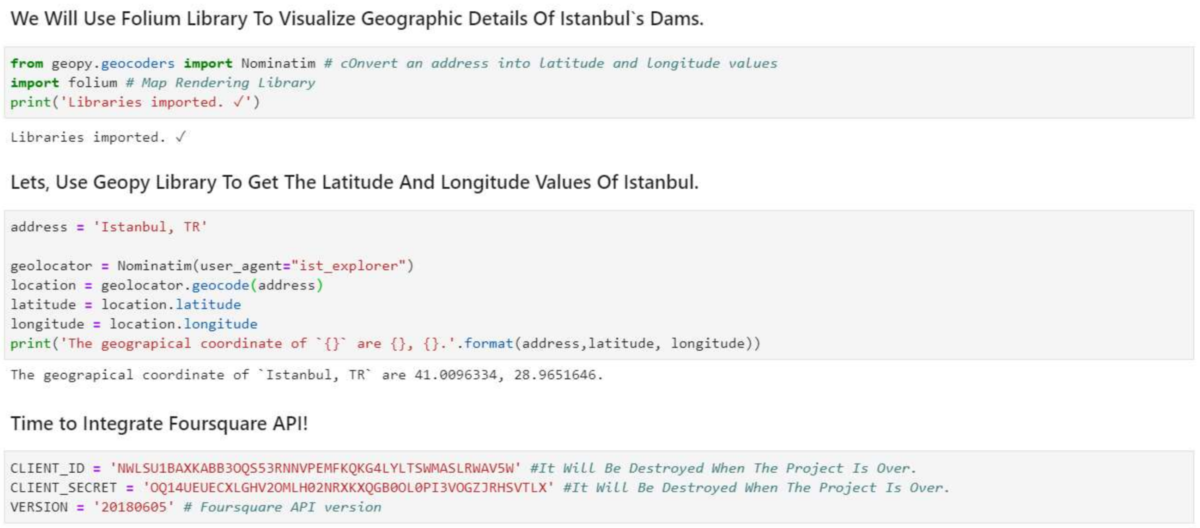


B.2. Controlling Dam Coordinates with Foursquare API

We need to import some external libraries for mapping of the datas and we will work on Forsquare API.

```
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # library to handle JSON files

from pandas.io.json import json_normalize # Tranform JSON file into a pandas dataframe

# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors

# import k-means from clustering stage
from sklearn.cluster import KMeans

print('Libraries imported. ✓')
```

```
Libraries imported. ✓
```



We Will Use Folium Library To Visualize Geographic Details Of Istanbul`s Dams.

```
from geopy.geocoders import Nominatim # cOnvert an address into latitude and longitude values
import folium # Map Rendering Library
print('Libraries imported. ✓')
```

```
Libraries imported. ✓
```

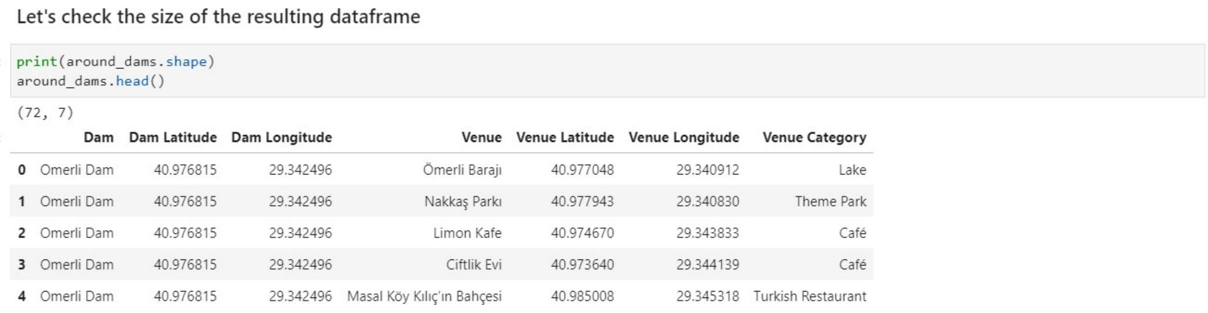
Lets, Use Geopy Library To Get The Latitude And Longitude Values Of Istanbul.

```
address = 'Istanbul, TR'

geolocator = Nominatim(user_agent="ist_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of `{}` are {}, {}.'.format(address,latitude, longitude))
```

```
The geograpical coordinate of `Istanbul, TR` are 41.0096334, 28.9651646.
```

Time to Integrate Foursquare API!

```
CLIENT_ID = 'NWLSU1BAXKABB3OQS53RNNVPEMFKQKG4LYLTSWMASLRWAV5W' #It Will Be Destroyed When The Project Is Over.
CLIENT_SECRET = 'OQ14UEUECXLGHV2OMLH02NRXKXQGB0OL0PI3VOGZJRHSVTLX' #It Will Be Destroyed When The Project Is Over.
VERSION = '20180605' # Foursquare API version
```

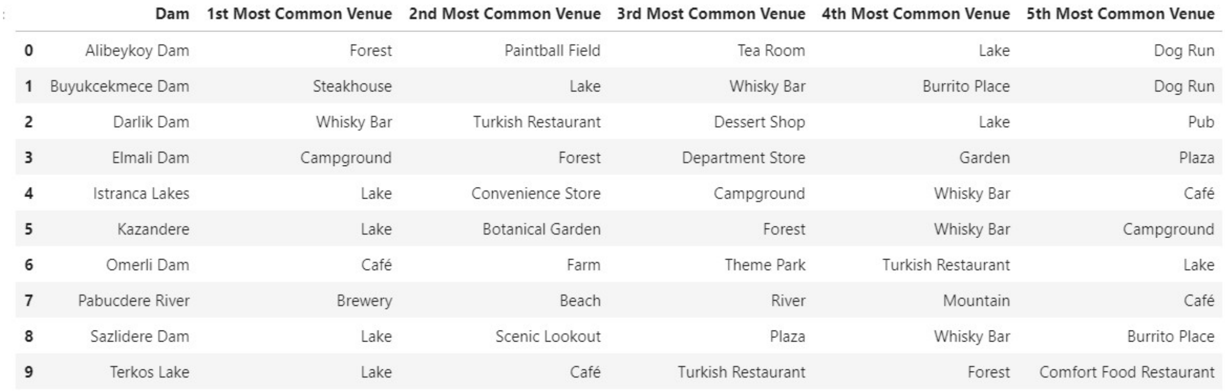


Let's check the size of the resulting dataframe

```
print(around_dams.shape)
around_dams.head()
```

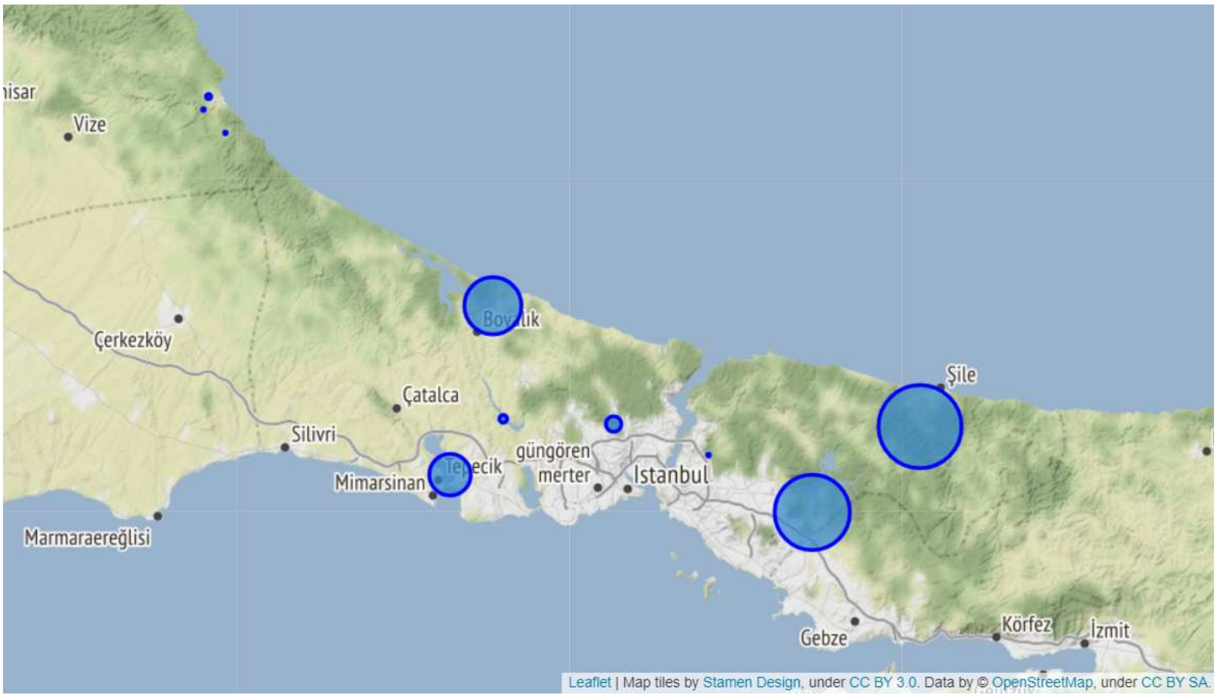
```
(72, 7)
```

	Dam	Dam Latitude	Dam Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Omerli Dam	40.976815	29.342496	Ömerli Barajı	40.977048	29.340912	Lake
1	Omerli Dam	40.976815	29.342496	Nakkaş Parkı	40.977943	29.340830	Theme Park
2	Omerli Dam	40.976815	29.342496	Limon Kafe	40.974670	29.343833	Café
3	Omerli Dam	40.976815	29.342496	Ciftlik Evi	40.973640	29.344139	Café
4	Omerli Dam	40.976815	29.342496	Masal Köy Kılıç'ın Bahçesi	40.985008	29.345318	Turkish Restaurant



	Dam	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	Alibeykoy Dam	Forest	Paintball Field	Tea Room	Lake	Dog Run
1	Buyukcekmece Dam	Steakhouse	Lake	Whisky Bar	Burrito Place	Dog Run
2	Darlik Dam	Whisky Bar	Turkish Restaurant	Dessert Shop	Lake	Pub
3	Elmali Dam	Campground	Forest	Department Store	Garden	Plaza
4	Istranca Lakes	Lake	Convenience Store	Campground	Whisky Bar	Café
5	Kazandere	Lake	Botanical Garden	Forest	Whisky Bar	Campground
6	Omerli Dam	Café	Farm	Theme Park	Turkish Restaurant	Lake
7	Pabucdere River	Brewery	Beach	River	Mountain	Café
8	Sazlidere Dam	Lake	Scenic Lookout	Plaza	Whisky Bar	Burrito Place
9	Terkos Lake	Lake	Café	Turkish Restaurant	Forest	Comfort Food Restaurant

Mission Successfully Completed! We Detected Lake or River Record within 1.2 Km of the Coordinates We Received from the Database. Now is the time to integrate our fresh water resources on a map according to their size.



Leaflet | Map tiles by Stamen Design, under CC BY 3.0. Data by © OpenStreetMap, under CC BY SA.

This map shows all the sources that provide fresh water to Istanbul. Flat sizes are proportional to the total volume of the dam.

C. Istanbul's Water Crisis Prediction

After determining the location of our dams, it was time to estimate the thirst.

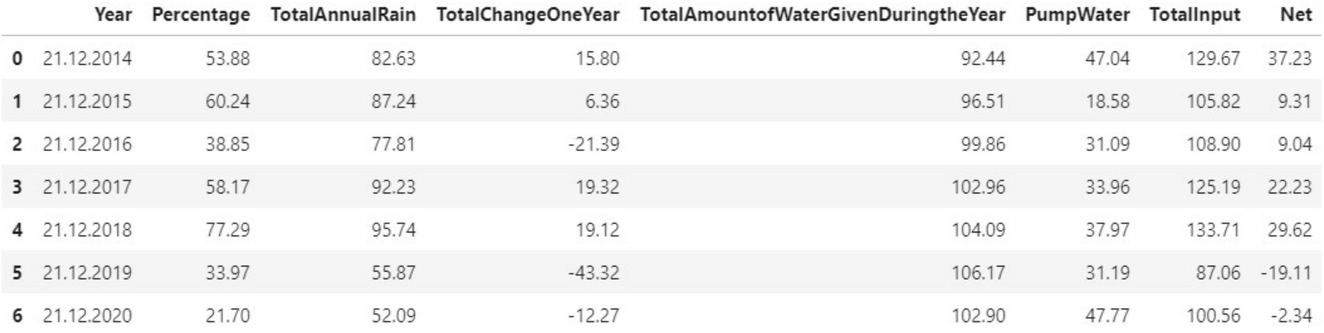
For this, I will use the data provided by the Istanbul municipality. These are basically "Dam Occupancy Rates Of The Last 6 Years For December.21.202X", "Total Annual Rain", "Annual Amount of Water Pumped from Surrounding Provinces" and Finally, "Annual Consumption". I will calculate the balance of the water coming out.

Then I will try to establish a proportion between the "Net Water Balance" in the City and the 1-year change percentages of the dams.

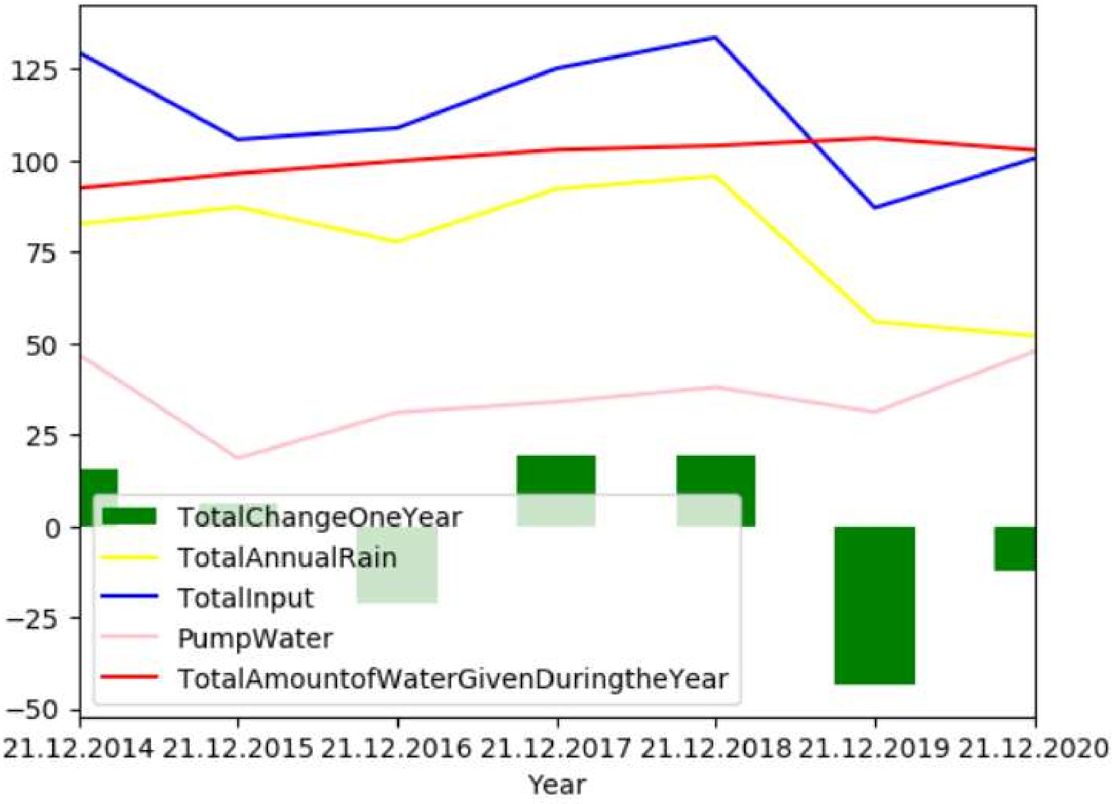
Small Reminders

- **TotalChangeOneYear** = [Percent Of This Year] - [Percent Of Last Year] Istanbul Municipality gives occupancy rates for the same day.
- **PumpWater** = The total water provided to the city by the 2 pump systems established by the Municipality to meet the water requirement
- **TotalAmountofWaterGivenDuringtheYear** = Total amount of water provided for all services(individual, corporate) throughout the year according to official figures
- **TotalInput** = [PumpWater] + [TotalAnnualRain] Total amount of fresh water coming to the city by pumps and precipitation, official records.
- **Net** = [TotalInput] - [TotalAmountofWaterGivenDuringtheYear]

All water values have been found by dividing 10 million cubic meters. Data standardization has thus been achieved.

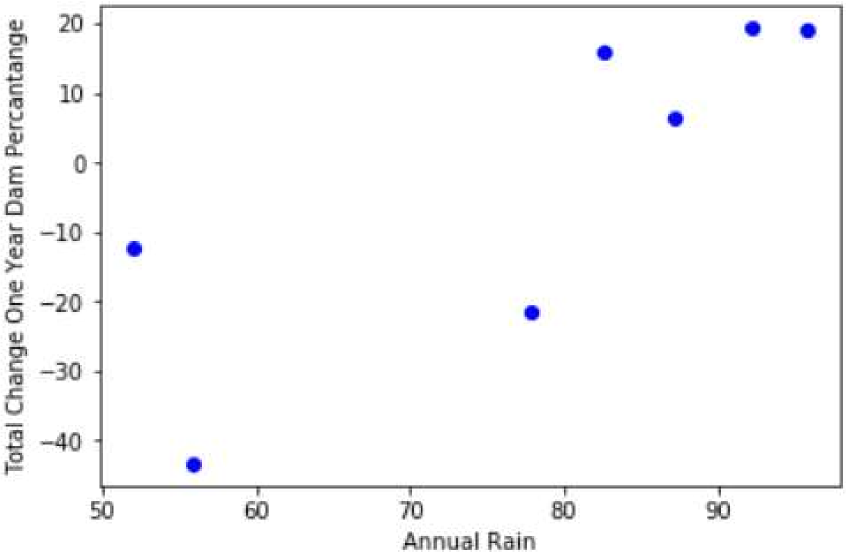


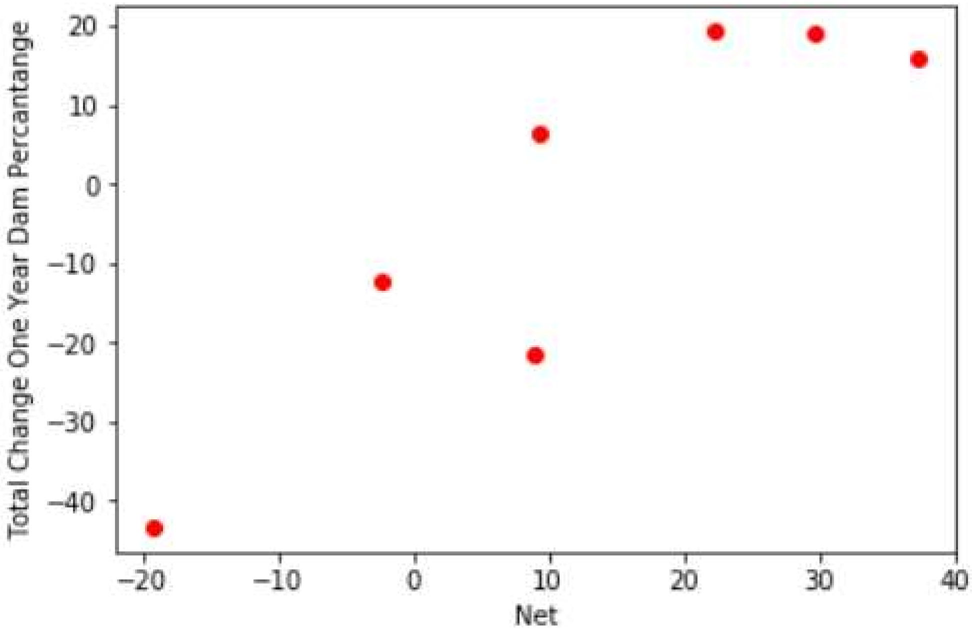
	Year	Percentage	TotalAnnualRain	TotalChangeOneYear	TotalAmountofWaterGivenDuringtheYear	PumpWater	TotalInput	Net
0	21.12.2014	53.88	82.63	15.80	92.44	47.04	129.67	37.23
1	21.12.2015	60.24	87.24	6.36	96.51	18.58	105.82	9.31
2	21.12.2016	38.85	77.81	-21.39	99.86	31.09	108.90	9.04
3	21.12.2017	58.17	92.23	19.32	102.96	33.96	125.19	22.23
4	21.12.2018	77.29	95.74	19.12	104.09	37.97	133.71	29.62
5	21.12.2019	33.97	55.87	-43.32	106.17	31.19	87.06	-19.11
6	21.12.2020	21.70	52.09	-12.27	102.90	47.77	100.56	-2.34



While the annual rainfall should be proportional to the dam percentage, some irregularities strike us. Of course, the reasons for this may be the water pumped from the surrounding provinces, low consumption or the factors that we cannot control.

It would be best to build this algorithm on the city's net water balance for 1 year and the percentage change in the 1-year dams.



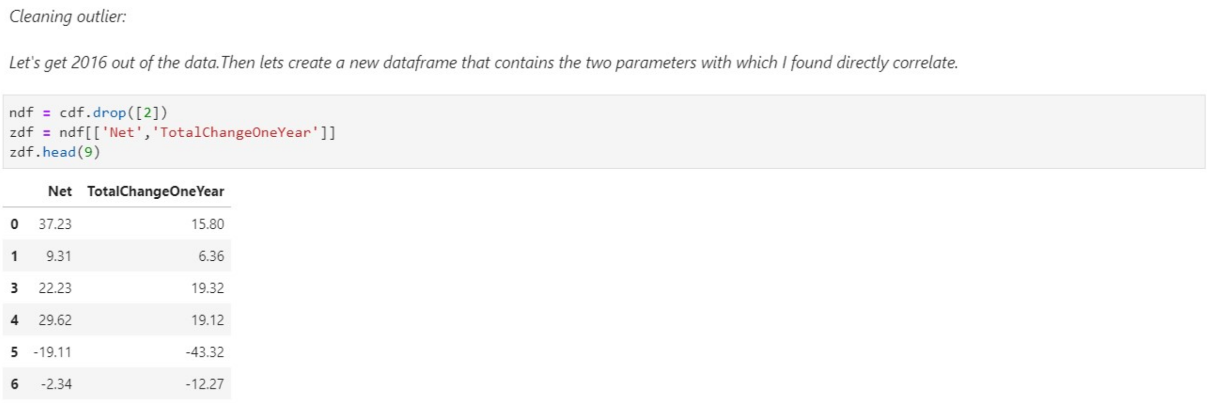


There are two main problems with the chart.

- The first is an outlier in 2016(Net+10): A mistake in the official figures may be due to malfunctions, losses or factors that we cannot calculate. So I'm going to take out 2016 while I'm designing artificial intelligence.

- The second important problem is the slowdown in dam occupancy rates when the net water inflow to the city exceeds 30. This again can be related to various problems. Maybe it's an annual thing.

- But the most logical theory that comes to mind may be that more water ingress than the dams of the city can store makes no sense. Yes, water is coming, but it cannot be stored. Of course, there may be unfamiliar factors and errors in my measurement. However, this is not a big outlier. The net water inflow has rarely exceeded 30 in the last 20 years. It does not prevent our algorithm working.



Cleaning outlier:

Let's get 2016 out of the data.Then lets create a new dataframe that contains the two parameters with which I found directly correlate.

```
ndf = cdf.drop([2])
zdf = ndf[['Net','TotalChangeOneYear']]
zdf.head(9)
```

	Net	TotalChangeOneYear
0	37.23	15.80
1	9.31	6.36
3	22.23	19.32
4	29.62	19.12
5	-19.11	-43.32
6	-2.34	-12.27

C.2 Use Of Supervised Neural Network Models - Multi-layer Perceptron For Predictions

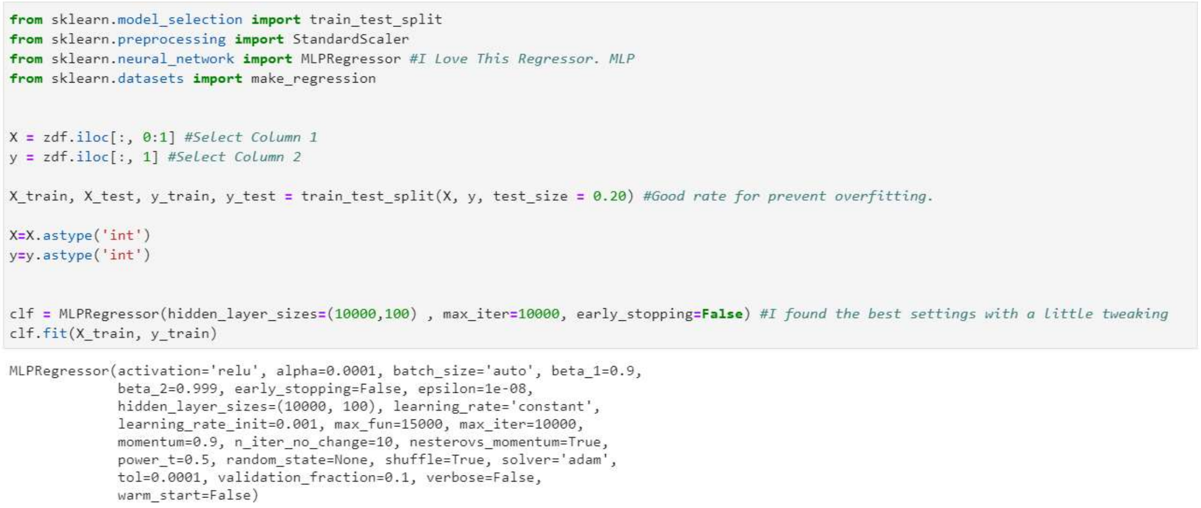
I will be honest. I had a hard time finding the algorithm that could most accurately cover and predict the relevant data.

It was obvious that Linear-Regression was not going to work. Because when there are more than 30 net water inflows, our dam percentage does not increase proportionally. There are also deviations in the data.

I tried logorithmic and sigmoid functions in non-linear regression. However, when the net water intake was negative, none of them could provide the arithmetic decrease.

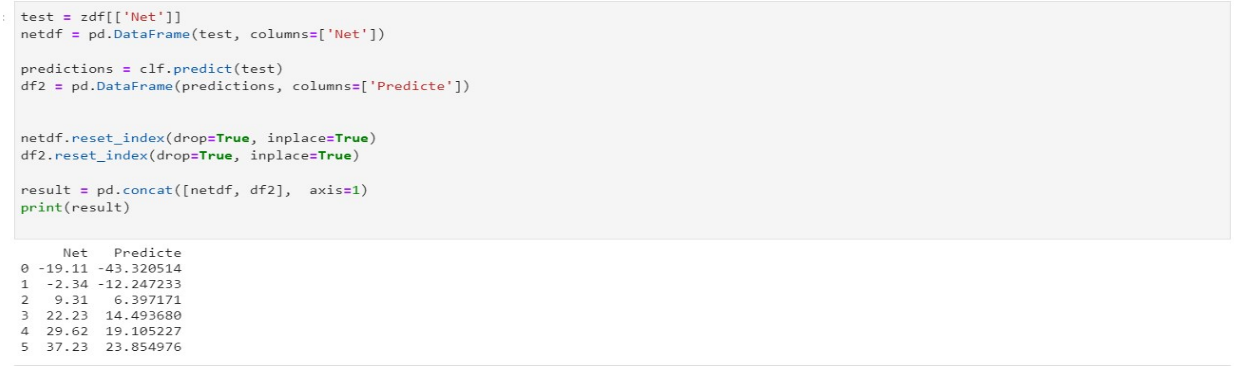
Then I used Multi-layer Perceptron, a simple neural network application. I have tried various test - train ratios. This algorithm provided the most accurate results.

Don't worry, I'll prove it with a graph.



```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPRegressor #I Love This Regressor. MLP
from sklearn.datasets import make_regression


X = zdf.iloc[:, 0:1] #Select Column 1
y = zdf.iloc[:, 1] #Select Column 2

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20) #Good rate for prevent overfitting.

X=X.astype('int')
y=y.astype('int')


clf = MLPRegressor(hidden_layer_sizes=(10000,100) , max_iter=10000, early_stopping=False) #I found the best settings with a little tweaking
clf.fit(X_train, y_train)
```

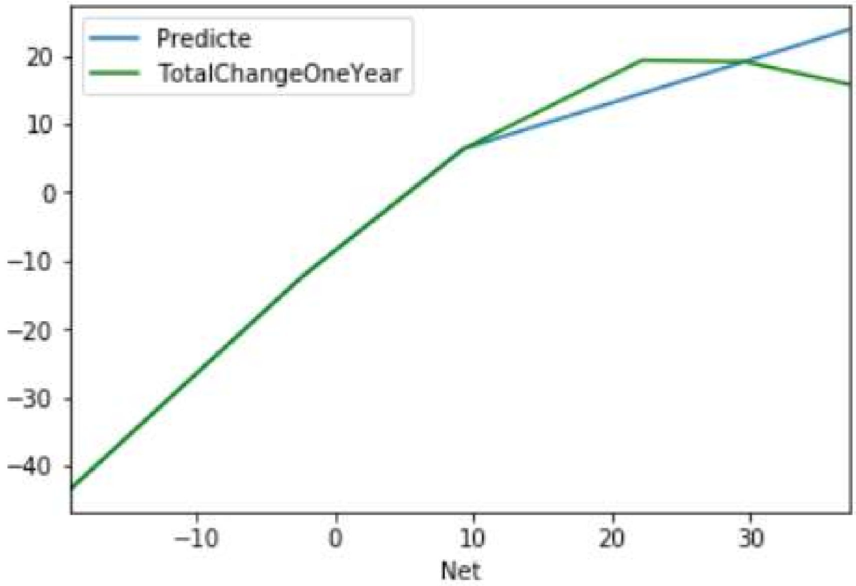
```
MLPRegressor(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9,
             beta_2=0.999, early_stopping=False, epsilon=1e-08,
             hidden_layer_sizes=(10000, 100), learning_rate='constant',
             learning_rate_init=0.001, max_fun=15000, max_iter=10000,
             momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
             power_t=0.5, random_state=None, shuffle=True, solver='adam',
             tol=0.0001, validation_fraction=0.1, verbose=False,
             warm_start=False)
```



```
test = zdf[['Net']]
netdf = pd.DataFrame(test, columns=['Net'])

predictions = clf.predict(test)
df2 = pd.DataFrame(predictions, columns=['Predicte'])


netdf.reset_index(drop=True, inplace=True)
df2.reset_index(drop=True, inplace=True)

result = pd.concat([netdf, df2],  axis=1)
print(result)
```

```
     Net   Predicte
0 -19.11 -43.320514
1  -2.34 -12.247233
2   9.31   6.397171
3  22.23  14.493680
4  29.62  19.105227
5  37.23  23.854976
```

Results:



In this graph, We can see that 2 data were taken for the test of the algorithm. However, we can still see that he made very small estimates with ~ 5 errors for those values. If he had taken the last data for education, it would decrease the percentage change as the amount of water increased. This is like an impossible thing in real life. Yes, the dams can reach the saturation point, but they do not fall to negative just because there is too much rain. So this version of the Algorithm is pretty good. Anyway, a total net water inflow of 30 - 40 above is not possible under these conditions.

When the Net Water balance drops to negative, the algorithm gives excellent results even with 3 training sets. We can say that he is an expert estimator about the rate of water decrease. So let's use this model to predict the water problem of Istanbul.

Predictions:

- ✓ Occupancy rate of Istanbul dams as of 21.12.2020: 21.66
- ✓ If our net water balance is 0 next year, our Dam occupancy rate on 21.12.2021: [14.27300521]
- ✓ If the next year will pass exactly like this year , our Dam occupancy rate on 21.12.2021: [9.41276697]
- ✓ If we can achieve water balance as in 2014, the rainiest time of the last 6 years, our Dam occupancy rate on 21.12.2021: [45.99567619]
- ✓ If we can achieve water balance as in 2010, the rainiest time of the last 10 years, our Dam occupancy rate as of 21.12.2021: [52.240123]
- ✓ If so much water consumption and low rainfall like in 2019 `the worst water balance of the last 10 years`, our Dam occupancy rates as of 21.20.2021: [-21.45358778]
- ✓ If the next year will pass exactly like this year , but if everyone saves 10% water, our Dam occupancy rate on 21.12.2021: [27.10021104]

D. Discussion

At the beginning of my study, I checked with the foursquare api the dam coordinates. The results were pretty accurate. Only one dam was marked as "Campground" instead of "Lake". Then I visualized the related water resources on the map.

In the second part of my study, I established a link between the annual amount of net water entering the city and the annual dam occupancy rate change. Classical regression was somewhat inadequate in this regard. Because dams have a saturation point. However, I found very close results in test data with MLP.

However, this is a project prepared in a short time. For example, water leaks in a municipality can deviate 20% of the total water use. Let's even think a little deeper. How many days are sunny in a year can affect the amount of evaporation. Undoubtedly, this will affect the annual percentage change.

In addition, the majority of water use is made by agricultural lands. (Approximately 50 percent of water consumption in Istanbul). In this case, other factors such as the water need of the planted plants, soil yield, number of snowy days may affect the net water balance.

However, I still managed to forecast on an annual basis with the city's net water balance.

Maybe we cannot affect the annual rainfall directly. However, water consumption may affect the net water balance shifting to a large positive.

In the last part of the study, I made a presentation to raise public awareness. Let's say there is as much rain in 2021 as in 2020 and water intake from the surrounding provinces. However, everybody use their water 10% sparingly throughout 2021. In this case, the total water consumption will decrease by 1/10. This leads to an increase of 10.20 in the net water balance. In 2020, the net water balance was -2.34. With 10% reduced water consumption our net water balance for 2021 would be +7.88. In this case, we will see that our dams are 27% in 21.12.2021 . Who wouldn't want their dams to be full!

F. Conclusion

As a result, data science helps us at many points in everyday life. For example, even visualizing the dam data we collect from the internet can make people aware that "my water comes from here, they are not too big, I should have been careful".

In addition, according to the estimates we have calculated with MLP, the occupancy rate of our dams can increase by 13% if each person consumes 10% less water, while remaining the same. For exactly this reason, we must use our water very carefully. It is wrong to say that nothing will happen with my consumption. Algorithms have proven this.

Global warming and climate change forces us to use our water resources more rationally. Detailed usage details can be obtained with data science projects. Even, the leak water in the city can be detected and where dam investments can be evaluated.

References:

[1] Istanbul - Britannica

[2] Istanbul Water Department Official Site

[3] Forsquare API

[4] Istanbul Dam Occupancy Rates

[5] Google Map

[6] Falling Water Level In Istanbul Dams Increases Risk Of Disease