Project report

Manufacturing Data Collection and Analytics

National Research University

Higher School of Economics

Option II

Performed by students

Ajay Mishra

Shipon Nath

Koushik

Project supervisor: KOVALEV ILYA ALEXANDROVICH

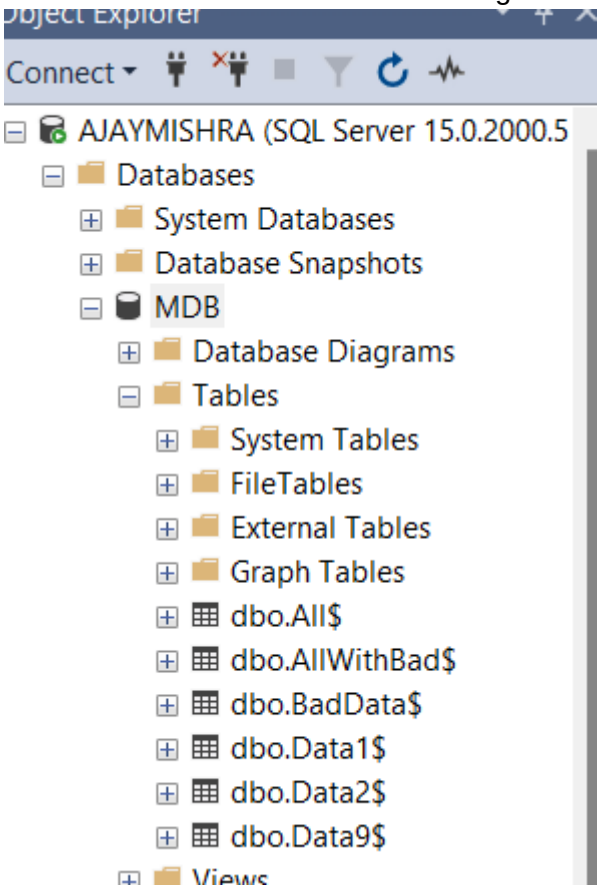
Table OF Content:

Task: -	**Done By: -**
Create DB- **Load All Data-** **Query 1, 2-**	Ajay Mishra
Load xml file in Colab- **Visualize the Data-** **Analysis the data-**	Shipon nath
Run the xml file with Python-	Kawshiqe Ahamed

Creation of DB-

-Created a data base including bad and good data, uploaded a file in Microsoft SQL, Management Server DB name is MDB

Created Database with bad file and 3 good files



Data Manipulation to understand about bad data -

- Took the 3 good files and find out how many location data are matching with correct location in bad file and found 308 locations are correct in bad data
-

```
select Count (Distinct BadData$.Location) as Count bad Data location from
BadData$
left join All$ ON BadData$.customer =All$.customer and BadData$.location
=All$.location

where BadData$.Location=All$.location
```

-

SQLQuery1.sql - A...YMISHRA\User (56))*

```
select Count (Distinct BadData$.Location)  as CountbadDataLocation from BadData$
left join All$ ON BadData$.customer =All$.customer and BadData$.location =All$.location

where BadData$.Location=All$.location
```

100 %

Results | Messages

	CountbadDataLocation
1	308

SQLQuery1.sql - A...YMISHRA\User (56))*

```
select BadData$.Location as LocationNames  from BadData$
left join All$ ON BadData$.customer =All$.customer and BadData$.location =All$.location

where BadData$.Location=All$.location
```

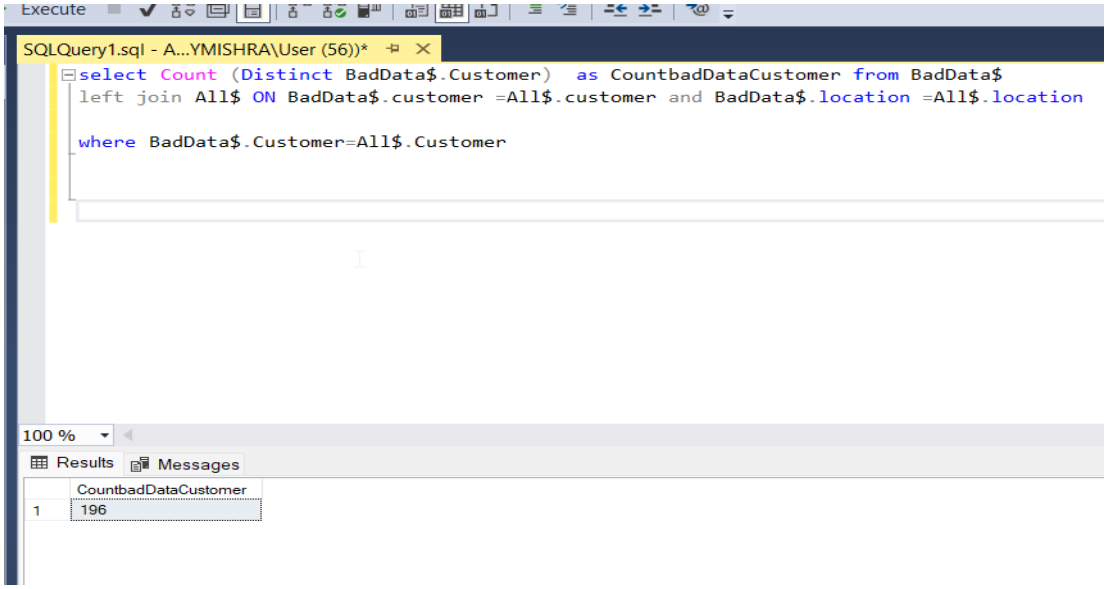
100 %

Results | Messages

	LocationNames
1	Bosch#14
2	Bosch#14
3	Bosch#14
4	Bosch#14
5	Bosch#17
6	Bosch#17
7	Bosch#19
8	Bosch#19
9	Bosch#19
10	Bosch#19
11	Bosch#20
12	Bosch#20
13	Bosch#20
14	Bosch#20

Query executed successfully.

196 data are correct in customer column in bad file while comparing with 3 good files

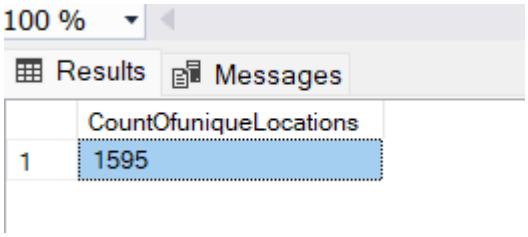


Answers of tasks with respect to SQL and their screenshot

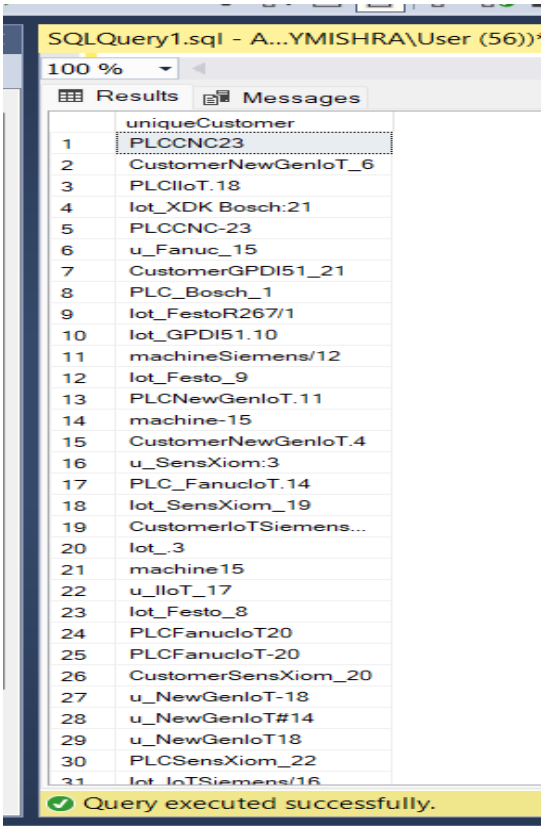
Took 3 tables and based on 3 tables answers these task

Questions1 - **Count the number of unique Locations**

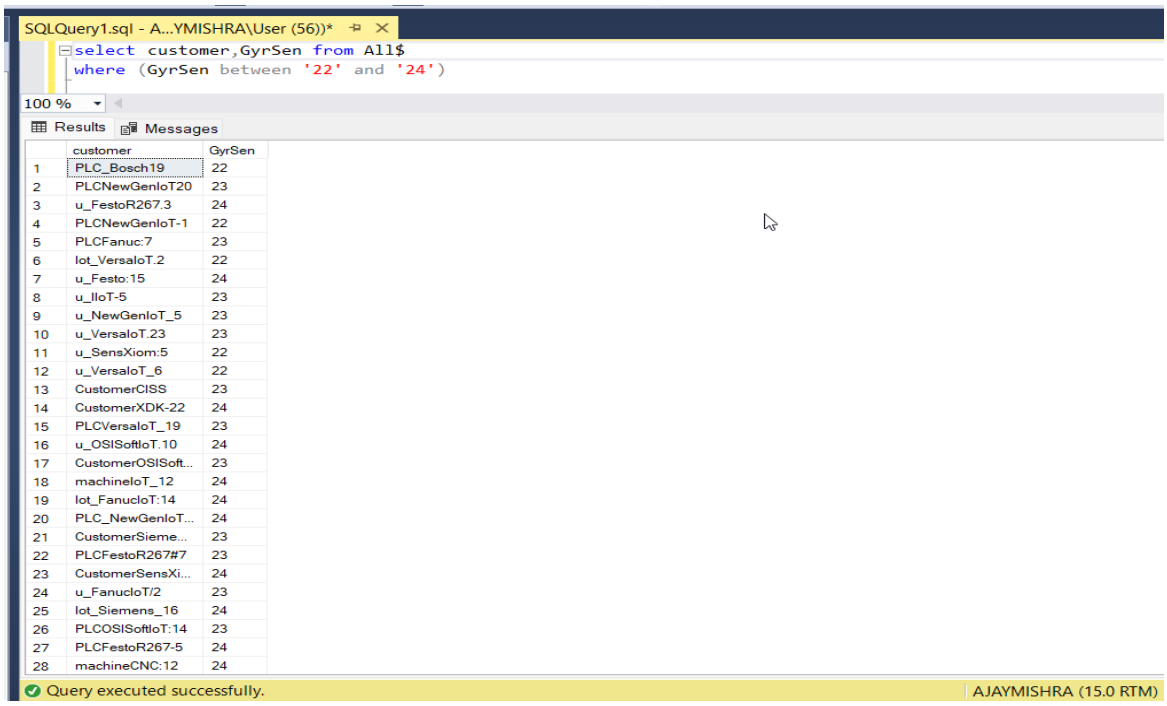
1-Count of Unique Locations - 1595



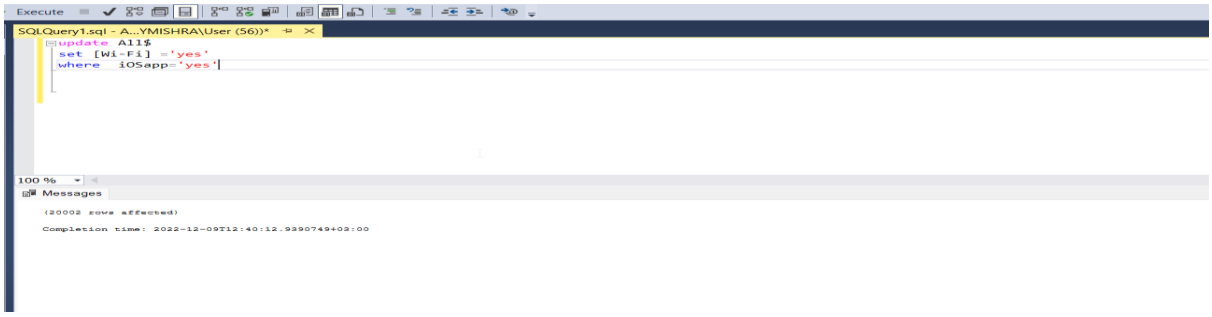
2-List Of Unique customer_- Count- 14134

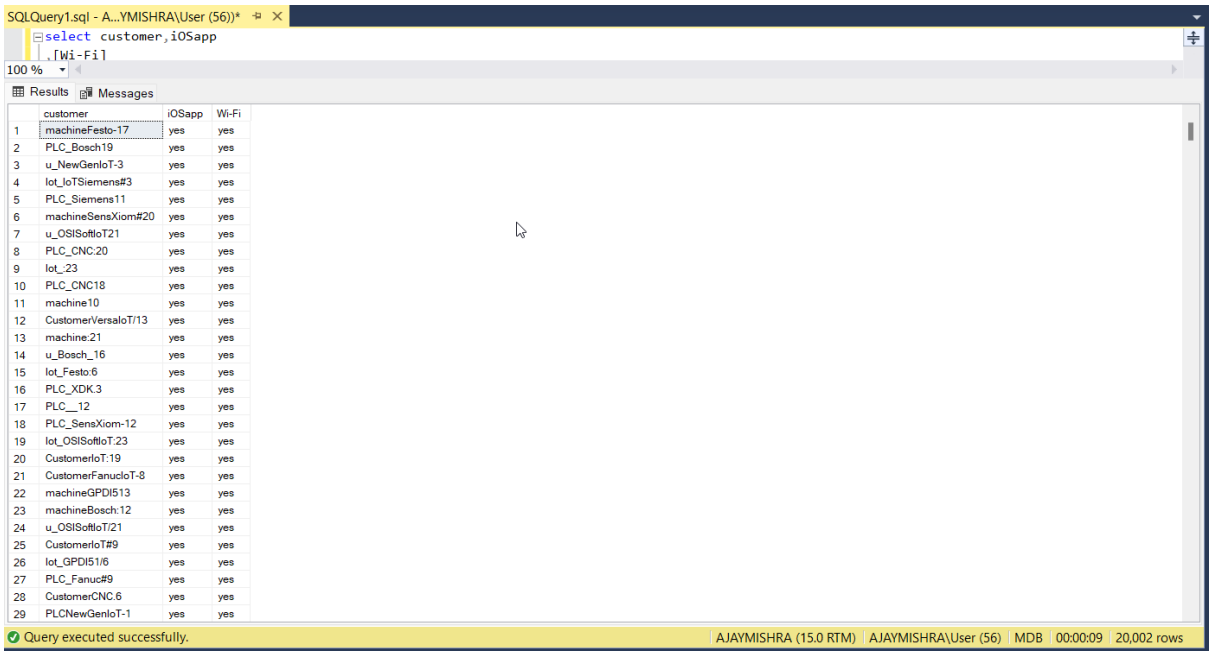


3- Find all Customers with GyrSen in the range between 22-24

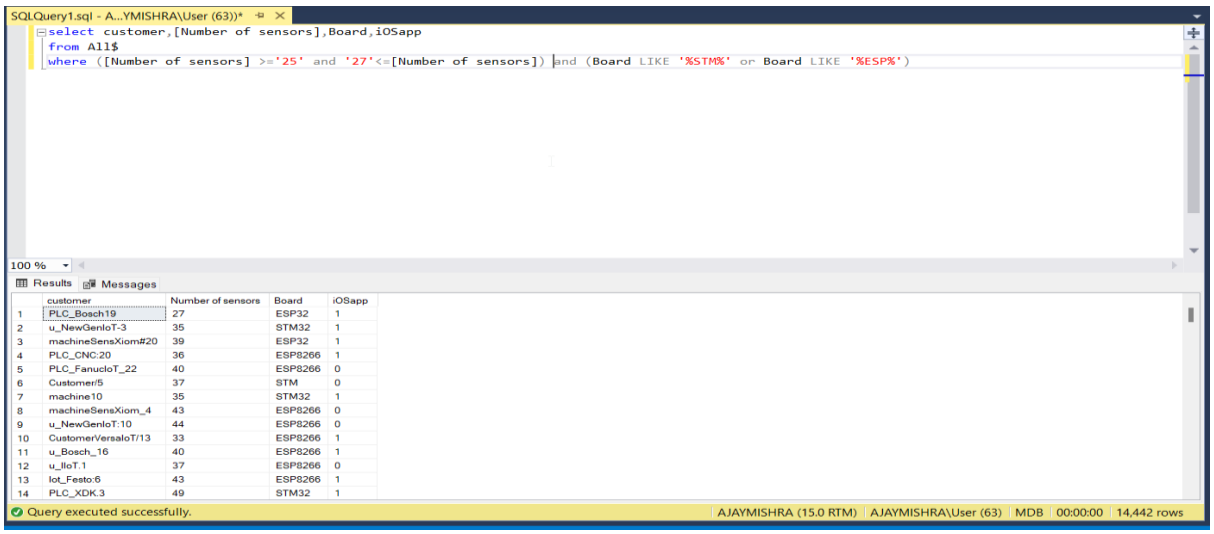


4-Find all ios app Customers that do not support wi-fi and update them to "yes"





5-Find all Customers with a Number of sensors in the range between 25-27, and for all devices with STM or ESP, regardless of digits further instead of Yes in the iOS App column display "1", and for No - "0" (Count of Customers- 14,442)



Run xml files with python

```
[3] import pandas_read_xml as pdx
    from pandas_read_xml import flatten, fully_flatten, auto_separate_tables
```

```
df = pdx.read_xml(text_xml, ['root'])

df
```

	@xmlns:xsi	@Comment	@IndraWorksVersion
0	http://www.w3.org/2001/XMLSchema-instance	Persistence file of IndraWorks Oscilloscope	14.22.1035.0

```
[9] df = df.pipe(flatten)

    df
```

	@xmlns:xsi	@Comment	@IndraWorksVersion	@CompatibilityVersion	@xmlns
0	http://www.w3.org/2001/XMLSchema-instance	Persistence file of IndraWorks Oscilloscope	14.22.1035.0	10.6	http://www.boschrexroth.de/indraworks.textform...

1 rows × 29 columns

```
[14] df = df.pipe(flatten)

     df
```

	@xmlns:xsi	@Comment	@IndraWorksVersion	@CompatibilityVersion	@xmlns
0	http://www.w3.org/2001/XMLSchema-instance	Persistence file of IndraWorks Oscilloscope	14.22.1035.0	10.6	http://www.boschrexroth.de/indraworks.textform...
1	http://www.w3.org/2001/XMLSchema-instance	Persistence file of IndraWorks Oscilloscope	14.22.1035.0	10.6	http://www.boschrexroth.de/indraworks.textform...
2	http://www.w3.org/2001/XMLSchema-instance	Persistence file of IndraWorks Oscilloscope	14.22.1035.0	10.6	http://www.boschrexroth.de/indraworks.textform...
3	http://www.w3.org/2001/XMLSchema-instance	Persistence file of IndraWorks Oscilloscope	14.22.1035.0	10.6	http://www.boschrexroth.de/indraworks.textform...

4 rows × 85 columns

```
[18] key_columns = ['@xmlns:xsi']
     data1 = df.pipe(auto_separate_tables, key_columns)

[19] data1.keys()
     dict_keys(['Devices|Device|Measurements|Measurement|Signals|Signal|GraphicProperties|GraphicSignalProperties|PredefinedBitDescriptions|BitDescriptions',
     'Devices|Device|Measurements|Measurement|Signals|Signal|GraphicProperties|GraphicSignalProperties|UserdefinedBitDescriptions|BitDescriptions', 'OscilloscopeData'])

[20] data1['Devices|Device|Measurements|Measurement|Signals|Signal|GraphicProperties|GraphicSignalProperties|PredefinedBitDescriptions|BitDescriptions']
```

	@xmlns:xsi	@id
0	http://www.w3.org/2001/XMLSchema-instance	ref-9
1	http://www.w3.org/2001/XMLSchema-instance	ref-13
2	http://www.w3.org/2001/XMLSchema-instance	ref-17
3	http://www.w3.org/2001/XMLSchema-instance	ref-21

```
[21] data1['Devices|Device|Measurements|Measurement|Signals|Signal|GraphicProperties|GraphicSignalProperties|UserdefinedBitDescriptions|BitDescriptions']
```

	@xmlns:xsi	@id
0	http://www.w3.org/2001/XMLSchema-instance	ref-10
1	http://www.w3.org/2001/XMLSchema-instance	ref-14
2	http://www.w3.org/2001/XMLSchema-instance	ref-18
3	http://www.w3.org/2001/XMLSchema-instance	ref-22

```
[22] data1['OscilloscopeData']
```

	@xmlns:xsi	@Comment	@IndraWorksVersion	@CompatibilityVersion	@xmlns	@id	Devices|@xsi:type
0	http://www.w3.org/2001/XMLSchema-instance	Persistence file of IndraWorks Oscilloscope	14.22.1035.0	10.6	http://www.boschrexroth.de/indraworks.textform...	ref-1	Array
1	http://www.w3.org/2001/XMLSchema-instance	Persistence file of IndraWorks Oscilloscope	14.22.1035.0	10.6	http://www.boschrexroth.de/indraworks.textform...	ref-1	Array
2	http://www.w3.org/2001/XMLSchema-instance	Persistence file of IndraWorks Oscilloscope	14.22.1035.0	10.6	http://www.boschrexroth.de/indraworks.textform...	ref-1	Array
3	http://www.w3.org/2001/XMLSchema-instance	Persistence file of IndraWorks Oscilloscope	14.22.1035.0	10.6	http://www.boschrexroth.de/indraworks.textform...	ref-1	Array

4 rows × 83 columns

```
[23] print(type(data1['OscilloscopeData']))
```

```
<class 'pandas.core.frame.Dataframe'>
```

```
pd = data1['OscilloscopeData']
```

```
print(pd.shape)
```

```
(4, 83)
```

```
[26] pd.describe()
```

	@xmlns:xsi	@Comment	@IndraWorksVersion	@CompatibilityVersion	@xmlns	@id	Devices|@xsi:type	Devices|@id	Computations|@xsi:type	Computat
count	4	4	4	4	4	4	4	4	4	
unique	1	1	1	1	1	1	1	1	1	
top	http://www.w3.org/2001/XMLSchema-instance	Persistence file of IndraWorks Oscilloscope	14.22.1035.0	10.6	http://www.boschrexroth.de/indraworks.textform...	ref-1	Array	ref-2	Array	
freq	4	4	4	4	4	4	4	4	4	

4 rows × 83 columns

```
[29] xval = pd["Devices|Device|Measurements|Measurement|Signals|Signal|XValue"]
```

```
[30] xval.to_frame()
```

1 to 4 of 4 entries Filter

index	
0	-409|-408|-407|-406|-405|-404|-403|-402|-401|-400|-399|-398|-397|-396|-395|-394|-393|-392|-391|-390|-389|-388|-387|-386|-385|-384|-383|-382|-381|-380|-379|-378|-377|-376|-375|-374|-373|-372|-371|-370|-369|-368|-367|-366|-365|-364|-363|-362|-361|-360|-359
1	-409|-408|-407|-406|-405|-404|-403|-402|-401|-400|-399|-398|-397|-396|-395|-394|-393|-392|-391|-390|-389|-388|-387|-386|-385|-384|-383|-382|-381|-380|-379|-378|-377|-376|-375|-374|-373|-372|-371|-370|-369|-368|-367|-366|-365|-364|-363|-362|-361|-360|-359
2	-409|-408|-407|-406|-405|-404|-403|-402|-401|-400|-399|-398|-397|-396|-395|-394|-393|-392|-391|-390|-389|-388|-387|-386|-385|-384|-383|-382|-381|-380|-379|-378|-377|-376|-375|-374|-373|-372|-371|-370|-369|-368|-367|-366|-365|-364|-363|-362|-361|-360|-359
3	-409|-408|-407|-406|-405|-404|-403|-402|-401|-400|-399|-398|-397|-396|-395|-394|-393|-392|-391|-390|-389|-388|-387|-386|-385|-384|-383|-382|-381|-380|-379|-378|-377|-376|-375|-374|-373|-372|-371|-370|-369|-368|-367|-366|-365|-364|-363|-362|-361|-360|-359

Show 25 per page
Like what you see? Visit the data table notebook to learn more about interactive tables

```
[31] yval = pd["Devices|Device|Measurements|Measurement|Signals|Signal|YValue"]
```

```
yval.to_frame()
```

1 to 4 of 4 entries Filter

index	
0	-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-
1	-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-9.9999|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-10|-
2	2.441|-7.324|2.441|-7.324|2.441|-7.324|0|-7.324|2.441|2.441|-7.324|-7.324|2.441|-7.324|-7.324|2.441|-4.883|2.441|2.441|-9.766|2.441|-7.324|4.883|2.441|-7.324|-9.766|2.441|2.441|2.441|2.441|0|-2.441|-7.324|2.441|0|2.441|-7.324|-2.441|2.441|2.441|2.441|-7.32
3	0.322|0.301|0.238|0.345|0.330|0.235|0.319|0.346|0.365|0.34|0.254|0.433|0.45|0.335|0.368|0.244|0.273|0.265|0.35|0.337|0.301|0.355|0.253|0.344|0.353|0.244|0.368|0.375|0.337|0.261|0.351|0.258|0.392|0.371|0.375|0.347|0.357|0.261|0.46|0.238|0.355|0.277|0.3

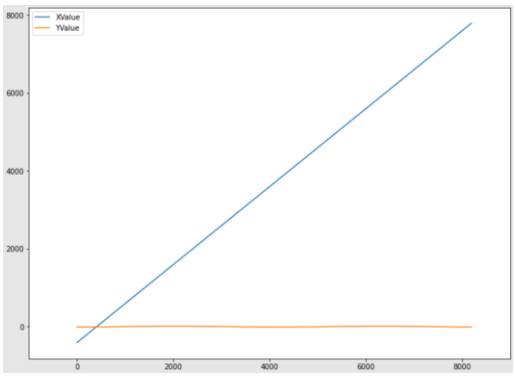
Analysis The Data;

For Analysis of the xml files, we use python. And we did this work on a Colab platform. In these 5 xml files there was the data received from the motor. There we got four types of parameters. 1) Position command value 2) Position feedback value 3) Velocity feedback value 4) Actual output current value (Absolute value). And every file we got 2 values. Data of X- Time Ms and Y - value of parameter.

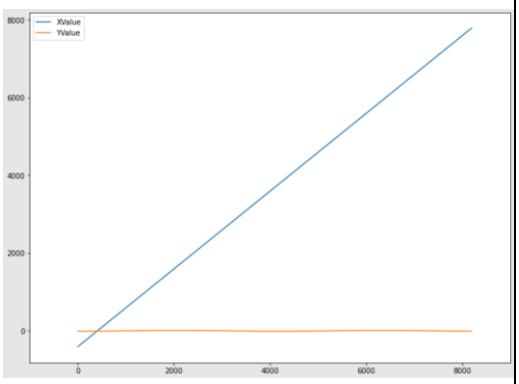
Using python tried to understand the values of X and Y through a graph. Here are all the Graphs that we got by using python.

Position Command Value:

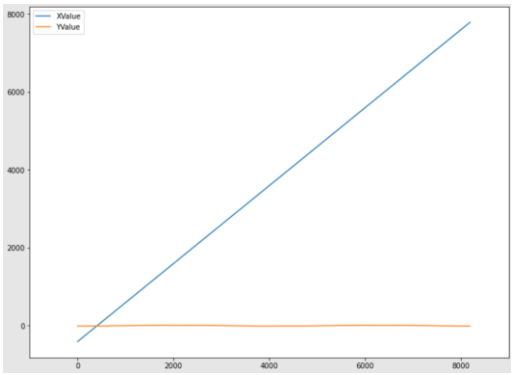
Xml file 1



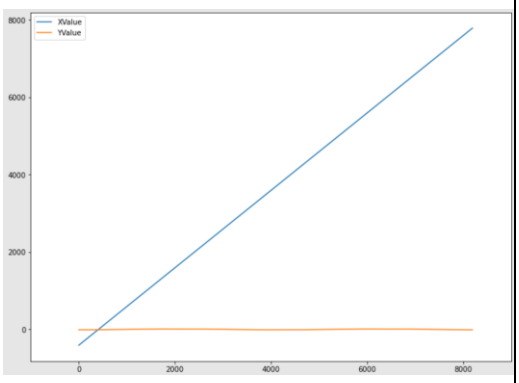
Xml file 3



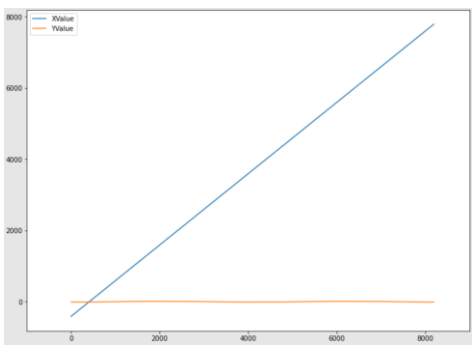
Xml file 2:



Xml file 4:

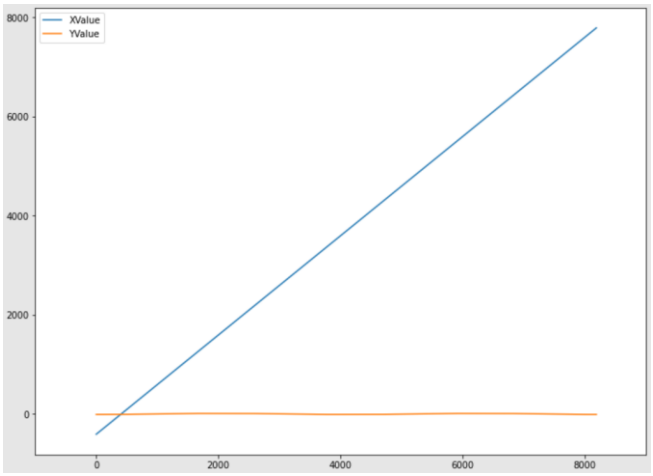


Xml file 5:

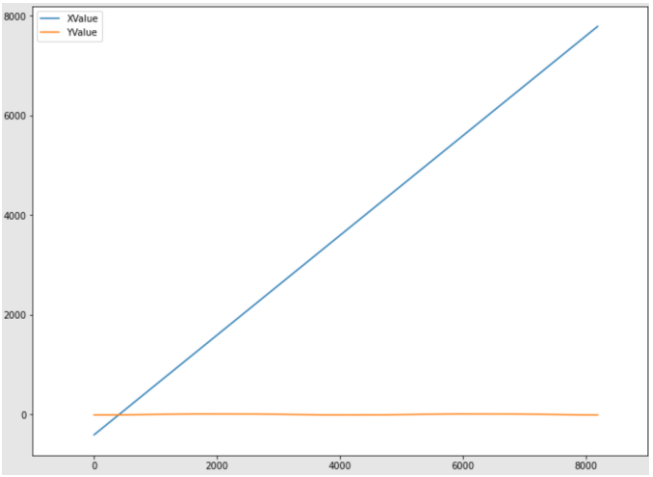


Position feedback value:

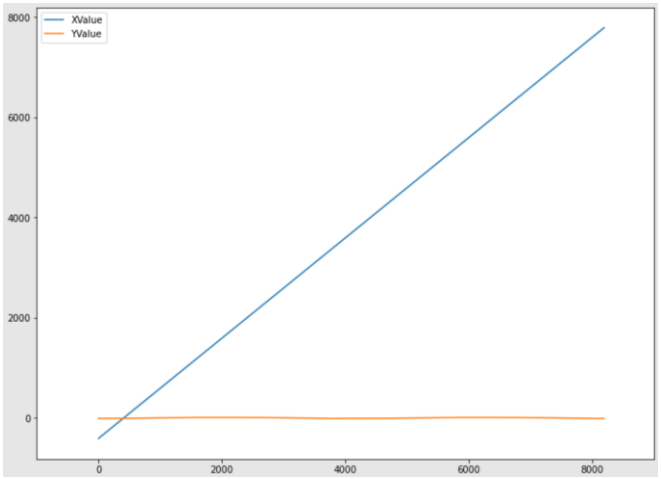
Xml file 1:



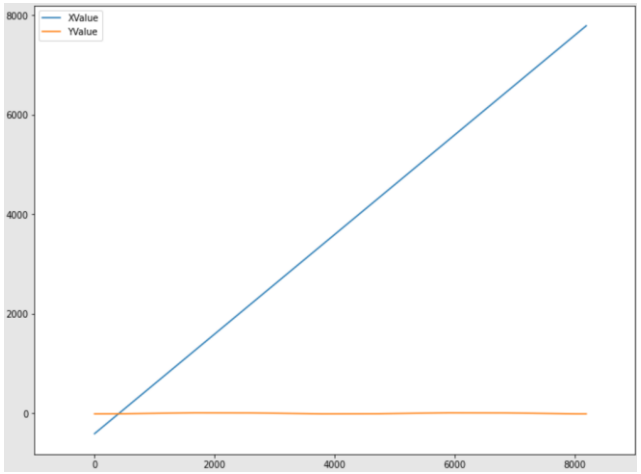
Xml file 2:



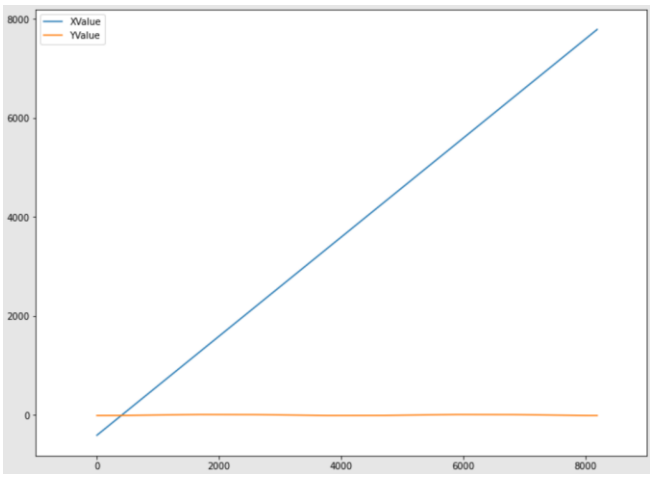
Xml file 3:



Xml file 4:

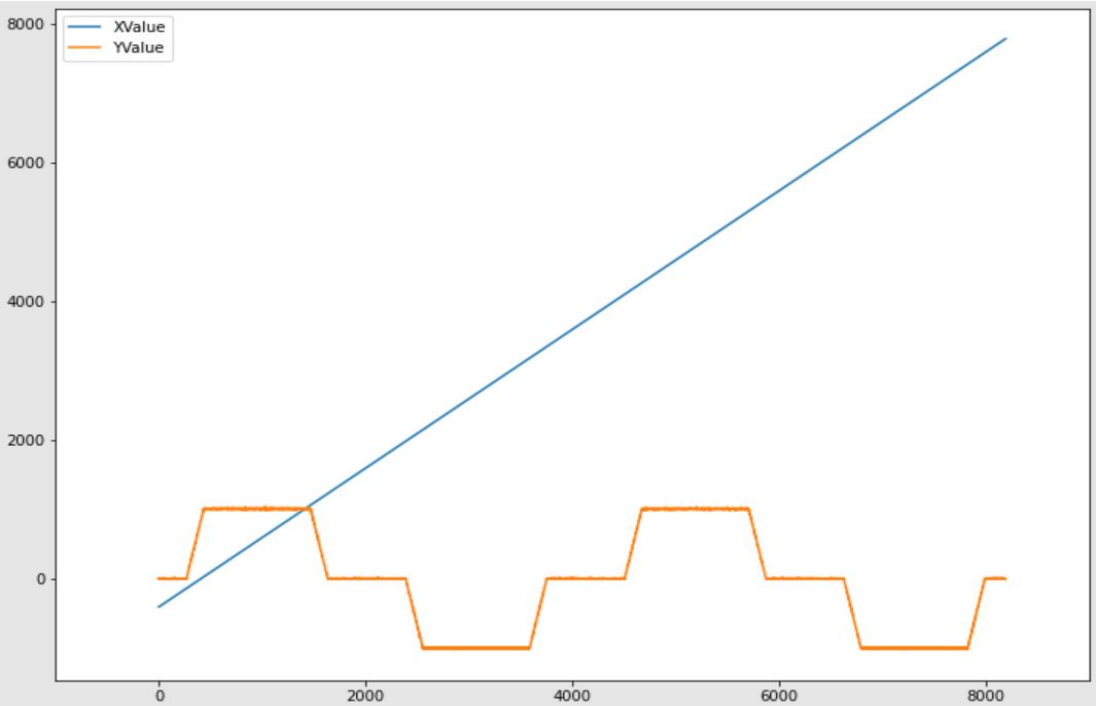


Xml file 5:

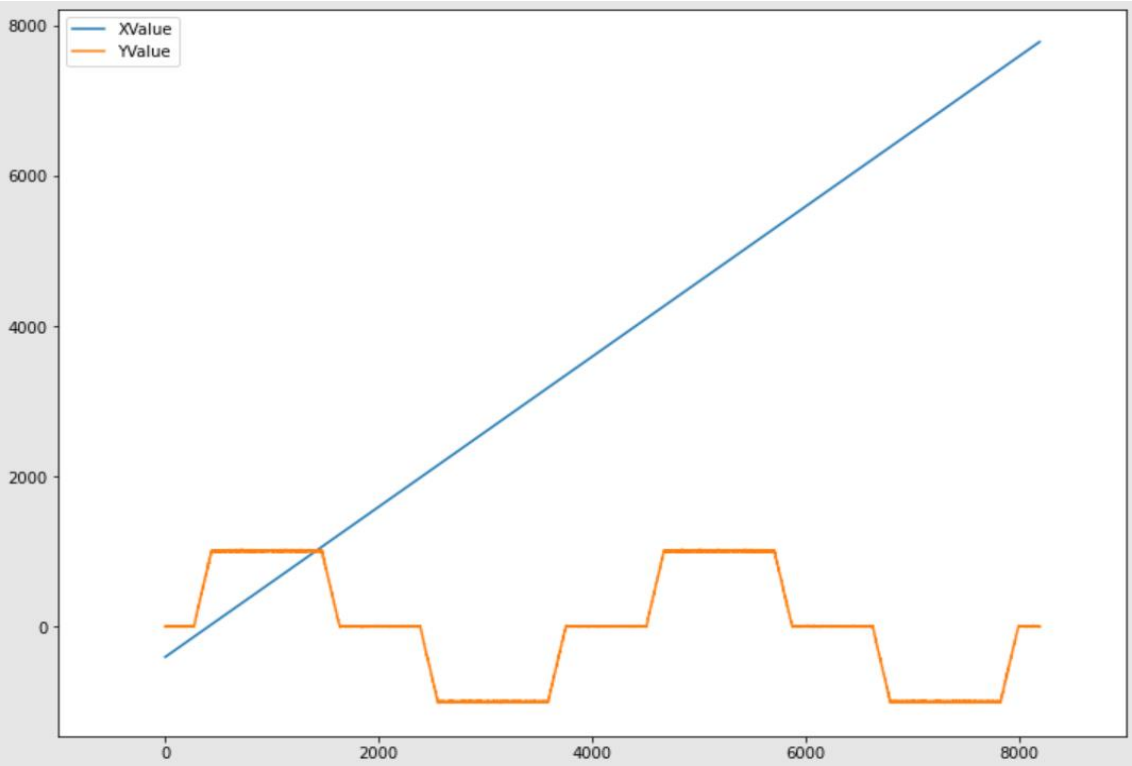


Velocity Feedback value:

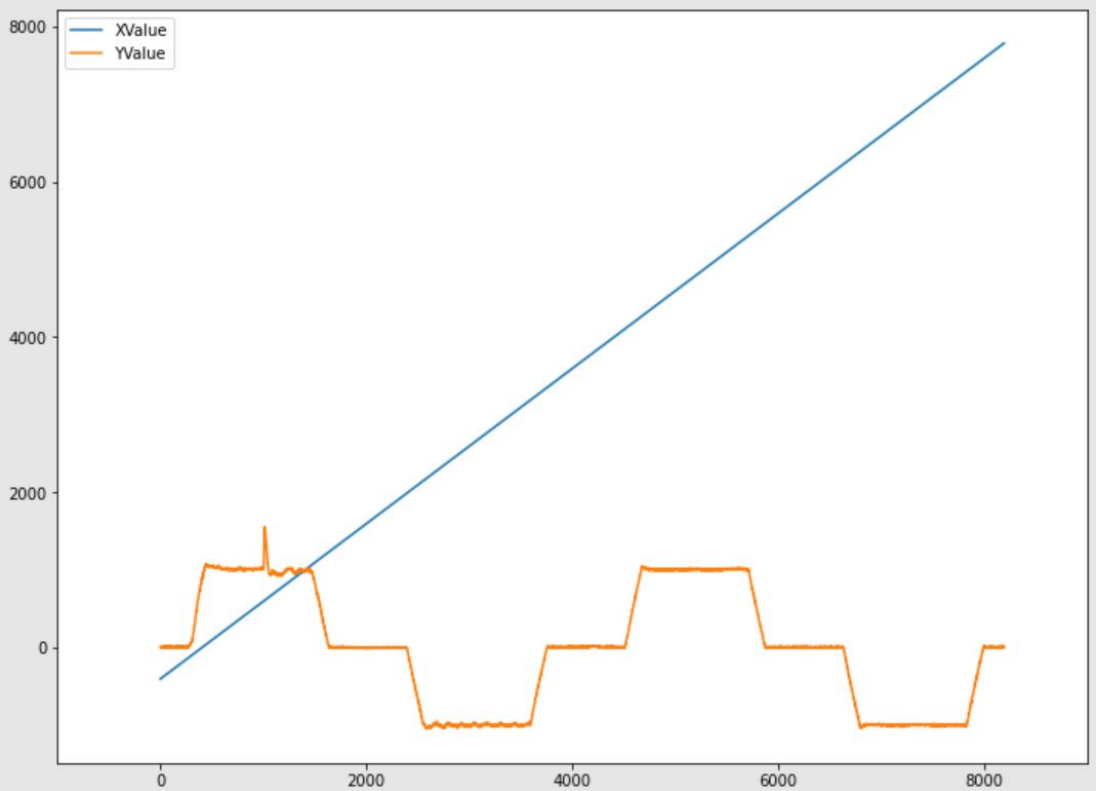
Xml file1:



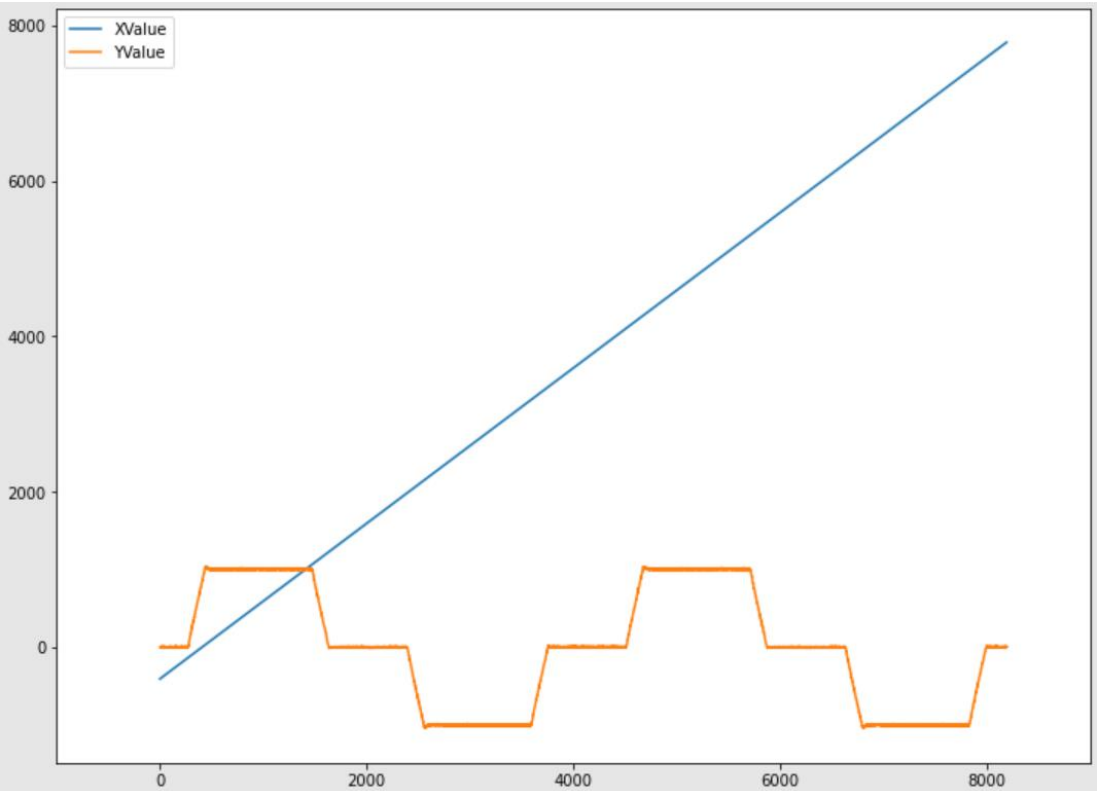
Xml file 2



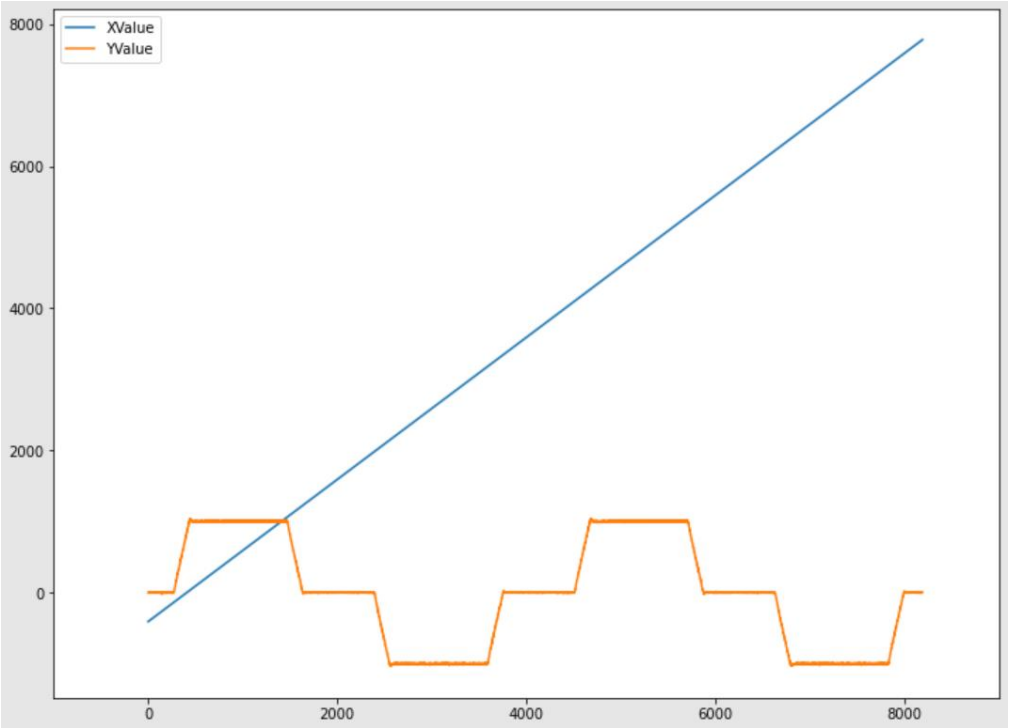
Xml file3



Xml file 4:

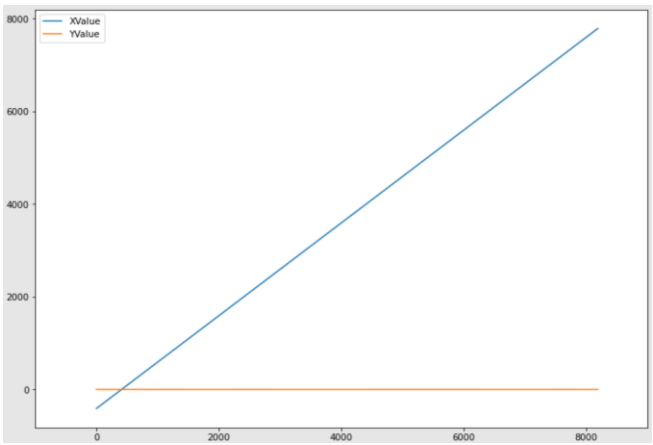


Xml file 5:

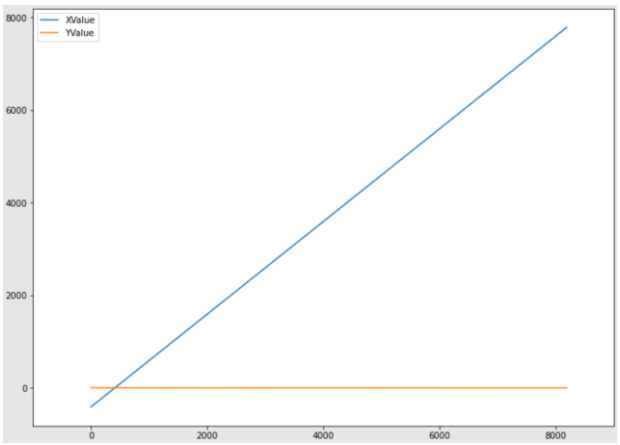


Actual output current value:

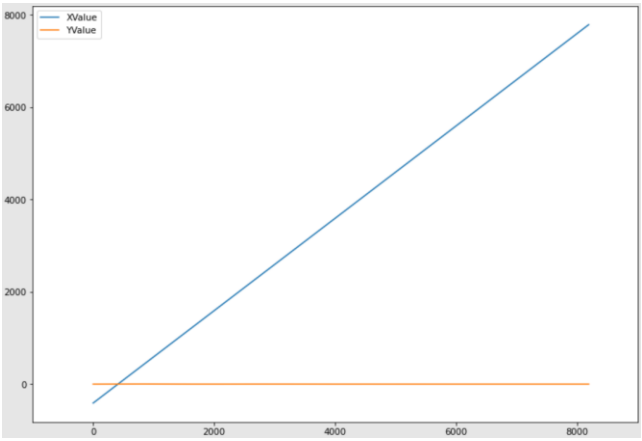
Xml file 1:



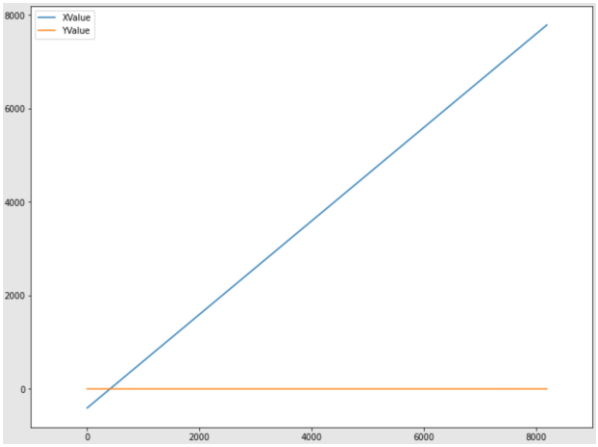
Xml file 2:



Xml file 3:



Xml file 4:



Xml file 5: