Final Report for MCI project (Scripted Data Cleaning)

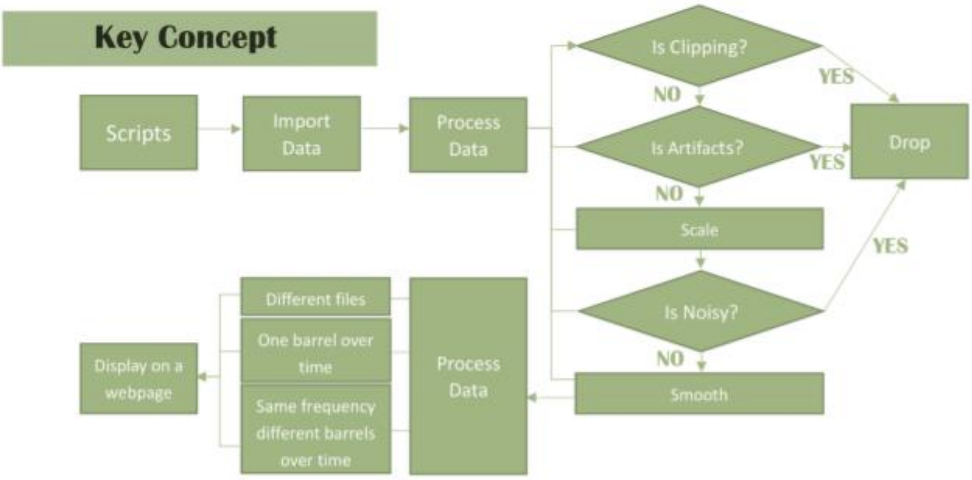
A1742398 Xiangnan Zhang

1. Introduction

Nowadays, machine learning becomes more and more popular, for example, face detection, voice modifying, auto-translation are widely used in our daily life. However, in order to get a more accurate result, we must clean the data which doesn't obey the rules or just recorded falsely, therefore we could use them in the train set of machine learning. Here comes the question, how can we pick up the data unwanted rapidly or transform them into a format which obeys the rules in a large set of data? Well, in this project, we created the "Scripted Data Cleaning", people could use our written scripts to filter the data and transform them, then use webpage to check the data which is after processed. More accurately, I finished the most of cleaning part by python scripts and my teammate realized most of visualization by chartjs. What's more, this project enable people inspect one specific item of data, which is quite awesome.

2. Project Aims

In this project, we make a corporation with a company, which focuses on agricultural products, such as beef, mutton, pork, etc. As we know, Adelaide is an excellent place for wine industry, there are many wine factories in different locations of Adelaide. At present, people check the wine quality in a modern way: They shoot different colors of lasers into wine and check the frequencies from different wave length of lasers, then they can put the results into machine learning and make a quick way to judge the quality of wine. Our task is to clean the data which doesn't obey the rules provided with company, and transform the data into the format by company's requirements. What's more, we build a webpage to visualize the data and add the function that picking up a specific wavelength to see the average spectrums with time.

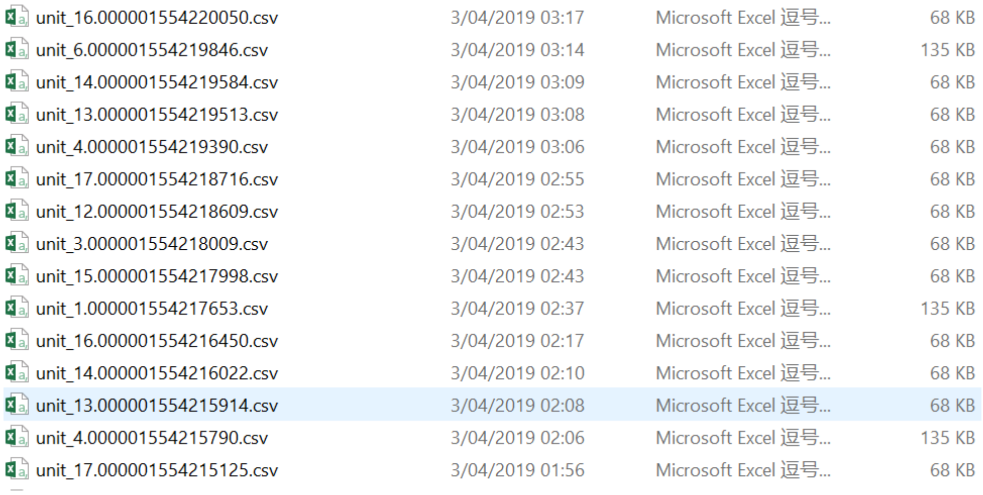


3. Approach (Done by myself)

I finished processing data and adopt following approaches to realize those functions:

(1) Preprocess the data

Since the data given by the company contains many files whose size is not the same, I have to make each file into the same size therefore I can process it further.



unit_16.000001554220050.csv	3/04/2019 03:17	Microsoft Excel 逗号...	68 KB
unit_6.000001554219846.csv	3/04/2019 03:14	Microsoft Excel 逗号...	135 KB
unit_14.000001554219584.csv	3/04/2019 03:09	Microsoft Excel 逗号...	68 KB
unit_13.000001554219513.csv	3/04/2019 03:08	Microsoft Excel 逗号...	68 KB
unit_4.000001554219390.csv	3/04/2019 03:06	Microsoft Excel 逗号...	68 KB
unit_17.000001554218716.csv	3/04/2019 02:55	Microsoft Excel 逗号...	68 KB
unit_12.000001554218609.csv	3/04/2019 02:53	Microsoft Excel 逗号...	68 KB
unit_3.000001554218009.csv	3/04/2019 02:43	Microsoft Excel 逗号...	68 KB
unit_15.000001554217998.csv	3/04/2019 02:43	Microsoft Excel 逗号...	68 KB
unit_1.000001554217653.csv	3/04/2019 02:37	Microsoft Excel 逗号...	135 KB
unit_16.000001554216450.csv	3/04/2019 02:17	Microsoft Excel 逗号...	68 KB
unit_14.000001554216022.csv	3/04/2019 02:10	Microsoft Excel 逗号...	68 KB
unit_13.000001554215914.csv	3/04/2019 02:08	Microsoft Excel 逗号...	68 KB
unit_4.000001554215790.csv	3/04/2019 02:06	Microsoft Excel 逗号...	135 KB
unit_17.000001554215125.csv	3/04/2019 01:56	Microsoft Excel 逗号...	68 KB

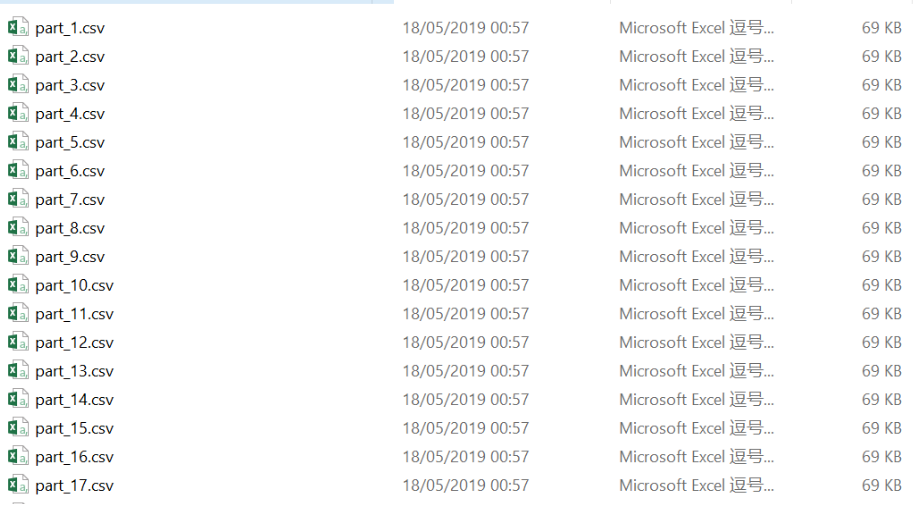
Figure 1. Initial data files

In order to make the size of each file the same, I do the following steps:

Combine all the files into one big csv file.

Split the big csv file by row counts, if the row number is the integer multiples of 1034, a new file will be created.

After doing this part, there are thousands of files were created, each of them owns the same size.

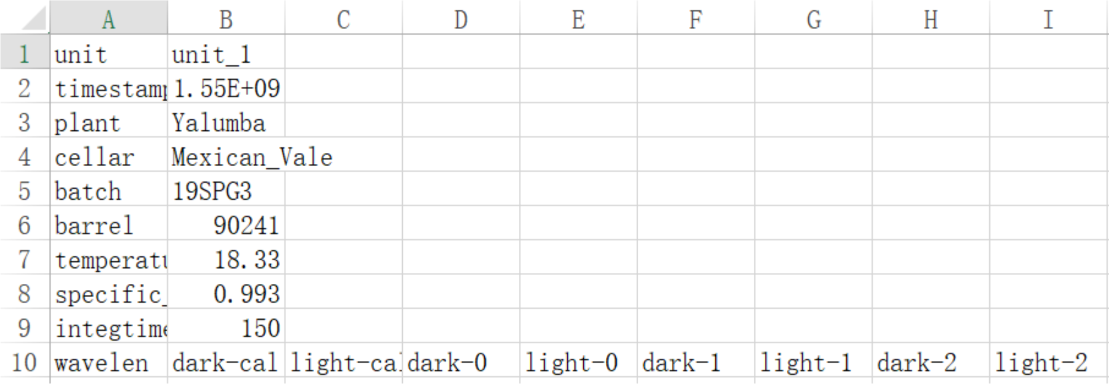


part_1.csv	18/05/2019 00:57	Microsoft Excel 逗号...	69 KB
part_2.csv	18/05/2019 00:57	Microsoft Excel 逗号...	69 KB
part_3.csv	18/05/2019 00:57	Microsoft Excel 逗号...	69 KB
part_4.csv	18/05/2019 00:57	Microsoft Excel 逗号...	69 KB
part_5.csv	18/05/2019 00:57	Microsoft Excel 逗号...	69 KB
part_6.csv	18/05/2019 00:57	Microsoft Excel 逗号...	69 KB
part_7.csv	18/05/2019 00:57	Microsoft Excel 逗号...	69 KB
part_8.csv	18/05/2019 00:57	Microsoft Excel 逗号...	69 KB
part_9.csv	18/05/2019 00:57	Microsoft Excel 逗号...	69 KB
part_10.csv	18/05/2019 00:57	Microsoft Excel 逗号...	69 KB
part_11.csv	18/05/2019 00:57	Microsoft Excel 逗号...	69 KB
part_12.csv	18/05/2019 00:57	Microsoft Excel 逗号...	69 KB
part_13.csv	18/05/2019 00:57	Microsoft Excel 逗号...	69 KB
part_14.csv	18/05/2019 00:57	Microsoft Excel 逗号...	69 KB
part_15.csv	18/05/2019 00:57	Microsoft Excel 逗号...	69 KB
part_16.csv	18/05/2019 00:57	Microsoft Excel 逗号...	69 KB
part_17.csv	18/05/2019 00:57	Microsoft Excel 逗号...	69 KB

Figure 2. Data after being preprocessed

As we can see in Figure 2, all the file’s size is 69 KB,

(2) Transform original data

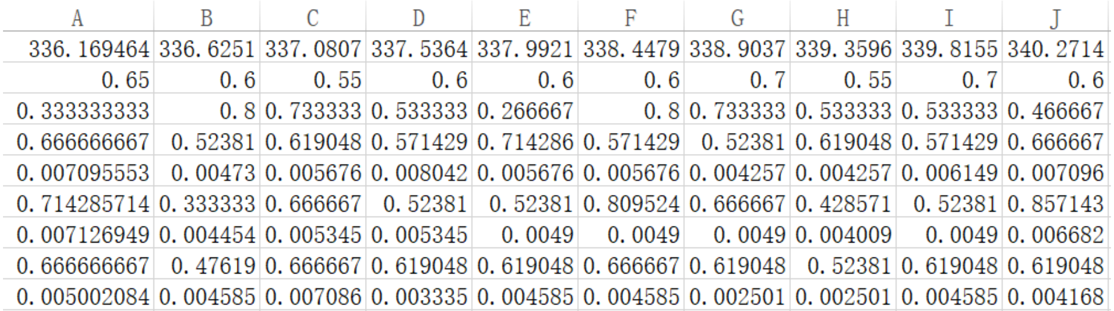


	A	B	C	D	E	F	G	H	I
1	unit	unit_1							
2	timestam	1.55E+09							
3	plant	Yalumba							
4	cellar	Mexican_Vale							
5	batch	19SPG3							
6	barrel	90241							
7	temperatu	18.33							
8	specific_	0.993							
9	integtim	150							
10	wavelen	dark-cal	light-ca	dark-0	light-0	dark-1	light-1	dark-2	light-2

Figure3. Initial data content

Figure3 show us the format of initial data, I need to process wavelength and different lights' frequencies.

I processed the data to make them realize the transform function, after being transformed, all the curves own the characteristic that zero maps to 0 and peak maps to 1, in order to realize that function, I made a simple but useful solution, the function is just as the followings: spectrum = (spectrum-spectrum.min()) / (spectrum.max() – spectrum.min()). After being processed, the peak of spectrum is 1 and the bottom of the spectrum is 0, the data's value range should be 0 to 1, which is quite small and easy to check. The sample of the data become this:



A	B	C	D	E	F	G	H	I	J
336.169464	336.6251	337.0807	337.5364	337.9921	338.4479	338.9037	339.3596	339.8155	340.2714
0.65	0.6	0.55	0.6	0.6	0.6	0.7	0.55	0.7	0.6
0.333333333	0.8	0.733333	0.533333	0.266667	0.8	0.733333	0.533333	0.533333	0.466667
0.666666667	0.52381	0.619048	0.571429	0.714286	0.571429	0.52381	0.619048	0.571429	0.666667
0.007095553	0.00473	0.005676	0.008042	0.005676	0.005676	0.004257	0.004257	0.006149	0.007096
0.714285714	0.333333	0.666667	0.52381	0.52381	0.809524	0.666667	0.428571	0.52381	0.857143
0.007126949	0.004454	0.005345	0.005345	0.0049	0.0049	0.0049	0.004009	0.0049	0.006682
0.666666667	0.47619	0.666667	0.619048	0.619048	0.666667	0.619048	0.52381	0.619048	0.619048
0.005002084	0.004585	0.007086	0.003335	0.004585	0.004585	0.002501	0.002501	0.004585	0.004168

Figure 4. The data sample after being transformed

The first line is wave length, the following is frequency. As we can see, all the frequencies are bigger than 0 and smaller than 1, that's just what we want after transforming.

(3) Noisy Detection

In this part, I adopt an easy way to check whether the spectrum noisy or not. After being checked, if the spectrum is noisy, it should be removed. In order to realize that function, I set a noise threshold, if the difference of the data is bigger than the threshold, it is noisy and should be removed, if the difference of the data is smaller than the threshold, it is not noisy and should be kept.

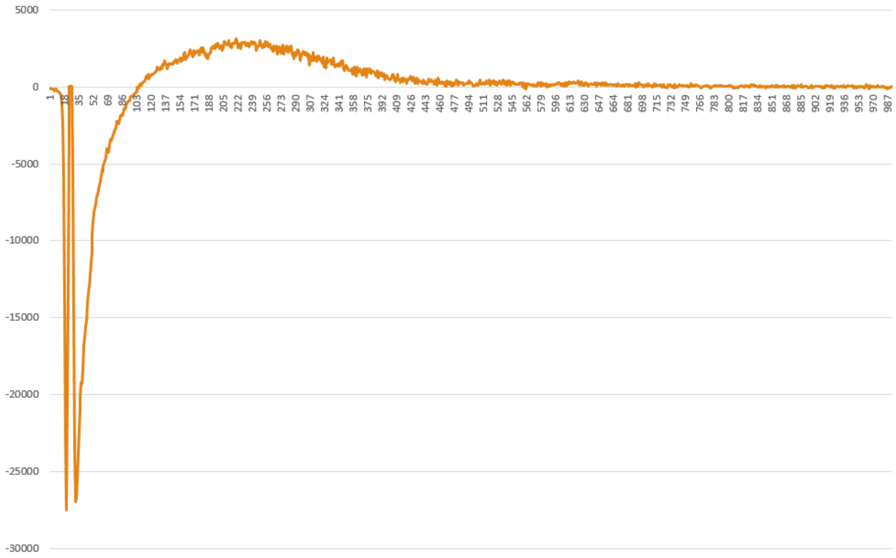


Figure 5. Noisy Spectrum

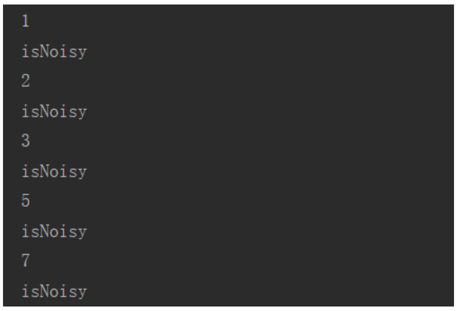
The function of is just as the followings steps:

Find the difference between each point in the spectrum and need the average

Take the absolute difference to avoid negative values, which we don't want when averaging

Check if the spectrum's average noise difference exceeds thresh.

After being processed, the sample of the data was removed, and can be shown on the console of Pycharm IDE, which is just I wanted.



```
1
isNoisy
2
isNoisy
3
isNoisy
5
isNoisy
7
isNoisy
```

Figure 6 Console results

(4) Remove Artifacts

In this part, I removed spectrums which changed too fast or too big in short wavelength range. The steps are the followings:

Iterate through the spectrum and check if a given point is greater than the points 2 to the left and right by 100 units, for each given point.

If so then I have identified an artifact and must reduce it.

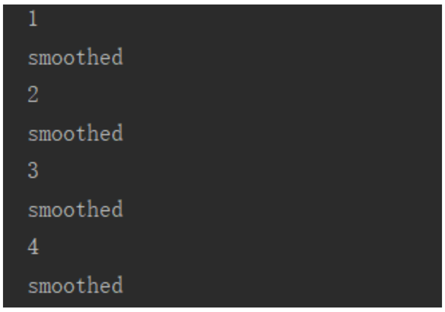
Remove the artifact by setting it to closest non-artifact point

(5) Smooth

I was struggled with this part and ask help for my teammate, therefore I knew it’s very easy because python has a built-in function.

I or we used Savitzky-Golay filter in this part, enabling increasing the precision of the data without distorting the signal tendency. The built-in function of python is like this:

e = savgol_filter(e, 51, 3)



```
1
smoothed
2
smoothed
3
smoothed
4
smoothed
```

Figure7.Smoothed result

If the data is smoothed, console of Pycharm IDE will show us.

(6) Check clipping

The clipping spectrum means the top of the spectrum is flat therefore there are at least two peaks in that curve, the Figure 6 just shows that kind of spectrum clearly, which should be removed from the data.

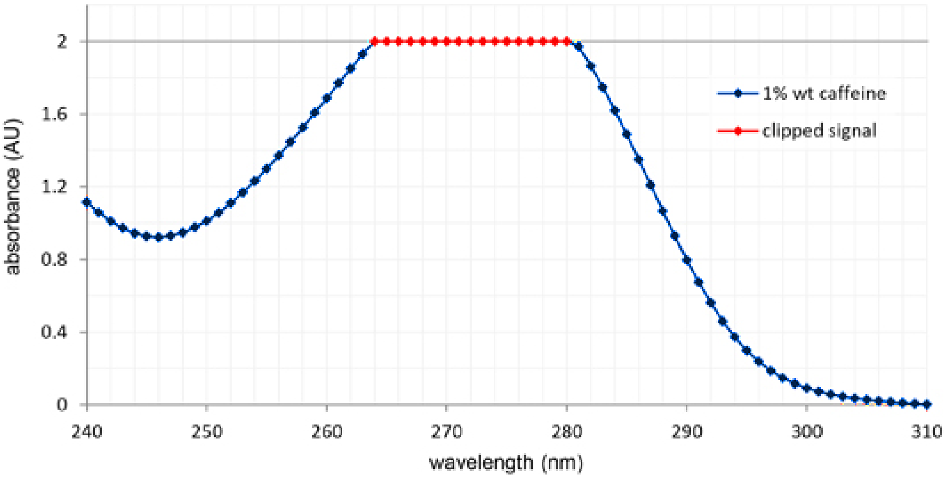
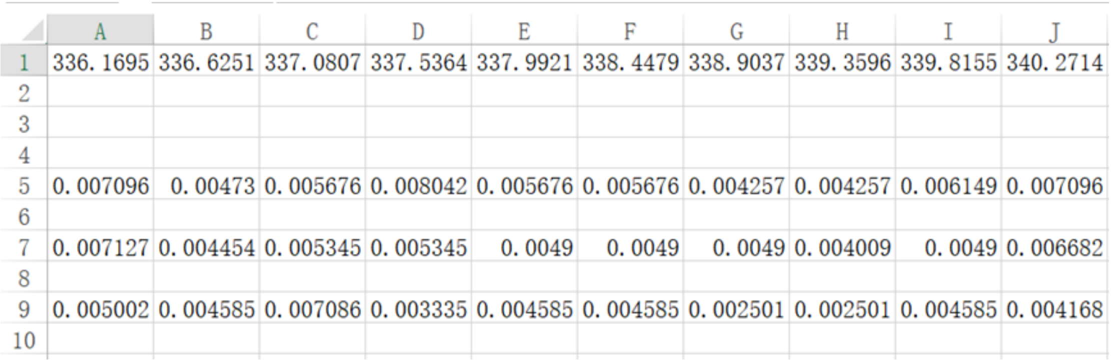


Figure 8 Clipping Spectrum

Figure 9 is the final data after being cleaned, the first row is wavelength and following rows are the frequencies.



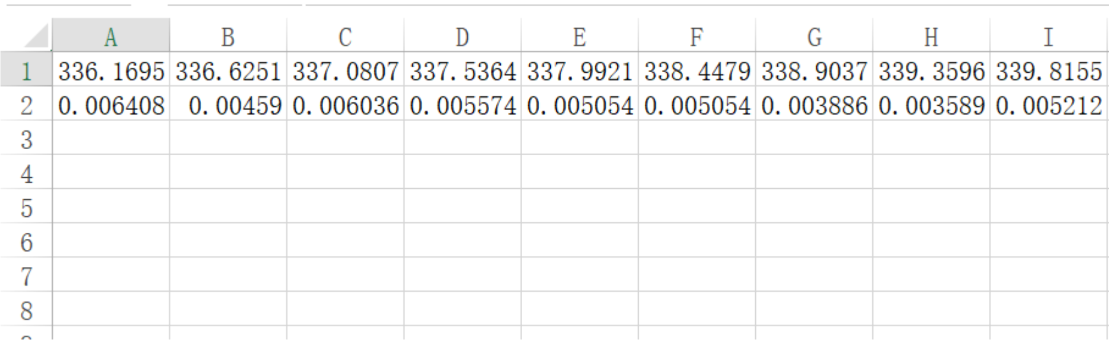
	A	B	C	D	E	F	G	H	I	J
1	336. 1695	336. 6251	337. 0807	337. 5364	337. 9921	338. 4479	338. 9037	339. 3596	339. 8155	340. 2714
2										
3										
4										
5	0. 007096	0. 00473	0. 005676	0. 008042	0. 005676	0. 005676	0. 004257	0. 004257	0. 006149	0. 007096
6										
7	0. 007127	0. 004454	0. 005345	0. 005345	0. 0049	0. 0049	0. 0049	0. 004009	0. 0049	0. 006682
8										
9	0. 005002	0. 004585	0. 007086	0. 003335	0. 004585	0. 004585	0. 002501	0. 002501	0. 004585	0. 004168
10										

Figure 9 data after being cleaned

(7) Average spectrums

The company asked to process light 0, 1, 2, and discard all the dark-cal and light-cal spectrums, then average kept light to be utilized further.

Figure 10 shows the data of specific light after being averaged.

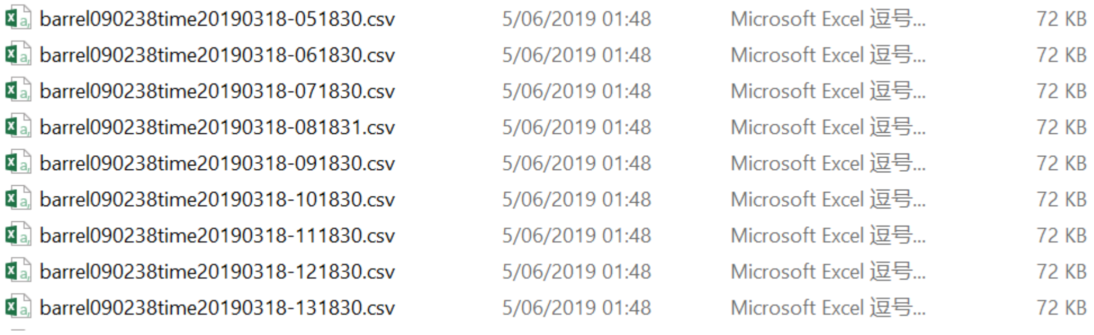


	A	B	C	D	E	F	G	H	I
1	336. 1695	336. 6251	337. 0807	337. 5364	337. 9921	338. 4479	338. 9037	339. 3596	339. 8155
2	0. 006408	0. 00459	0. 006036	0. 005574	0. 005054	0. 005054	0. 003886	0. 003589	0. 005212
3									
4									
5									
6									
7									
8									

Figure 10 data after being averaged

(8) Rename files with barrel number and date.

Users want to inspect the specific barrel of wine with time, so they should choose the processed file based on barrel number and time, so I write a script to rename files with barrel number and time, therefore it is very convenient for people to lookup.



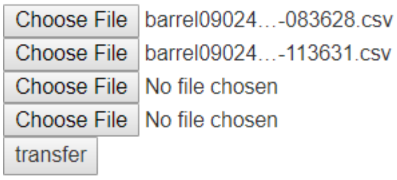
barrel090238time20190318-051830.csv	5/06/2019 01:48	Microsoft Excel 逗号...	72 KB
barrel090238time20190318-061830.csv	5/06/2019 01:48	Microsoft Excel 逗号...	72 KB
barrel090238time20190318-071830.csv	5/06/2019 01:48	Microsoft Excel 逗号...	72 KB
barrel090238time20190318-081831.csv	5/06/2019 01:48	Microsoft Excel 逗号...	72 KB
barrel090238time20190318-091830.csv	5/06/2019 01:48	Microsoft Excel 逗号...	72 KB
barrel090238time20190318-101830.csv	5/06/2019 01:48	Microsoft Excel 逗号...	72 KB
barrel090238time20190318-111830.csv	5/06/2019 01:48	Microsoft Excel 逗号...	72 KB
barrel090238time20190318-121830.csv	5/06/2019 01:48	Microsoft Excel 逗号...	72 KB
barrel090238time20190318-131830.csv	5/06/2019 01:48	Microsoft Excel 逗号...	72 KB

Figure 10 File after being renamed

4. Results(Done by the whole team)

Using csv file like Figure 8 and my teammate's visualization tool, we can check the results for the whole data given by the company.

Just as the following picture shows, cleaned spectrums can be shown on the webpage directly, people can check whether the spectrum meet the requirements, if something is wrong, they can look up this csv file and find out the problem.



Choose File barrel09024…-083628.csv
Choose File barrel09024…-113631.csv
Choose File No file chosen
Choose File No file chosen
transfer

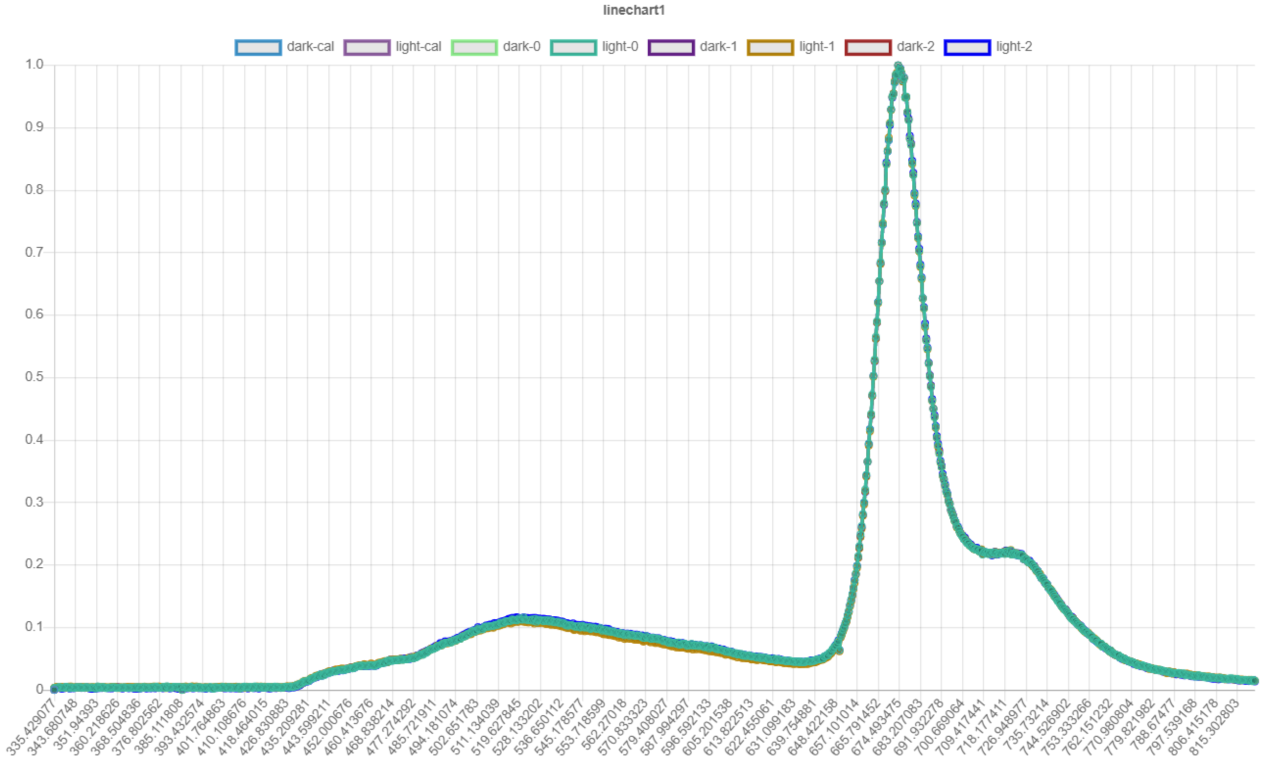


Figure 11. Visualization for one specific barrel (not averaged)

Webpage can be utilized with csv file which is not averaged, just as Figure 11, different kinds of lights can be shown with different colors, since the value of different light is very close, it is not easy to distinguish them, but it can be see easily if different colors vary a lot.

What’s more, averaged spectrum can also be shown on webpage, Figure 12 is the averaged result of one spectrum, we can use this result in the next step by selecting specific wave length to check the frequencies.

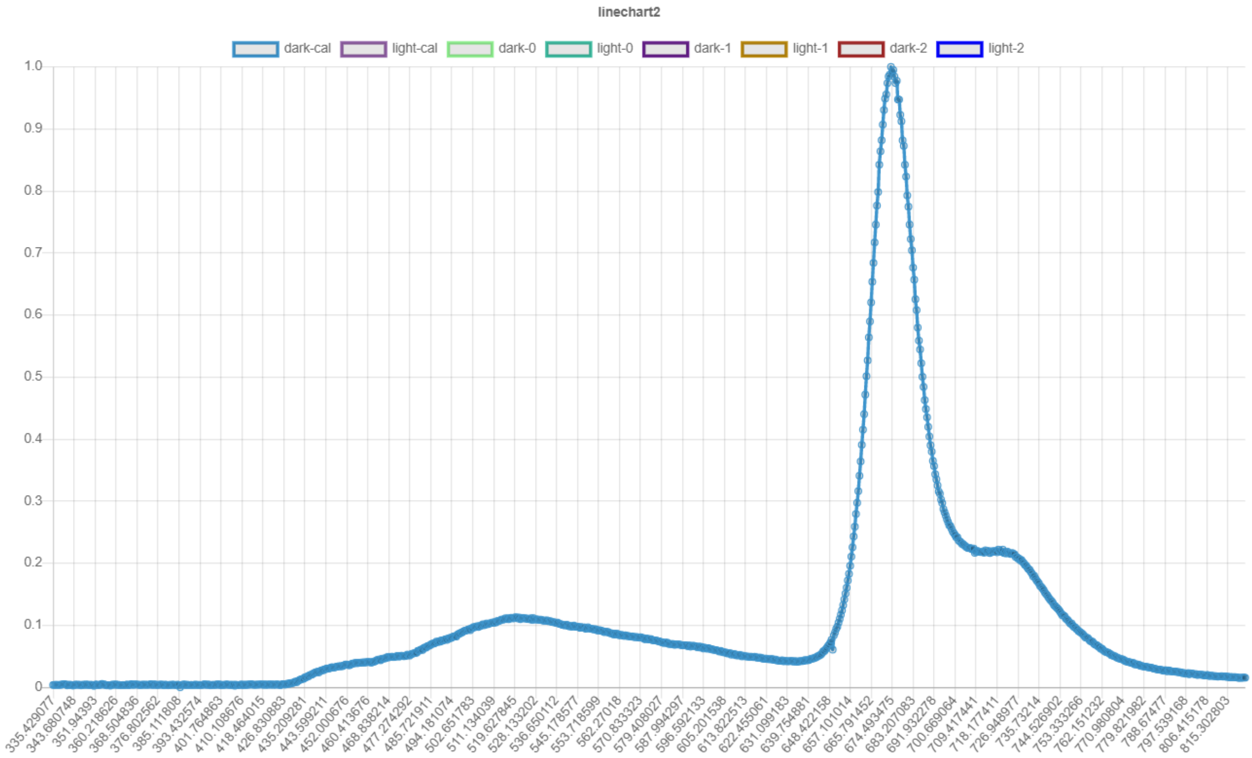
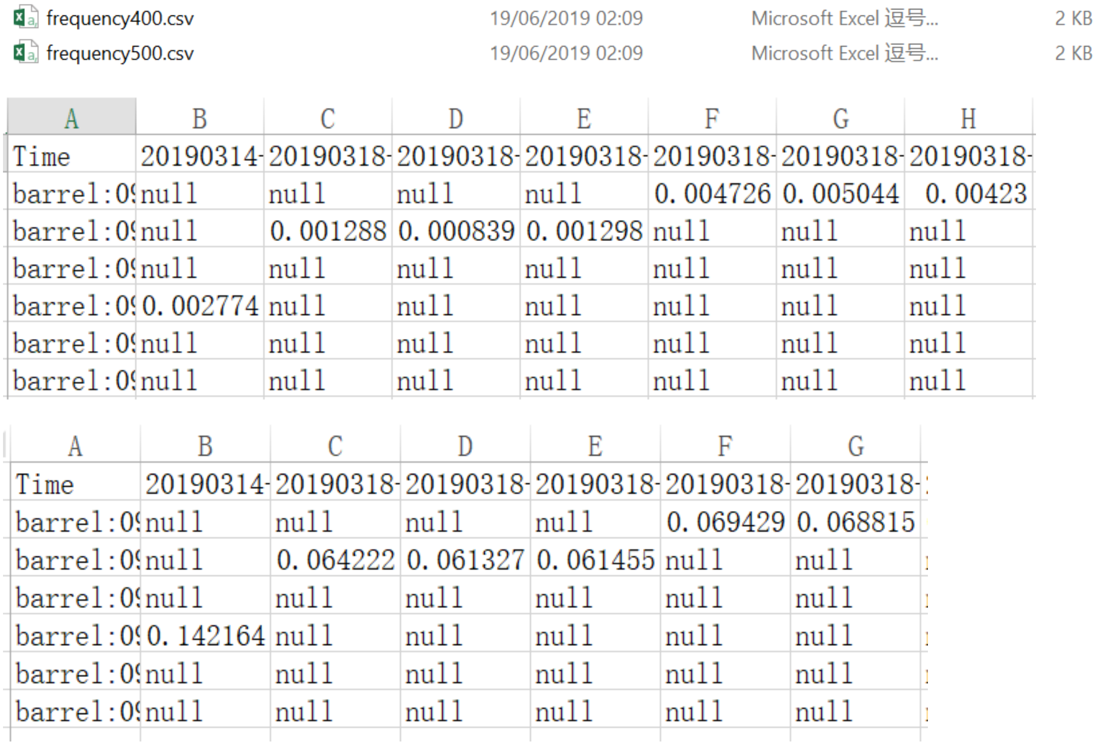


Figure 12. Visualization for one specific barrel (averaged)

We can select specific wavelength to check the frequencies. Entering a frequency value, my teammate's script will scan all the files and create a csv file.



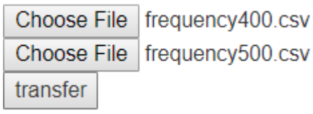
frequency400.csv	19/06/2019 02:09	Microsoft Excel 逗号...	2 KB
frequency500.csv	19/06/2019 02:09	Microsoft Excel 逗号...	2 KB

A	B	C	D	E	F	G	H
Time	20190314-	20190318-	20190318-	20190318-	20190318-	20190318-	20190318-
barrel:0(	null	null	null	null	0.004726	0.005044	0.00423
barrel:0(	null	0.001288	0.000839	0.001298	null	null	null
barrel:0(	null	null	null	null	null	null	null
barrel:0(	0.002774	null	null	null	null	null	null
barrel:0(	null	null	null	null	null	null	null
barrel:0(	null	null	null	null	null	null	null

A	B	C	D	E	F	G
Time	20190314-	20190318-	20190318-	20190318-	20190318-	20190318-
barrel:0(	null	null	null	null	0.069429	0.068815
barrel:0(	null	0.064222	0.061327	0.061455	null	null
barrel:0(	null	null	null	null	null	null
barrel:0(	0.142164	null	null	null	null	null
barrel:0(	null	null	null	null	null	null
barrel:0(	null	null	null	null	null	null

Figure 13. csv files of frequency

Just as the Figure13 shows, two csv files are created. Barrel number and time is stored in the file. User can import files into the webpage and transfer.



Choose File frequency400.csv

Choose File frequency500.csv

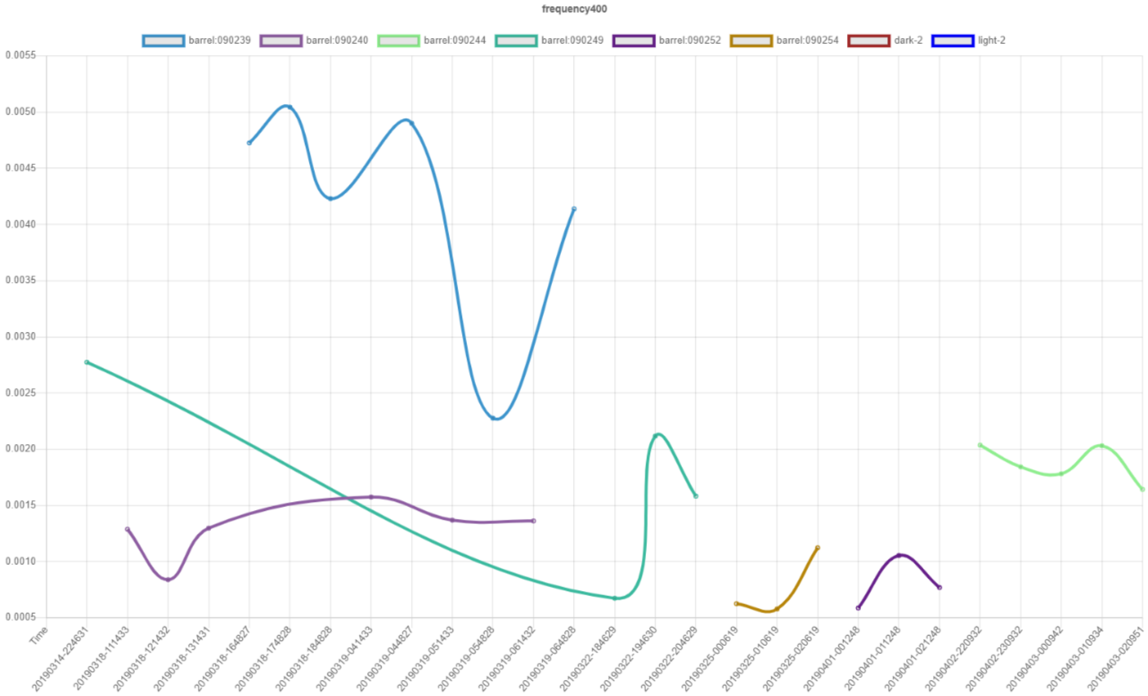


Figure 14. Frequency 400(6 barrels)

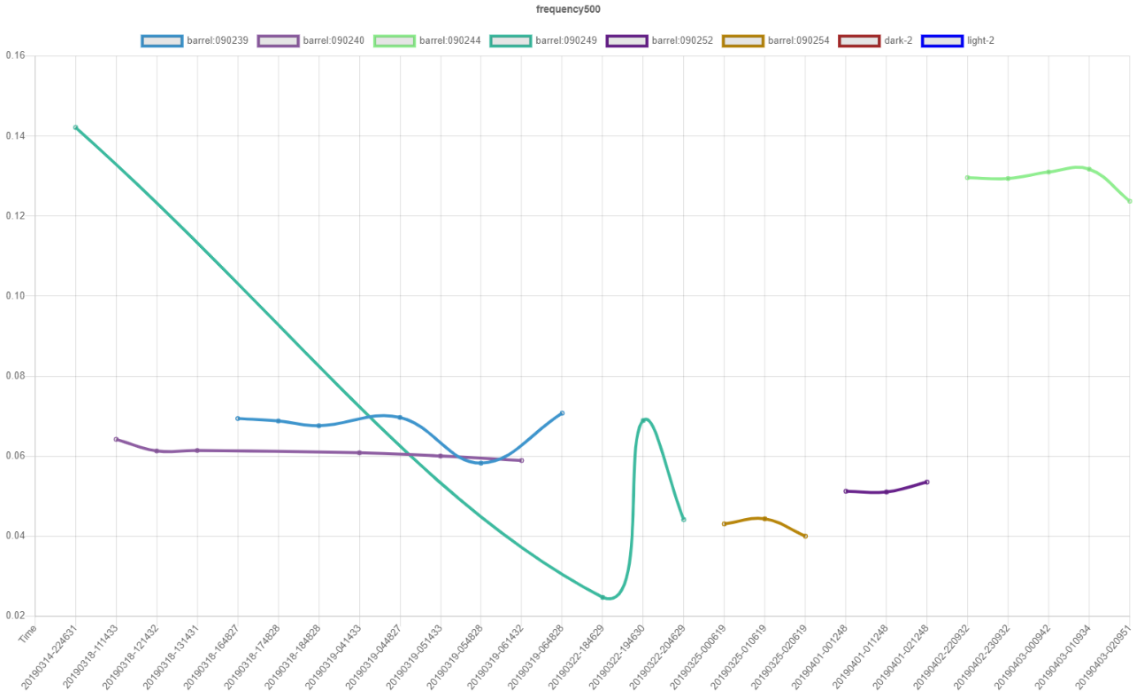


Figure 15. Frequency 600(6 barrels)

Figure 14 and 15 directly show us the frequency change with time of specific wavelength, since that company didn’t tell us how to identify the quality from spectrum, I personally think the barrel 090240’s (the purple one) quality is good because it changed little with time.

5. Conclusion

I fully meet the requirements of the wine company that clean data and visualize it, data can be processed by python scripts and visualized by webpage.

This project can be improved in the followings:

I can optimize the UI design to build a more user-friendly webpage, adding more functions such as animation or words-speaking, therefore color-blind or totally-blind people could also use our tool.

Since different companies have different requirements, I must modify our scripts to meet them, which is quite annoying. However, if I could use database to store those data, then I do not need to modify scripts anymore, database could be extremely helpful.

Reference:

[1] Bill Lubanovic, O'Reilly Media, Inc. *Introducing Python* ISBN: 9781449361167