Data Quality Testing (Fall 2019)

● Shlok Gopalbhai Gondalia ● CS498 ● shlok@rams.colostate.edu

I. INTRODUCTION

Data quality testing is one of the test approaches to check the authenticity of entries in a large dataset. To verify these huge datasets manually requires a lot of time and also it is not humanly possible when the dataset becomes really large. Most of the time, many types of errors occur when working with real-life datasets. Therefore, in our research, we use the ADQuaTe data quality test approach to solve the issues which might arise in a large dataset. So, for testing our tool, we have used the UCI real-life datasets for experiments and then we analyzed those results to make our tool more efficient and reliable. Our tool uses the Machine Learning algorithm, which includes multiple nodes and layers to figure out the faulty records in a dataset. This tool is designed by Hajar Homayouni and I have used this tool to generate results with the new datasets and analyzed those results if there is any ambiguity.

II. EXPERIMENTS & RESULTS

Before I even started working on the actual datasets, I didn't know how to install the tool and how to get it setup on my personal computer (Windows 10). During this Dr. Sudipto Ghosh along with Hajar Homayouni helped me and after spending enough time, we didn't succeed. So, in the end, we ended up installing the tool in my CS Linux machine and then I started working on the actual tool, and I faced many challenges but was also able to get some good results.

The first step in the experiments is preprocessing the datasets from the ODDS (Outlier Detection Datasets) website. Sometimes, it requires adding the id column and removing the class column. It also requires adding the names of the attribute before it is ready to run through the tool. During this process, I faced many difficulties which include not knowing which file contains actual datasets, not knowing the outliers and how they are defining the outliers. The only pattern I saw is that they define outliers to whatever class which appeared the least in the dataset. During this, Hajar Homayouni along with Dr. Sudipto Ghosh helped me a lot to understand these different sets and how should I preprocess this dataset.

Almost, after half of the semester gone, I was now finally understanding what I need to do and how I need to do those. So, I used the real-life datasets given on the ODDS website and started working on them one-by-one. Here is a brief summary of all the datasets present on the ODDS website along with some comments which include the problems I am facing as of right now.

Name	Run	Comment
Lympho	YES	OLD
WBC	YES	OLD
Glass	YES	OLD

Vowels	NO	Multivariate Time Series
Cardio	YES	NEW
Thyroid	NO	Don't Know which File to Download
Musk	YES	Weird Result
Satimage-2	YES	NEW
Letter	NO	Don't know the outliers
Speech	NO	Don't know how to download the Data
Pima	NO	Dataset no longer available
Satellite	YES	Same Dataset as Satimage-2, but this uses different outliers (Produced Good Results) NEW
Shuttle	YES	NEW
BreastW	YES	OLD
Arrhythmia	NO	Too many columns, the tool can't run it right now (Weird Results)
Ionosphere	YES	OLD
Mnist	NO	Don't know how to download the Data
Optdigits	YES	Not satisfied with the results, this may be due to too many columns (Weird Results)
Http	NO	Don't know the outliers
ForestCover	NO	Data too large (Weird Results)
Mulcross	NO	Dataset no longer available
Smtp	NO	Don't know the outliers, Also it has the same data as the Http one
Mammography	NO	Don't know how to download the Data
Annthyroid	NO	Don't Know which File to Download
Pendigits	NO	Don't know the outliers
Ecoli	YES	OLD
Wine	YES	NEW
Vertebral	YES	OLD
Yeast	NO	Don't know the outliers
Seismic	YES	Weird Results
Heart	YES	Weird Results

*OLD: Results already use in the last paper.

*NEW: Datasets I ran without any issues and it produced good results.

*Weird Results: For example, it doesn't show much improvement as the tool runs.

*Don't know the Outliers: Not enough information is given in the Dataset to determine the outliers.

From this summary, we can clearly see that there are very few datasets that ran successfully with the current tool we have. There are many problems in these datasets, like not knowing the outliers at all, not enough information in which file to download out of dozens from the datasets or not redirecting to UCI datasets. I have also mentioned weird results in many of the datasets which means that the results produced by the tool didn't make any sense. For example, True positive Rate not showing enough improvement. Below you can see an example from the MUSK Dataset.

dataset_id	time	previously_ detected	suspicious_ detected	undetected	newly_ detected	true_ negative_ rate	false_ negative_ rate	false_positive_ rate	true_positive_ rate
musk_5336	35:14.2	0	0	0	0	1	0	1	0
musk_5336	37:37.3	0	1	1	0	1	0	1	0
musk_5336	39:59.8	0	1	1	0	1	0	1	0
musk_5336	42:25.5	0.010309278	0.989690722	0.989690722	0	1	0	0.989690722	0.010309278
musk_5336	44:57.1	0.020618557	0.979381443	0.979381443	0	1	0	0.979381443	0.020618557
musk_5336	47:27.4	0.020618557	0.979381443	0.979381443	0	1	0	0.979381443	0.020618557
musk_5336	49:50.7	0.020618557	0.979381443	0.979381443	0	1	0	0.979381443	0.020618557
musk_5336	52:24.3	0.020618557	0.979381443	0.979381443	0	1	0	0.979381443	0.020618557
musk_5336	54:57.8	0.072164948	0.927835052	0.927835052	0	1	0	0.927835052	0.072164948
musk_5336	57:33.7	0.082474227	0.917525773	0.917525773	0	1	0	0.917525773	0.082474227
musk_5336	00:07.3	0.082474227	0.917525773	0.917525773	0	1	0	0.917525773	0.082474227

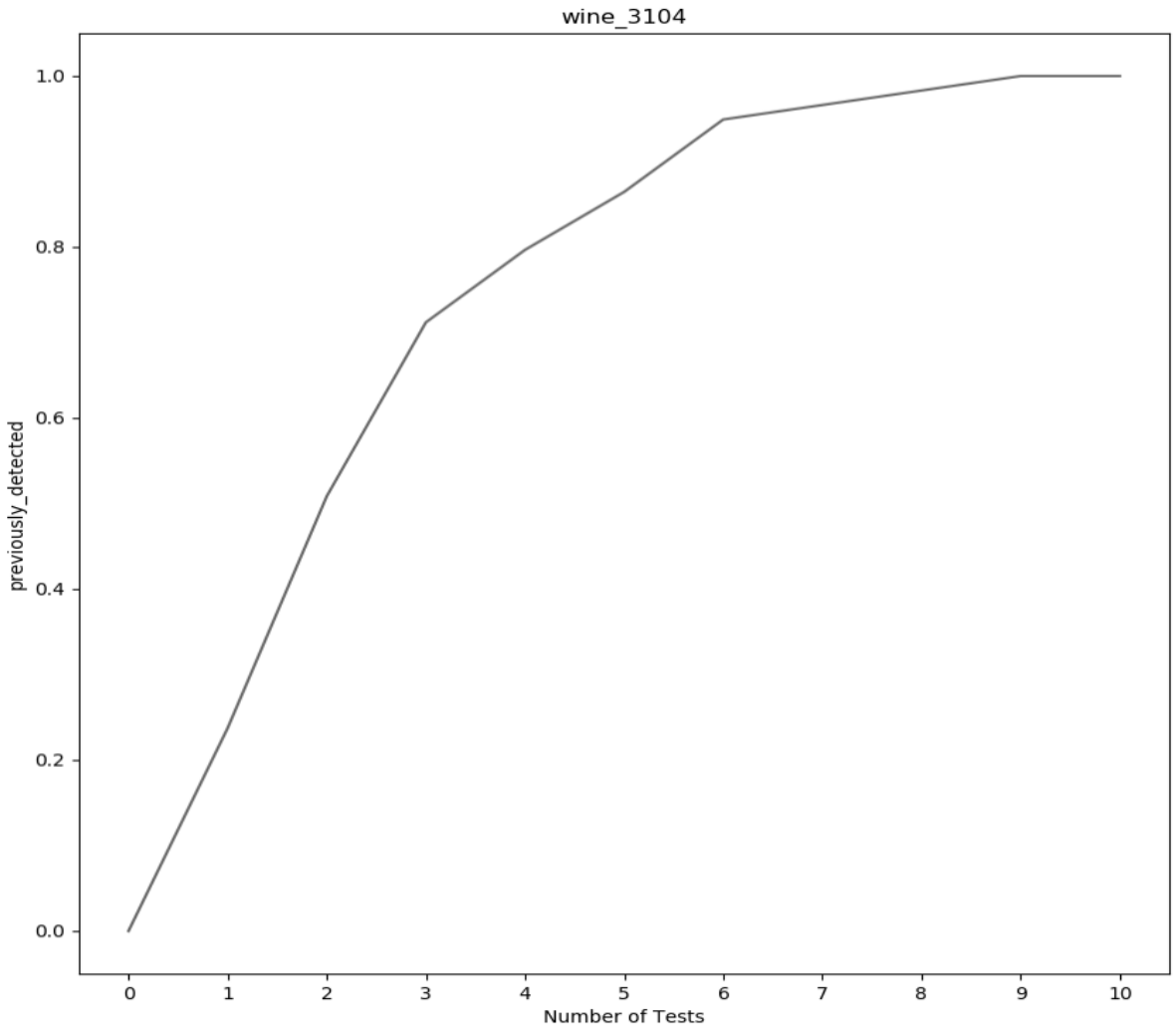
From the results, we can immediately notice that the True Positive Rate is not improving at all, as after the complete test, the result is only improved by 8%, but some other datasets improved by more than 95%, for example, Satimage-2 dataset.

dataset_id	time	previously_ detected	suspicious_ detected	undetected	newly_ detected	true_ negative_ rate	false_ negative_ rate	false_positive_ rate	true_positive_ rate
satimage_4789	09:11.5	0	0	0	0	1	0	1	0
satimage_4789	09:30.9	0.352818372	0.647181628	0.647181628	0	1	0	0.647181628	0.352818372
satimage_4789	09:49.6	0.701461378	0.298538622	0.298538622	0	1	0	0.298538622	0.701461378
satimage_4789	10:08.3	0.8434238	0.1565762	0.1565762	0	1	0	0.1565762	0.8434238
satimage_4789	10:26.8	0.939457203	0.060542797	0.060542797	0	1	0	0.060542797	0.939457203
satimage_4789	10:45.4	0.954070981	0.045929019	0.045929019	0	1	0	0.045929019	0.954070981
satimage_4789	11:04.2	0.960334029	0.039665971	0.039665971	0	1	0	0.039665971	0.960334029
satimage_4789	11:22.8	0.964509395	0.035490605	0.035490605	0	1	0	0.035490605	0.964509395
satimage_4789	11:42.0	0.964509395	0.035490605	0.035490605	0	1	0	0.035490605	0.964509395
satimage_4789	12:01.2	0.966597077	0.033402923	0.033402923	0	1	0	0.033402923	0.966597077
satimage_4789	12:20.7	0.972860125	0.027139875	0.027139875	0	1	0	0.027139875	0.972860125

Therefore, something is clearly wrong with the datasets like the Musk dataset. So, from running all these different datasets, I found a pattern/trend in the datasets that whenever there are more than 26-30 attributes or more than 50000 entries the datasets, then the tool produces weird results like the Musk dataset. Interestingly the results didn't matter in terms of the percentage of outliers. So, I clearly don't know why the tool is not working properly for these datasets. One more thing, which I also noticed is that all the datasets used for the last paper were small enough compared to these new datasets, to not notice this problem at all.

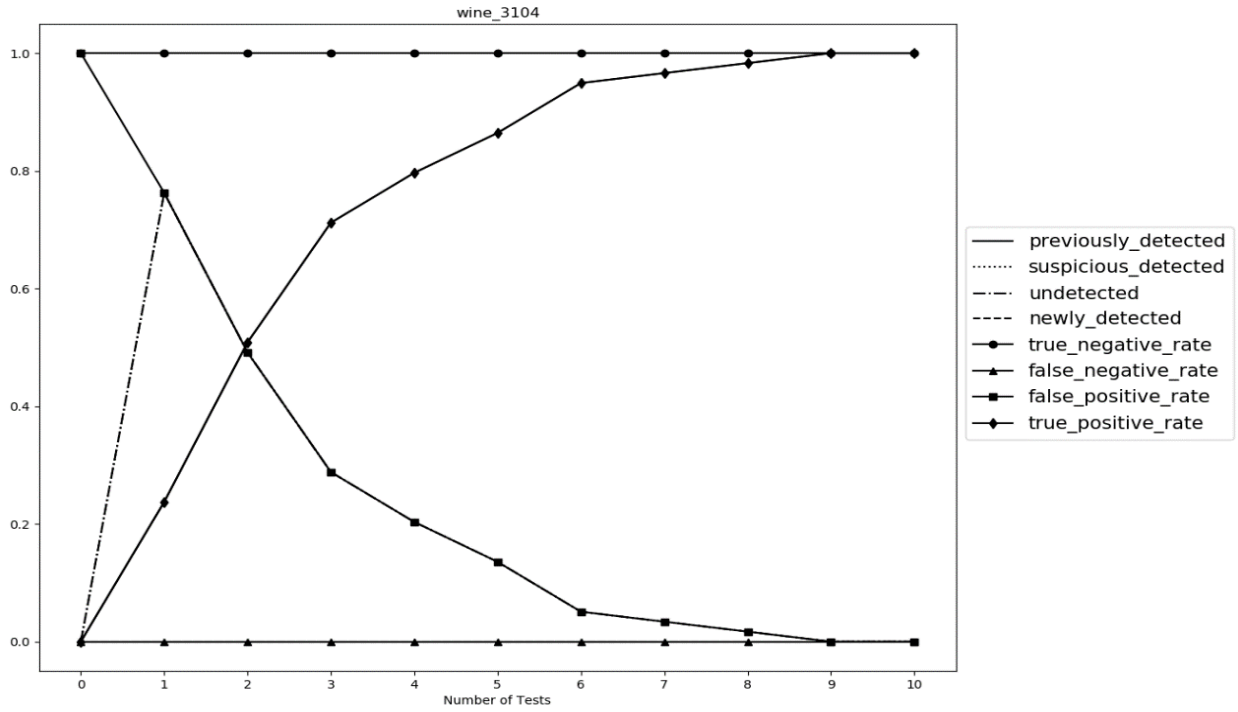
III. SCRIPTS FOR PLOTTING GRAPHS

Along with working on these datasets, Dr. Sudipto Ghosh also gave me a task to make python scripts that can plot graphs of these results in different ways, so that we have don't have to create these graphs manually when we write our final report. Therefore, this semester I have created three different python scripts that can plots graphs with each script having different functionality. The first script is the plot.py, this script uses different libraries like pandas, DataFrame. Matplotlib and CSV to successfully create the graphs of these datasets based on columns. For example, below is the graph created by plot.py which shows the Previously detected attribute of the Wine dataset.

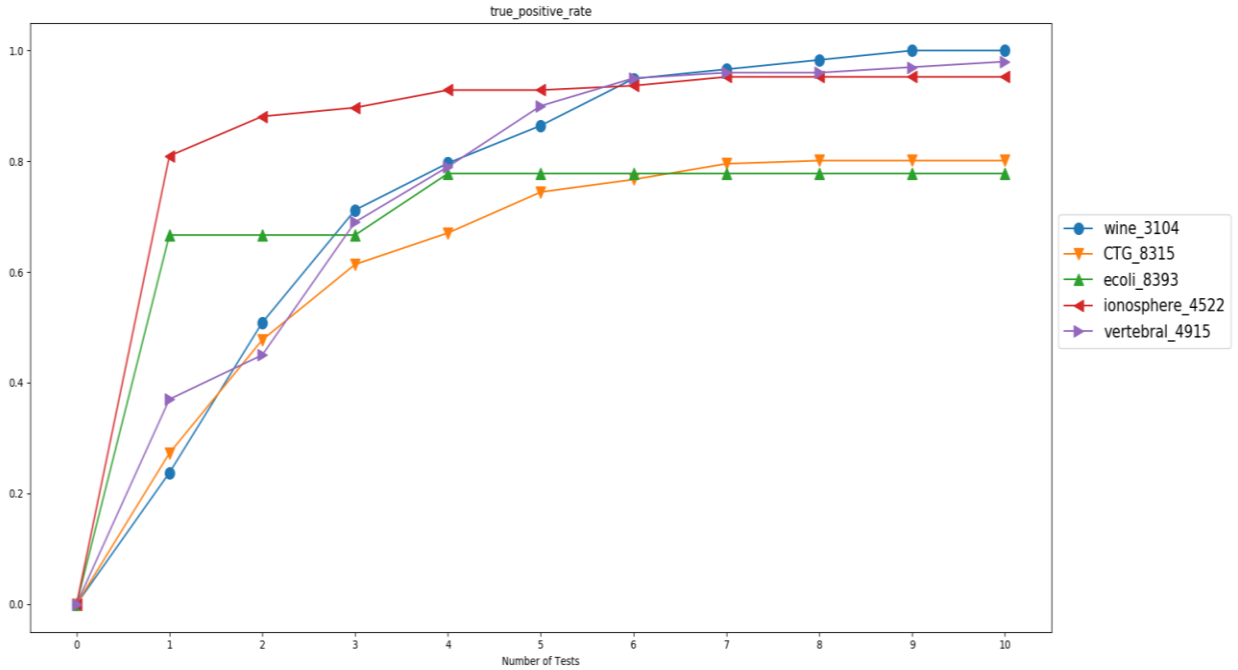


The second script is called plot_merge.py and this script also uses the same libraries, but it functions a little bit differently than plot.py. This script creates graphs of all the attributes of particular datasets in a single plot, showing all the properties of a dataset. For example, below is both the dataset (for reference) and the graph created by plot_merge.py which shows all the attributes of the Wine dataset.

dataset_id	time	previously_ detected	suspicious_ detected	undetected	newly_ detected	true_ negative_ rate	false_ negative_ rate	false_positive_ rate	true_positive_ rate
wine_3104	55:44.8	0	0	0	0	1	0	1	0
wine_3104	55:46.4	0.237288	0.762712	0.762712	0	1	0	0.762712	0.237288
wine_3104	55:47.8	0.508475	0.491525	0.491525	0	1	0	0.491525	0.508475
wine_3104	55:49.3	0.711864	0.288136	0.288136	0	1	0	0.288136	0.711864
wine_3104	55:50.7	0.79661	0.20339	0.20339	0	1	0	0.20339	0.79661
wine_3104	55:52.2	0.864407	0.135593	0.135593	0	1	0	0.135593	0.864407
wine_3104	55:54.6	0.949153	0.050847	0.050847	0	1	0	0.050847	0.949153
wine_3104	55:56.2	0.966102	0.033898	0.033898	0	1	0	0.033898	0.966102
wine_3104	55:57.7	0.983051	0.016949	0.016949	0	1	0	0.016949	0.983051
wine_3104	55:59.1	1	0	0	0	1	0	0	1
wine_3104	56:00.6	1	0	0	0	1	0	0	1



The last script I created is called plot_trend.py and this script again uses the same libraries as the other two scripts. This script is the most important out of all the scripts, and probably we will be using this script a lot, to conclude our results. This script creates graphs based on a particular attribute and combines all the datasets (given by the user) to create one graph which shows the general trend of all the datasets. For example, below is the graph created by plot.py which shows the True Positive Rate attribute of the Wine, Cardio (CTG), Ecoli, Ionosphere, and Vertebral dataset.



Overall, this was side thing and I am hoping that these scripts will help towards making the final report for the research, as making all these graphs manually consumes a lot of time.

IV. THINGS I LEARNED

This semester has turned out to be one of the greatest semesters I ever had at CSU. Research experience helped me a lot to learn new things and also helped me to expand my connections in the CS department. Due to research, I learned to use python and its libraries which I didn't know before. I also got some knowledge about Machine Learning like nodes and layers, with the help of Dr. Sudipto Ghosh. Learning both these things will help me a lot as I go along my career. Research also helped me personally on my time management skill as taking 18 credits is not easy, but then also I managed to get through by giving time to all the important things along with the research weekly tasks. I learned how to handle big data with help from Hajar Homayouni, which I would have never learned in my general classes this early. I also developed a skill of analyzing these datasets and their results, which I am guessing will help in my future classes. As I didn't do much work over this semester as some of the things didn't go well as I assumed to be. Much of the time went on setting up the tool, learning how it works, and also managing my other classes. But I am confident that in the next semester I will be learning a lot, as I already know how different things work and also, I am taking fewer credits which will give me extra time to focus on research tasks.

V. CONCLUSION AND WHAT's NEXT

This is my first exposure to any time of research, and I have done so many things over the course of this semester. I ran most of the datasets from the ODDS website and also created a summary, so that it is easy to solve the problems and also, we can get an overall idea of all these datasets. I also analyzed these datasets and found several things which were going wrong when running through the tool, like weird results which appeared, if the number of entries in the datasets were too large or the number of the attribute. Now, we also don't have to worry about plotting the results manually, as those three scripts would help to achieve those graphs. There are a lot of things which I gained through this research fellowship, which I would have never learned or experienced if I was not a part of this research group.

Next semester, I am planning to get more involved in this research by handling time-series datasets and get to know more about these datasets work and how do we preprocess and analyze these types of datasets. One thing which I definitely want to do is to read more research papers so that I can know what different things are going on in this particular direction and by doing that it will also help me to learn and understand how to read research papers.

V. REFERENCES

[1] H. Homayouni, S. Ghosh, I. Ray, M. Kahn, 2019. “An Interactive Data Quality Test Approach for Constraint Discovery and Fault Detection”, submitted as a full paper to IEEE Big Data.

[2] H. Homayouni, S. Ghosh, I. Ray, 2019. “ADQuaTe: An Automated Data Quality Test Approach for Constraint Discovery and Fault Detection”, In IEEE 20th International Conference on Information Reuse and Integration for Data Science.