DATA QUALITY TESTING

Name: Shlok Gopalbhai Gondalia
Class: CS498
Email: shlok@rams.colostate.edu
Semester: Fall 2019

INTRODUCTION

What is Data Quality Testing?

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.

My Understanding of this Research

For testing we are using 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 are any ambiguity.

EXPERIMENTS & RESULTS

Before Hand Problems

- 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
- In the end, we ended up installing the tool in my CS Linux machine and then I started working on the actual tool
- Faced many challenges but was also able to get some good results

SUMMARY OF DATASETS

NAME	RUN	COMMENT
Lympho	YES	OLD
WBC	YES	OLD
Glass	YES	OLD
BreastW	YES	OLD
Ionosphere	YES	OLD
Ecoli	YES	OLD
Vertebral	YES	OLD

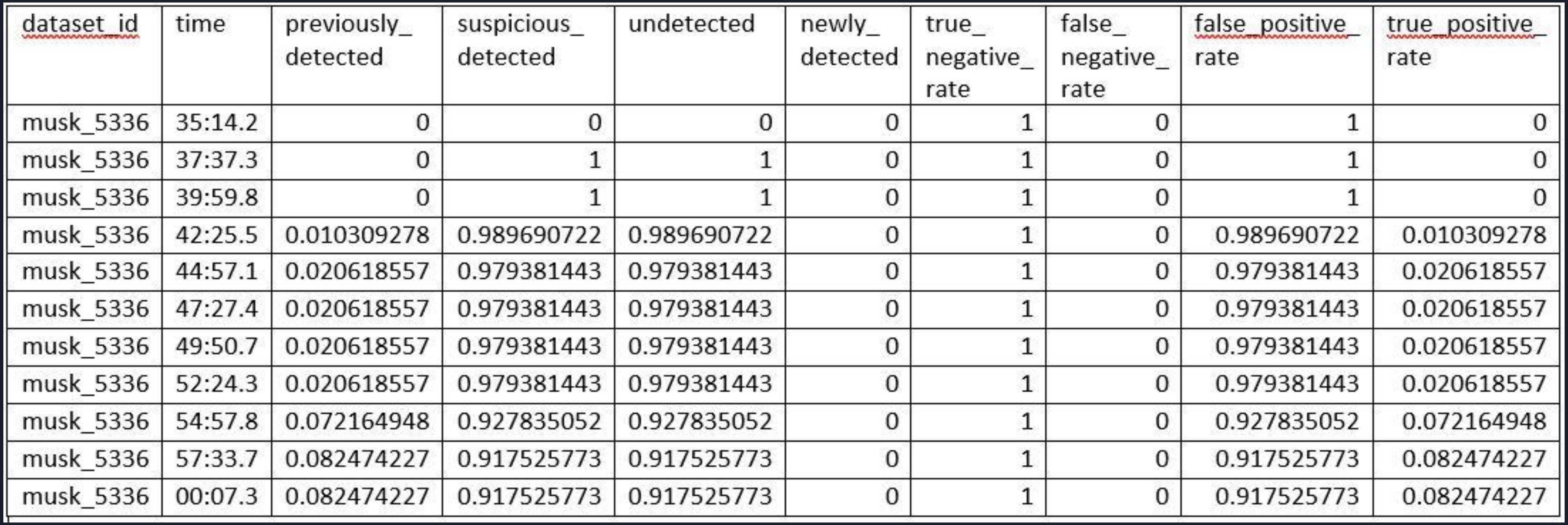
*OLD = DataSets Already
Used for Last Paper
(All having Good Results)

Not Interesting Right!

NAME	RUN	COMMENT
Musk	YES	Weird Results
Optdigits	YES	Weird Results
Seismic	YES	Weird Results
Heart	YES	Weird Results

Weird Results of the datasets means that the results produced by the tool didn’t make any sense. For example, True positive Rate not showing enough improvement.

Example of a Dataset With a Weird Result



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

Why Weird Results???

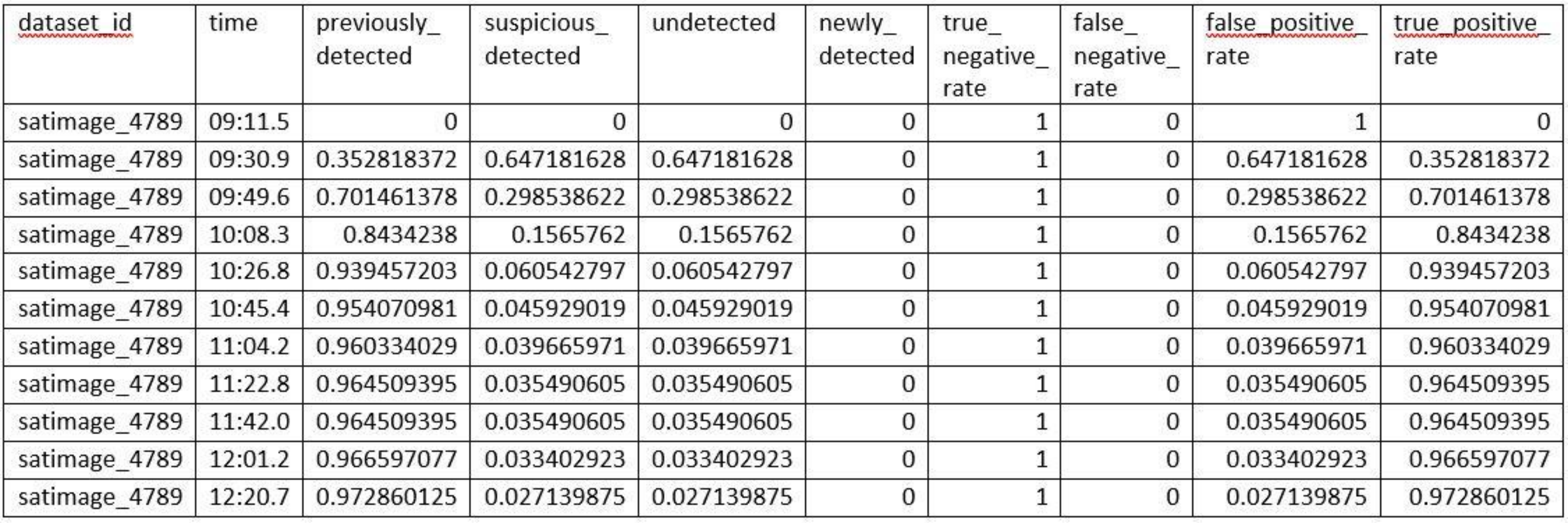
- It is clearly that something is definitely wrong with the datasets like the Musk dataset
- Found a pattern/trend in the datasets that whenever there are more than 26-30 attributes or more than 50000 entries in the datasets, this problem arises
- Interestingly the results didn't matter in terms of the percentage of outliers
- 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.

NAME	RUN	COMMENT
Cardio	YES	NEW
Satimage-2	YES	NEW
Satellite	YES	NEW
Shuttle	YES	NEW
Wine	YES	NEW

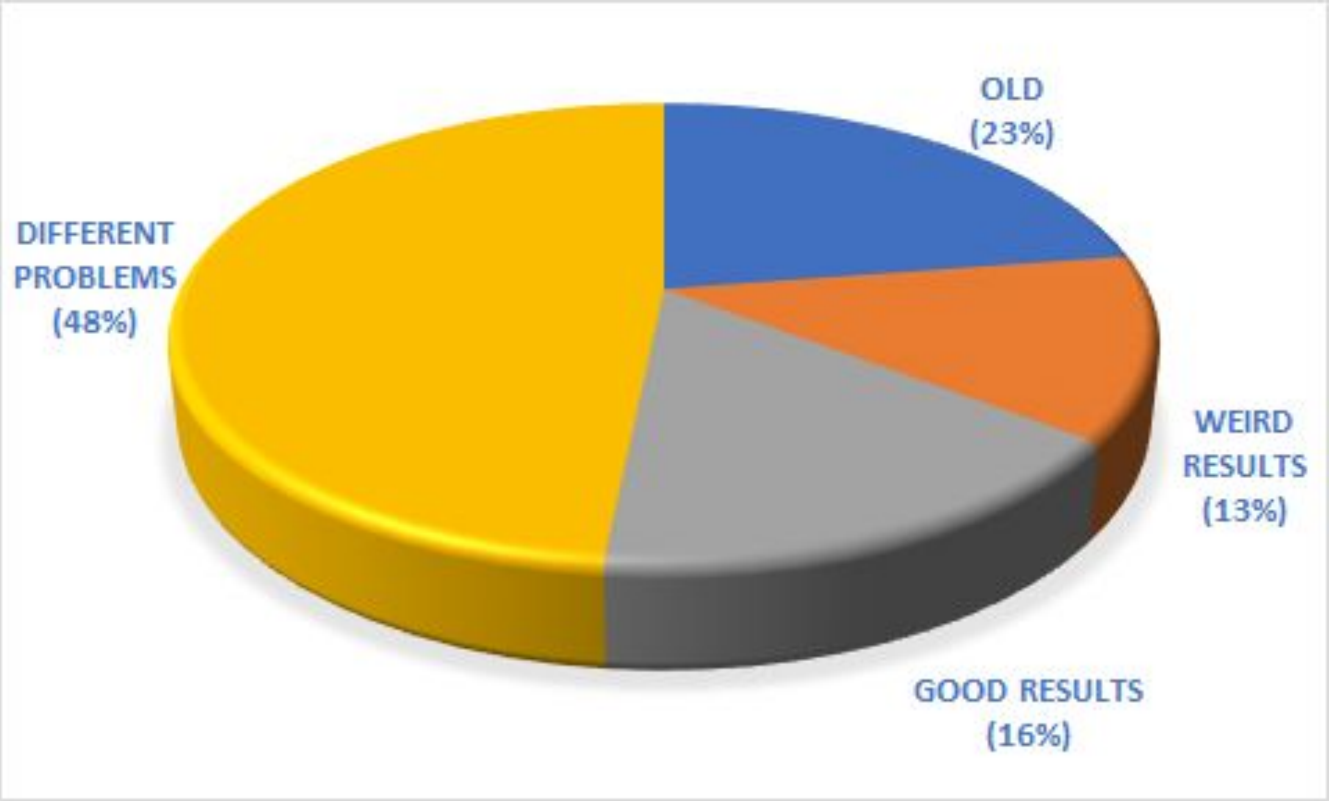
All the Datasets which ran successfully without any problems or weird results

Example of a Dataset With a Good Result



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

NAME	RUN	COMMENT
Pima	NO	Dataset no longer available
Mulcross	NO	Dataset no longer available
Speech	NO	Don’t know how to download the Data
Mnist	NO	Don’t know how to download the Data
Mammography	NO	Don’t know how to download the Data
Letter	NO	Don’t know the outliers
Http	NO	Don’t know the outliers
Pendigits	NO	Don’t know the outliers
Yeast	NO	Don’t know the outliers
Smtp	NO	Don’t know the outliers, Also it has the same data as the Http one
Thyroid	NO	Don’t Know which File to Download
Annthyroid	NO	Don’t Know which File to Download
Vowels	NO	Multivariate Time Series
ForestCover	NO	Data too large
Arrhythmia	NO	Too many columns, tool can’t run it right now

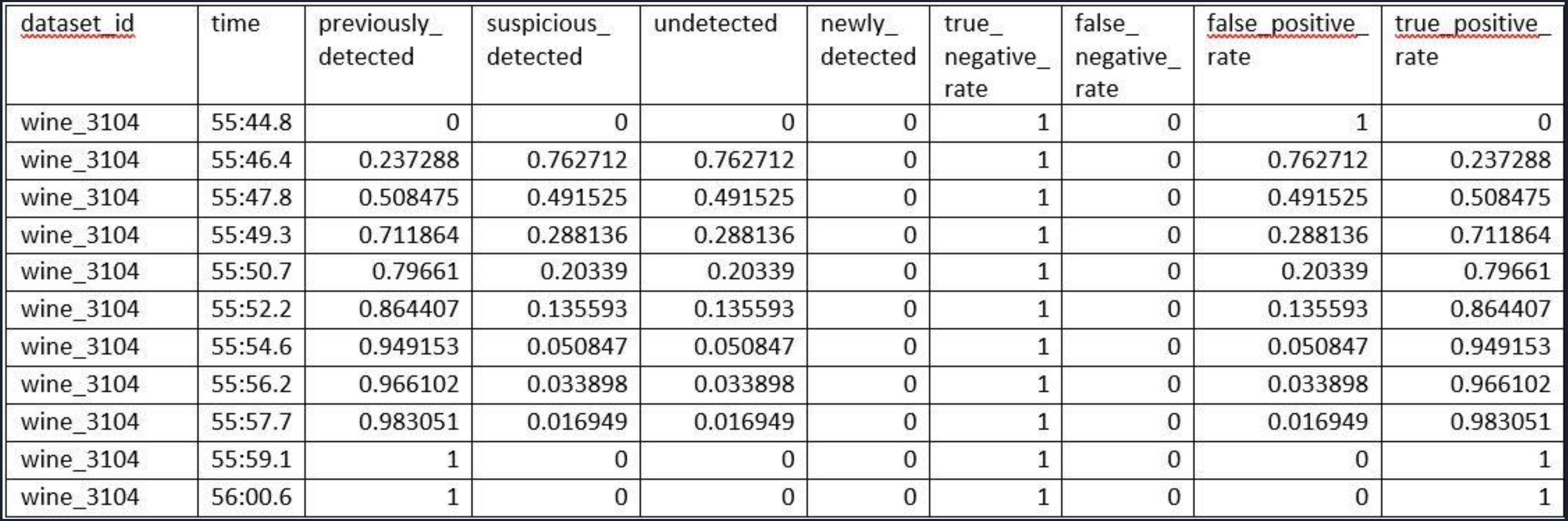

OLD
(23%)
DIFFERENT
PROBLEMS
(48%)
WEIRD
RESULTS
(13%)
GOOD RESULTS
(16%)

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
- This way we have don’t have to create these graphs manually when we write our final report
- This semester I have created three different python scripts that can plots graphs with each script having different functionality

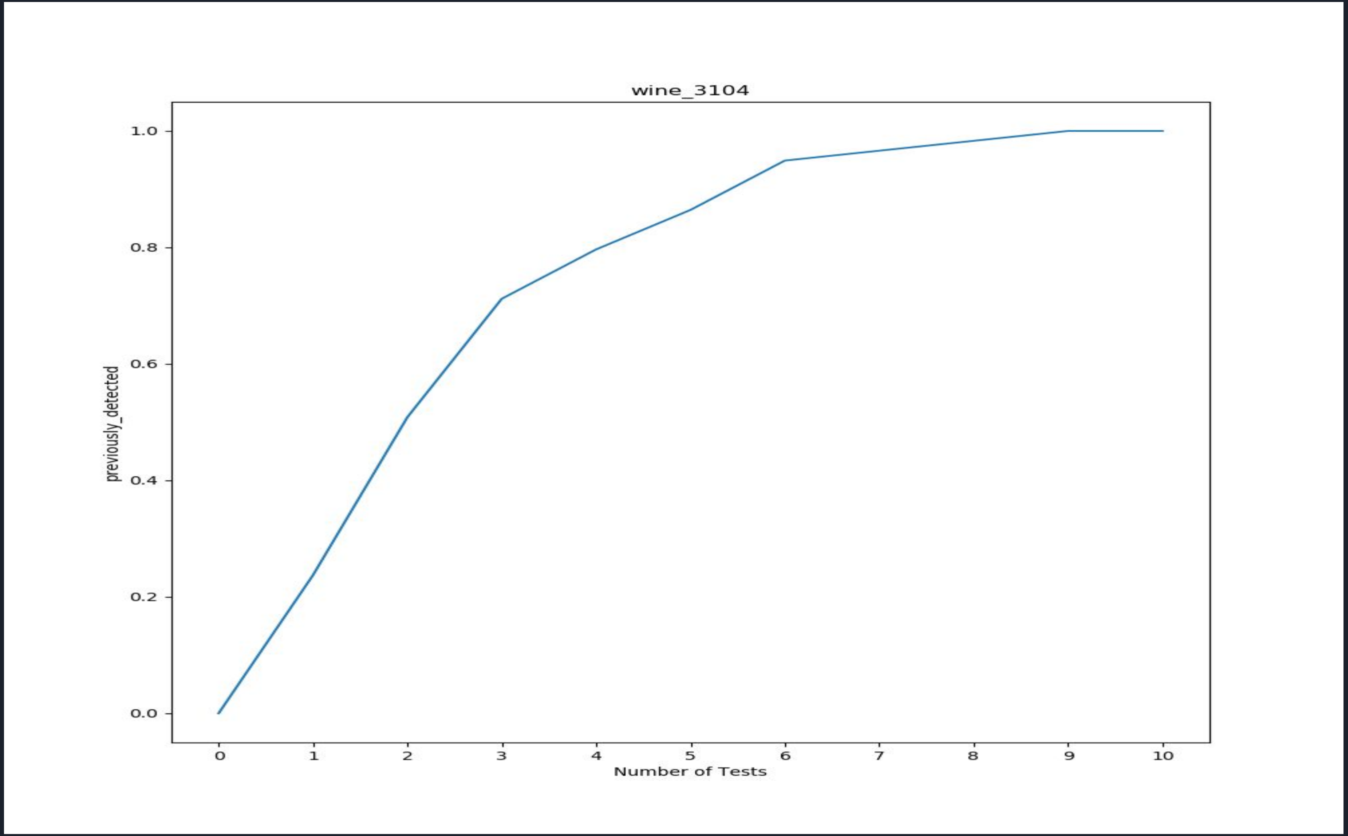
First Script (plot.py)

This script uses different libraries like pandas, DataFrame. Matplotlib and CSV to successfully create the graphs of these datasets based on columns.



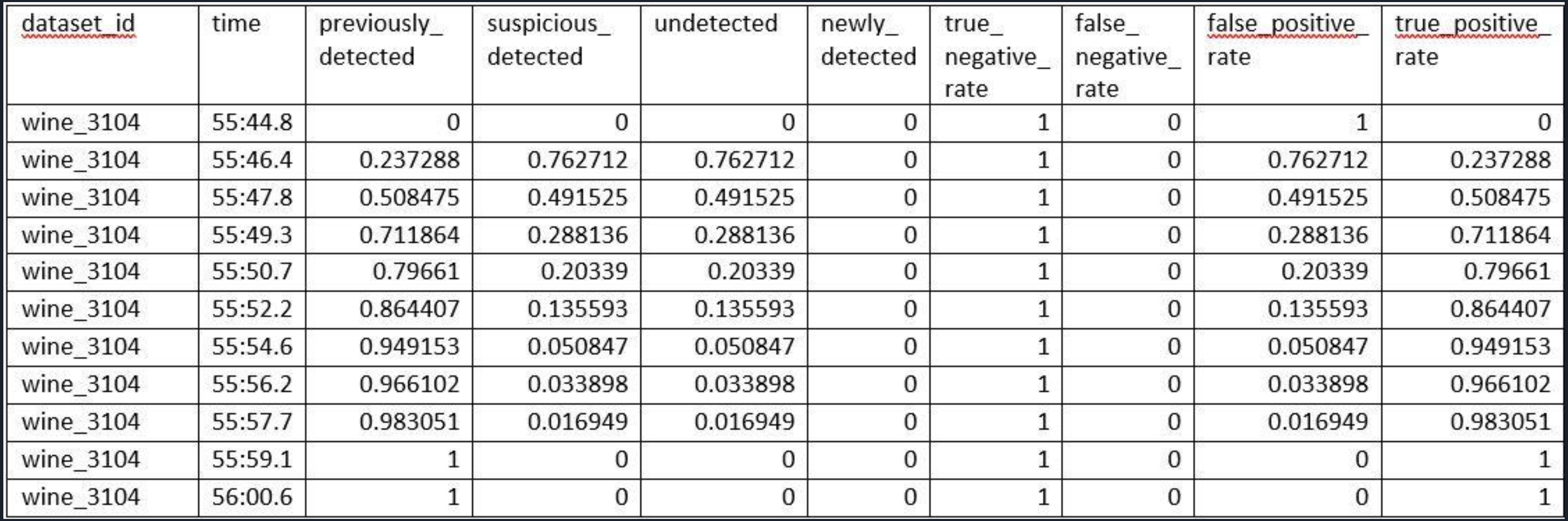
dataset_id	time	previously_detected	suspicious_detected	undetected	newly_detected	true_negative_rate	false_negative_rate	false_positive_rate	true_positive_rate
wine_3104	55:44.8	0	0	0	0	1	0	1	0
wine_3104	55:46.4	0.237288	0.762712	0.762712	0	1	0	0.762712	0.237288
wine_3104	55:47.8	0.508475	0.491525	0.491525	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

Graph created by plot.py which shows the Previously detected attribute of the Wine dataset.



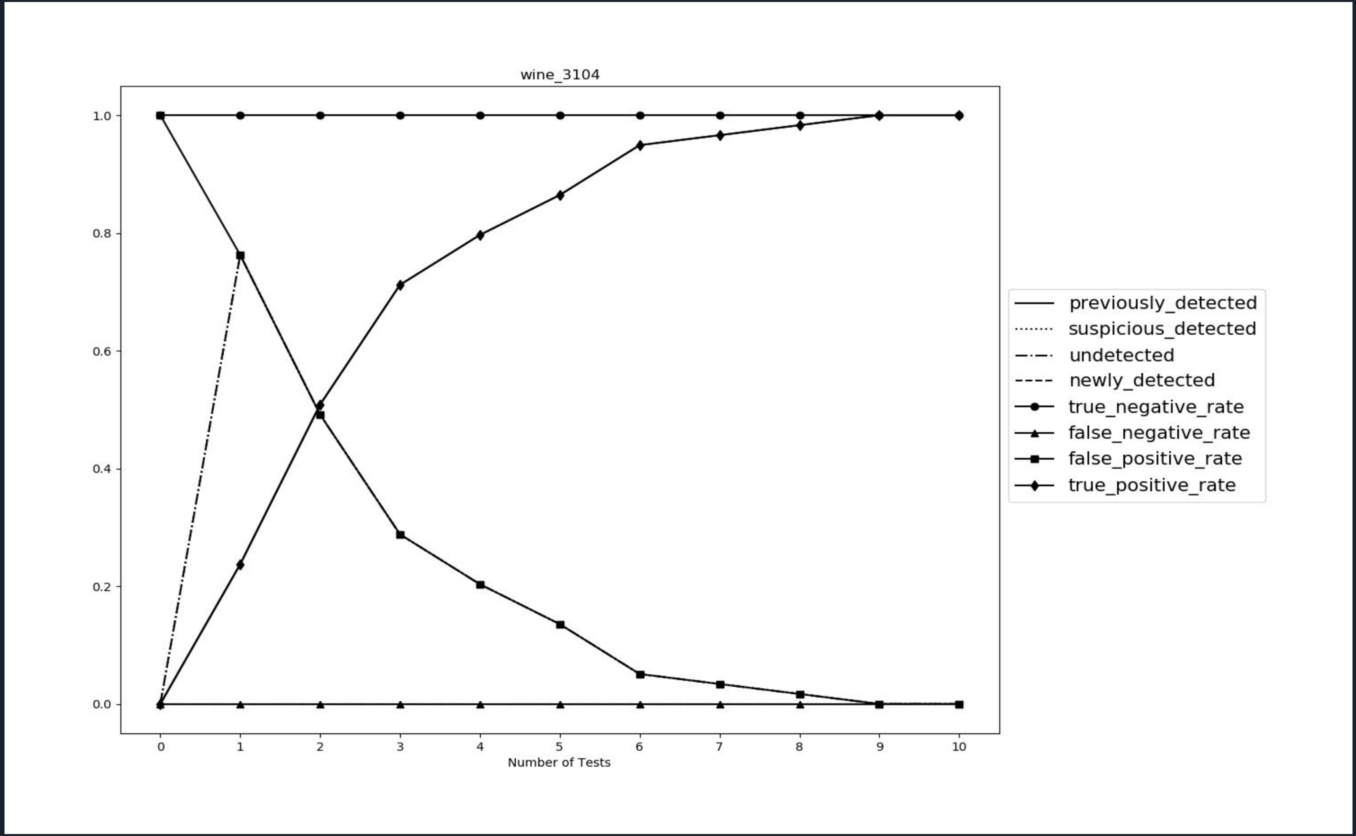
Second Script (plot_merge.py)

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.



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

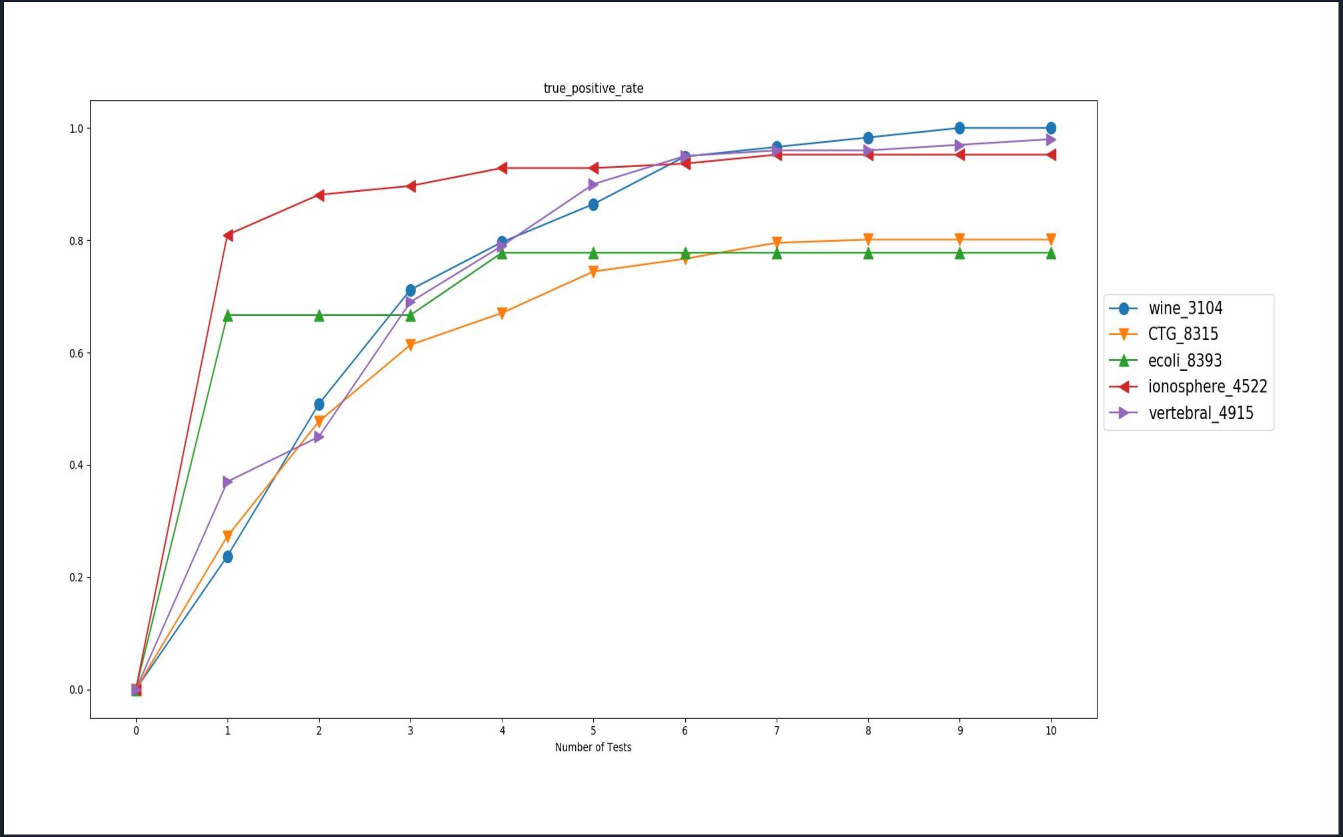
Graph created by plot_merge.py which shows the all the attributes of the Wine dataset.



Third Script (plot_trend.py)

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.

Graph created by plot_trend.py which shows the True Positive Rate attribute of the Wine, Cardio (CTG), Ecoli, Ionosphere, and Vertebral dataset.



Things I Learned/Gained Over the Semester

- Expand my connections in the CS Department
- Improve Python Skills and learned to use different libraries in Python
- Got some brief insight of Machine Learning like nodes and layers
- Handle big data and how to preprocess those data.
- Skill of analyzing these datasets and their results.
- Improved my time management skill, with all the things going on this semester. (Taking 18 credits is not easy to manage)

Things that didn’t go well….

- So much time spent on setting up the tool
- Was not able to give enough time as I also had to manage my other classes
- Faced problems which I had no clue about

Next Semester….

- Confident that I will be learning a lot, as I already knows how different things work
- Taking fewer credits, so it will give me extra time to focus on different research tasks

QUESTIONS

THANK YOU