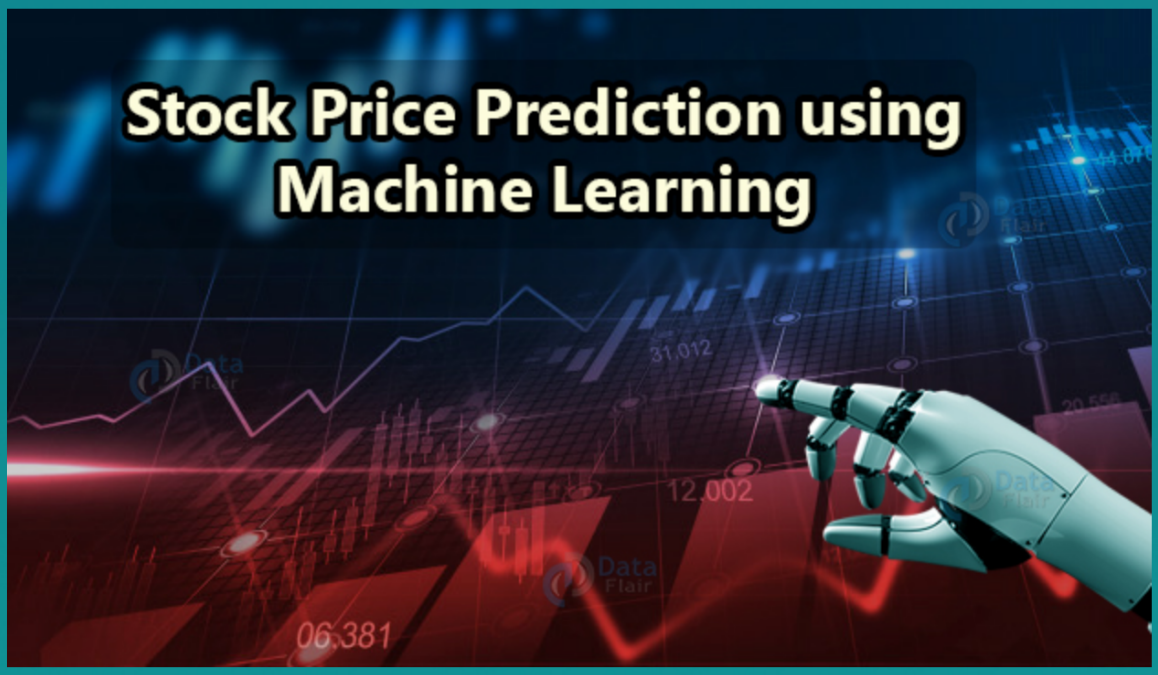
CPTS 315

Course Project Report

A RUDIMENTARY ANALYSIS OF THE STOCK MARKET

CHARLES NORDEN, 011606177
ABHIMANYU RAJ PUT, 011691878



CHARLES NORDEN, 011606177
ABHIMANYU RAJ PUT, 011691878

A RUDIMENTARY ANALYSIS OF THE STOCK MARKET

Introduction

The generation and accumulation of financial data has reached an all-time high thanks to the age of economic globalization and the convenience of digital technology. The capacity of a person to manually evaluate the rapidly increasing amount of data has far outpaced that of a computer. Owing to long-term patterns, cyclical fluctuations, seasonal variations, and erratic movements, financial time series data are more complex than other statistical data.Many external factors, such as many highly interrelated economic, political, social, and even psychological factors, have a significant impact on these. The constant growth of such highly fluctuating and erratic data has highlighted the crucial need for more automated approaches to efficiently analyze such vast financial data and extract meaningful statistics. Data mining has carved its own niche in financial time series analysis as a method of uncovering valuable secret information. [1] It paves the way for investors to make constructive, data-driven decisions that lead to successful gains with lower investment risk.

An investor's primary aim in the stock market is to make a large profit. In a stock market, there are numerous investment opportunities such as trading (i.e. purchasing and selling) stocks, securities, foreign exchanges, and precious metals, among others. One of the most common forms of financial investment is stock trading. Stock market investors can increase their profits by purchasing or selling their investments at the right time. Finding the best trading time with the least amount of risk is the secret to making a lot of money in stock trading.However, because of the stock market's highly fluctuating and dynamic nature, determining the best time to buy or sell is often difficult. Most researchers are interested in technical metrics to track stock prices and to assist investors in setting up trading rules for buy–sell–hold decisions. On the basis of historical stock data, technical indicators are developed. As a result, trading decisions based on specific technical metrics are not always more profitable. Various data mining and artificial intelligence techniques have been used to

analyze data in the literature.Apart from scientific and fundamental research, machine learning and its implementations have been crucial in financial analysis. In recent years, machine learning (ML) has found widespread use in finance, and it has proven to be an effective method for predicting stock market direction. ML algorithms, one of the main methods, rose to prominence in finance in the early 1980s, when they were first used for financial forecasting and stock price analysis. [2] This method is a broad subgroup of Artificial Intelligence, and these algorithms have been used in portfolio optimization, stock betting, and credit lending, in addition to stock price behavior prediction and stock analysis. The question our team had before we started on this project were how can we implement the Machine leaning algorithm when it comes to the stock prediction. What are challenges in building the stock market anlysis and Prediction and what should be the datasources for such a project. When it comes to the challenges and aprroach, challenge was to obtain this kind of data and converting it into the form we wanted. When it comes to the approach we used the data and then converted into numpy array of vectors and then used this data in a standard perceptron classifier. Results include graphs, bar graphs and data generators for timeseries, technical indicators , fundamental data etc. Output also includes the weight vector.

Data Mining Task

Input Data

We used the Alpha vantage Api to get our input data related to the stcks of various companies. Time Series Stock,Technical indicators and the Sector Performance API’s were used by our team to implement the futher tasks. Technical indicators are valuable methods for furthering technical research. These metrics assist investors in making stock buying and selling decisions, allowing them to gain a clearer understanding of market action by deciding which stocks to purchase and which stocks to sell, as well as when to do so. We collected the Time Series data and

the Technical Indicators at a time interval of sixty minutes using the API calls into pandas dataframes which was later converted into csv files related to respective companies. This data was later vectorized to be used into the perceptron classifier. When it comes to the Sector Performance data it was solely used to plot the graphs and threads for the microsoft company.

Task Details

At first we focused on getting the datasets that we are required to use in this project, as we were building this project from scratch. After doing some research we found out that we can use differet API's to get this type of data for different companies.

The first stage of the final project plan was for us to read through the literature on this specific issue and industry-tested solutions. We looked at papers that described how they used data mining and classifier-based algorithms to make predictions and create tools that helped with stock market analysis, understanding the current state of the market, suggested algorithms, and the challenges that come with trying to effectively analyze and forecast stocks and the cryptoworld. After this the task was to make a classifier which was in a working condition with good accuracy for the provided dataset. We use ensemble learning algorithms to forecast the trend in stock market price change. Ensemble learning algorithms or techniques incorporate many machine learning algorithms into a single predictive model to provide better predictive results than any single model could provide. The main aim of employing these techniques is to address modeling issues related to time series forecasting and, as a result, improve the machine learning algorithm's consistency and accuracy, as well as achieve better results.After researching a bit about it we came to know that for predicting market prices and movements, Multi layer perceptron (MLP) and Artificial Neural Networks (ANN) are commonly used, each with different method for learning patterns and then predicting them. So we decided to use the perceptron classifier algorithm to learn patterns and predict as it was the one thought to us in class in a detailed manner and also it was close to the MLP used in the real time industries.

So far on the knowledge our team has acquired we divided the work into smaller chunks:

- Figur out how to get data sets related to the current problem.
- Convert the data ino vectors
- Use the vectorized data ro ghet the predictions.
- Choose the best-performing choice and output the test data's expected labels.

After that, we evaluated the performance in order to obtain a clearer understanding of the outcomes of each of these tasks.

Questions

As we worked on the coding for this project, we had a few more questions. Why was a certain classifier the best, and is it still the best for this specific classification problem or other word-based problems of this nature? I was also curious as to how we can tell how conservative a particular classifier is when it comes to analyzing and forecasting stocks. The majority of these questions emerged from attempting to find out how to code and complete the tasks mentioned above. Although they weren't directly linked to the outcome of my project, they did provide us with some insight into the thread I should pursue to enhance our understanding of industry implementations.

Challenges

Due to non-stationary, blaring, and volatile data, stock market prediction is a major challenge, and as a result, investors find it difficult to invest their money for profit. Several methods for predicting stock market movements have been devised in the current techniques. We struggled a lot in trying to understand and read through the documentations of the libraries I considered in order to help me achieve my project goals, as it is for any project that relies heavily on a system you don't fully understand. Many of Python's packages are open source, but there are few examples of how to use them on which to base our implementation. So it was a struggle to find the best library that could

both do what we needed and had documentation describing how to use it. Another challenge was to figure out how to convert the data we obtained from the API's will be convertd into a numpy array of vectors. Another challenge as to decide which algorithm to apply to get the best result.

Technical Approach

At first we used the API calls to obtain the data and later ploted the Timeseries, TechIndicators, SectorPerformances and also used this data to implement the threads. These plots helped us analyze the trends and worked as tools for us. We had different algorithms in our minds to proceed with

- Market Basket algorithm
- Perceptron classifier
- Reccomender's system
- Random Forest classifier.

We decided to approach the implemntation of our idea by chosing the perceptron classifier. At first in this approach we got our data using the following Function calls:

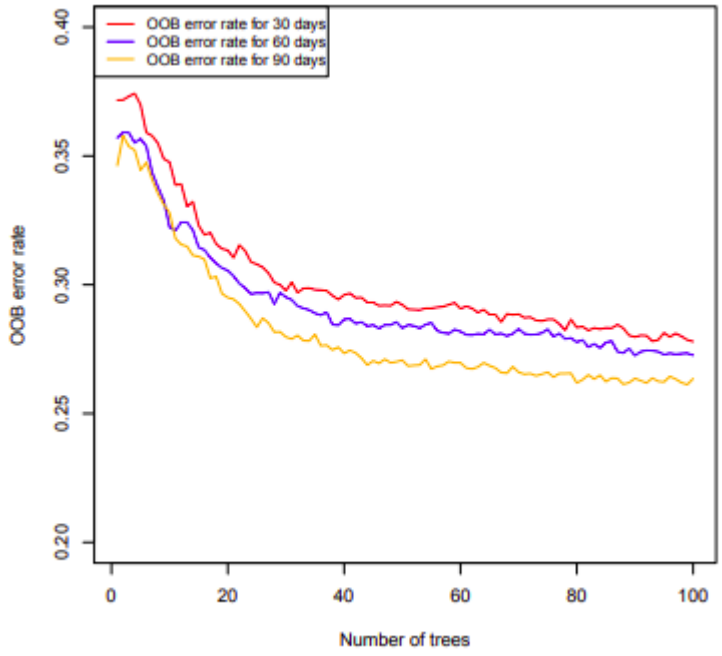
```
ts = TimeSeries(key=f"{MY_API_KEY}", output_format='pandas', indexing_type='date')
tsdata, tsmeta_data = ts.get_intraday(symbol=equity, interval='60min', outputsize='full')
```

Here the function calls the timeseries API function, in the output format of pandas(other options like json and csv were also available), with indexing equal to dates, and also it has been set to intraday with time interval of 60 mins. Now we converted the data(raw lists of lists) into numpy array of vectors using the following snippet of code.

```
def convert2Vectors(raw: list):
    """
    this function converts a raw list of lists to a numpy array of vectors

    :param raw [list]: the raw input
    """
    vectors = list()
    for entry in raw:
        vectors.append(entry[1:-1])


    # flip the table to start from the oldest date and end with the latest entry
    vectors = list(reversed(vectors[1:-1]))


    """
    next we need to normalize the vectors:
    """
    # convert to numpy array for ease of processing
    vectors = np.array(vectors, dtype='float')

    v_end   = vectors[-1]
    v_start = vectors[0]
    normed  = (vectors - v_start)/abs(v_start - v_end)

    return normed, normed[0], normed[-1]
```

After the data is converted into the vectors we can feed this data as input and feed it to the standard perceptron to get the desired output by training the dataset.

Another approach we thought of implying but didnot imply was the Random Forest classifier.In this method of implementation, First the response variable is transformed into a binary variable. When we have a binary result, RFC is used to predict the course of the AAPL stock price. Exponential smoothing, on the other hand, has been recommended in previous studies to improve the accuracy of ensemble learning algorithms. As a result, since the smoothing factor is expected to have a major impact on the model's efficiency, the RFC results are divided into three parts, each of which investigates the effect of the smoothing factor (alpha α) with three different values. (0.0095,0.2,0.95). We look forward to apply this method in the future, after a significant amount of experience.



(Alpha (α)= 0.2)
Figure 1

Evaluation Methodology:

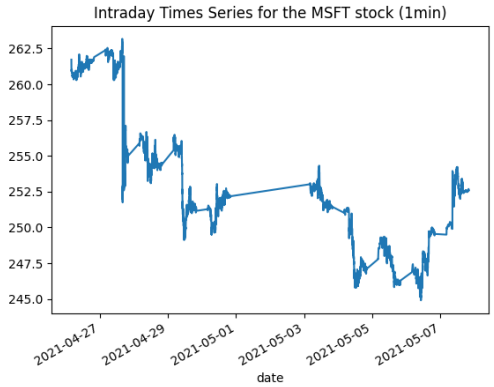
The input dataset we used was obtained from the Alpha Vantage Api, it contained Time Series , sector performance and Technical indicators. The Timeseries data gives the information regarding the open, high, close , low and volume, whereas the Technical Indicators gives informtion regarding different bands.

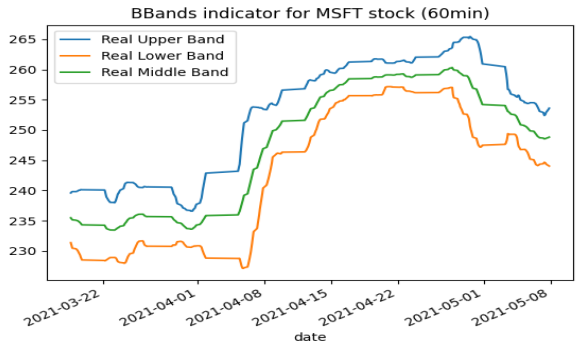
Figure 2

```
                     1. open   2. high    3. low  4. close  5. volume
date
2021-05-07 20:00:00  252.6800  252.6800  252.6000  252.6000       216.0
2021-05-07 19:52:00  252.6899  252.6899  252.6899  252.6899      1035.0
2021-05-07 19:51:00  252.6899  252.6899  252.6899  252.6899      3025.0
2021-05-07 19:48:00  252.5500  252.5500  252.5500  252.5500       210.0
2021-05-07 19:47:00  252.6000  252.6000  252.6000  252.6000       623.0
...                       ...       ...       ...       ...         ...
2021-05-06 10:28:00  245.6700  245.7068  245.4800  245.5900     61545.0
2021-05-06 10:27:00  245.6400  245.7300  245.5000  245.7000     59332.0
2021-05-06 10:26:00  245.5700  245.6800  245.4700  245.6300     66594.0
2021-05-06 10:25:00  245.2600  245.6100  245.2100  245.5700     87881.0
2021-05-06 10:24:00  245.1300  245.3000  245.1100  245.2500     48990.0

[1000 rows x 5 columns]
                     Real Lower Band  Real Middle Band  Real Upper Band
date
2021-05-07 20:00:00           244.0411          248.8219          253.6027
2021-05-07 19:00:00           244.0892          248.7803          253.4714
2021-05-07 18:00:00           244.1238          248.7594          253.3950
2021-05-07 17:00:00           244.1630          248.7364          253.3099
2021-05-07 16:00:00           244.2062          248.7096          253.2130
...                                ...               ...               ...
2021-03-17 20:00:00           232.5926          235.8827          239.1728
2021-03-17 19:00:00           232.5428          235.8349          239.1269
2021-03-17 18:00:00           232.4162          235.7689          239.1215
2021-03-17 17:00:00           232.2688          235.6962          239.1236
2021-03-17 16:00:00           232.1799          235.6450          239.1102

[581 rows x 3 columns]
                      Rank A: Real-Time Performance  ...  Rank J: Year Performance
Energy                                       0.0189  ...                   -0.2728
Real Estate                                  0.0121  ...                       NaN
Industrials                                  0.0105  ...                    1.7075
Materials                                    0.0093  ...                    1.2524
Consumer Discretionary                       0.0080  ...                    3.4625
Information Technology                       0.0078  ...                    4.7378
Health Care                                  0.0072  ...                    2.5224
Communication Services                       0.0061  ...                    0.9224
Financials                                   0.0054  ...                    1.8948
Utilities                                    0.0025  ...                    0.9884
Consumer Staples                             0.0001  ...                    1.2394
```

When it comes to the output the vectorized input when feeded to the standard perceptron algorithm we get the trained weight vectors. Due to our limited knowledge we were unable to decide what would be the best metric to evaluate our output.

Results and Discussion





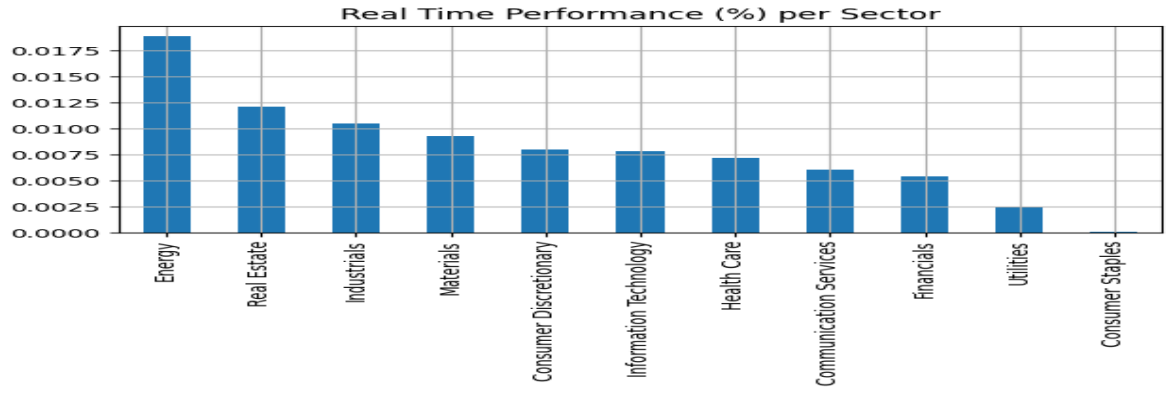


Figure set 3

The above images are the output of the mp_stockdata which represents Microsoft data graphically.

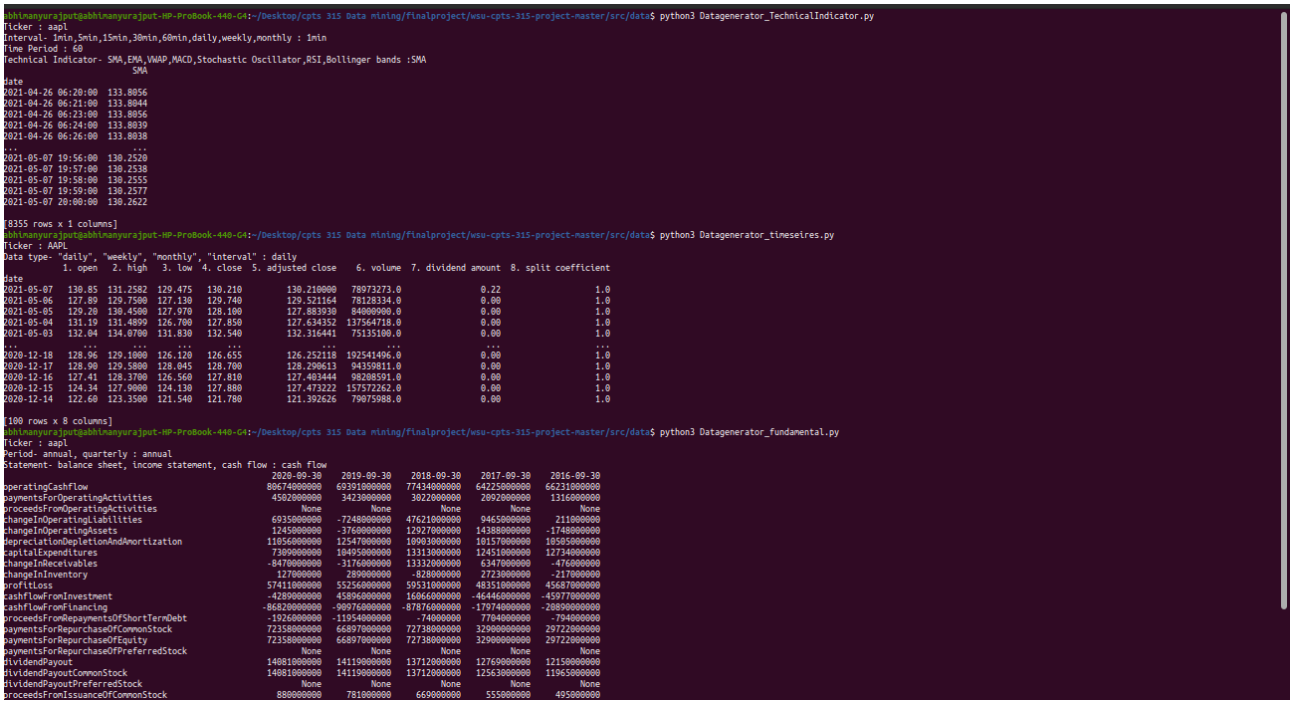


Figure 4

These are the datagenerators which can provide the data related to any company as per the inputs given by the user such as ticker, time interval , time period etc.

When it comes to the components of the project that worked well everthing worked, the plot code, the data generator code , vector generator code and also the perceptron algorithm. The data generator code was able to generate data as per the inputs given by the user, we thought that creating something like this would be really helpful for any individual who wants to learn about the stock market using Datamining. The plots and bar graphs were matching the expectations and helped us analyze few things while implementing the algorithm. The perceptron algorithm worked well enought with vectorized dataset as input producing the weight vector.

Whereas when it comes to what didnot go well, we were unable to evaluate the correctness of our output due to our limited knowledge.

Some of the question we researched were:

1.) What type of data sources would be best to implement this idea with?

For our research, we'll need two sources of data: first, news sentiments, and second, historical prices. From January 1, 2008, to December 31, 2017, Reuters tick data and news data for five separate stocks were collected over a ten-year period.[2] AAPL stands for Apple stock, GOOGL for Google stock, AMZN for Amazon stock, and FB for Facebook stock. As a result, a tick is a unit of measurement for the smallest upward or downward price movement. In certain instances, a one-second timeframe requires a large number of ticks, up to 30.Tick data was gathered, which included the open bid, close bid, high bid, and low bid, as well as the time stamp. This high-frequency data is gathered in order to perform intra-day short-term forecasting. Since we group our data hourly, our model needs at least one tick to be released every hour. This massive amount of data necessitates some preprocessing to account for the large volume of data (7 trading hours * 3600 = 25200 tick prices per day) as well as the difference in tick intervals.Multiple prices may be released at the same time in tick data, and some ticks may be missed in other seconds.

2.) What are challenges in building the stock market anlysis and Prediction ?

Since market assumptions shift over time, models can easily go from useful to useless. Since this is a highly competitive field, any valuable knowledge will be gained in the private sphere and will almost never be made public. Accurate data for training can be costly to acquire, and public data is largely worthless for serious purposes. Thousands of covariates are used in accurate models, and efficient models require training.

Lessons Learned

By allowing our software to search through hundreds of entries for stocks to forecast, it was able to find a few that performed well during the validation period. There was no overarching trend to speak of. We had cherry-picked a few cases in which the model had simply gotten lucky by chance. Machine learning cannot predict a random series, and when training models, you must be aware of your own biases. Validation must be done with care. We are confident that we will not be the last to succumb to the allure of the attic's old treasure map, but proceed with caution. If you want to read, there are far less random time series to play with. Simulate, double-check your assumptions, and be mindful of your own prejudices.

Bibliography

1.) Rajashree Dash(2016, March). A hybrid stock trading framework integrating technical analysis with machine learning techniques. Retrieved from https://www.sciencedirect.com/science/article/pii/S2405918815300179

2.) Mariam Moukalled. Automated Stock Price Prediction Using Machine Learning. Retrieved from https://www.aclweb.org/anthology/W19-6403.pdf

3.) Dattatray P. Gandhmal (2019, November). Systematic analysis and review of stock market prediction techniques. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S157401371930084X#:~:text=Stock%20market%20prediction%20is%20a,predict%20the%20stock%20market%20trends.

4.) Dharmaraja Selvamuthu(2019, March 21). Indian stock market prediction using artificial neural networks on tick data. Retrieved from https://jfin-swufe.springeropen.com/articles/10.1186/s40854-019-0131-7#:~:text=Support%20Vector%20Machines%20(SVM)%20and,learning%20patterns%20and%20then%20predicting.

Link to github Repository:

https://github.com/nkarl/wsu-cpts-315-project