Stock Market Analysis Using Twitter Sentiment

Apoorva Kshirsagar
Computer Science
University at Albany
001348110
akshirsagar@albany.edu

Abstract— Using social media has reached an excessive degree, where twitter is called the most popular micro-blogging service which permits users to express their opinion approximately an occasion. in this project, we used numerous strategies to evaluate sentiment of tweets and then locate correlation with the stock rate movement the use of visualization. those duties had been carried out by means of mining tweets the use of twitter's API and then system similarly for analysis. For tweets sentiment techniques have been used Vader sentiment analysis and support vector machine and the result pondered that Vader sentiment changed into giving better result and we continue further with it. subsequent step changed into to mine tweets which changed into used to taking records immediately from national stock exchange. Later both sentiment and stock values were merged into a identical record in which we are able to without difficulty find correlation between two values.

Keywords—stock market analysis; sentiment analysis, twitter; Prediction

I. INTRODUCTION

In the present year expansive measure of information is transmitted utilizing diverse online networking stage and this information contains data about each occasion or point, twitter being the most well-known out of every single social medium destinations which gives smaller scale blogging administration. Many clients sign in every day to tweets about any occasion they need to express their emotions about and twitter gives that stage to them. Each tweet contains maximum of 140 characters, each tweet may not be helpful but rather still we can analyze the assumption from those tweets about how the temper of public relating that topic is.

In the field of Data Sciences, it has turned out to be well known to gather information from online networking site and process that information and later investigation it is utilizing diverse strategies to extract applicable information.

In the stated project, information from stock market is expected to make forecast about the stock costs. There are distinctive elements engaged with prediction of stock costs and in addition tweets sentiment and in this undertaking, we utilized diverse methods to arrange information and analyze it, gathering tweets is by a wide margin first imperative and after that later discovering correlation between tweets sentiment and stock prices.

II. MOTIVATION

When reading about the stock market and predicting the rate through historical data, turned into simplest alternative and later generation modified and debunked the use of historical examine of market - as stock market is fluctuating on large basis. The Efficient Marketplace Hypothesis (EHS) states that stock marketplace relies upon on certain key factors the ones are modern-day event, assessment & score of product can be impact stock value of employer and contradicting to it that those factors can be or most probably be inaccurate and consequently market price can't be predicted (at most 50% accuracy).

With the progression in innovation in the field of web and web-based social networking there rose an online element where individuals spend many hours to connect with each other and one of that web-based social networking stage is 'TWITTER'. It is a continuous data organize that interfaces clients to the most recent data about the subject intriguing them and take after the specialists or other individuals or anybody where they tweet and share their experience.
Our purpose is to develop a system which pursuits on predicting the future price or market value of stocks by way of taking account users feelings after which determining about the stock buying or selling.

III. PROBLEM STATEMENT

Using social networking to get to tweets and find sentiment analysis of each tweet and predict the stock cost of companies, market and stock indexes to recognize the opportunity of exchanging. Classify polarity of every content record, sentences or highlight level and decide if opinion is positive, negative or nonpartisan Making a recommendation framework for client with the goal that our framework can prescribe client about which organization ought to develop.

In conclusion, utilizing cross-validation and regression to analyze both the sentiment of tweet and stock cost of and discovering relationship amongst opinion and stock costs.

IV. SIGNIFICANCE OF PROBLEM

The finding and application of this project will be redounding to the benefit of people who invest a lot in share market and this will provide an important role that twitter can be used for doing such a task which further benefits computer science world and especially data science world who is currently taking over every technology.

V. RELATED WORK

The most well-known publication in this area is by Bollen. They investigated whether the collective mood states of public (happy calm, anxiety) derived from twitter feeds are correlated to the value of the Dow Jones Industrial Index. They used a Fuzzy neural network for prediction. Their outcomes show that public mood states in twitter are strongly correlated with Dow Jones Industrial Index. Chen and Lazer derived investment techniques by using looking at and classifying the twitter feeds. Bing et al. studied the tweets and concluded the predictability of stock values

primarily based on the kind of industry like Finance, IT and so forth.

Zhang found out a high negative correlation between mood states like hope, fear and worry in tweets with the Dow Jones Average Index. These days, Brian et al. investigated the correlation of sentiments of public with stock growth and decreases the usage of Pearson correlation coefficient for stocks. in this paper, we took a singular method of predicting rise and fall in stock costs primarily based on the sentiments extracted from twitter to locate the correlation. The middle contribution of our findings is the improvement of a sentiment analyzer which works higher than the only in Brian's work and a novel approach to discover the correlation. Sentiment analyzer is used to classify the sentiments in tweets extracted. The human annotated dataset in our paintings is likewise exhaustive. we've got shown that a strong correlation exists among twitter sentiments and tomorrow stock values in the results section. We did so by way of thinking about the tweets and stocks establishing and remaining charges of Microsoft over 12 months.

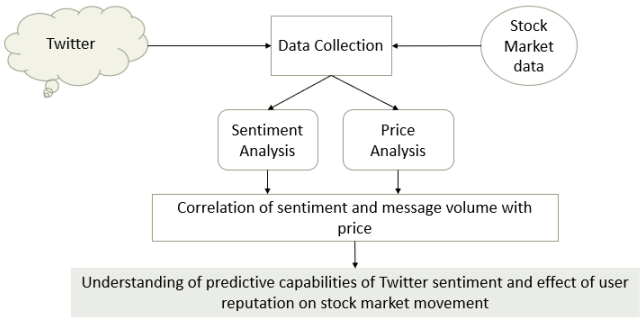
VI. Proposed Approaches

The problem statement stated before turned into briefly discussed, right here on this section we can talk about how to address the ones trouble and could speak approximately proposed answer for hassle statement. Our undertaking particularly has goals.

1. Collect data from twitter and finance data from national stock exchange.
2. Process and clean tweet corpus.
3. Generate feature vector and train data from given text document.
4. Analyze and predict tweets sentiment using support vector machine and Vader sentiment analyzer techniques.
5. Merge data for both tweets sentiment and stock data.
6. Predict the stock market trend from twitter sentiment using linear regression technique and fining correlation between sentiment and stock values.

VII. System Design & Implementaion

A. Architecture



The architecture design for the system is simple and robust which does its purpose without any problems. This architecture design was chosen for the project because it's far most commonly used layout and has simple and easy to enforce capabilities and libraries which will not motive any hassle whilst jogging on older computer versions..

B. Datasets

Our system particularly includes datasets, one from twitter which can be the tweets for companies which had been stored in a separate csv file, tweets are used for reading emotion of customers approximately companies and how market is speaking about it, and next dataset is from national stock exchange which presents information approximately open and close price of businesses, in each datasets undertaking are finished one at a time and analyzed until they may be ready for merging, then these two datasets are merged as a single csv record in which final prediction is accomplished and later correlation is located among them..

C. Major Component

Major components for our system are:

a) Tweets Collection: For tweets collection, Twitter provides sturdy API, there are two methods to get this completed through Twitter streaming API and Twitter REST API, we used REST API because it lets in to find tweets related to a query of recent tweets. The request of JSON item contains the tweets and their metadata, which incorporates many statistics that is time of tweets, place where tweets was written, retweets etc., our essential recognition changed into on tweets textual content and time while it became made. API requires users to have API key which may be obtained by means of twitter developer website. The textual content of tweets carries an excessive amount of extraneous words which are not do not forget in a part of sentiment they're specially URLS, tags, to get correct sentiment we have to filter out the ones tweets or dispose of noisy words.

Table I. Twitter Example

Company Name	**Tweet status/Text**
Reliance	Merger with Idea not because of Reliance Jio: Vodafone CEOhttps://t.co/SID7WAQ4jn -via @RajivMessage British Rulers Leaving Bharat
Bajaj	#excise Bajaj Auto Ltd. Versus Union of India through Secretary, Ministry of Commerce and Industry, Government ofâ€¦ https://t.co/SQHsxlTMfb
TATA	RT @up100: Tata Consultancy Services (TCS) Officers visiting at "UP-100" Bhawan Lucknow @uppolice @anilarch https://t.co/JDQ3ot0IOv
Hero	RT @HeroMotoCorp: Coming up...a major announcement Hero MotoCorp #FIFA U-17 World Cup India 2017 https://t.co/6Ae3mV6IAi

b) Tweets Pre-Processing: There are number of steps to achieve this as follows:

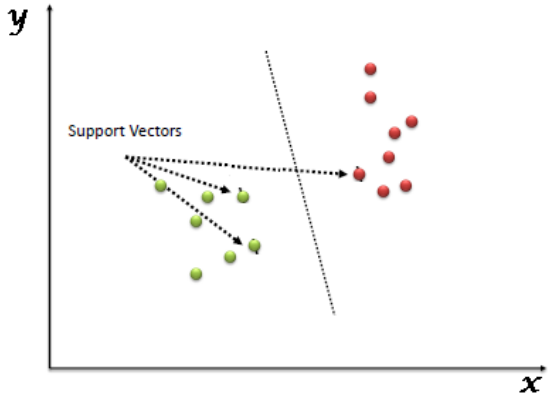
- First step is to split the text by space, which forms a list of words for each text and they are called feature vectors or most occurring word in a tweet, which will be used later to train data for support vector model,
- Next step is to remove stop words from tweets, python library known as NLTK is used for this purpose.
- Stop words contains articles, punctuation, and few other words, which do not pose any sentiment in a tweet and should be removed. Stop words list is stores in a dictionary which check each tweet with the stop words and if tweets contain those words it will be removed immediately, and tweets will be filtered.
- Tweets also have some extra symbols like "@", "#", and URLS, any text next to "@" symbol is a username of user who is writing a tweet which does not add any purpose in sentiment and therefore should be removed or stored in different file.

- *Sentiment Analysis:* Sentiment analysis was first major part of our system as this will allow us to compare stock price, we approached this with the aid of two approach out of which is one could be very famous called support vector machines and different as Vader sentiment analyzer. Sentiment analysis is straightforward method of figuring out whether a chunk of textual content is positive, negative or impartial. most important methods takes one in all two form, polarity primarily based or valance based totally, in which intensity is taken into account. for instance, the phrases 'excellent' and 'high-quality' might be treated the identical in a polarity-based totally method, whereas 'superb' would be treated as more tremendous than 'precise' in a valence-based method. both of these have their personal characteristic which lead them to very powerful tool for sentiment evaluation. we will discuss approximately both in detail.

- *Vader sentiment analyzer is known as Valence aware dictionary and sentiment reasoner*. it's far and effective tool in python. It belongs to the type of sentiment evaluation that is based totally on lexicons of sentiment-associated words. in this approach, each phrase in the lexicon is rated as positive negative. Vader analyze a text after which take a look at for words which can be in lexicon and that they have a rating that is given automatically. Vader produces four sentiment metrics from those phrases ratings which might be positive, negative, neutral and compound. Vader is incredible for social media textual content which that is due to the fact tweet textual content contains words or symbols which can produce needless information and can trade sentiment of tweet from one manner to other, Vader handles this sort of trouble by means of adding such symbols in lexicon. firstly, Vader bundle become hooked up into the project folder and then the use of one function referred to as SentimentIntensityAnalyser, it is an item from Vader package. in the end, we can use polarity_scores() technique which again from Vader sentiment which presents metrics for a of textual content.

```
def sentiment_cal(tweet):
    value = SentimentIntensityAnalyzer()
    score = value.polarity_scores(tweet)
    score = float(score['compound'])
    return score
```

- Support vector machine (SVM) is a supervised machine learning algorithm, which can be used for both regression and classification, but it is specially used in classification problems. inside the SVM model we plot statistics points on a n-dimensional space wherein n represents the quantity of capabilities you've got and then we carry out type by means of finding the hyper-plane that differentiate two classes.



- Support vectors are best way to separate two classes. Scikit-learn is a widely used and popular python library. collect a big tweet corpus is the most mission as we want to split classes in there, after collecting big set of tweets we need to manually label tweet textual content and store them in a text report. After amassing tweets, we ought to build and teach a classifier for sentiment evaluation, for classifier we then find function vector for every piece of textual content. To manner every text, we procedure and find meaningful feature whose frequency is greater than 20 and create an array call X= [] and y= []. next, we use 10-fold cross validation to decide fine w and b for the help vector equation to be able to determine the how two magnificence may be get separated. similarly, we want a testing facts set on the way to assist us to investigate records from education set and expect tweet sentiment. ultimately, we used metrics category record to generate result for precision and consider.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	14
1	1.00	1.00	1.00	301
avg / total	1.00	1.00	1.00	315

With this we finish our sentiment evaluation in which we find that using Vader is better as it's miles an inbuild function of python and we don't need to manually work on documents and it saves masses of time, plus the usage of SVM we should train data a good way to fail if we had much less quantity of information to be had.

c) Stock price analysis: We simply took data from National stock exchnage and proceed further with it, as NSE data is robust and we can trust it without any issues as that data is used by many users and companies.

d) Correlation: Correlation, in the finance and investment industries, is a statistic that measures the degree to which two values pass with regards to every different. Correlation is calculated that's recognise as correlation coefficient, whose price falls among -1 to 1. 1 suggest a super positive correlation this means that that if one asset moves up or down, different asset will circulate in lockstep with it, inside the same route. -1 manner a super bad correlation which means that if one asset flow up or down, different asset will flow in contrary route from it, and there's a zero correlation sometimes which means there's no relation between two.

This become the final step in our system, after completing all the records collection and pre-processing, we must merge both the data from sentiment evaluation and NSE statistics which is pretty a rigorous assignment as both the records does now not have any issue in common except the date, so we decided to discover some relation among statistics and were given successful. Stock Market opens at 9AM and closes at 4PM, so we made a easy calculation that we can computer common sentiment for each day this is when is the first tweets came in morning approximately 9AM and which was the final tweet for the equal organisation that is 4PM, we later calculate open_score and close_score for tweets that have been written among those timeline after which it have become without difficulty to relate two document.

e) Implementation Detials: Our application has mainly four files which needs to be run one at a time. The program will run in following manner.

- Get tweets from twitter using REST API, and store data in csv file which is easy to work on, the csv file contains time, username, tweet text, company name, sentiment, confidence and date, initally confidence and sentiment will be NULL as we will compute it in next program code.
- Next step is to implement sentiment analysis on csv file made earlier, tweets are process and the result is stored in separate csv file which contains date, time, username, sentiment.
- We find average sentiment for each day where we change the tweets sentiment data to form a relation with other dataset which is from NSE and this data is stored in different csv file, which has three columns, open_score, close_score and data.
- Next, we have a merge data from both data files, this was done manually, and this data is saved in a separate csv file, which contains data from both average sentiment file and NSE data.
- Process tweets remove stop words and create feature set for training data.
- Classify training data and test classified data model on test data for tweet sentiment analysis.
- Find correlation between sentiment of tweets and stock prices.

VIII. RESULTS AND ANALYSIS

We have implemented this application in 2 phases which are checkpoint 1 and checkpoint 2. In each phase we completed the tasks by distributing it among team members and later merging them together.

In the first checkpoint, we first collected tweets using twitter REST API, which responses in JSON format, we need authentication to fetch data from twitter, and need following configuration for it.

```
{
        "consumer_key": "",
        "consumer_secret": "",
        "access_token": "",
        "access_token_secret": ""
}
```

After fetching data from twitter, we need to store that data in a file for which we used csv file. The next step was to process tweet for which we had some basic steps and they are as follows:

1. Tokenize tweets
2. Remove extraneous keywords from tweets
3. Remove stop words from tweets

After fetching data and preprocessing, now comes task to do twitter sentiment analysis which we accomplished using Vader sentiment package by python and storing the data in a separate CSV file which will be used later for calculating average sentiment for tweets. The generated tweets csv file contains company name, date, time, username and sentiment. Following is a small part from that file.

	company	date	time	username	sentiment
0	Reliance	########	6:01	RshDvvrma	0.1027
0	Reliance	########	6:03	NewsBossIndia	0.3818
0	Reliance	########	6:04	EtemaadDailyNew	0
0	Reliance	########	6:05	shyamjiagrawal	0.3818
0	Reliance	########	6:05	Rampraw4121	0.1027
0	Reliance	########	6:06	AjNewspaper	0.3818
0	Reliance	########	6:06	thus_spake	0.1027
0	Reliance	########	6:06	alan_abraham8	0.1027
0	Reliance	########	6:06	ravindarthakur0	0.1027
0	Reliance	########	6:08	Rampraw4121	0.1027
0	Reliance	########	6:09	BroderkMarshall	0.3818
0	Reliance	########	6:10	zithastechno	0
0	Reliance	########	6:11	IndiLeak	0.3818
0	Reliance	########	6:12	SubratN	0

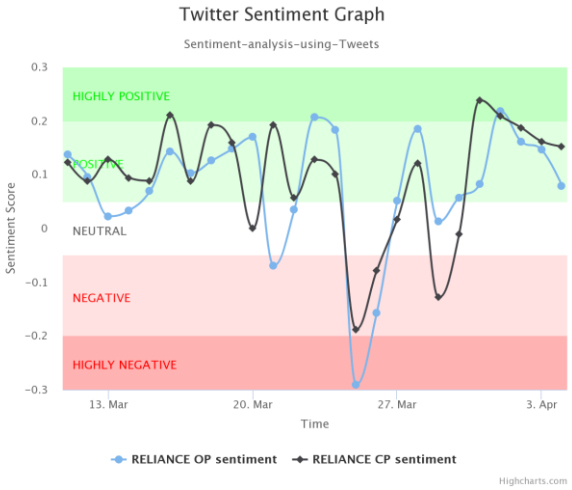
These were the tasks done for checkpoint 1, then we continued to fetch more tweets on daily basis as we got less number of tweets initially. Approximately, we got around 100 tweets for each company each day.

For checkpoint 2, we first checked that if we had sufficient data for the analysis as this could give exceptional result and whole system can fail. Then we manually went through the file and checked that we have completed every task from checkpoint 1 which turned out that we still has less number of tweets for few companies so we decided to fetch more tweets for next week and finally we had enough data that can be analyzed.

We then started working on next task, to find average sentiment of tweets for each day which was stored in a separated CSV file, following is an example from that file.

close_score	date	open_score
0.12312973	########	0.136453704
0.087479012	########	0.095978947
0.127274074	########	0.022238614
0.093629201	########	0.033530404
0.088488556	########	0.068960784
0.209550361	########	0.142469877
0.086928977	########	0.103239024
0.191771865	########	0.125527311
0.158684524	########	0.1474
0	########	0.169215217
0.191236019	########	-0.069225828
0.057127092	########	0.034364609
0.127134896	########	0.206134987
0.100737838	########	0.182294865
-0.187957585	########	-0.290353573
-0.078752041	########	-0.157434361

The above results indicated that the output is same as the expected one, and we can proceed with our application, we made a graph for this example which was easy to visualize.



Here we can see sentiment score for open price and close price.

For the next task we had to manually merge two files and we faced some problems while trying to do it via code and somehow data could not be merged so we decided to do it manually. Doing this was not a difficult task as we just matched data and put data accordingly into their field. Following is the example from that file.

close_score	date	open_score	Close	Date	High	Low	Open
0.093629201	########	0.0335304	1289.5	########	1319	1285.25	1318.75
0.088488556	########	0.0689608	1304.95	########	1316.3	1290.4	1291.05
0.209550361	########	0.1424699	1297.65	########	1310.4	1293.6	1310
0.086928977	########	0.103239	1300.7	########	1319.95	1298.05	1308
0	########	0.1692152	1280.8	########	1306.25	1278.35	1306
0.191236019	########	-0.069226	1263.8	########	1283.9	1259.3	1282.2
0.057127092	########	0.0343646	1259.7	########	1265.9	1246.55	1252
0.127134896	########	0.206135	1273.3	########	1277.55	1258	1263.15
0.100737838	########	0.1822949	1286.75	########	1292	1268.45	1274.1
0.016734194	########	0.0506032	1251.1	########	1278.75	1247.2	1271.1
0.120644571	########	0.1841203	1245.75	########	1264	1242.1	1258
-0.12878068	########	0.0120903	1256.65	########	1260	1233.35	1251.7
-0.01074244	########	0.0571418	1270.65	########	1274.75	1253	1255
0.238220802	########	0.0816654	1320.9	########	1337.65	1266	1266
0.161634081	4/3/2018	0.1458255	1374.65	4/3/2018	1380.5	1337.05	1342

Left three columns are from average sentiment file and remaining are the columns from National Stock Exchange dataset. For checkpoint 2, this was the final step for all the data collection and processing which was completed. Then we started using python techniques for predicting stock market movement using twitter sentiment analysis.

To analyze this data was the last step, and it was completed using linear regression, but first we found correlation between two values that is open_score from sentiment analysis and open_price from NSE dataset and similarly for close price and following are results from that analysis.

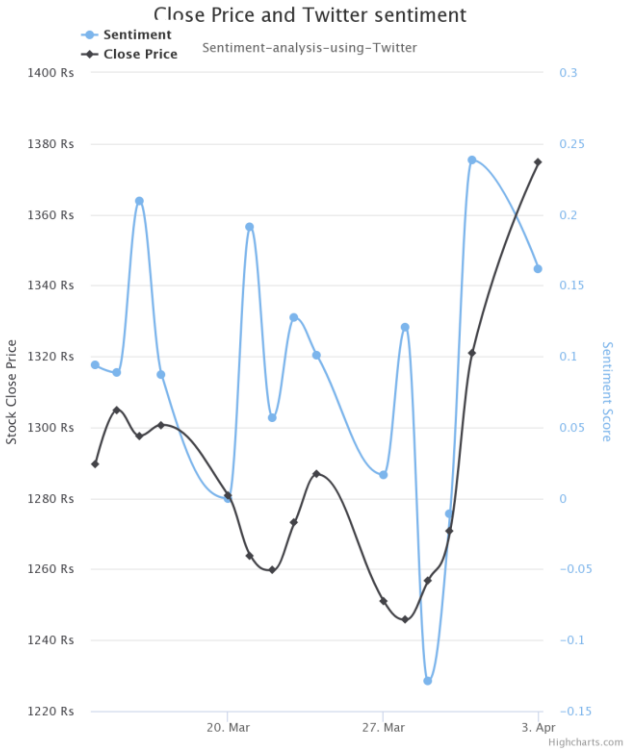
```
('Open price correlation:', 0.16501586120961761)
('Close price correlation:', 0.4749685257404374)
```

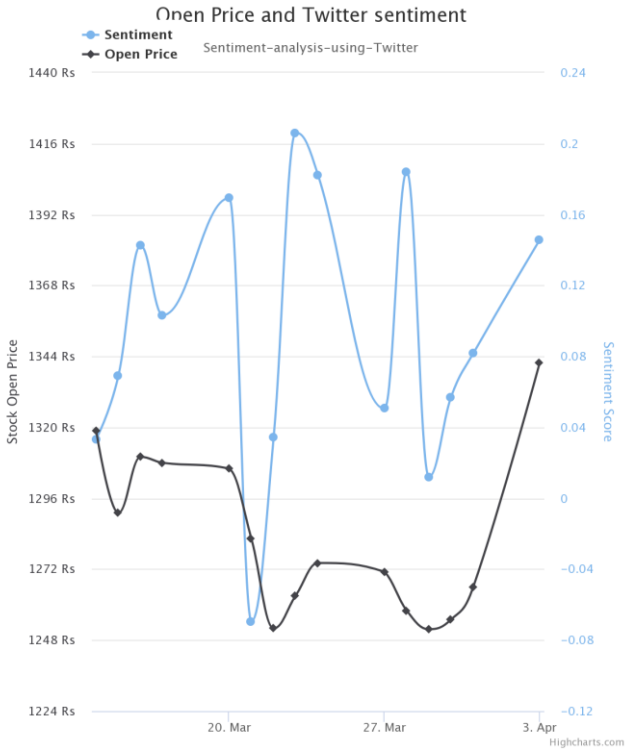
Here, we can see that the value for correlation is positive which means that people emotions and actual value are moving in same direction which can be either up or down it depends on the stock values of that company in market.

Last step was to find accuracy which was achieved by using linear regression and following are the result from it.

```
('Traditional Method Accuracy for Open Price: ', 97.2293482223652)
('Sentimental Method Accuracy for Open Price: ', -67.309776790104)
('Hybrid Method Accuracy for Open Price: ', -1987.9226003079382)
('Traditional Method Accuracy for Close Price: ', 8.630060243282312)
('Sentimental Method Accuracy for Close Price: ', -148.40353450931948)
('Hybrid Method Accuracy for Close price: ', -49.450841170768435)
```

We have three different accuracy methods implemented which makes it easy to determine which method is producing more accurate data and how value can fluctuate. Following are the graphs for correlation between prices.





IX. Recommendation System

We also implemented recommendation system for our application. A recommender system is a technology that is deployed in the environment where items (products, movies, events, articles) are to be recommended to users (customers, visitors, app users, readers) or the opposite. Typically, there are many items and many users present in the environment making the problem hard and expensive to solve. Imagine a shop. Good merchant knows personal preferences of customers. Her/his high-quality recommendations make customers satisfied and increase profits. In case of online marketing and shopping, personal recommendations can be generated by an artificial merchant: the recommender system. To build a recommender system, we need a dataset of items and users and ideally also interactions of users with items. There are many application domains—typically, users are customers, items products and interactions are individual purchases.

Content-based system: Such systems are recommending items like those a given user has liked in the past, regardless of the preferences of other users. Basically, there are two different types of feedback.

Knowledge-based system: Both users and items have attributes. The more you know about your users and items, the better results can be expected.

In recommendation system we do not consider any historical data or data from other user, we only look interaction of a user with itself. For our system we focused on collaborative filtering and the techniques used is call Association rule mining which a rule-based mining machine learning method for discovering interesting relation between variable in a dataset.

Association Mining Rule:

Let be an itemset, an association rule and a set of transactions of a given database.

1) Support

Support is an indication of how frequently the itemset appears in the dataset. The support of with respect to is defined as the proportion of transactions in the dataset which contains the itemset X.

In the example dataset, the itemset X= {beer, diaper} has a support of 0.2 since it occurs in 20% of all transactions (1 out of 5 transactions). The argument of supp () is a set of preconditions, and thus becomes more restrictive as it grows (instead of more inclusive).

2) Confidence

Confidence is an indication of how often the rule has been found to be true. The *confidence* value of a rule, X=>Y , with respect to a set of transactions T , is the proportion of the transactions that contains X which also contains Y.

Confidence is defined as:

Conf(X=>Y) = supp(XUY)/supp(X).

For example, the rule {butter, bread} => {milk}has a confidence of 1.0 in the database, which means that for 100% of the transactions containing butter and bread the rule is correct (100% of the times a customer buys butter and bread, milk is bought as well).

For our system we used raw data which consist of all the rules.

```
raw_data = ["Bajaj-Auto, Reliance, Heromotocorp, BajFinance",
            "Tata-Motors, Heromotocorp, Maruti, Axisbank",
            "BajFinance, Reliance, Maruti, Cipla",
            "Bajaj-Auto, ITC, Reliance",
            "LT, Heromotocorp, Tata, Cipla, Wipro",
            "Reliance, Maruti, BajFinance, Tata-Motors"]
```

We used orange package from python which is interactive data analysis tool, we stored this raw data into a file whose extension is given as file_name.basket, we then load data from this file and apply function from orange library which is Orange.associate.AssociationRulesSparseInducer(), which will produce result and below is the result from it.

```
Supp Conf  Rule
 0.3  0.7  Maruti -> Tata-Motors
 0.3  1.0  Tata-Motors -> Maruti
 0.3  0.7  Maruti -> BajFinance
 0.3  0.7  BajFinance -> Maruti
 0.3  0.7  Maruti -> BajFinance Reliance
 0.3  1.0  Maruti BajFinance -> Reliance
 0.3  1.0  Maruti Reliance -> BajFinance
 0.3  0.7  BajFinance -> Maruti Reliance
 0.3  0.7  BajFinance Reliance -> Maruti
 0.3  0.5  Reliance -> Maruti BajFinance
 0.3  0.7  Maruti -> Reliance
 0.3  0.5  Reliance -> Maruti
 0.5  1.0  BajFinance -> Reliance
 0.5  0.8  Reliance -> BajFinance
 0.3  0.5  Reliance -> Bajaj-Auto
 0.3  1.0  Bajaj-Auto -> Reliance

User 0: Bajaj-Auto, Reliance, Heromotocorp, BajFinance
 0.3  0.7  BajFinance -> Maruti
 0.3  0.7  BajFinance -> Maruti Reliance
 0.3  0.7  BajFinance Reliance -> Maruti
 0.3  0.5  Reliance -> Maruti BajFinance
 0.3  0.5  Reliance -> Maruti

User 1: Tata-Motors, Heromotocorp, Maruti, Axisbank
 0.3  0.7  Maruti -> BajFinance
 0.3  0.7  Maruti -> BajFinance Reliance
 0.3  0.7  Maruti -> Reliance

User 2: BajFinance, Reliance, Maruti, Cipla
 0.3  0.7  Maruti -> Tata-Motors
 0.3  0.5  Reliance -> Bajaj-Auto

User 3: Bajaj-Auto, ITC, Reliance
 0.3  0.5  Reliance -> Maruti BajFinance
 0.3  0.5  Reliance -> Maruti
 0.5  0.8  Reliance -> BajFinance

User 4: LT, Heromotocorp, Tata, Cipla, Wipro

User 5: Reliance, Maruti, BajFinance, Tata-Motors
 0.3  0.5  Reliance -> Bajaj-Auto
```

X. LIMITATTION AND CHALLENGES

During the implementation of this system we came across several challenges, which are mentioned below.

- **Missing stock indices:** As the stock market is closed on weekends and US holidays, there are no open/close prices for any of the stocks on those days. Which affects our DJIA values as well. We have used interpolation of the prices to fill in the missing values. For interpolation implementation, we have used the interpolate method from pandas package.
- **High fluctuations in prices:** As the prices of the stocks fluctuate a lot, we have used a technique called smoothing which is used in financial markets to take a moving average of the values, which results in comparatively smooth curves. For moving average implementation, we have used the EWMA method from pandas package.

XI. References

[1] http://t-redactyl.io/blog/2017/04/using-vader-to-handle-sentiment-analysis-with-social-media-text.html

[2]https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

[3] https://www.investopedia.com/terms/c/correlation.asp

[4] Machine learning in prediction of stock market indicators based on historical data and data from Twitter sentiment analysis. http://ieeexplore.ieee.org.ezproxy.gl.iit.edu/stamp/stamp.jsp?tp=&arnumber=6753954&tag=1

[5] Stock Prediction Using Twitter Sentiment Analysis **htt**p://cs229.stanford.edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis.pdf

[6] Qian, Bo, Rasheed, Khaled, Stock market prediction with multiple classifiers, Applied Intelligence 26 (February (1)) (2007) 2533, http://dx.doi.org/10.1007/s10489-006-0001-7.

[7] E.F. Fama, The behavior of stock-market prices, The Journal of Business 38 (1) (1965) 34105, http://dx.doi.org/10.2307/2350752

[8] J. Leskovec, L. Adamic and B. Huberman. The dynamics of viral marketing. In Proceedings of the 7th ACM Conference on Electronic Commerce. 2006

[9] https://arxiv.org/pdf/1610.09225.pdf

[10] https://github.com/biolab/orange3

[11] https://en.wikipedia.org/wiki/Association_rule_learning