University of Ljubljana – School of Economics and Business  
  
Course: Big Dana Management and Technologies

**Stock Sentiment Analysis using News Headlines: GameStop example**  
Course professor: Jurij Jaklič

Student: Jakov Čevizović  
Student ID: 19243046

Ljubljana, 12.12.2021

**Introduction**

This seminar will give a detailed explanation of stock sentiment analysis using a news headline from the Nasdaq website. The analysis will be performed on GameStop (NYSE: GME), primarily because of its recent popularity to retail investors. Because of that, there is a hypothesis that news headlines due to the increased reporting of the stocks prices, might influence the direction of a stocks price. The data that is going to be analyzed is extracted via Octoparse and contains more than 1500 rows of data. The dataset from Octoparse one column that contains both dates and headlines, where later it would be separated and renamed to “Date” and “Headlines” columns. The purpose of this research is that with automatic extraction of high-quality data we analyze which words in the headlines of articles have the biggest impact on the changes in stock prices. The research is based on simple machine learning algorithms and text mining from Rapid Miner, web scraping from Octoparse, and the basic use of Excel for data formatting.

**Web scraping and excel formating**

Diagram

Description automatically generated

In order to do web scraping through Octoparse, it was necessary to create a new task in *Advance Mode* where the [link](https://www.nasdaq.com/market-activity/stocks/gme/news-headlines) of the website was pasted. To start the web scarping, it was required to click the *I Accept* and within “Tips” click to *Click button* to accept the cookies to get access to the website within the Octoparse. In the options of the *Go to Web Page* step, it was important to enable *Use cookie* and to select *Use cookie from the current page*. After these steps, the step *Click Item* is deleted from the workflow.

Secondly, to loop and select the data from all the pages, the necessary step was to do pagination. By clicking on the arrow A picture containing text

Description automatically generated, the next step was to pick *Loop click single button* from “Tips”. The following XPath for pagination was used; //button[@class="pagination\_\_next"]. Due to Octoparse's automatic creation of the Xpath, the correctness was checked by inspecting the page in Google Chrome, . After the pagination, it was required to pick what data we want to be scrapped from the website. To do that, we selected the whole “box” that contains the date and the headline. To select all headlines, it was necessary to either click on two different “boxes”, or to click on *Select All* inside the “Tips”.

Graphical user interface, text, application, email

Description automatically generated  
In the “Loop Item” process for extracting data, the following XPath was used: //body/div[2]/div[1]/main[1]/div[2]/div[4]/div[3]/div[1]/div[1]/div[1]/div[1]/ul[1]/li. After the selection of elements, we extract the text by clicking on *Extract text of the selected links* from “Tips”. To extract the data, the process was run, and after looping through all 507 pages, the final dataset contained more than 1500 rows, in one collumn called Field1\_text. Before extracing data to the Excel (xlsx) all duplicates were removed.

In excel, the data set was separated into two separated columns through the *Text to Columns* function. After getting two different columns, one contained dates, and the other contained headlines. They were renamed to *Date* and *Headline*. In order to have the right data type when importing the data to the Rapid Miner, the column date was also changed to the date data format.

Graphical user interface, text, application

Description automatically generated

**Data preparation process in rapid miner**

The first step in Rapid Miner was to import the data. When importing the data, the *Define header row* was selected, so that Rapid Miner automatically puts the Date and Headline as the name of the columns. It was required that the Data column is a date\_time type, and that Headline is a polynomial data type.

Diagram, schematic

Description automatically generated  
  
After importing the data set, operator *Aggregate* was used in order to combine the headlines with the same date into one document, for that the concatenation function was used as an aggregation function. To proceed with data preparation, the operator *Read Yahoo Finance* was used to get the data set with the price (open, close, low, high, volume) data about a GameStop stock, and such data was loaded from 16.10.2009 to 02.12.2021.

In order to get a column that would display a price increase/decrease in comparison with the previous day, the operator *Lag* was used to create an additional price column that would show the open price of the stock for the next day, which then could be compared to the price of the previous day. Used settings for *Lag* were; attribute filter type was set to subset and attributes Date and Open were used, and to show the next day default lag was put to -1. Table

Description automatically generated  
  
Operator *Generate Attributes* was used to create a new additional column called *Price Change*, where 1 indicates that the price of the next day is higher than the previous day, and 0 indicates that the price did not increase. To create the attribute the following formula was used: if([Open+1]>Open,"1","0").  
Table

Description automatically generated

In order to successfully merge two tables together, the attributes from the *Read Yahoo Finance* data set were reduced to only *Price* and *Price Change* attributes with the use of the *Select Attributes* operator. To merge two datasets together, the *Join* operator was used with the join type as inner, and with key attributes as *Date* from both left key attribute and right key attribute. After merging two tables together, concatenated headlines were renamed to *Headlines* with operator *Rename* for easier analysis. After that, the new data set was stored in the repository to prevent the loading of the *Read Yahoo Finance* operator. The data set that will be further processed in this project looks like this;

Table

Description automatically generated

**Creating k-NN prediction model in Rapid Miner**

After retrieving the previous data set, to transform the data from unstructured (text) into structured data (vector) and for creating a word vector a *Process Documents from Data* operator was used, the importance of *Process Documents from Data is* that it can go through dataset row by row and it can process it row by row (it will go in the first row, and pick the headlines for the first row, the second row and pick the headline of the second row, …). Within the parameters, creation of word vector, add meta-information, keep text and select attributes and weights were selected. As a source attribute, Headlines were selected, with the weight of 1.0. For the vector creation model, TF-IDF was used, which tells us more how important a particular word is for a particular document. Even if the relative frequency is high, the word is also telling us less about the document (how important is it) if it appears in many documents, it is less significant if the word appears in many documents for one particular document. Because it is a subprocess, to get the cleaner dataset, operators *Tokenize* (to split documents into words), *Transform Cases* (to make everything lower case letters), *Filter Tokens by Length* (to remove characters like s, t,…), *Filter Stopwords (English)* (to filter out stop words like a the, …) and *Stem Porter* (to get roots of the words, and after stemming we have tokens that are not words). After getting structured data, the newly created attribute *text* was renamed back to *Headlines,* and the role was put back to regular by using the operator *Set role*.

*Extract sentiment* operator is used in order to find sentiments and to see if the score is high or low, either showing us if the headline is positive or negative. Within the *Extract Sentiment* operator, show advanced output was selected to be able to see more than just a polarity, to easier understand the results. Because of the newly formed attributes due to the *“add meta information”* in *Process Documents from data,* a *Select Attribute* operator was used to only include the attributes that have no missing values, to get rid of columns with missing values. With the operator *Set Role,* we selected attribute *Price Changes* as our label, which means that that is the attribute that we want to predict the result of.

To properly split the data into training and testing we use operator *Cross Validation*, with the number of folds set to 10, which means that all the datasets will be split into 10 equal beans, where 1 bin would be used for testing, and the other 9 are used for training, the iteration continues until all bins are involved in the training and testing phase. Inside the sub-process, *k-NN*, *Apply model*,and *Performance* operator were used. *k-NN* is an algorithm that is based on similarity, the idea is that if we do not know the value of the output variable (Price Changes) for the particular case of headline, *k-NN* tries to find labeled units for which we know the value of prediction variable, and try to find some most similar reviews from those for which we know the value or the score. In the parameters of *k-NN*, k was set to 3 because it is prone to overfitting, so it is better to select the lower number to get better results.

**Interpretation of results**

From the WordList result that we have received from Process Documents from Data, it is possible to conclude that besides the word gamestop, the most commonly mentioned words are; stock, earn, corp, investor, share,… Since the GME has been a hot topic recently due to the popularity of short squeeze and its increased volatility which are closely related to the WallStreetBets subreddit, the words reddit, meme, squeeze and robinhood are also important in the analysis.

Table

Description automatically generated

From the performance vector, the following results are received:

Table

Description automatically generated

As the accuracy is calculated as , which is = 0.5317164179, we get an overall accuracy of 53.19%. The accuracy is satisfactory, because it is above 50%, which means that the investor has a probability of having slightly better chances than flipping a coin. Additionally, the true 1 and true 0 columns represent the values that were set in the column *Price Changes*, as 1 and 0, where 1 represents the increase of prices, while 0 represents the no change or decrease of stock price. In column true 1, a k-NN model predicted 138 price increases, and in 126 examples it predicted wrongly price increase instead of price decrease. For the column true 0, 125 examples were predicted wrongly as price increase instead of the price decrease, and 147 examples were correctly predicted as price decrease. The results give us the accuracy of predicting true 1 to 52.27% (where the price increased) and accuracy of predicting true 0 of 54.04% (where price decreased).

From the sentiment analysis, we can see the influence of negative and positive words on our prediction. The most important result is the *Score* column, which are sentiment scores, where the higher the value shows positive sentiments, and negative values show negative sentiments. The calculation of the score is based on *Scoring String* which is composed out of words based on dictionaries, sentiment score is associated with each word in the dictionary. Not all words from headlines can be found in the dictionary for the calculation of sentiment score. Words such as avoid, stress, fear, suffer, pain is considered negative words by Rapid Miner, while the words like spark, gain, worth, winner, great are considered positive by Rapid Miner.

Graphical user interface

Description automatically generated with medium confidence

**Conclusion**

Known how complex it is to predict a price movement in a stock market, the models performed in Rapid Miner lack additional complementary information and can be improved with the use of fundamental and technical analysis of stock markets. Such additional information can increase investors' edge over the market. Despite having an overall accuracy of 53.19%, it is not advisable to solely rely on such a result, because most people tend to lose money in stock markets. The model lacks the information on what to do in certain cases; when they buy the stock when to sell the stock, and at which price levels should the trade be taken. Despite its imperfections, such a model can be a perfect additional source of information in the analysis of the stock market which would help investors analyze the sentiment of people towards the certain stock.