**SUMMARIZING AND PERFORMING SENTIMENT ANALYSIS ON FINANCE NEWS**

**Group Name:** Great Mates

**Group Members:**

|  |  |  |
| --- | --- | --- |
| **First Name** | **Last Name** | **Student Number** |
| Abhinav | Duvvuru | C0810417 |
| Sandeep Reddy | Alla | C0824160 |
| Sivapriya | Tatireddy | C0817271 |
| Uma Maheswara Rao | Gutta | C0814571 |

**Submitted to:** Prof. Parisa Naraei

**Submission date:** December 18, 2021.

Table of Contents

[1. Abstract 4](#_Toc90657936)

[2. Introduction 4](#_Toc90657937)

[Figure 1. Sample text of a news article 6](#_Toc90657938)

[Figure 2. Summarized output text of summarization model 6](#_Toc90657939)

[3. Methods 7](#_Toc90657940)

[3.1 Hugging face transformers summarization model 7](#_Toc90657941)

[3.2 Beautiful Soup 7](#_Toc90657942)

[3.3 Regular expressions 8](#_Toc90657943)

[3.4 Hugging face transformer sentiment analysis pipeline 8](#_Toc90657944)

[3.5 Spacy library 8](#_Toc90657945)

[3.6 Machine Learning algorithms 8](#_Toc90657946)

[4. Summarization Pipeline Building 9](#_Toc90657947)

[4.1 Data Collection 9](#_Toc90657948)

[Figure 3. Tesla news from yahoo finance website 9](#_Toc90657949)

[4.2 Data Cleaning 10](#_Toc90657950)

[Figure 4. Uncleaned URLs of news articles 10](#_Toc90657951)

[Figure 5. Cleaned URLs of news articles 11](#_Toc90657952)

[4.3 Collecting the textual data from URLs 11](#_Toc90657953)

[Figure 6. Content of an article gathered from URL 12](#_Toc90657954)

[4.4. Summarizing the Articles 12](#_Toc90657955)

[Figure 7. Summarized text of various articles 12](#_Toc90657956)

[4.5. Sentiment Analysis 13](#_Toc90657957)

[Figure 8. Sentiment scores of summarized articles 13](#_Toc90657958)

[4.6. Converting the data into CSV format 13](#_Toc90657959)

[Figure 9. Dataframe of summarized articles and sentiment labels 14](#_Toc90657960)

[4.7. Visualization using Spacy library 14](#_Toc90657961)

[Figure 10. Parts of speech tagging using Spacy 15](#_Toc90657962)

[Figure 11. Named entity recognition 15](#_Toc90657963)

[4.8. Text Classification 15](#_Toc90657964)

[Figure 12. Count of sentiment labels of articles 16](#_Toc90657965)

[4.8.1. Logistic Regression 16](#_Toc90657966)

[4.8.2. K Nearest Neighbors 16](#_Toc90657967)

[4.8.3. Support Vector Classifier 17](#_Toc90657968)

[4.8.4. Hyperparameter Tuning 17](#_Toc90657969)

[5. Results 17](#_Toc90657970)

[Figure 13. Confusion matrix of Linear Support Vector Classifier 18](#_Toc90657971)

[Figure 14. Classification report of Linear Support Vector model 19](#_Toc90657972)

[6. Conclusions and Future Work 19](#_Toc90657973)

[7. References 19](#_Toc90657974)

# 1. Abstract

In this project, we performed the summarization on various news articles collected from yahoo finance website. We have gathered the news articles related to the companies like Bitcoin, Tesla, and GameStop from the Yahoo Finance website. We summarized all the data from the news articles separately and used the data to perform sentiment analysis. We performed sentiment analysis on the individual summaries of the articles and gathered information of the sentiment scores of the articles. We then converted the whole data into a readable format for further analysis.

We used the summaries of the articles and sentiment scores to perform text classification. We used various supervised Machine Learning algorithms and created a model to perform text classification on our data. We analyzed the predictions of our model to see how it was able to classify positive and negative sentiments based on the labels that were already available in our data based on sentiment analysis. We were able to achieve an accuracy of 77 percent with our model in predicting the sentiments.

# **2**. Introduction

Reading newspapers and getting to know the day-to-day happenings is a natural habit for most of us. With the advent of technology and internet, we are now trying to learn lot of things from internet using mobile devices. We are surfing the internet for all the improvements happening around and gaining necessary knowledge in very less time and minimal energy. While the old and natural habit of reading newspapers still exist in most of the places, we are slowing getting habituated to reading the news through online resources to save time and energy.

As mobile phones became mandatory part of our lives in recent times, we are using them to gain the knowledge and insights over various topics.

Finance sector plays an important role in our world as the world revolves around the money being made through large companies and industries. Large companies and industries like Tesla come up with a lot of new inventories and it plays an important role in stock markets. To see the developments happening in the companies and to check the stock values and investments of the companies, we need to constantly follow the news related to the companies. This will help the stockers, investors and other common people gain the insights of the company structure and flow of currency related things. Gathering the news from various articles and other resources can help us to understand the exact scenario and will also help us improve the decision-making aspects.

But the above point of going through various articles and resources to read the news and understand various other aspects is time consuming. There are several articles in various websites and each news resource will provide something or the other related to the company on daily basis. This makes us think about unnecessary aspects of the related company and may lead to wrong decision making. In-order to avoid going through multiple aspects and noting the point of time constraints in reading multiple articles, we created a summarization pipeline to give the summary of related news article.

We have collected the news articles of various companies like Tesla, Bitcoin and GameStop. All the articles were lengthy and have a lot of information around the company stocks and other news. We used hugging face transformers model to convert the all-lengthy articles to summarized form. This will give us the summaries of articles to investigate and with those summaries we can see what exactly the articles are all about. This will help us reduce the time

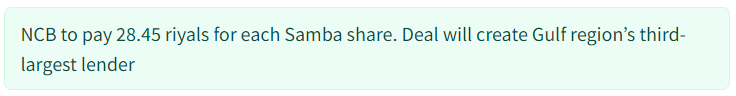
of reading the article on a whole and will give the necessary insights required by checking into the summary. We also performed sentiment analysis on the summaries of the articles that were collected from our pipeline to see if the related news is positive or negative. This will help us in making appropriate and correct decisions in very short span of time.

Now let us consider an example to see what exactly is happening by using our model. The below figure 1 shows us the text of an article.



### Figure 1. Sample text of a news article

After performing the summarization model, we are going to get the summarized text of the article as shown below figure 2.



### Figure 2. Summarized output text of summarization model

The example shown in figure 2 is performing summarization on one single article. We performed summarization on various articles from different companies. We created a pipeline to get the summaries of required number of articles from any company or organization. We can include multiple companies to get summaries and sentiments of different news articles according to our requirements.

# **3**. Methods

We used various python libraries, hugging face transformers summarization model, regular expressions, and machine learning algorithms for building our summarization pipeline. The main components used in our model are briefly explained as follows:

## 3.1 Hugging face transformers summarization model

We are using hugging face transformers financial summarization model for our project. This pretrained “**human-centered-summarization/financial-summarization-pegasus**” is used specifically for summarizing the financial news using the Pegasus tokenizer. This will help us in taking the text as input and will provide the summarized text as output. This is the best summarization model available for dealing with the financial related articles and blogs for summarization.

## 3.2 Beautiful Soup

Beautiful Soup is python library for web scraping and scraping the data from various resources like HTML and XML documents. This will help us in getting the required data from the websites. We used this library to get the news articles required for our model from the yahoo finance website. This is a very useful tool and is also easy to access the html contents required for analysis.

## 

## 3.3 Regular expressions

Regular expressionsare used to extract contents from the text data. They are very useful in the process of text preprocessing which will help us in getting the required items from the text data. We used regular expressions in our data to clean the gathered data while scraping the data using beautiful soup. This helped us in separating only necessary contents for further analysis.

## 3.4 Hugging face transformer sentiment analysis pipeline

Hugging face sentiment analysis pipeline is available in the official hugging face transformers website. This pipeline is helpful in categorizing the textual data into positive or negative sentiments based on the context of the data. This will also provide us with the strength of the polarities like how much positive or negative is our data rated on a -1 to +1 scale. If the score is +1 or very near to 1, then the text is in extremely positive context. In the same way, if the score is -1 or very near to -1, then the text is in extremely negative context. We used this sentiment analysis pipeline to check the sentiment scores of our summarized articles to check on the polarities of news articles.

## 3.5 Spacy library

Spacy is a Natural Language processing library used for text pre-processing like removal of stop words, lemmatization, stemming, parts of speech tagging, named entity recognition and other important aspects of textual data. We are using Spacy library for visualizing the textual data based on the named entity recognition and parts of speech tagging of our data.

## 3.6 Machine Learning algorithms

We used the below machine learning algorithms to perform text classification on our summarized articles data. We also used **TF-IDF Vectorizer** to convert the textual data into a

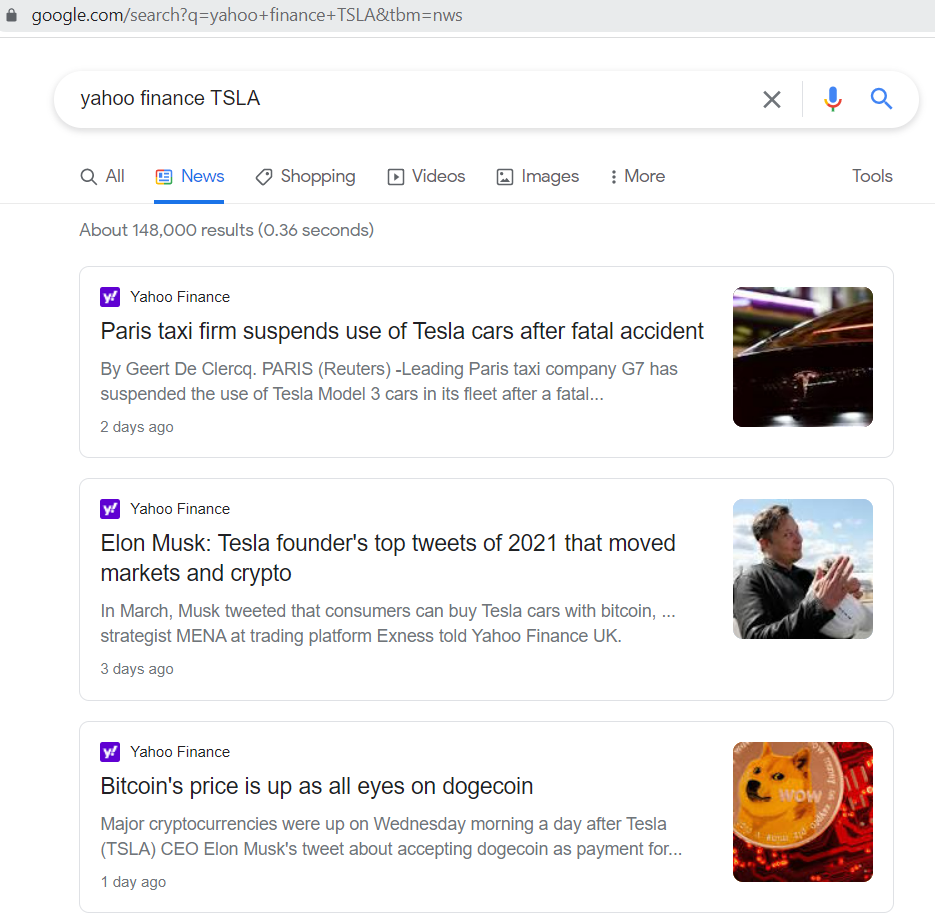
sparse matrix of number which will support the numerical data format of most machine learning algorithms.

* Logistic Regression
* K Nearest Neighbors Classifier
* Linear Support Vector Classifier

# **4. Summarization Pipeline Building**

## 4.1 Data Collection

We collected the data from yahoo finance website using the python beautiful soup library. We collected the html contents of the website and the links of the news articles that are available related to the companies Bitcoin, Tesla, and GameStop. We collected nearly 100 articles for each company mentioned above.



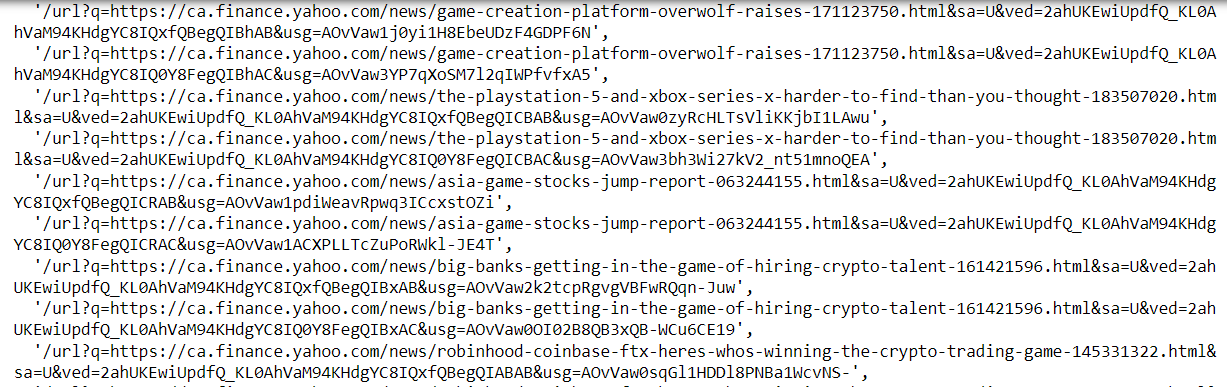
### Figure 3. Tesla news from yahoo finance website

In the above figure 3, we are taking the links of each article related to TESLA company for our further analysis. We are also collecting the links for other companies mentioned above in the same way.

## 4.2 Data Cleaning

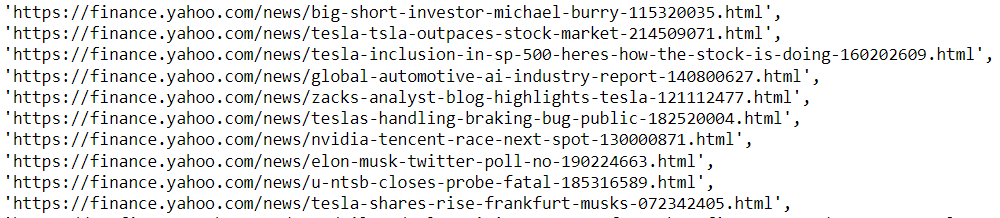
We have collected the URLs for the articles using the beautiful soup library of python. Our data now have the required links for the websites but there are many unnecessary items were also scraped as part of the process. We need to clean the links gathered from the website for further processing. We used python regular expressions package to deal with the unnecessary aspects of our data.

Before cleaning, the links of the articles are as shown in the below figure 4. There is a lot of unnecessary parts in the URLs collected.



### Figure 4. Uncleaned URLs of news articles

After using the regular expressions, we have got the data with much more clarity compared to the previous data. The cleaned URLs are as shown in figure 5.



### Figure 5. Cleaned URLs of news articles

From the above figure 5, we can clearly see that the URLs data is cleaned, and it is evident in comparison to the previous structure.

## 4.3 Collecting the textual data from URLs

From the data cleaning process, we got the necessary URLs for our analysis. Now, we are again using beautiful soup library and requests module to get the text inside the URLs of the news articles. Now, we are gathering the exact textual data under each of the individual articles from the existing URLs. We are only collecting the limited data (350 words after splitting the paragraphs data) here instead of full text from the articles as per our tokenizer requirements. Specifically, we are collecting paragraphs of the data using beautiful soup html parser. Our pipeline will take more time for processing as we have nearly 300 articles to scrape the textual data.

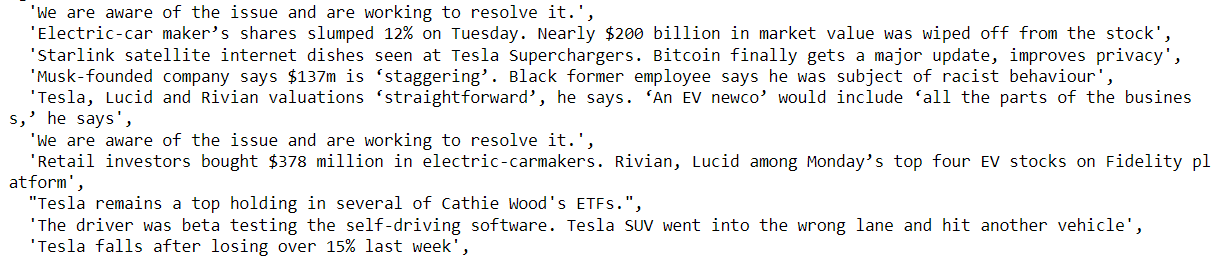
We have now gathered the textual data from the websites having news articles as shown in the below figure 6. This is the scraped data from one of the Tesla company’s related news articles.



### Figure 6. Content of an article gathered from URL

## 4.4. Summarizing the Articles

Now we have got the required data for performing summarization, we are now going to summarize all the textual data related to the news articles of various companies. We are using the Pegasus tokenizer and pre-trained summarization model of human centered summarization from hugging face transformers for performing the summarization operation.



### Figure 7. Summarized text of various articles

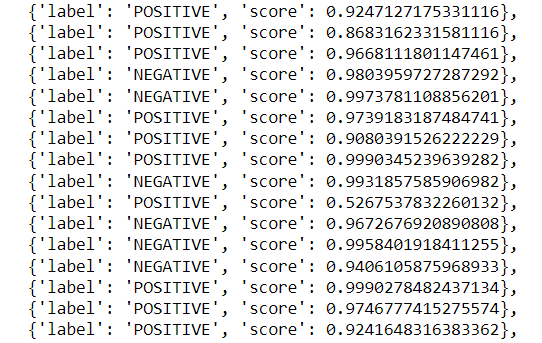
In the above figure 7, we are showing on how the articles were summarized and are converted to small statements that give us the idea of exactly what happened in the story of articles. The hugging face pre-trained Pegasus model is giving us better results as seen above from our

summarization pipeline. We can see how good our summarization model like “Car shares slumped by 12% on Tuesday” and how much market value is wiped off for the company.

Now that we have the summarized data of the news articles, we will now perform sentiment analysis on the available data to see if the related article is positive or negative based on the results.

## 4.5. Sentiment Analysis

For the sentiment analysis on summarized data, we are using the pre-trained sentiment-analysis pipeline of hugging face transformers. This will give us the positive and negative sentiments of the summaries of articles along with the sentiment scores of the data. We are getting the scores as shown in below figure 8.



### Figure 8. Sentiment scores of summarized articles

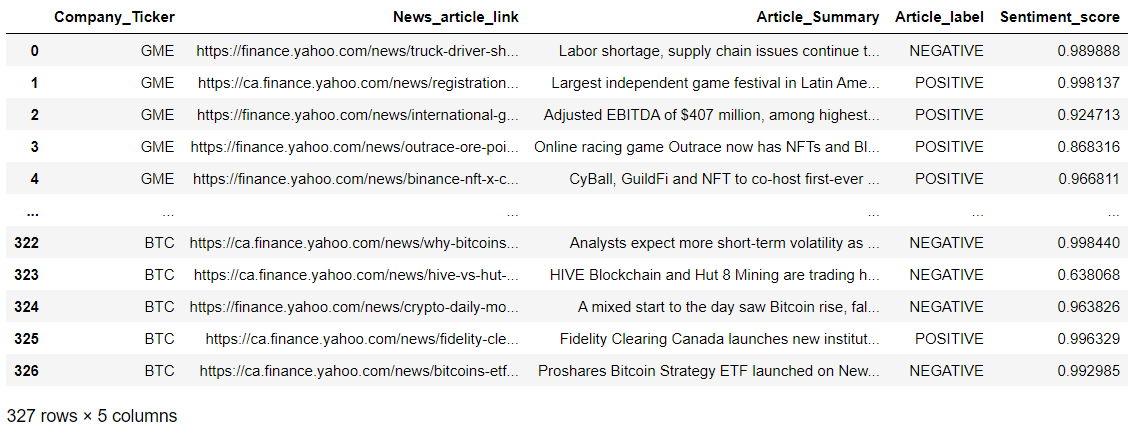
From the above figure 8, we can see that the labels are positive and negative. The respective scores show us the strength of positivity or negativity.

## 

## 4.6. Converting the data into CSV format

As we are done with the preparation of our data using the summarization model and sentiment analysis, we now need to convert the data into accessible format. So, we are using csv module to convert the data into a comma separated format. After converting the data into csv format.

we will create a dataframe for our further analysis. The data is created using the pandas library of python.

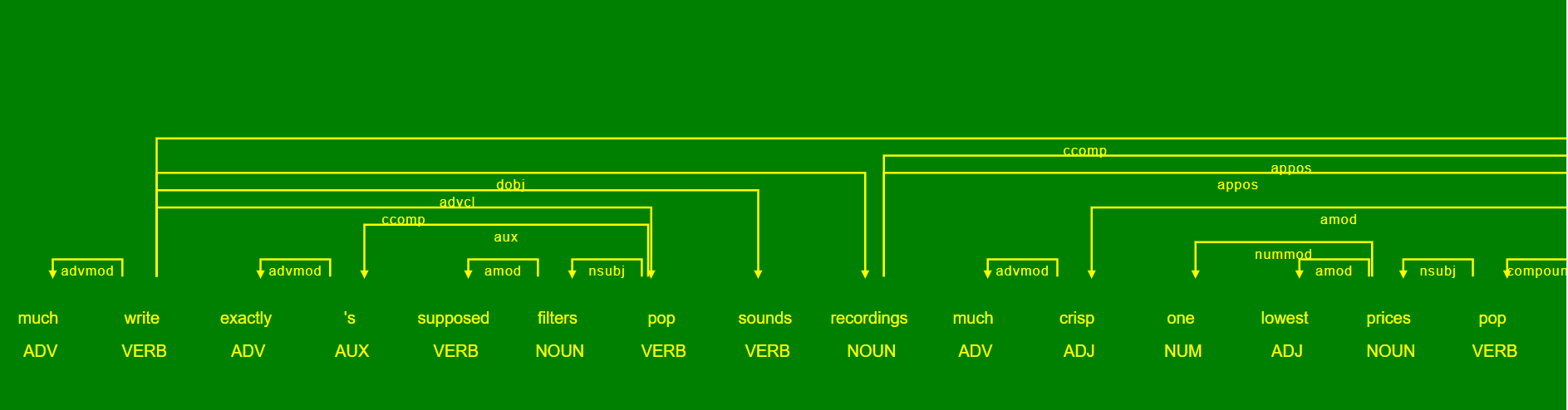


### Figure 9. Dataframe of summarized articles and sentiment labels

The above created dataframe (figure 9) have the company ticker, link of news article, summary of the respective article, sentiment label of the article and the sentiment score columns.

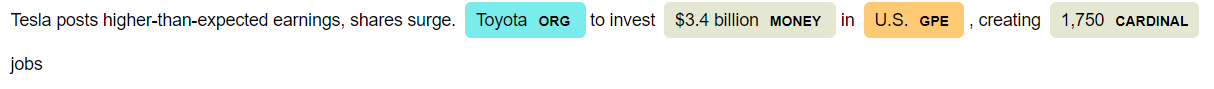
## 4.7. Visualization using Spacy library

We can investigate the parts of speech tagging using the spacy library for our summarized data as shown in below figure 10.



### Figure 10. Parts of speech tagging using Spacy

We can also investigate the named entity recognition scenarios from the displacy module of spacy library. The named entity patterns are shown as follows in figure 11.

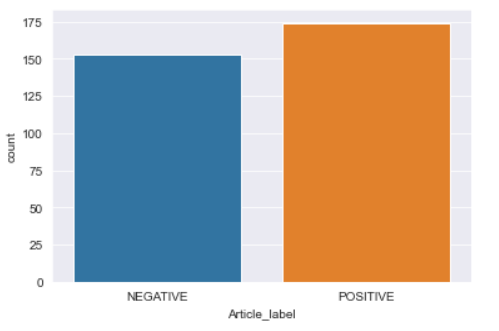


### Figure 11. Named entity recognition

From the above figure 11, we can see that the organization is Toyota as entity, money as an entity and other things shown in the figure.

## 4.8. Text Classification

The data now has output label of positive and negative sentiments. The count of them in the dataset is shown as below figure 12.



### Figure 12. Count of sentiment labels of articles

We have the required data from the sentiment analysis pipeline and converted it into dataframe format. Now, we will perform the text classification based on the text summary and the sentiment label columns of our summarization data. We used TF-IDF Vectorizer to convert the textual data into a sparse matrix of normalized vector features of words based on the frequency and inverse document frequency.

We used various machine learning algorithms to predict the output label column that is positive or negative.

### 4.8.1. Logistic Regression

We are using Logistic Regression for classifying the output labels into positive and negative. Logistic regression will predict the probability of the classes and will use sigmoid function to predict the data into highest probability classes. We got an accuracy of nearly 74 % in predicting the classes and labels.

### 

### 4.8.2. K Nearest Neighbors

We will use the k nearest neighbors classifier to predict the classes when the data have more numerical values. We will classify the data by using the nearest classes of the point. We will define the appropriate k value to check the nearest k points of our data to classify the data point. We got an accuracy of nearly 74 % with the k nearest classifier.

### 4.8.3. Support Vector Classifier

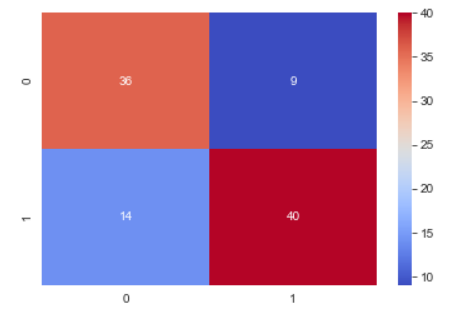
We will use support vector classifier on our data to predict the classes. Support vector classifier will divide all the points based on the hyperplane that will go on to separate data points into different clusters. We used this method to classify our data into positive and negative classes. We got an accuracy of nearly 77 % with the usage of support vector classifier model.

### 4.8.4. Hyperparameter Tuning

We used various machine learning algorithms in modelling our summarized data. To improve the performance of our model, we are tuning the parameters of our Support vector model. As we got better accuracy for support vector model, we are tuning the parameters of the model using GridSearchCV algorithm. We tuned various parameters like kernel of the model using polynomial, radial basis functions, sigmoid functions apart from Linear model. We got better results using the Linear kernel for the model.

# **5. Results**

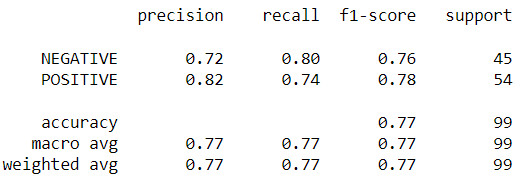
We got the results of the summarization pipeline using the Pegasus tokenizer of the hugging face transformers human centered summarization. Additionally, we applied the text classification on the results of sentiment analysis pipeline to perform predictive analysis on our data. We got an accuracy of nearly 77% with Linear Support vector classifier model and the confusion matrix of the data is shown in the below figure 13.



### Figure 13. Confusion matrix of Linear Support Vector Classifier

From the above confusion matrix (figure 13), we can see that our model is predicting well with only 23 wrong predictions out of 100 instances. Here, 0 represents negative summary and the label 1 represents positive summary of our data.

The classification report of our predicted data is shown as below. Here, we can see the model accuracy as 77%. Along with the accuracy, we have better precision for positive summary predictions and better recall for negative summary predictions. Overall, the results were quite good as shown below figure 14.



### Figure 14. Classification report of Linear Support Vector model

# **6. Conclusions and Future Work**

In conclusion, we can say that the summarization pipeline established through our project is working extremely well and can be used to grab the data easily from the websites using the model created. We can also investigate the sentiments of the summaries for better decision making of company stocks and other related insights.

Coming to the future work, we are planning to deploy the model using the Flask API and present it in a better way for users. We are also planning to implement the sentiment analysis part by using other sentiment tools instead of using the pre-trained pipeline from Hugging face transformers.

# 

# **7. References**

PEGASUS for Financial Summarization. (n.d.). Retrieved from <https://huggingface.co/human-centered-summarization/financial-summarization-pegasus>

Summary of the tasks. (n.d.). Retrieved from <https://huggingface.co/docs/transformers/task_summary>

## Sentiment Analysis with BERT and Transformers by Hugging Face using PyTorch and Python. (April 2020). Retrieved from <https://curiousily.com/posts/sentiment-analysis-with-bert-and-hugging-face-using-pytorch-and-python/>

# Regular expression operations. (n.d.). Retrieved from <https://docs.python.org/3/library/re.html>

# Text Classification with Machine Learning & NLP. (n.d.). Retrieved from <https://monkeylearn.com/text-classification/>

Yahoo finance. (n.d.). Retrieved from <https://www.google.com/search?q=yahoo+finance&tbm=nws>