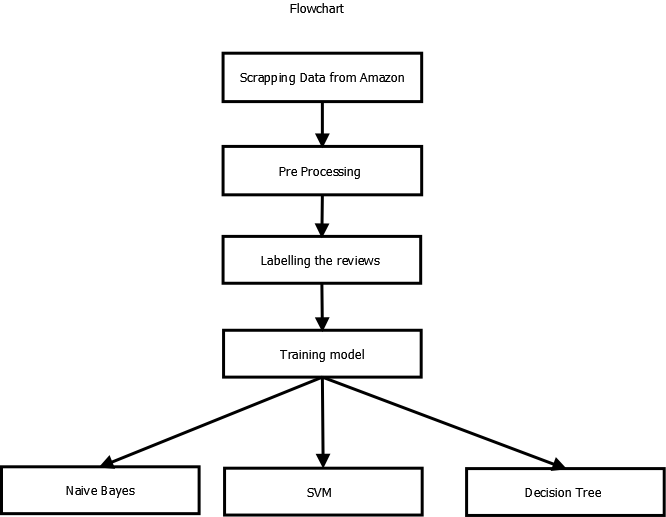
Report on sentiment analysis- part 1

We, Vinod Ghanchi and Sidharth Nair from KJ Somaiya college of Engineering, performed this task/project as a part of internship under the guidance of Asst. professor Mr. Vaibhav Vasani.

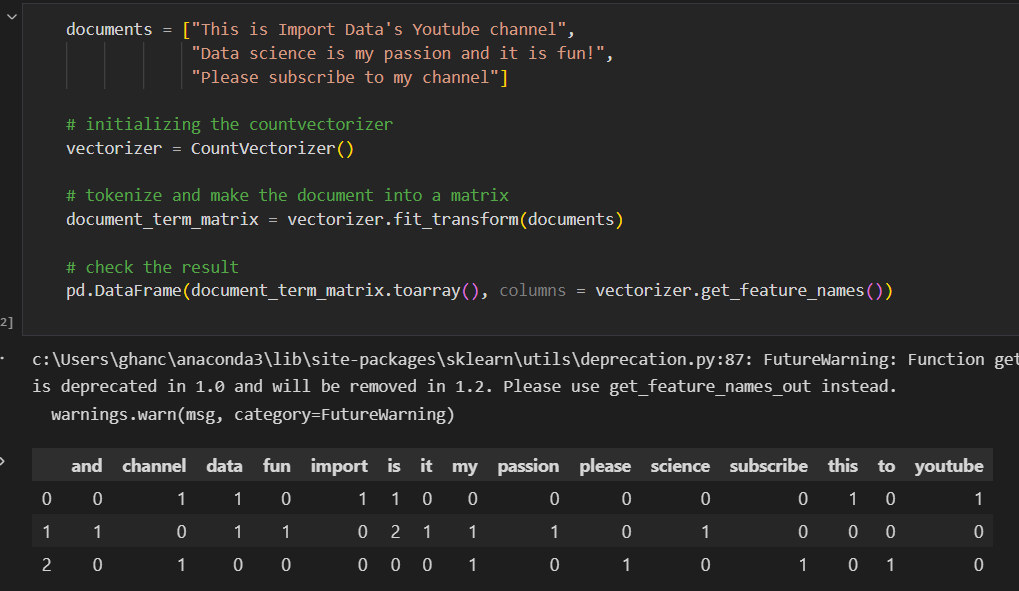
The aim of this task was to do sentiment analysis of reviews of products on amazon (online shopping website) using various models like svm, naïve bayes, etc.

Flowchart of the project/task:



Steps involved in completing this task:

1. **Scrapping the data**: We scrapped the data from amazon.com using beautiful soup in python. In this step we created different function for extracting the single page data using data asin number(unique identification number) of products. Using this data-asin number, we were able to extract all the reviews of a particular product and also similar products from all the pages available. In this manner we extracted all the reviews of all the products from all the pages and stored them into a form of list. We converted this list into dataframe using pandas and then converted it to excel and csv.
2. **Preprocessing the data**: The data we extracted was not clean data. We need to clean the data first before we could apply any classification techniques on it. For this used “spacy” module. We checked for all the null values, and replaced it with empty string “”. Firstly we converted all the reviews from the reviews column to lowercase and stored it into a new column new\_reviews. Next, we remove punctuations, then emojis, pictographs and stopwords like a, the, etc and many more. We then lemmatized the words into their root words. This is the cleaned dataset on which we can work on.
3. **Labelling the reviews dataset**: For this step we imported modules “vader” and “nltk”. Using vader module, we calculated the polarity scores of each rows in reviews column. We got the values of positive, negative, neutral and compound. Depending on the compound values, we can label the reviews as having positive and negative polarity. For the compound values more than -0.5, we treated them as positive and labelled it as 0 and for the values less than -0.5, we treated them as negative review and labelled it as 1.
4. **Naïve Bayes model**: Using sklearn module, we applied naïve bayes classifier on our labelled data.Firstly we splitted the dataset into train and test set as 70% and 30%. Using the count vectorizer function, we tokenized the document into a matrix. We have also entered a user defined list to see the result of this function and it showed us as below

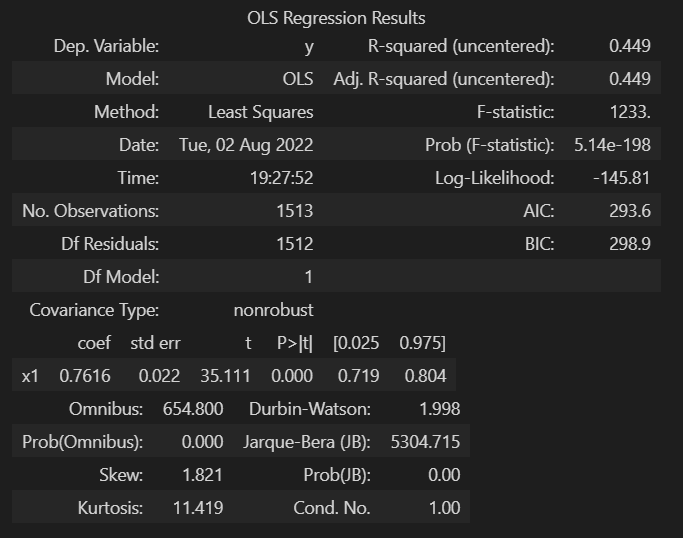


As shown, it tokenize each word in the list and shows us the frequency. We applied the same to our train and test dataset.

For this we used multinomial naïve bayes classifier model. After fitting the vector to the model, we got the following confusion matrix.



The accuracy score using this model was 92.33%.

Using the statsmodel module, we got the summary of the predicted result. 

As we can see all the values are shown.

**Interpretation of OLS results**

**R squared value:**

R2is the coefficient of determination that tells us that how much percentage variation independent variable can be explained by independent variable. Here, 44.9 % variation in Y can be explained by X. The maximum possible value of R2 can be 1, means the larger the R2value better the regression.

**F-statistic value:**

F test tells the goodness of fit of a regression.

The null hypothesis under this is “all the regression coefficients are equal to zero”. Prob(F-statistics) depicts the probability of null hypothesis being true.

The test is similar to the t-test or other tests we do for the hypothesis. The F – statistic is calculated as below –

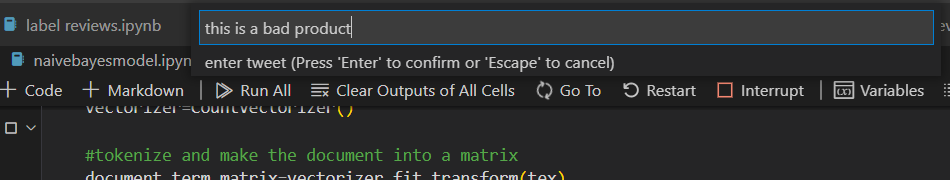


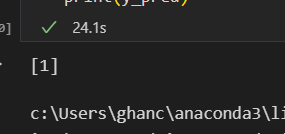
Inserting the values of R2, n and k, F = (0.449/1) / (0.551/1512) = 1232.1.

You can calculate the probability of F >1232.1 for 1 and 1512 df, which comes to approx. 0. As per the above results, probability is close to zero. This implies that overall the regressions is meaningful.

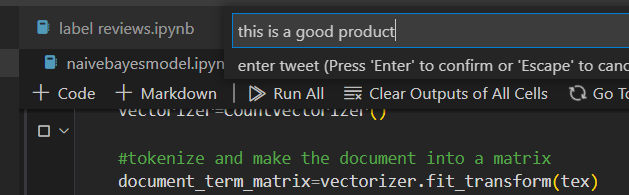
The remaining terms are not often used. Terms like Skewness and Kurtosis tells about the distribution of data. Skewness and kurtosis for the normal distribution are 0 and 3 respectively. Jarque-Bera test is used for checking whether an error has normal distribution or not.

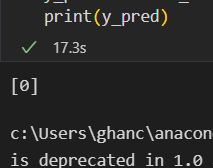
Using pickle module, we deployed our model or project in python to a .pkl format file.





As we can see it showed us the expected result i.e. negative sentiment.

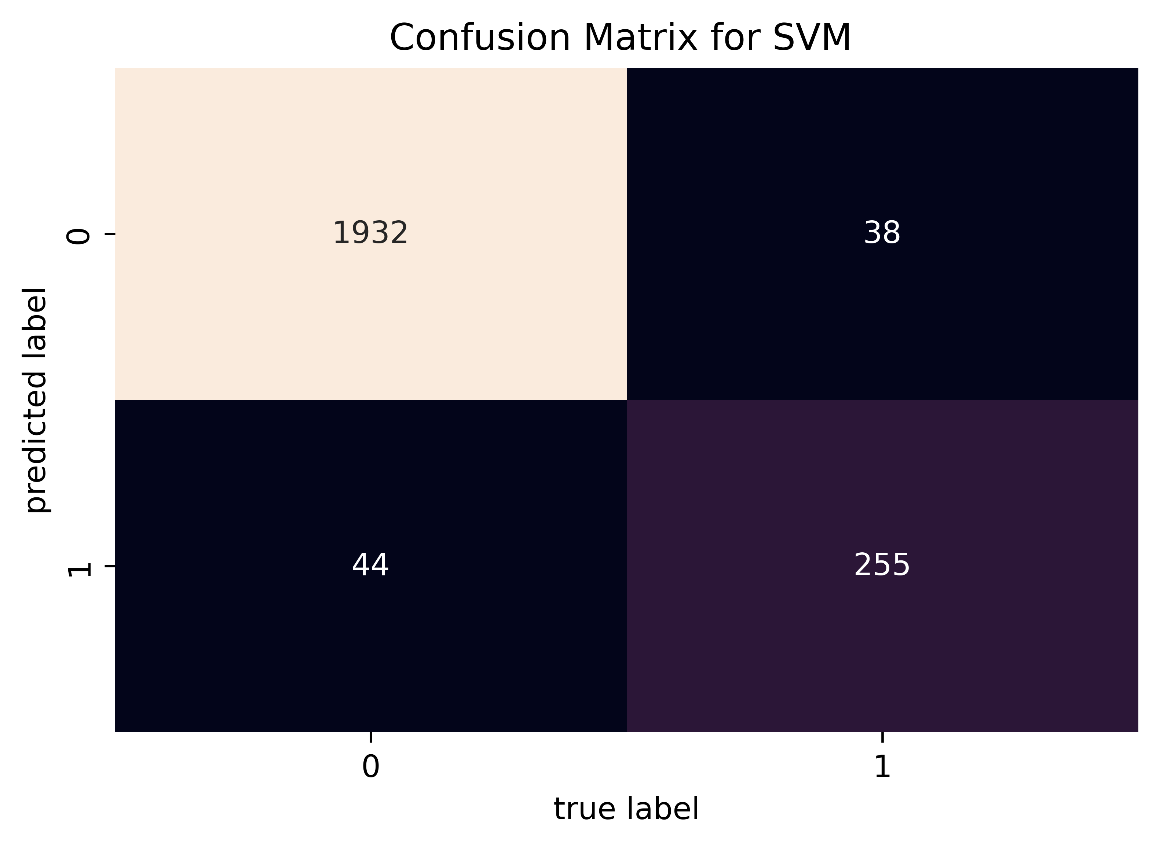




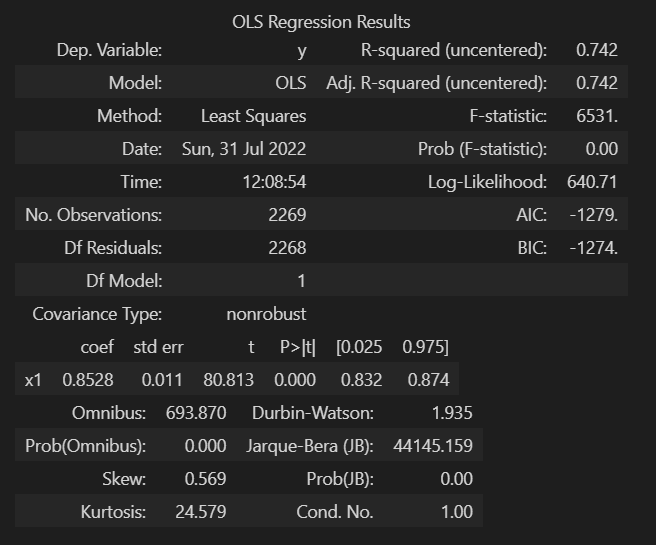
As we can see it showed us the expected result i.e. positive sentiment.

1. **SVM model**: The second model that we used was svm model. All the function and the code till the model selection is the same as the naïve bayes model code. Using sklearn we selected svm linear model. The accuracy using this model came out to be 96.38%.

After fitting the vector to the model, we got the following confusion matrix.



The OLS results are as below:



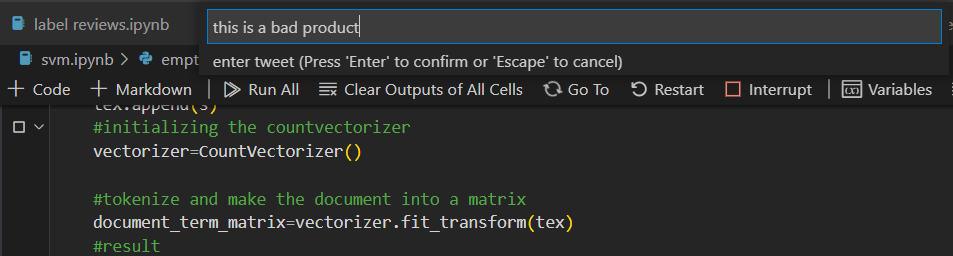
**Interpretation of OLS results:**

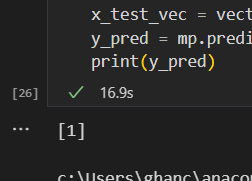
**R squared value:** Here, 74.2 % variation in Y can be explained by X. The maximum possible value of R2can be 1, means the larger the R2value better the regression.

**F-Statistic value:** Inserting the values of R2, n and k, F = (0.742/1) / (0.258/2268) = 6522.7.

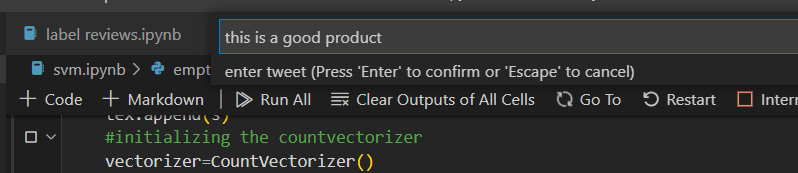
You can calculate the probability of F >6522.7 for 1 and 2268 df, which comes to approx. 0. As per the above results, probability is close to zero. This implies that overall the regressions is meaningful.

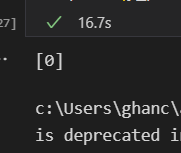
As above step, we deployed this model too using pickle module.





As we can see it showed us the expected result i.e. negative sentiment.

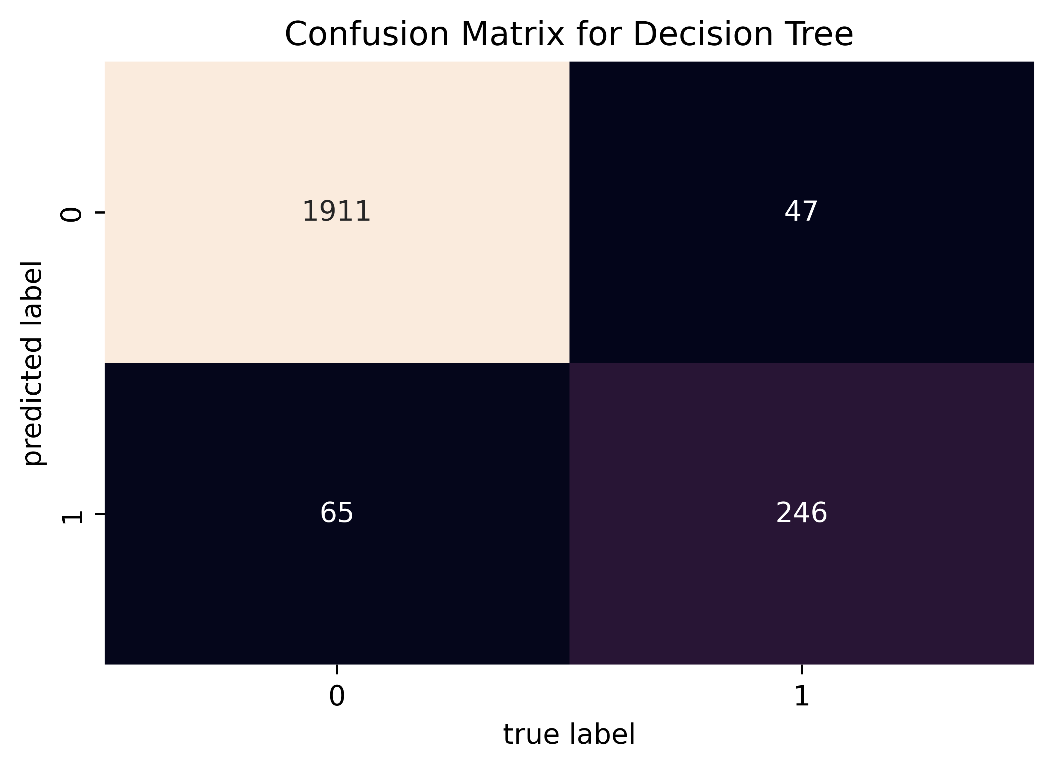




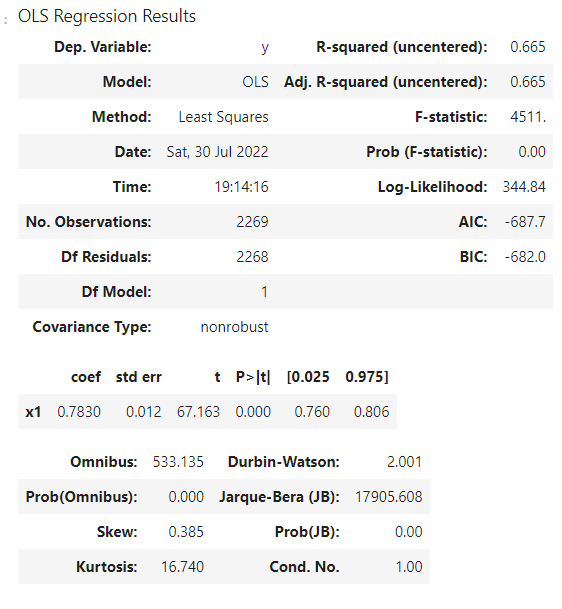
As we can see it showed us the expected result i.e. positive sentiment.

1. **Decision tree model**: The last model that we used was decision tree model. All the function and the code till the model selection is the same as the naïve bayes model code. Using sklearn we selected decision tree model and set the criterion to entropy. The accuracy using this model came out to be 95.01%.

The confusion matrix of using this model is as below:



The OLS results are as below:



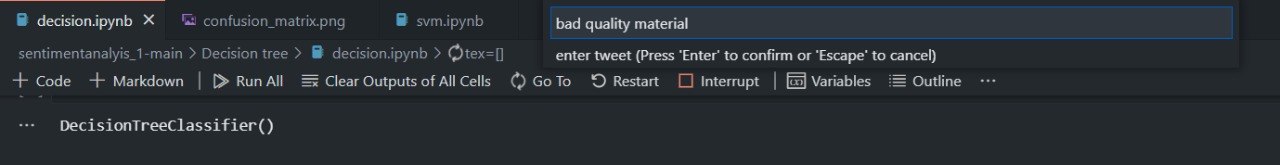
**Interpretation of OLS results:**

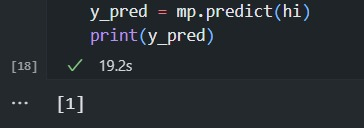
**R squared value:** Here, 66.5 % variation in Y can be explained by X. The maximum possible value of R2can be 1, means the larger the R2value better the regression.

**F-Statistic value:** Inserting the values of R2, n and k, F = (0.665/1) / (0.335/2268) = 4502.1.

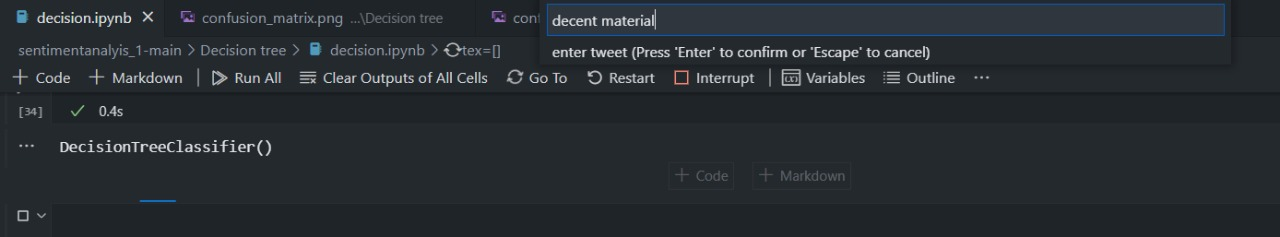
You can calculate the probability of F >4502.1 for 1 and 2268 df, which comes to approx. 0. As per the above results, probability is close to zero. This implies that overall the regressions is meaningful.

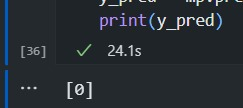
As above step, we deployed this model too using pickle module.

`



As we can see it showed us the expected result i.e. negative sentiment.





As we can see it showed us the expected result i.e. positive sentiment.

Comparison of the three models:

Conclusion: As we can see from both the graphs, svm has the highest r squared value and accuracy, so we conclude that in our task of sentiment analysis, SVM model is of best use.

\*We have added comments in all the code files so that it would be easy to interpret\*