Sarcasm Detection: Natural Language Processing

Artificial Intelligence Project

Instructions for Program:

- Extract project zip file
- Run naive bayes and logistic regression using the command line. Python filename.py

Contributors:

- Arpankumar Vijaykumar Rajpurohit: worked on code and report
- Krishna Sai Panthala : worked on code and report
- Siva Sai Kumar Paritala: worked on code and report

Introduction:

- Sarcasm hides in plain sight of words as sentiment which requires intelligence to understand the meaning behind the words. This project goes over various natural language processing algorithms to gain the insights into the sarcasm hidden in the news headings and how we can analyze and develop different algorithmic models such as naive bayes, logistic regression to get the insights into sarcasm detection.

Overview:

- This project contains the dataset where headlines, link articles and info of whether the headline is sarcastic or not is given.The model we are using from the given dataset is detection model (Supervised learning - Classification).
- Firstly, Naive Bayes is a machine learning algorithm which works on probability. It is a probabilistic model and a gaussian probabilistic model works on the continuous variable.
- Secondly, logistic regression is implemented in the project. The method of modeling the likelihood of a discrete result given an input variable is known as logistic regression. Here, logistic regression works as a binary outcome model. Binary outcome is estimated by maximum likelihood estimation.

Dataset:

- Link to dataset is given below, dataset is a json file containing three properties is_sarcastic, headline, article_link. The dataset contains 26,709 rows.
- https://www.kaggle.com/datasets/rmisra/news-headlines-dataset-for-sarcasm-detection?resource=download

Approach:

- For programming, python will be used because it contains many libraries for machine learning processing. Furthermore, language processing functions are imported from sklearn, to load json data pandas library is used and matrix operations are supported by numpy.
- For Preprocessing, firstly, the data as headlines is filtered with quotations, exclamation marks and questions then it is transformed into sufixed versions and lowercase letters. Then data is split into a train and test set which are used to create a model and test the model. After testing the model, a confusion matrix is generated and output is presented in visual presentation denoting accuracy, error, precision and recall.
- For Naive Bayes, a given equation is implemented where x is a vectorised processed word dataset and output is binary of sarcasm detected or not detected.

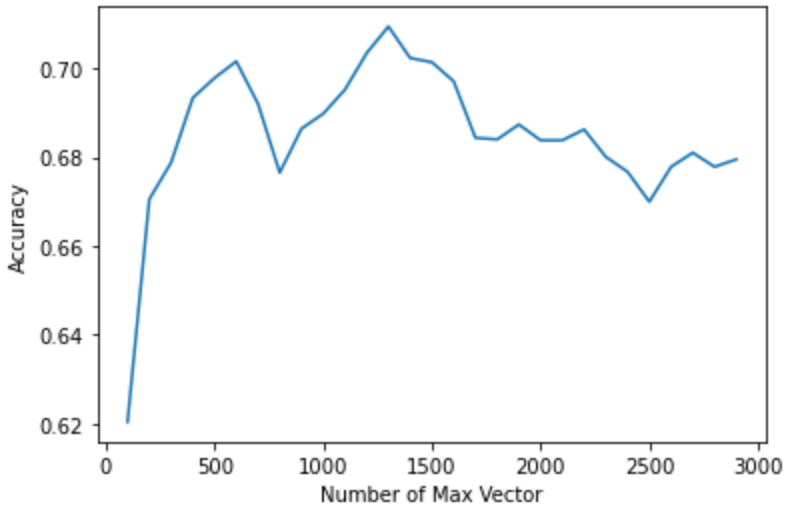
$$P(X|Y=c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{\frac{-(x-\mu_c)^2}{2\sigma_c^2}}$$

- For logistic regression, sigmoid activation function is implemented where layer input is calculated of theta and input which is served to calculate the cost between predicted output and output. Using cost and learning rate theta is updated until given iterations.

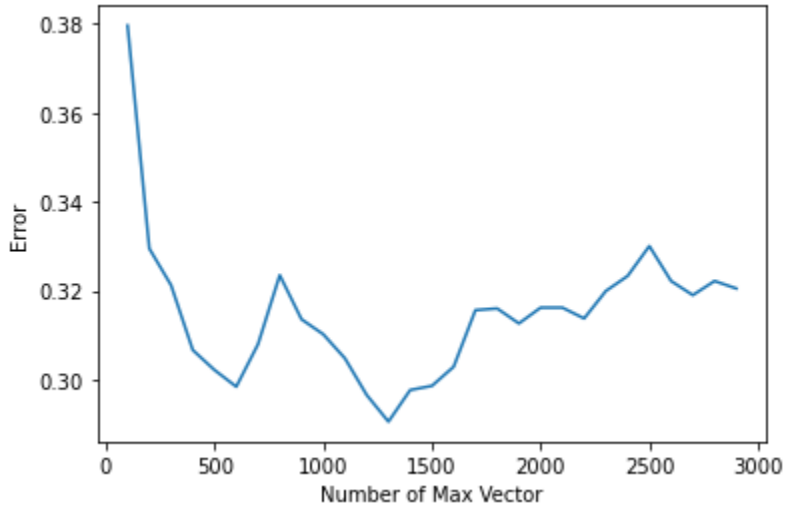
$$S(x) = \frac{1}{1+e^{-x}}$$

Analysis:

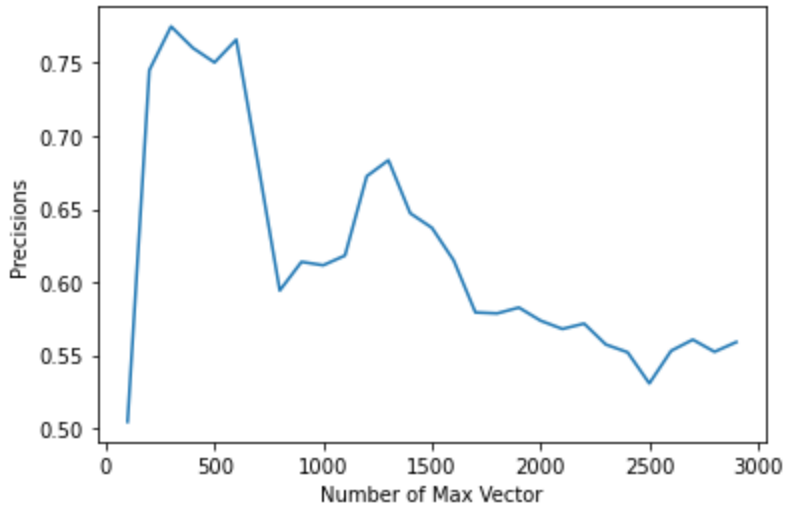
- Naive Bayes accuracy graph



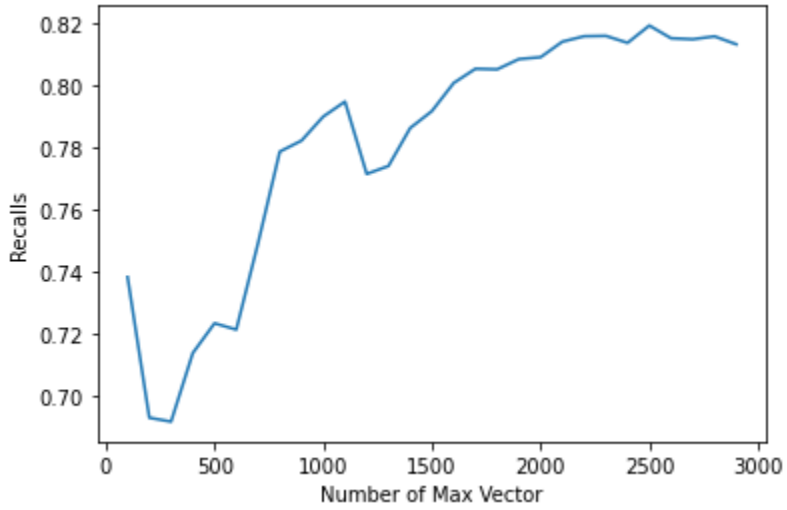
- Naive Bayes Error graph



- Naive Bayes Precision graph



- Naive Bayes recall graph



- For Naive Bayes, maximum accuracy was gained using 1300 max vectors with an accuracy of 71%, error rate of 29%, precision of 68% and recall of 77%.
- For Logistic regression, we achieved accuracy of 80%, 20% error rate, 88% precision and 78% recall.

-
- For logistic regression, confusion matrix can be seen above

- From analysis we can see that logistic regression gives more accurate results where accuracy increased by 9%.

Challenges:

- **Data preparation:** to prepare the data that we can utilize in the model such that it gives more accuracy and stays relevant to the output was the first challenge we faced.
- **Data Processing:** at the data processing phase trying to convert the data from string to vector , the new transformed array was large in memory size so implementing operation of type conversion stopped program progress altogether.

Future Work:

- In future, to understand the insights better and increase the accuracy of the model, we would like to add a whole article from fetched links and train it on the state of the art models like GPT - 2 and BERT. Furthermore, the progress can also be made with the focus of how to give feedback in realtime and optimize the model for faster processing using parallel programming.

Conclusion:

- In conclusion, this project attempts to understand the different accuracies and learning capabilities of different natural language models by implementing them and analyzing the results to better serve the problem statement of how to detect the sarcasm which is hidden in the words and can be skipped in normal understanding. Here, sarcasm analysis can be viewed as one kind of sentiment analysis.

Reference:

- Gaussian Naive Bayesian Data Classification Model Based on Clustering Algorithm
- Read this paper if you want to learn logistic regression