Interim Project Report [Group: 6]
CSE343: Machine Learning, Winter 2022

Mohd Naki
(2018052)
naki18052@iiitd.ac.in

Vaibhav Gupta
(2019341)
vaibhav19341@iiitd.ac.in

1. Introduction

This section will contain the problem statement and the motivation.

1.1. Problem Statement

The objective of our project is to classify the positive and negative reviews of consumers over different types of products and build a supervised learning model to polarize huge amounts of reviews.

1.2. Motivation

The motivation behind implementing this technique is that organisations always want to find consumer opinions and emotions about their services and products. Prospects also want to know the opinions and emotions of existing consumers before they purchase the product or service.

2. Related Work

We referenced a research paper from Stanford which attempts to do sentiment analysis on the same dataset: https://cs229.stanford.edu/proj2018/report/122.pdf
The best performing model from the above paper is a LSTM model which achieved 73.5% training accuracy and 71.5% testing accuracy

3. Dataset and Evaluation

The dataset is a list of 34,660 consumer reviews for Amazon products like the Kindle, Fire TV Stick, and more provided by 'Datafiniti's Product Database'. The dataset includes basic product information, rating, review text, and more for each product. This dataset is a sample from a larger dataset.
After removing samples with unwanted and empty values, 34,627 samples were left. These were further split as:

- Train set: 64% = 22161 samples
- Validation set: 16% = 5540 samples
- Test set: 20% = 6926 samples

Dataset url: https://www.kaggle.com/datasets/datafiniti/consumer-reviews-of-amazon-products?select=1429_1.csv

3.1. Features

The dataset has 21 features. Some of which are ID, Name, brand, category, manufacturer, review date, review id, review rating, review text, user city, user province, user name.
Out of these we have extracted review rating (as target) and review text (as independent feature) to train the models.

3.2. Pre-processing

Sentiments are classified as follows:

$$Positive : rating > 3$$
$$Neutral : rating = 3$$
$$Negative : rating < 3$$

We have used the 'Bag of Words' strategy to turn the data into numerical features. We pre-processed the text using:

- **Tokenization:** Breaking the review text into words.
- **Stop-words Filtration:** Words like "the", "are" are filtered.
- We used occurrence counting, which built a dictionary of features from integer indices with word occurrences.
- We converted this dictionary of texts into a feature vector.

The above was achieved by CountVectorizer() from scikit-learn.
We have 27701 training samples with 12487 distinct words. In longer text documents, we see a higher average count value of words that are insignificant, this particular problem will overweigh the text documents that have lower average counts with same frequency. To overcome this redundancy, we used Tfidf-Transformer() from scikit-learn.

3.3. Evaluation metrics

- Accuracy
- Confusion matrix
- F1 score
- Precision score
- Recall score
- Log loss
- ROC-AUC plots
- Learning Curve
- Validation Curve

4. Analysis & Progress

This section contains the progress so far and the analysis and observations made.

4.1. Models trained

2 simple models:

- Multinomial Naive Bayes
- Logistic Regression

and 1 advanced model:

- Linear Support Vector Machine

4.2. Analysis

From *Figure 1* and *Figure 2* it can be infered that the dataset is unbalanced.
Figure 3 show the separability of the dataset by plotting the frequent and infrequent words occurring in positive and negative reviews. For example: 'love', 'highly recommended', 'perfect', 'excellent' occur in positive reviews. And 'defective', 'returning', 'poor', 'disappointing' occur in negative reviews.

4.3. Observations

Figure 4 shows that the training and validation set converge very quickly. However, *Figure 5* shows that with more samples higher accuracy can be achieved. And *Figure 6* shows that after 20,000 samples the training and validation accuracy start to diverge.
Figure 7, Figure 8 and Figure 9 show that with more samples lower loss can be achieved but they also show that the training set is unrepresentative. Meaning that it is a lot easier for the model to learn than the validation set. This means that the training set requires a more varied amount of samples. Currently the models are **under-fitting**.
Figures 10, 11 & 12 show the ROC-AUC curves for the different classes and models. From these graphs we can infer that all models have room for improvement as larger the Area Under the Curve, higher is the performance of the model.
From *Figure 13* we can observe that a smoothing parameter of 0.7 is sufficient for optimal training for the Naive Bayes model. From *Figure 14* we can observe that 'L1' regularization gives us the best result.

5. Results

Results for the test set. All models performed reasonably well. The Naive Bayes model being the simplest had the lowest accuracy and highest error. The Linear SVM being the most complex of the bunch had the highest accuracy and lowest error. The Logistic Regression model's performance was better than Naive Bayes but not as good as the Linear SVM.

Score	**Multinomial Naive Bayes**	**Linear SVM**	**Logistic Regression**
Accuracy	0.931	**0.935**	0.934
max. F1 score	0.964	0.967	0.966
avg. F1 score	0.321	0.405	0.445
max. Precision score	0.931	1.0	0.938
avg. Precision score	0.310	0.9	0.702
max. Recall score	1.0	0.999	0.997
avg. Recall score	0.33	0.382	0.407
Log loss	2.358	**2.169**	2.229

5.1. Comparison

The state of the art model mentioned in Section 2 had an accuracy of 71.5% on the test set whereas our linear SVM was able to achieve an accuracy of 93.5% on the test set.

6. Future Work

This section contains the work that will be done for the final report.

Learning techniques

- K-Nearest Neighbour Classification
- Neural Network

Dataset: Train models on full or a larger sample of the dataset.
Metrics: We will be adding the following metrics for the final report: *Heatmap & Error tree*

6.1. Analyses

- Understand the working of existing and implemented models
- Diagnose the results obtained
- Refine and improve the results

6.2. Team member roles

Mohd Naki:

- Implement Neural Network
- Improve Linear SVM
- Improve Multinomial Naive Bayes
- Heatmap
- Perform accuracy and error analysis

Vaibhav Gupta:

- Implement K-Nearest Neighbour Classifier
- Improve Linear SVM
- Improve Logistic Regression
- Error tree
- Perform accuracy and error analysis

7. Figures, Plots & Tables

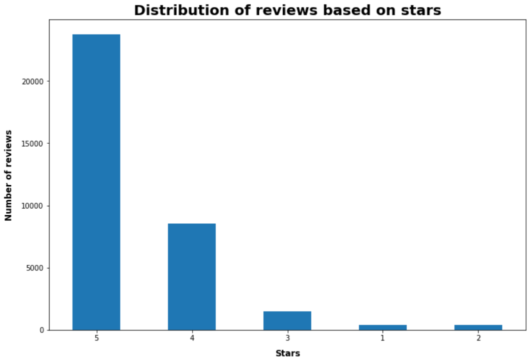

Figure 1. Number of reviews in each rating

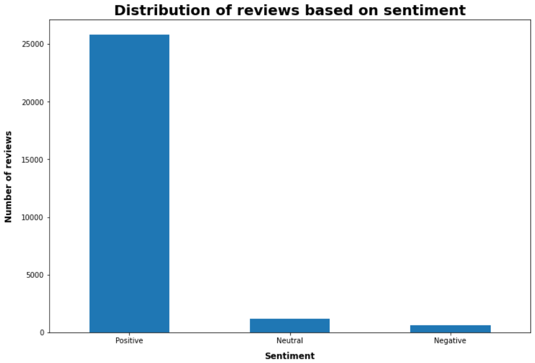

Figure 2. Number of reviews in each sentiment

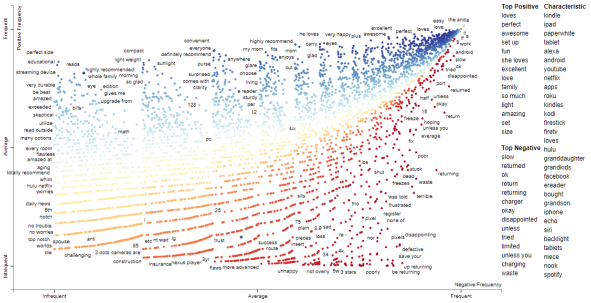

Figure 3. Separability of dataset

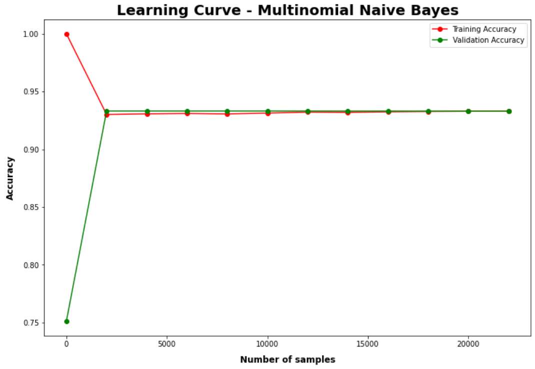

Figure 4. Accuracy of Naive Bayes model with increasing samples

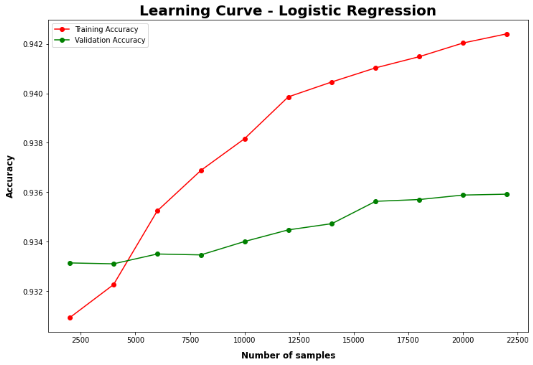

Figure 5. Accuracy of Logistic Regression model with increasing samples

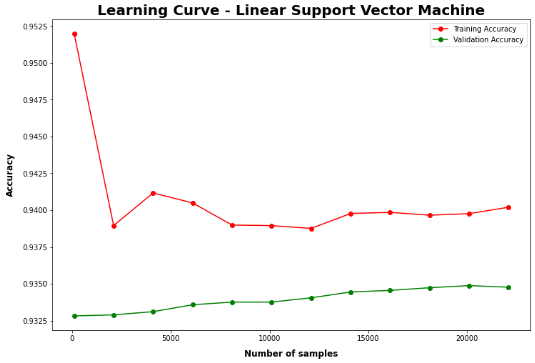

Figure 6. Accuracy of Linear Support Vector Machine model with increasing samples

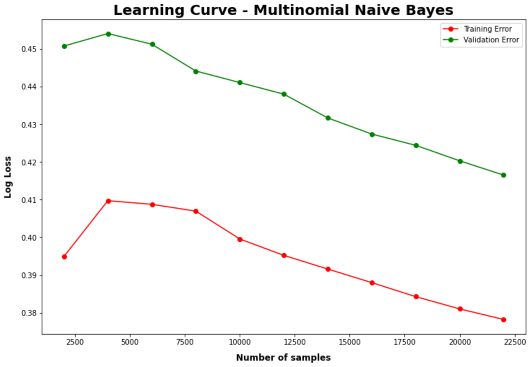

Figure 7. Error of Naive Bayes model with increasing samples

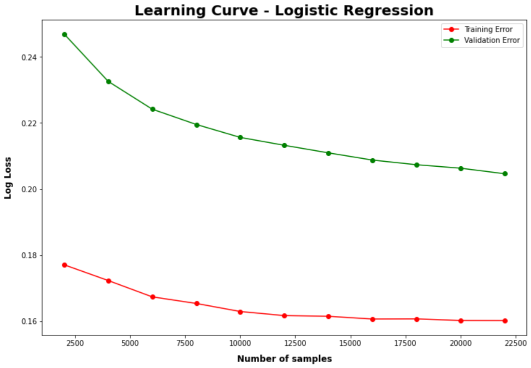

Figure 8. Error of Logistic Regression model with increasing samples

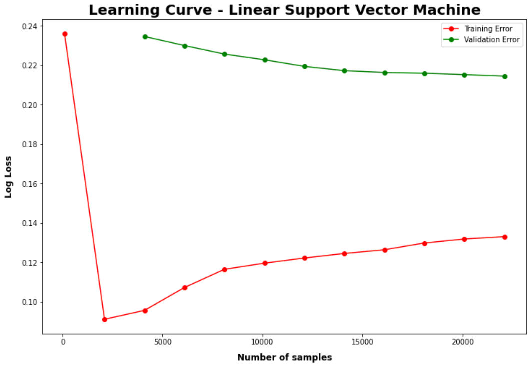

Figure 9. Error of Linear Support Vector Machine model with increasing samples

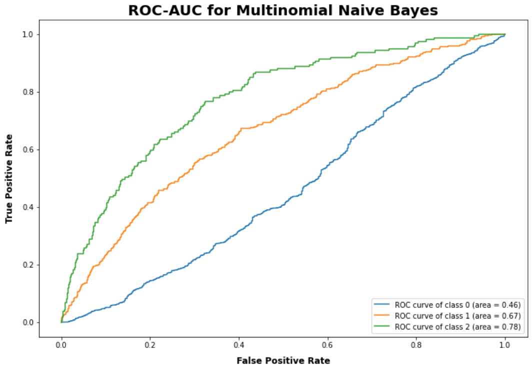

Figure 10. ROC-AUC curve for Naive Bayes model

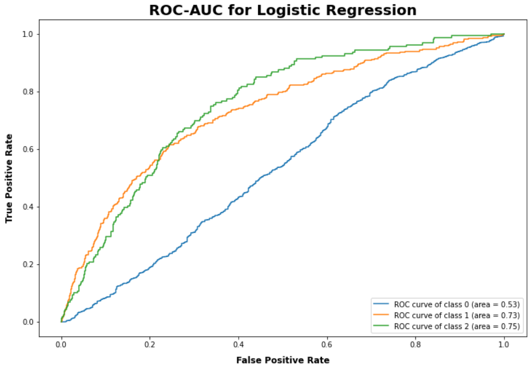

Figure 11. ROC-AUC curve for Logistic Regression model

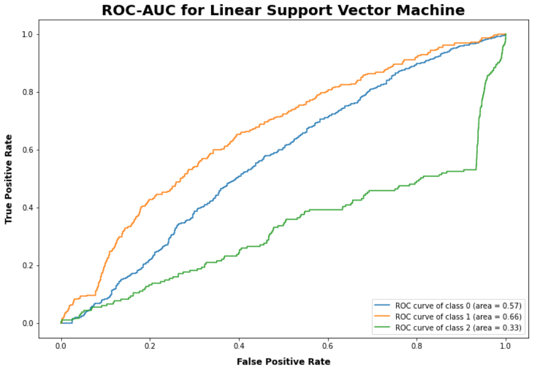

Figure 12. ROC-AUC curve for Linear Support Vector Machine model

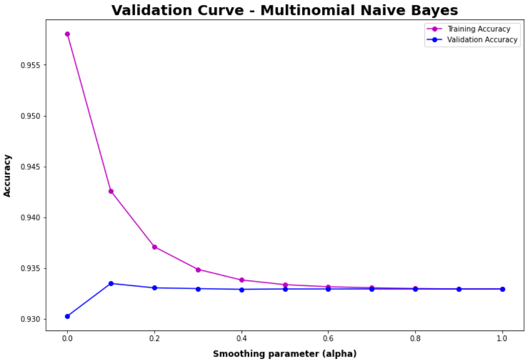

Figure 13. Accuracy of Naive Bayes model with varying hyper-parameter

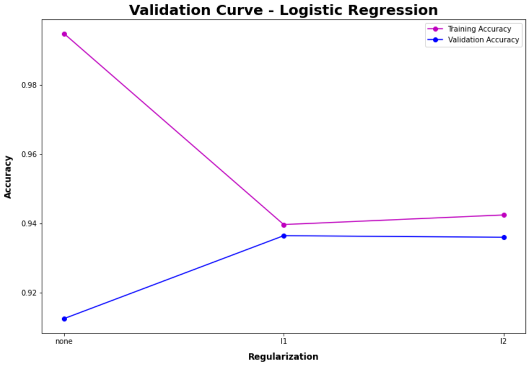

Figure 14. Accuracy of Logistic Regression model with varying hyper-parameter