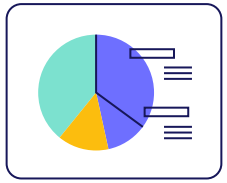
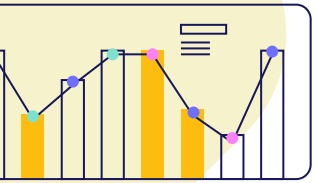
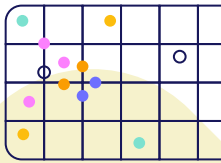


Sentiment Analysis with Naive Bayes Classifier: Insights from Amazon Reviews

By - Manpreet Kaur and Anayna Singh

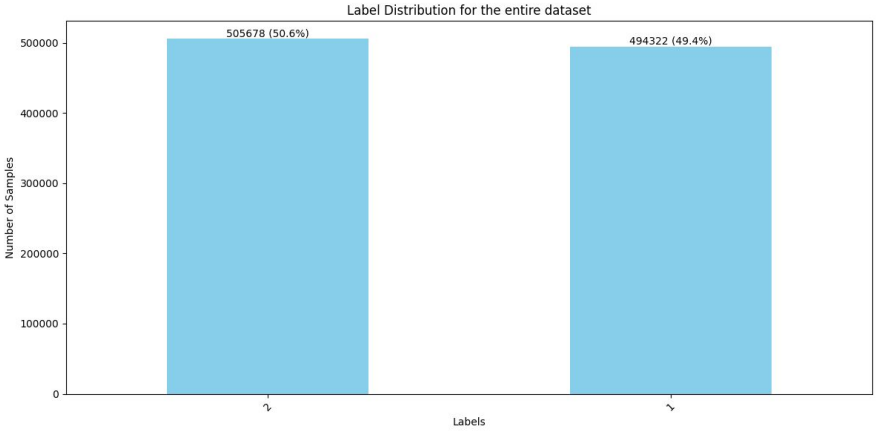


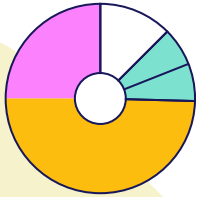


Data Set Information

- **Description:** Amazon reviews dataset for Sentiment Analysis (.csv format).
- **Usage:** Training and testing a Naive Bayes classifier for sentiment analysis.
- **Size:** Total 3.5 million reviews – used 1 million for training and testing.
- **Labels:** Label 1 and Label 2 for positive and negative sentiments, respectively.
- **Split:** Train/Test: 80/20 and 60/20
- **Challenges:** Possible inconsistencies, necessitating preprocessing.
- **Preprocessing:** Removal of HTML tags, URLs, non-alphanumeric characters, stop words, and lemmatization.
- **Link to dataset:** https://www.kaggle.com/datasets/bittlingmayer/amazonreviews

Training / Test Sets





Our Approach

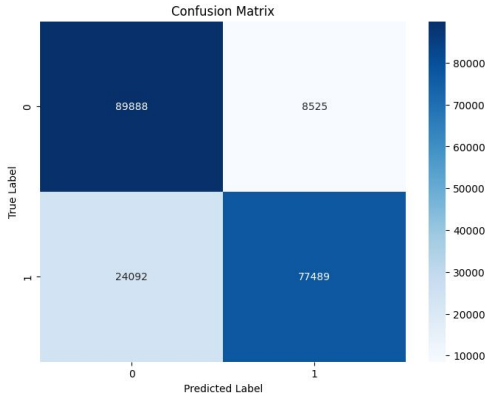
- **Algorithm Pseudocode:**
  - Load and preprocess dataset.
  - Split into training and test sets.
  - Train Naive Bayes classifier.
  - Test classifier and calculate metrics.
  - Classify user-entered sentences.

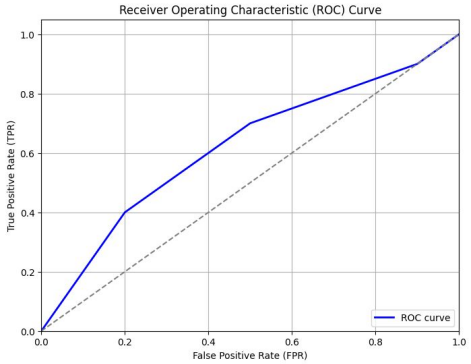
- **Details:**
  - Implemented interactive user interaction.
  - Allowed adjusting training set size via CLI.

- **Implementation:**
  - Utilized pandas, sklearn, and nltk.
  - Employed Laplace smoothing for unseen words.
  - Removed HTML tags, URLs, and stopwords.
  - Calculated metrics like accuracy and F1-score.

Evaluation

Train Size = 80





Sensitivity (recall): 0.763

Specificity: 0.913

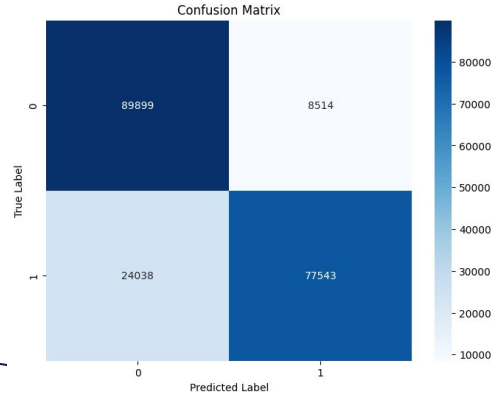
Precision: 0.901

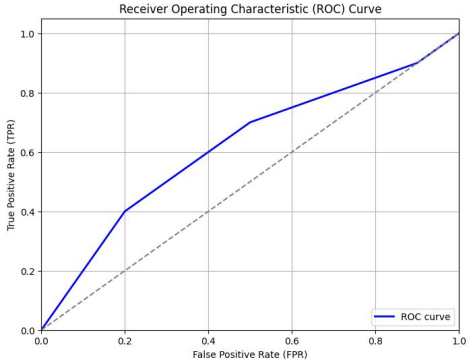
Negative predictive value: 0.789

Accuracy: 0.837

F-score: 0.826

Train Size = 60





Sensitivity (recall): 0.763

Specificity: 0.913

Precision: 0.901

Negative predictive value: 0.789

Accuracy: 0.837

F-score: 0.826

Demo:

```
C:\Users\kaur6\Downloads\NLP_CSV>python pa2.py 60
C:\Users\kaur6\Downloads\NLP_CSV\pa2.py:3: DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major release of pa
(to allow more performant data types, such as the Arrow string type, and better int
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

  import pandas as pd
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\kaur6\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\kaur6\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Kaur, Manpreet, A20551672 solution:
Training set size: 60 %

Training classifier...
Testing classifier...
Test results / metrics:

Number of true positives: 77543
Number of true negatives: 89892
Number of false positives: 8519
Number of false negatives: 24032
Sensitivity (recall): 0.7634063499876939
Specificity: 0.9134344737884993
Precision: 0.9010132230252609
Negative predictive value: 0.7890523506899336
Accuracy: 0.8372336063524447
F-score: 0.8265214216812249
```

```
Enter your sentence:
it was the best product ever i have bought

Sentence S:

it was the best product ever i have bought

was classified as Label 2.
P(Label 2 | S) = 1.0395981367474991e-14
P(Label 1 | S) = 2.096794714350705e-15

Do you want to enter another sentence [Y/N]? y
Enter your sentence:
i hate the quality of the fabric

Sentence S:

i hate the quality of the fabric

was classified as Label 1.
P(Label 2 | S) = 3.224061437067225e-12
P(Label 1 | S) = 1.117241825198671 2e-11

Do you want to enter another sentence [Y/N]? y
Enter your sentence:
it is neither good not bad

Sentence S:

it is neither good not bad

was classified as Label 1.
P(Label 2 | S) = 2.076693480860348e-08
P(Label 1 | S) = 3.642125393218617e-07

Do you want to enter another sentence [Y/N]? n

C:\Users\kaur6\Downloads\NLP_CSV>
```

Summary

What did you observe? Did it match your expectations?

- The model's performance reaches a plateau, showing minimal improvement with increased training data from 60% to 80%.
- Achieving around 84% accuracy, the model effectively classifies most data.
- Using 80% training data doesn't offer a significant advantage over 60%.

What surprised you?

- The lack of significant improvement in performance despite increasing the training data size.
- The consistency of performance metrics across different training sizes.

Challenges

- Balancing recall and precision to achieve optimal performance.
- Identifying factors that limit further improvements in model performance.

Possible Improvements

- Fine-tuning model parameters to optimize performance.
- Implementing advanced techniques such as ensemble learning or neural networks for better performance.

Thank you