SentimentSphere: Sentiment Classification of E-commerce Reviews Using Textual Data Analysis

Ahsanullah University of Science and Technology

Department of Computer Science And Engineering

CSE4114 | Pattern Recognition and Machine Learning Lab

Presented To:

Mr. Ashek Seum

Lecturer, Dept. of CSE, AUST

Mr. Sajib Kumar Saha Joy

Lecturer, Dept. of CSE, AUST

Presented By:

Name: **Saikat Das**

ID: **2021010158**

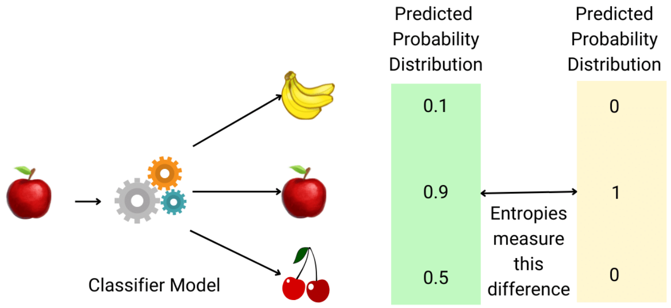
Section: **C2**

Group: **08**

Introduction to Sentiment Analysis

Sentiment analysis is the process of determining the sentiment (positive or negative) expressed in a piece of text.

Classify text into **positive or negative sentiments**, such as customer reviews or social media comments.



Used in **customer feedback**, **social media monitoring**, and **e-commerce reviews** to understand opinions and improve services/products.

SentimentSphere: Project Summary

Implement a sentiment analysis model using **Logistic Regression** to classify comments as **positive** or **negative**.

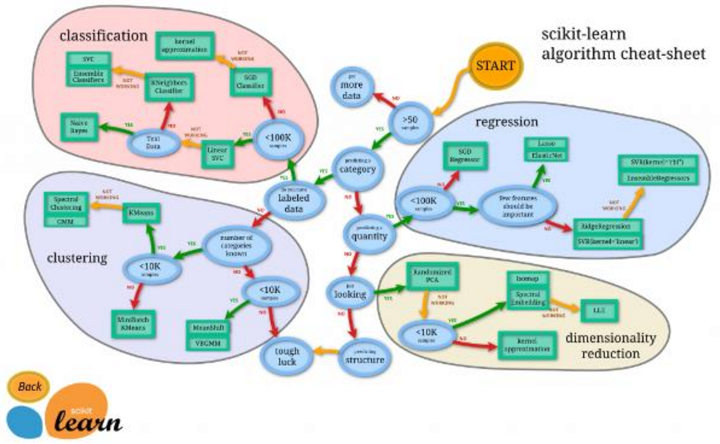
Dataset:
3.6 million labeled Amazon customer reviews, each with a sentiment label (1 = negative, 2 = positive).

And **0.4 million** labeled test dataset.

Expected Accuracy:
Aim for **88% - 90%** accuracy in sentiment classification.

Evaluation Metrics:

- **Precision**
- **Recall**
- **F1-Score**



Dataset Overview

Dataset Size:

- Total of **4 million reviews**
- **3.6 million** reviews for **training**
- **0.4 million** reviews for **testing**

Format:

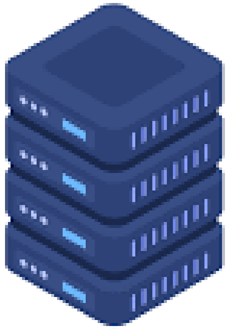
- CSV files: **train.csv** and **test.csv**

Sentiment Labels:

- **1:** Negative sentiment
- **2:** Positive sentiment

Source:

- Dataset sourced from **Kaggle**



	Label	Comments	Description
0	2	Stuning even for the non-gamer	This sound track was beautiful! It paints the ...
1	2	The best soundtrack ever to anything.	I'm reading a lot of reviews saying that this ...
2	2	Amazing!	This soundtrack is my favorite music of all ti...
3	2	Excellent Soundtrack	I truly like this soundtrack and I enjoy video...
4	2	Remember, Pull Your Jaw Off The Floor After He...	If you've played the game, you know how divine...

Data Preprocessing

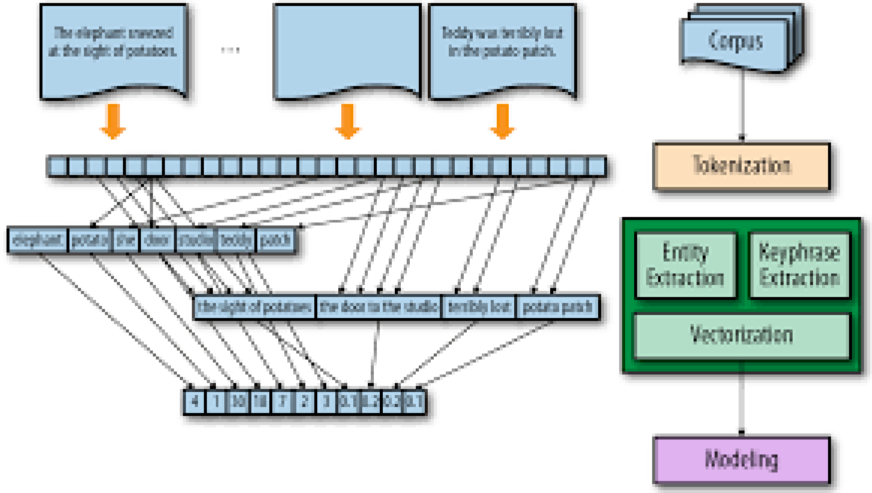

```python
# Fill missing values with empty string
df_train.fillna("", inplace=True)
df_test.fillna("", inplace=True)
```

Feature Engineering

TF-IDF Vectorization:

• Explains Term Frequency-Inverse Document Frequency method.
• Importance of weighing words based on frequency and document occurrences.

Purpose: Converts text into numerical data suitable for Logistic Regression.



Model Selections

Log Loss: A measure of how well the predicted probabilities match the actual labels.
• Lower Log Loss indicates better model performance.

Models Evaluated:
• **Logistic Regression**: Measures the probability of each class and calculates the log loss.
• **SGD (Stochastic Gradient Descent)**: Also predicts probabilities for each class.
• **Naive Bayes**: Provides probabilities for class predictions.

Log Loss Results:
• **Logistic Regression**: 0.345
• **SGD**: 0.402
• **Naive Bayes**: 0.467

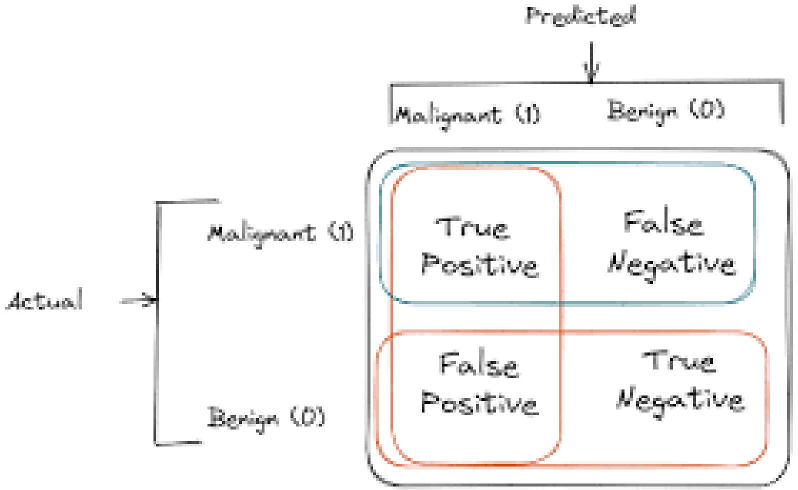
Conclusion: Logistic Regression performed the best with the lowest Log Loss.

Evaluation Metrics

- **Accuracy:**
  Percentage of correct predictions (Overall performance of the model).
- **Precision:**
  Ratio of correct positive predictions out of all predicted positives.
- **Recall:**
  Ratio of actual positive cases correctly predicted by the model.
- **F1-Score:**
  The harmonic mean of Precision and Recall, providing a balanced measure.

Results:

- **Accuracy:** 89%
- **Precision (positive class):** 0.90
- **Recall (positive class):** 0.88
- **F1-Score (positive class):** 0.89



Challenges & Limitations

- **Imbalanced Dataset**: Unequal distribution of positive and negative comments.
- **Noisy Data**: Informal language, slang, spelling errors, typos.
- **Computational Complexity**: Large dataset requiring significant resources.
- **Possible Solutions**: Techniques like oversampling, undersampling, or mini-batch gradient descent.

Future Work & Conclusion

Future Improvements:

• Hyperparameter tuning for better results.
• Use deep learning models (e.g., LSTM, CNN) for better performance.
• Real-time deployment in customer feedback systems.
• Multiclass classification for neutral sentiment.

Conclusion

• Successfully implemented the sentiment analysis model.
• Achieved 89% accuracy and balanced performance in precision and recall.
• Future efforts will focus on enhancing robustness and deploying the model.

Questions?