Classifying sentiment on the sentiment140 tweets dataset

I. Dataset

In this project, I use the Sentiment140 dataset, which consists of 1,600,000 tweets extracted using the Twitter API. This dataset is designed for sentiment analysis, classifying the sentiment of tweets as either positive or negative. [2].

The dataset has applications in areas such as product research and brand monitoring, where companies can analyze public sentiment towards their products or services based on real-time social media feedback. [1].

II. Classification Pipeline

The classification pipeline includes preprocessing, feature engineering, and model selection.

In preprocessing, mentions, links, punctuation, and stopwords were removed. Lemmatization was applied to normalize word forms.

For feature engineering, text was vectorized using:

- **CountVectorizer (Bag of Words)**: Counts word frequency.
- **TfidfVectorizer (TF-IDF)**: Assigns importance based on word frequency across the dataset.

The models used were:

- **Naive Bayes**: Assumes word independence, simple but struggles with context.
- **Logistic Regression**: Considers word interactions, better with negations but may miss sarcasm.

III. Evaluation

Among the four pipelines, Logistic Regression + TfidfVectorizer achieved the highest average accuracy of 77.55%. Therefore, I decided to evaluate this model in more detail.

Key Words for Classification

The most important words for both the negative and positive classes provide insight into how the model makes decisions.

- **Top Negative Words**: *sad, miss, unfortunately, sadly, died* — These words align strongly with negative emotions such as sadness and loss, indicating that the model effectively captures sentiment cues related to negative sentiment.
- **Top Positive Words**: *smile, thanks, thank, proud, welcome* — These words are typical indicators of happiness and gratitude, suggesting that the model correctly associates these terms with positive sentiment.

These words make sense and correspond well with the sentiment labels, meaning the model is not merely classifying based on coincidences but capturing meaningful patterns in the data.

Model Errors The model's performance varies slightly between the two sentiment classes:

- **Negative sentiment**: 23.81% error.
- **Positive sentiment**: 20.27% error.

IV. Dataset size

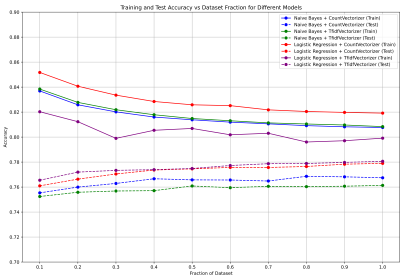

Fig. 1. Training and Test Accuracy vs Dataset Fraction for Different Models.

The difference between the training mean accuracy (80.41%) and test mean accuracy (77.55%) indicates a 2.86% gap. This suggests that increasing dataset size could improve accuracy by 1-2%.

Although there is a difference between train and test accuracy, there is limited room for improvement, which may not justify the cost of labeling more tweets.

V. Topic Analysis

To refine the results, I applied a two-layer classifier. Documents were first classified into topics, and then a sentiment classifier was applied to each topic individually. The overall accuracy of the two-layer classifier was **0.7514**.

A. Topic-Specific Accuracies

- **Topic 0**: 72.82%
- **Topic 1**: 75.93%
- **Topic 2**: 73.90%
- **Topic 3**: 74.08%
- **Topic 4**: 78.78%

B. Conclusion

The error rates vary across topics. The classifier performs better on certain topics (e.g., Topic 4 with 78.78% accuracy) than others (e.g., Topic 0 with 72.82% accuracy). This suggests that specific topics contain clearer sentiment cues, improving classification performance. The two-layer approach allows for more accurate predictions on certain subsets of the data.

Identify applicable funding agency here. If none, delete this.

REFERENCES

[1] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," CS224N Project Report, Stanford, 2009.

[2] Sentiment140 dataset on Kaggle. Available at: https://www.kaggle.com/datasets/kazanova/sentiment140

[3] Ramos, J. (2003). Using TF-IDF to determine word relevance in document queries.

[4] Harris, Z. S. (1954). Distributional Structure. Word, 10(2-3), 146-162.

[5] McCallum, A., Nigam, K. (1998). A comparison of event models for Naive Bayes text classification.

[6] Cox, D. R. (1958). The regression analysis of binary sequences. Journal of the Royal Statistical Society.