AML Midterm Project - Sentiment Analysis

Aditya Mahesha Ballaki (ab3237@cornell.edu)
Maria DeCaro (msd249@cornell.edu)

November 3, 2024

Abstract

In this project, we implemented a hybrid approach to sentiment analysis using a combination of supervised and unsupervised learning algorithms on a dataset of phrases labeled with sentiments ranging from 0 to 4. The dataset, denoted as D, consists of partially labeled instances reflecting varying degrees of sentiment intensity. To enhance the predictive capabilities of our models, we applied several preprocessing techniques to clean and vectorize the textual features. Our modeling strategy included K-Nearest Neighbors (KNN), Naive Bayes, and Logistic Regression algorithms, allowing for performance evaluation in sentiment prediction. Additionally, we employed techniques to augment missing labels in the dataset, which improved the comprehensiveness of our model validation.

After augmenting the dataset using predictions from Logistic Regression, the Naive Bayes classifier achieved an accuracy of 95.4% on the validation test set, making it our best result.

1 Methods

1.1 Pre-processing

1.1.1 Preprocessing for K-means

For K-means clustering, we enhanced input quality by cleaning the text: removing URLs, HTML tags, and punctuation, while converting everything to lowercase. Stop words were eliminated to retain meaningful terms, and we tokenized and lemmatized the text. The cleaned text was transformed into a Bag of Words model for structured representation suitable for clustering.

1.1.2 Preprocessing for Logistic Regression

In our preprocessing for Logistic Regression, we standardized phrases by converting them to lowercase, removing punctuation and special characters, and eliminating URLs. We dropped rows with missing values and applied lemmatization using the WordNetLemmatizer. Finally, we performed TF-IDF vectorization on the processed phrases to convert the text into numerical format for analysis.

1.2 Dataset Augmentation

1.2.1 Clustering using K-Means

We employed K-means clustering on the unlabeled dataset to generate pseudo-labels, using 5 clusters based on the `label_with_kmeans` function, which was set with `n_init=5` and `max_iter=300` for robust convergence. The clusters were subsequently mapped to known class labels using majority voting, assigning representative labels based on the most common true labels within each cluster. We validated the K-means clustering on the labeled training data, achieving an aligned accuracy with true labels of 0.5853 and an Adjusted Rand Index (ARI) of -0.0581. This validation indicated the effectiveness of our clustering method, providing an initial signal for additional model training.

In addition to K-means, we attempted to implement PCA and hierarchical clustering; however, both approaches were limited by RAM constraints,

which prevented their successful implementation. We specifically used Truncated SVD instead of PCA, as it is better suited for handling sparse matrices. Unfortunately, Truncated SVD yielded similar accuracy results to K-means, with an aligned accuracy of 0.5853 but a lower ARI of -0.0652. These results indicated that we could not rely on Truncated SVD for effective dimensionality reduction, leading us to decide against its use in favor of the K-means approach, which showed better overall performance. Despite these challenges, our experiments with K-means clustering suggest that there is potential for further exploration of clustering methodologies in future work, particularly if we have improved RAM capabilities to support more sophisticated techniques.

1.2.2 Logistic Regression

In this method, we apply Logistic Regression (without regularization) from the `sklearn` library, using TF-IDF to preprocess the text data beforehand. Logistic Regression is a statistical model commonly used for binary classification tasks; it predicts the probability of class membership by modeling the relationship between input features and a binary outcome using the logistic function, represented as

$$P(y = 1|x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n)}}$$

where β coefficients are learned from the data. We start by dividing the training dataset into labeled and unlabeled subsets, fitting the labeled subset to the Logistic Regression model to learn from known classifications. Then, we make predictions on the unlabeled data, augmenting the dataset with these predictions.

1.3 Supervised Learning Prediction

1.3.1 Logistic regression and Logistic Regression

We applied logistic regression to assess model performance on both augmented and non-augmented datasets, using validation data for testing. First, we trained a plain logistic regression model using only the labeled data. Then, as described in section 1.2.2, we generated augmented data using logistic regression and trained a second model on the combined labeled and augmented dataset. Finally, we compared the accuracy of both models to evaluate the impact of this data augmentation approach on model performance.

1.3.2 Naive Bayes and Logistic Regression

We applied Naive Bayes to assess model performance on both augmented and non-augmented datasets, using validation data for testing. First, we trained a plain Naive Bayes model using only the labeled data. Then, as described in section 1.2.2, we generated augmented data using logistic regression and trained a second Naive Bayes model on the combined labeled and augmented dataset. Finally, we compared the accuracy of both models to evaluate the impact of this logistic regression-based data augmentation on the Naive Bayes model's performance.

1.3.3 Supervised learning on K-means Augmentation: including Naive Bayes and Logistic Regression

To enhance the final model performance, we implemented a sequential pipeline following the K-means clustering phase. Due to the significant amount of RAM required for applying supervised models on large datasets, we recognized the need to reduce dimensionality for the code to run efficiently. This reduction alleviated memory constraints and contributed to improved accuracy, allowing the models to focus on the most relevant features of the data.

In the next step, we balanced the pseudo-labeled data by downsampling it to 20% and combining it with the true-labeled data. This approach created a smaller, manageable subset that preserved class balance while accommodating memory constraints, ensuring that both pseudo-labeled and true-labeled samples contributed meaningfully. The models trained on this semi-supervised dataset, including Logistic Regression and Multinomial Naive Bayes, exhibited further improvements in validation accuracy, indicating that incorporating pseudo-labeled

data was advantageous at this stage.

To further refine model training, I introduced sample weights to prioritize the true-labeled data, assigning a weight of 2.0 to the labeled data. This adjustment helped the model focus more on reliable, labeled examples while still leveraging pseudo-labeled data. After implementing this weighting, Logistic Regression and Multinomial Naive Bayes models demonstrated slight accuracy improvements, with Logistic Regression improving from 0.324 to 0.360, while Naive Bayes maintained around 0.555. These results suggested that weighting could refine model learning without compromising generalization.

Additionally, to address class imbalance, we applied Random Oversampling to the semi-supervised dataset, duplicating samples from minority classes. This step enhanced model generalization by balancing class distributions. As a result, Logistic Regression's accuracy increased to 0.532, while Multinomial Naive Bayes saw a significant boost to 0.674. These findings demonstrated that oversampling effectively mitigated class imbalances, leading to improved multi-class performance and reinforcing the importance of each step in our sequential pipeline for model enhancement.

2 Results

Kaggle Logistic regression: 0.92939
Kaggle Naive Bayes: 0.93520

Model	Accuracy	F1 Score
Logistic Regression + Logistic Regression	91.21%	91.22%
Logistic Regression + Naive Bayes	95.42%	95.41%
No Augmentation + Logistic Regression	90.91%	90.90%
No Augmentation + Naive Bayes	95.31%	93.33%
K Means Clustering + Logistic Regression	51.37%	56.00%
K Means Clustering + Naive Bayes	70.95%	71.02%

Table 1: Model Performance on Validation Data

Method	Acc. (LogReg)	Acc. (NaiveBayes)
Downsampled Dataset	0.3043	0.5648
Sample Weights	0.3618	0.5633
Random Oversampling	0.5137	0.7095

Table 2: Clustering and Augmentation Summary

Table 2 depicts initial clustering is performed on the dataset, followed by downsampling, sample weights, and random oversampling on the augmented dataset before supervised learning. Values are based on the validation set.

K-means Clustering Results: Aligned Accuracy with True Labels = 0.5853, ARI = -0.1169.

3 Discussion

The results presented in Table 1 indicate a notable performance difference between the models applied to the dataset. The combination of Naive Bayes with the logistic regression augmentation achieved the highest accuracy of 95.42% and an F1 Score of 95.41%. This suggests that the model effectively captured the underlying patterns in the data, likely benefiting from the logistic regression augmentation process that expanded the training dataset, allowing the Naive Bayes classifier to learn from a more diverse set of examples. On the Kaggle leaderboard, Naive Bayes recorded a score of 0.93520, reinforcing its competitive performance.

In contrast, Logistic Regression without the additional logistic regression augmentation prior to running the model, yielded an accuracy of 90.91% and an F1 Score of 90.90%, demonstrating solid performance but indicating that augmentation plays a significant role in improving model effectiveness. The Kaggle accuracy score for Logistic Regression was 0.92939, which is comparable but slightly lower than that of the Naive Bayes model. The results also reveal a marked decline in performance for both models when K-means clustering was incorporated.

Logistic Regression combined with K-means clustering augmentation dropped to an accuracy of 51.37%, while Naive Bayes when combined with K-means augmentation showed an accuracy of 70.95%. These findings suggest that the clustering approach may not have aligned well with the actual data distribution, compared to the other methods we attempted, leading to poor predictive performance.

One possible reason for the limited benefit of augmenting the dataset could be mis-classification during the augmentation process. If the logistic re-

gression or clustering model misclassified instances while generating augmented data, this could introduce noise into the training set, negatively impacting the performance of subsequent classifiers.

We implemented the sequential pipeline following the K-means clustering phase to enhance final model performance, recognizing the need for dimensionality reduction due to the significant RAM requirements of applying supervised models on large datasets. Additionally, to address the class imbalance, we applied Random Oversampling (instead of SMOTE due to RAM capabilities) to the augmented dataset, which alleviated memory constraints and contributed to improved accuracy by allowing the models to focus on the most relevant features of the data.

Table 2 highlights the challenges encountered during the augmentation process with K-means clustering, as well as the benefits of the additional steps we implemented to enhance the augmented data before incorporating the supervised model. The initial K-means clustering results show an alignment accuracy with true labels of 0.5853 and an Adjusted Rand Index (ARI) of -0.1169. The negative ARI indicates poor clustering performance, suggesting that the clusters formed do not correlate well with the actual class labels. This further underscores the complexity of the dataset and the necessity of employing effective augmentation strategies to improve model performance.

The accuracy scores with the semi-supervised models (those including k-means) both for the downsampled dataset and those utilizing sample weights were notably low, particularly for Logistic Regression, with accuracy scores of 30.43% and 36.18%, respectively. Conversely, random oversampling significantly improved the accuracy for both classifiers, achieving 51.3% for Logistic Regression and 70.95% for Naive Bayes. This implies that oversampling was more effective in balancing the dataset and enhancing the models' ability to generalize from the training data.

The varying performance of classifiers can be attributed to their inherent characteristics. Naive Bayes, for instance, assumes independence among features, which can be advantageous in scenarios with high-dimensional data and limited training samples. Its ability to leverage the augmented data effectively contributed to its high accuracy. On the other hand, Logistic Regression relies on the linear relationship between features and classes, which may not capture the complexities of the dataset as effectively, especially when the augmented data is noisy.

Overall, these results highlight the importance of choosing the right methods for data preparation and augmentation to maximize the predictive power of machine learning models.

3.1 Alternative Approaches Explored

We initially attempted to use PCA for dimensionality reduction; however, due to low RAM availability, this approach was not feasible. Consequently, we turned to Truncated SVD, which yielded results of an aligned accuracy with true labels of 0.585 and an Adjusted Rand Index (ARI) of -0.0008, indicating subpar clustering performance that mirrored our earlier results with a lower ARI. Despite these challenges, we remain hopeful that improved RAM capabilities in the future will allow us to revisit PCA and explore its potential benefits for our analysis.

Further experiments with hierarchical clustering, limited by computational constraints, suggested an interesting but less feasible option. To refine our evaluation process, we introduced label augmentation in the test and validation sets using Logistic Regression, which helped adjust model evaluation and offered a potential pathway for enhancing model performance despite the limitations encountered in our primary analysis.

References

[1] Peng, C. Y. J., Lee, K. L., Ingersoll, G. M. (2002). An introduction to logistic regression analysis and reporting. *The Journal of Educational Research*, 96(1), 3-14.

[2] Park, H. A. (2013). An introduction to logistic regression: from basic concepts to interpretation with particular attention to nursing domain. *Journal of Korean Academy of Nursing*, 43(2), 154-164.