Mohammed F. Murshid
Professor Galletti
CS506
28 October 2024
Kaggle Username: mohammedmurshid

CS506 Midterm Report Fall 2024

Introduction

This project aimed to develop a predictive model for Amazon movie review ratings by exploring a range of machine learning algorithms and progressively enhancing model performance through ensemble methods. Beginning with an examination of the dataset's structure and dimensions, I noted its substantial size and complexity. The dataset encompassed numerous textual and numerical features, providing rich information but also presenting challenges in terms of processing and model training. The initial approach, leveraging the KNeighborsClassifier included in the starter code, allowed for a baseline assessment of model performance on this dataset. However, this classifier yielded an accuracy of around 0.39, indicating limited effectiveness. This result is likely due to KNeighborsClassifier's sensitivity to large datasets, where computational complexity increases exponentially with dataset size. Furthermore, KNeighborsClassifier does not capture complex interactions effectively in high-dimensional data, as it treats each feature independently, which may dilute the significance of textual and contextual features in a dataset as extensive as this.

Experimentation with Random Forest and Model Refinement

The next step involved testing the Random Forest Classifier, chosen for its adaptability to complex datasets and ability to handle large feature sets. The Random Forest algorithm, an ensemble method that builds multiple decision trees, combines the predictions of each tree to produce a more stable and accurate result. This approach is particularly beneficial when dealing with large datasets, as it reduces overfitting by averaging the results of many weak learners. Implementing this classifier raised the accuracy to 0.52, a noticeable improvement over the initial baseline. This performance boost can be attributed to the model's ability to evaluate numerous features simultaneously, capturing the diverse information present in both text-based and numeric fields. Additionally, Random Forest is robust to missing data and noisy features, both of which are common in real-world datasets. However, the limitations of single-model classifiers, even with an ensemble approach like Random Forest, hinted at the potential for further improvement.

Further Algorithm Exploration and Feature Engineering

To build on the progress made with Random Forest, I examined other classifiers, including Logistic Regression, which I selected for its interpretability and ability to serve as a baseline for linear relationships. Logistic Regression provided insights into feature impact, helping to assess which variables held the most predictive power in distinguishing review ratings. However, the model yielded only a minor increase in accuracy to 0.53. Logistic Regression's simplicity can sometimes be advantageous, but in this case, the model struggled to capture the nuanced relationships within the dataset, as review data is rarely linearly separable. This experiment underscored the importance of feature selection and engineering, as complex datasets require meaningful representation for models to achieve high performance. I conducted feature engineering based on both numerical and textual attributes, creating additional metrics like Helpfulness, Text_length, Summary_length, and word counts. These features were chosen based on assumptions about review structure and content, with the hypothesis that metrics such as review length and helpfulness ratings would correlate with rating intensity. Specifically, reviews that are longer or rated more helpful may reflect a more detailed user experience, which could indicate a stronger opinion, positive or negative.

Beyond feature creation, I also experimented with hyperparameter tuning using grid search to optimize model settings across various classifiers. However, due to the extensive computational resources required for grid search on such a large dataset, this approach proved impractical within the development environment. The frequent memory overloads and slow processing highlighted the constraints of working with substantial datasets, as well as the importance of balancing hyperparameter tuning with computational feasibility.

Applying XGBoost and Stacking for Enhanced Performance

Driven to further optimize model performance, I implemented XGBoost, a gradient boosting algorithm renowned for its efficiency with large, complex datasets. XGBoost's sequential structure, where each new tree attempts to correct errors from previous ones, allows it to capture intricate patterns within the data that were previously overlooked. XGBoost outperformed previous models by improving overall accuracy due to its adaptive nature and ability to weight misclassified instances, thereby enhancing model resilience. However, while XGBoost improved results, its predictive power on its own did not fully meet the performance goals.

This realization led to experimenting with ensemble methods through stacking, which involved combining the strengths of multiple classifiers. By implementing a StackingClassifier with Random Forest, XGBoost, and Logistic Regression as a meta-learner, I leveraged each model's unique strengths to boost performance further. Random Forest excelled at identifying broad patterns, XGBoost managed more subtle interactions, and Logistic Regression offered a final layer of interpretability and linear distinction. The stacking approach yielded the highest accuracy at 0.59, underscoring the potential of ensemble methods for handling multi-faceted datasets with interdependent features. The improvement is likely due to the models' combined ability to address both high variance and bias, creating a more adaptable predictive framework capable of generalizing across varied input patterns.

Insights from Model Variability and Random States

Throughout these experiments, I observed fluctuations in the accuracy scores, often in the second decimal place, as a result of using random_state=42. While this fixed seed provided a consistent basis for comparing model variations, the random splits in the training and validation sets introduced slight variability in performance. This fluctuation illustrates the sensitivity of the model to data distribution in training and testing sets, a critical factor in models involving high-dimensional text data. Recognizing the model's sensitivity to these variations allowed me to validate the consistency of improvements while acknowledging the impact of dataset sampling.

Conclusion

Ultimately, the combination of Random Forest, XGBoost, and Logistic Regression in a stacked ensemble proved the most effective approach for this dataset. Each model contributed complementary strengths, resulting in a robust predictive framework. This iterative process, from KNeighborsClassifier to stacked ensembles, highlights the importance of exploring diverse algorithms and combining them in ways that amplify their individual capabilities. Key techniques, such as feature engineering, TF-IDF vectorization, and dimensionality reduction, were crucial in refining the data for optimal analysis, while ensemble methods allowed for balancing predictive accuracy with model stability. The final model, combining ensemble approaches and sophisticated feature selection, demonstrates a deepened understanding of the data's structure and its predictive potential.