**IMDB Movie Reviews**

**Unit 4 Capstone**

****



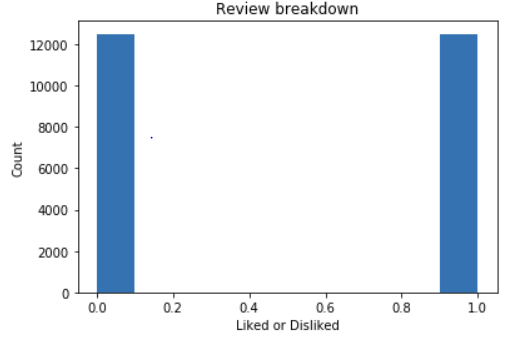
By: Elijah Fuller

*About the Dataset*

The data for my project comes from a Kaggle event sponsored by IMDB. The dataset can be found here <https://www.kaggle.com/kaushik3497/imdb-sentiment-analysis#labeledTrainData.tsv>. The dataset included two 25,000 word-based review datasets, I only used the train set due to the amount of time the models were taking to tune. The dataset’s purpose was not in order to find good movies, yet to determine whether the review was a positive or negative one.

*About the Project*

The purpose of this project was to determine was along the same idea as the originator and was to create accurate models that would decide whether a movie review was positive or not by using NLP. The reviews were a perfectly even split 50-50 split between positive and negative reviews, which made it easier to avoid having an unbiased model.



*The Process*

After cleaning the text, I used tfidfVectorizer to convert the dataset to a matrix of features. It was a simple feat to also remove the stop words using tfidfVectorizer. The stop words are extremely common words used in the English language that would not have been helpful data.

To tune the models, I used GridSearchCV to assist me in finding the best parameters. This process runs the models in a loop based on the parameters I provide and then notifies the best parameter found within the range of parameters that were given. Because of the amount of times a model is ran this process took a long time for tuning, especially for the more complex models such as SVC, random forest and gradient boosting.

*The Models*

**Logistic Regression Classification**

Logistic Regression used to describe data and to explain the relationship between one dependent binary variable and one or more independent variables. Logistic is both simple and an efficient model. Because of this model’s low complexity, it is normally a front-runner when in need of a model that is fast. While often the more complex models may often outperform logistic regression when it comes to accuracy, this is not always the case and was not the case for this project. My logistic regression model had the best accuracy and the second-best fit time. After tuning, I found the best inverse regulation strength (or “C”) to be 6.929999999999906.

**Random Forest**

Random Forest is a model that creates an ensemble of decision trees using the bagging method. While it is common for random forest to perform well with accuracy it was not the top performer in this project. This model is known for having slow performance speeds and this the case for my random forest model had the second slowest fit time of the models I used for this project. Another thing I always try to watch out for with random forest is that it is prone over-fitting, thankfully this was not a problem with my model and the test accuracy scores were very close to the training scores. The parameters that showed me the best results were entropy criterion, the max feature was square root, with a max depth of 28 and 183 n-estimators.

**Gradient Boosting**

Gradient boosting is similar to random forest as it also uses an ensemble of decision trees, however this model uses the boosting method rather than bagging. As advantage boosting as over bagging is that boosting is done serially and adjusts the model with corrections learned from the last model. This process tends to have a higher accuracy than the other models and, in this project, it had the second-best accuracy. The drawback is that this process is normally slow, even when comparing to random forest. My gradient boosting model ended up having second faster fit time than random forest, but this is not typical and the bagging method being able to run forests in parallel would normally have a faster fit time than gradient boosting’s running serially. The parameters that I found to have the best results was square root max features, a max depth of 17 and 188 n-estimators.

**KNN Classification**

K-nearest neighbors is simple algorithm that predicts an observation by comparing with the closest known observations in the training data. The comparison in my project use Euclidean distance, which is a common standard distance measurement using geometry. The parameters the gave the best results was 148 N-neighbors and the weights were based on distance rather than uniform. The most impressive observation that stood out to me with KNN was its fit time was extremely quick compared with all the other models.

**Support Vector Machine Classifier**

Uses a technique called the kernel trick to transform the data and utilizing the transformation it finds an optimal boundary between the possible outputs. Because of SVM’s complexity it can end up being a slow model, as was the case for this project SVM had the slowest fit time. SVM’s accuracy was the lowest for this project, but not by a large percentage. The best parameters I found used a polynomial kernel and 715 max iterations.

*Model Comparison*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Train set accuracy | Test set accuracy | Fit time | Overfit | False Positives | False Negatives |
| Logistic Regression | 89.42% | 87.6% | .5 seconds | 1.82% | 356 | 277 |
| Gradient Boosting | 86.28% | 84.56% | 17 seconds | 1.72% | 466 | 332 |
| Random Forest | 84.32% | 84% | 18 seconds | .32% | 591 | 349 |
| KNN | 85.14% | 81.78% | .06 seconds | 3.36% | 419 | 492 |
| SVC | 81.26% | 81.68% | 19.5 seconds | - | 668 | 615 |

*Conclusion*

After looking at all the results above it should come to no surprise my pick for best model in this project was logistic regression. Logistic regression had best accuracy and a fit time of half a second, it would be difficult for me to argue that any of the other models should be used over logistic regression with this dataset.

When I first learned of logistic regression in this course, I honestly did not think much of it as it was a very simple model. However, it is quickly becoming my favorite model because of its performance to accuracy is normally hard to beat in the projects I have done so far. I will make sure my supervised projects in the future will always include a logistic regression model, this model is just too reliable to overlook.