Machine Learning Report

Introduction

The objective of this project is to build a model that can accurately predict the rating on a scale of one to five of wine and other alcoholic drinks based on text reviews. The alcohol data is taken from Wine Reviews [1] on Kaggle. This dataset contains a list of over 2000 reviews of alcoholic drinks, containing information about brand, name, weight, review title, review date, review, recommendations, ratings, and if others found the reviews helpful or not. For the purpose of this experiment, only the text reviews were used as input to train the models, with the rating points as the outputted labels. A variety of neural network models as well as one type of logistic regression model were used to predict the drinks' ratings on a scale of one to five. The source dataset can be seen in Table 1 in the supplementary materials section of this paper.

Related Research

Research on existing related work was conducted. From the input data perspective, there are few additional projects that consider text reviews in the prediction of alcohol ratings. Most of the similar research used variety, chemical pH, and country of origin [2][3] as input features. Although these features can undoubtedly be used to predict alcohol ratings, they also assume that the quality of a wine is primarily, if not solely, attributed to its chemical consistency. This project argues that a far better input to consider is the past experiences of those who have already tried these drinks, and that it is reviewers who truly impact wine ratings.

Most previous studies have used similar models to the ones carried out in this project, namely, classification [4] and neural networks. Other projects treated this hypothesis as a linear regression problem [5] or implemented methods such as Random Forest [6] and Support Vector Machine. However, neural networks were rarely used. In this experiment, three types of models were used: a logistic regression model, a classification model, and a deep model. The ratings of one to five were treated as five categories and the text reviews were encoded using count vectorizers, TF-IDF vectorizers, and Keras tokenizer. This covered more options than previous experiments and explored the dataset in new ways.

Methodology

The source data consists of approximately 2,000 reviews which is admittedly a small dataset. The data use was further complicated by the uneven distribution of ratings, as can be seen in Figure 1. To make the dataset more readable, each column was renamed and columns that contained unnecessary information – such as the date of review and review number – were dropped. During pre-processing, the total amount of missing values was calculated, and the reviews which had no rating and recommendation given were dropped. The datatype was also changed; the "Rating" column from float to integer as only the numbers one to five were used, and the "Recommend" column from object to integer as only whole numbers used. To improve data quality, duplicated reviews were also removed. This resulted in approximately 1,750 data points. The updated, pre-processed dataset can be seen in Table 2 in the supplementary materials section of this paper.

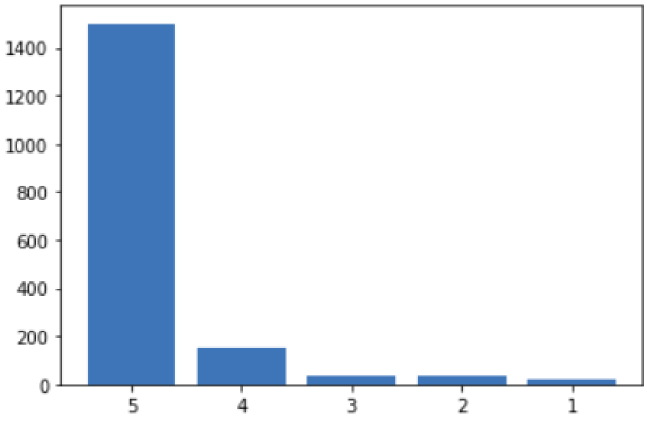


Figure 1 – Rating Distribution

After processing the data, the correlation between the numeric features was inspected. There was a weak positive correlation between both "Recommend" and "Helpful", and "Helpful" and "Rating", suggesting that reviews which were determined to be helpful, had higher alcohol ratings and recommendation ratings. There was also a strong positive correlation between "Recommend" and "Rating", which was expected. The higher the rating of the alcohol, the more likely it is that the reviewer recommends it to others. Before building the models, the text reviews were also pre-processed. The reviewers were transformed into lower-case only. After that, stopwords, which are words determined to have no real meaning such as "the", "and", "to", etc., were removed. This ensured that only the important words used in each review would be used by the machine learning models. The newly cleaned reviews were then stored in a new column, "Clean_Reviews".

The first model used was logistic regression. This is a linear, multiclass classifier model and is ideal for predicting one of five ratings, or classes, based on text reviews. A LBFGS solver was used, since it works best on smaller datasets when compared to other solvers, and in addition, it saves a lot of memory. The dataset was split into 66% train and 33% dev. Two experiments were then carried out to determine which vectorizer produced more accurate results. The Count Vectorizer focuses on the word frequency of reach review and resulted in an accuracy of approximately 84%. The TF-IDF Vectorizer also takes the importance of the words used in reviews into account and resulted in a slightly higher accuracy. Thus, this was the vectorizer used during model evaluation. In an attempt to increase accuracy, a pipeline was used, comprised of different hyperparameters of the logistic regression model. GridSearchCV was used to split the train data into different partitions in order to test for the best hyperparameters, as well as to perform cross-validation.

Next, a classification neural network model was used. The Keras tokenizer was used to change each word into an integer and store that integer in a dictionary. Given that there is approximately 50,000 words in the "Clean_Review" column, this dictionary was limited to the top 10,000 most common words. The reviews were re-vectorized into a NumPy array, with one review per row and one column per word. The ratings were reindexed from zero to four for readabilities sake, and then data was split into 30% train and 70% dev. The NumPy array was converted into PyTorch format before the multiclass model was built. CrossEntropyLoss was used to reduce the loss as much as possible in the classification model. The SGD optimizer performed a weight update for each train data example, and a learning rate of 0.01 was used for 150 epochs to train the model.

The final model built was a deep model comprised of stacks of dense layers as well as rectified linear units, in order to correct the overfitting seen in the previous classification model. Early stopping was also applied to interrupt the training as soon as the validation loss stopped improving. Again, the text reviews were tokenized and limited to the top 10,000 most common words, and the ratings were reindexed for readability. In this model, the data was split into 40% train and 60% dev to give the model more data to learn from. Instead of one linear layer, the deep model was comprised of multiple layers, executed one by one by nn.Sequential. In addition to early stopping, dropout was also applied. It masks 50% of the features used at each layer, to regularize the training.

Discussion & Results

The final logistic regression model returned an overall accuracy of 85%. It has a high F1-score of 78%, which is the weighted average of both precision and recall. The confusion matrix comparing the predicted labels versus the actual true labels can be seen in Figure 2. Although the pipeline did not increase the overall accuracy of this model, it did improve the accuracy of the individual ratings one and four, as seen in Figure 3. A learning curve was generated, showing the effect that adding more samples to the train data would have. With this dataset, the accuracy would decrease. To further evaluate the model, dummy classifiers were used. Returning the accuracy of a stratified model proved that the logistic regression model was more accurate than this baseline, however it proves to be less accurate than the most frequent strategy. This is likely due to the large amount of five ratings in the dataset, given that there is a very clear “most frequent” value.

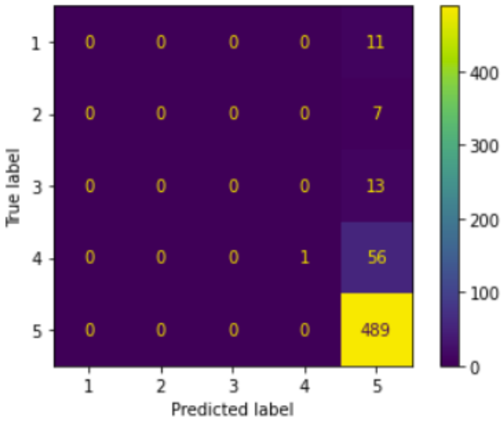


Figure 2 - Logistic Regression Confusion Matrix

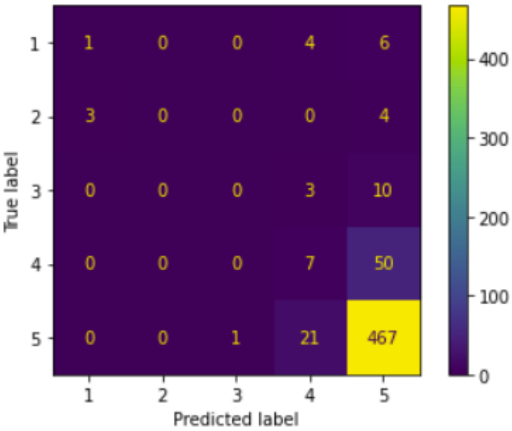


Figure 3 - GridSearchCV Confusion Matrix

Figure 4 shows the loss of the classification model. Although both the train loss and validation loss decrease with every epoch, the validation loss is slightly higher. This suggests that the model is slightly overfitting the training data. The val accuracy shown in Figure 5 does not change during training. The optimizer has found a local minimum for the loss, which is likely once again a result of the model primarily predicting a rating of five, given its common occurrence in the dataset. Two more metrics were defined to evaluate this model – mean square error (MSE) and mean absolute error (MAE). The MSE is the sum of the square of the differences between the predicted output and the real output. The MAE is the median of all the absolute differences between the predicted output and the real output. For this model, the MSE was 50%, meaning that it was making an error in its predictions for 50% of cases. The MAE was 25%, meaning that it was on average making an error in its predictions for 25% of the time. Although the MSE is quite high in this case, the MAE is low, meaning that on average, the model is correctly predicting ratings more often than not.

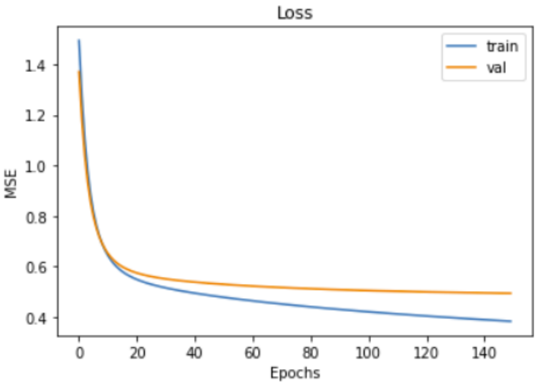


Figure 4 - Classification Model Loss

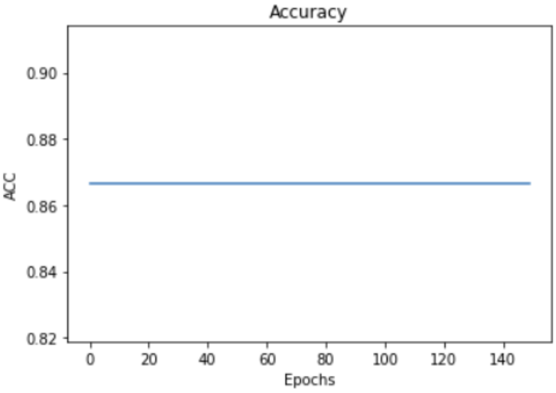
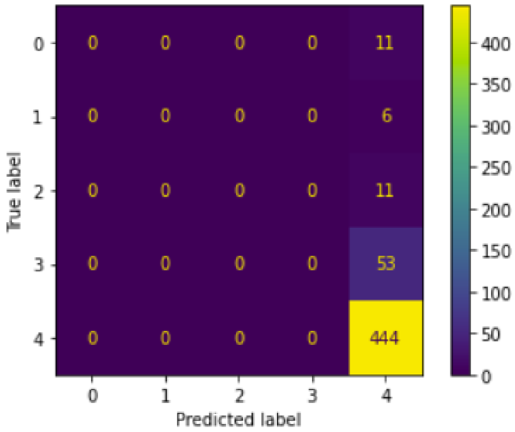


Figure 5 - Classification Model Accuracy

The accuracy of the model has slightly decreased from that of the logistic regression model, at 84.6%, and the F1-score has also marginally declined, now 77.5%. The MSE was 62% and the MAE was 26%, both numbers having worsened from the previous model. The overall accuracy remains the same, however, although the prediction accuracy for lower ratings has worsened, as can be seen in the confusion matrix in Figure 6. The model is still one per cent less accurate than the dummy classifier's most frequent strategy.



Fiaure 6 - Classification Confusion Matrix

Although the addition of ReLU layers and dropout regularization slowed down the training process of the deep model, the accuracy of the model, seen in Figure 7, has greatly improved. The data is no longer overfitted. Since regularization is applied only during training and not during validation, the train loss curve is now above the validation loss. The val accuracy, seen in Figure 8, has also improved, if only marginally. In the previous model, it was 87% but has now increased to 88%.

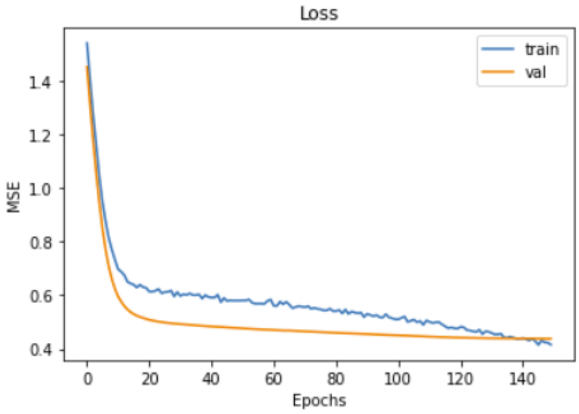


Figure 7 - Deep Model Loss

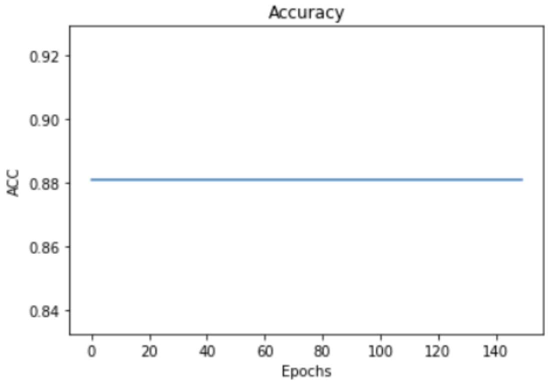


Figure 8 - Deep Model Accuracy

The accuracy of the model as well as the F1-score have also improved to 86% and 79% respectively, which the accuracy of each individual rating seen in Figure 9. The MSE decreased to 57% and the MAE decreased to 24%, both of which are good indicators that the model is making fewer mistakes and is returning more accurate predictions. The overall accuracy is also 86%, whereas the dummy classifier's most frequent model has decreased to 84%. This makes this deep model more accurate than the baseline model, as well as the most accurate model created during the course of this project.

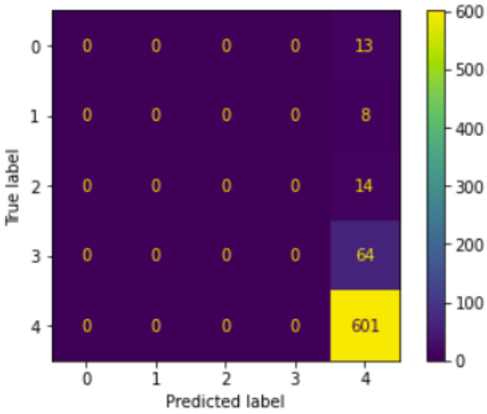


Figure 9 – Deep Model Confusion Matrix

Conclusion

Overall, as can be seen in Table 3, the deep learning model with its dense layers and regularization techniques proved to be the best model. It accurately predicted alcohol ratings based on text reviews 86% of the time, with a MSE average error of 24%. Although there were some complications with the dataset given it's uneven rating distributions, the deep model still achieved a high accuracy and performed better than the baseline models. This proves that text reviews are a valid indicator of the rating of alcoholic drinks, although in the future, this accuracy could be improved even further by taking other input features into account, such as the brand of alcohol, the recommendation categories, and the number of helpful votes given to each review.

Table 3 - Comparative Results

Evaluation Metric	Accuracy	Most Frequent	Stratified	F1-Score	MSE	MAE
Logistic Regression	0.82	0.86	0.76	0.79	0.50	0.25
Classification	0.85	0.86	0.74	0.78	0.62	0.26
Deep	0.86	0.84	0.71	0.79	0.57	0.24

References

1. Kaggle wine review dataset. https://www.kaggle.com/datasets/krrai77/wine-reviews
2. J. Ng. *Predicting wine ratings*. Jasmine Ng, 2020.
3. T. Shin. *Predicting wine quality with several classification techniques*. Medium, 2021.
4. M. Gu. *Sentiment analysis with wine reviews*. Medium, 2019.
5. Y. L. Chan, K. Gu, and S. Yang. *Predicting Wine Points using sentiment analysis*. University of California, 2019.
6. O. Goutay. *Wine ratings prediction using machine learning*. Medium, 2021.

Supplementary material

Table 1 - Source Dataset Example

Sl.No.	Brand	Name	Reviews Date Added	Reviews do Recommend
1	Gallo	Ecco Domani174 Pinot Grigio - 750ml Bottle	2018-01-09T13:24:04Z	TRUE
2	Fresh Craft Co.	Fresh Craft174 Mango Citrus - 4pk / 250ml Bottle	2018-01-09T17:31:52Z	TRUE
3	1000 Stories	1000 Stories174 Zinfandel - 750ml Bottle	2018-01-09T17:31:51Z	TRUE

Sl.No.	Reviews Num Helpful	Reviews Rating	Weight	Reviews Title
1	1	5	1.0 lbs	My Favorite White Wine
2		5	2.45 lbs	Yum!!
3		5	3.09 lbs	A New Favorite!

Sl.No.	Reviews Text
1	This a fantastic white wine for any occasion!
2	Tart, not sweet...very refreshing and delicious!
3	I was given this wine so it was a delightful surprise to find that it has a flavorful and delicious taste! A new favorite!!!!

Table 2 – Pre-Processed Dataset

Brand	Name	Recom-mend	Helpful	Rating	Weight	Review Title	Review
Gallo	Ecco Domani174 Pinot Grigio - 750ml Bottle	True	1.0	5.0	1.0 lbs	My Favorite White Wine	fantastic white wine any occasion
Fresh Craft Co.	Fresh Craft174 Mango Citrus - 4pk / 250ml Bottle	True	NaN	5.0	2.45 lbs	Yum!!	tart sweet refreshing delicious
1000 Stories	1000 Stories174 Zinfandel - 750ml Bottle	True	NaN	5.0	3.09 lbs	A New Favorite!	given wine delightful surprise find flavourful delicious taste new favourite