Text Analysis and NLP Final Report

By: Jeremy Glasgow

Introduction

The goal of this project was to apply various machine learning techniques, including text analysis and natural language processing, to accurately predict car review engagements. The dataset used for this project consisted of just over 110,000 car reviews for cars manufactured from 2007 to 2017. Each review had informational data such as the make, model, average MSRP, and year of the car the review was about. Each review also contained ratings for various vehicle attributes such as performance, comfort, fuel economy, how fun it is, interior, exterior, build quality, and reliability.

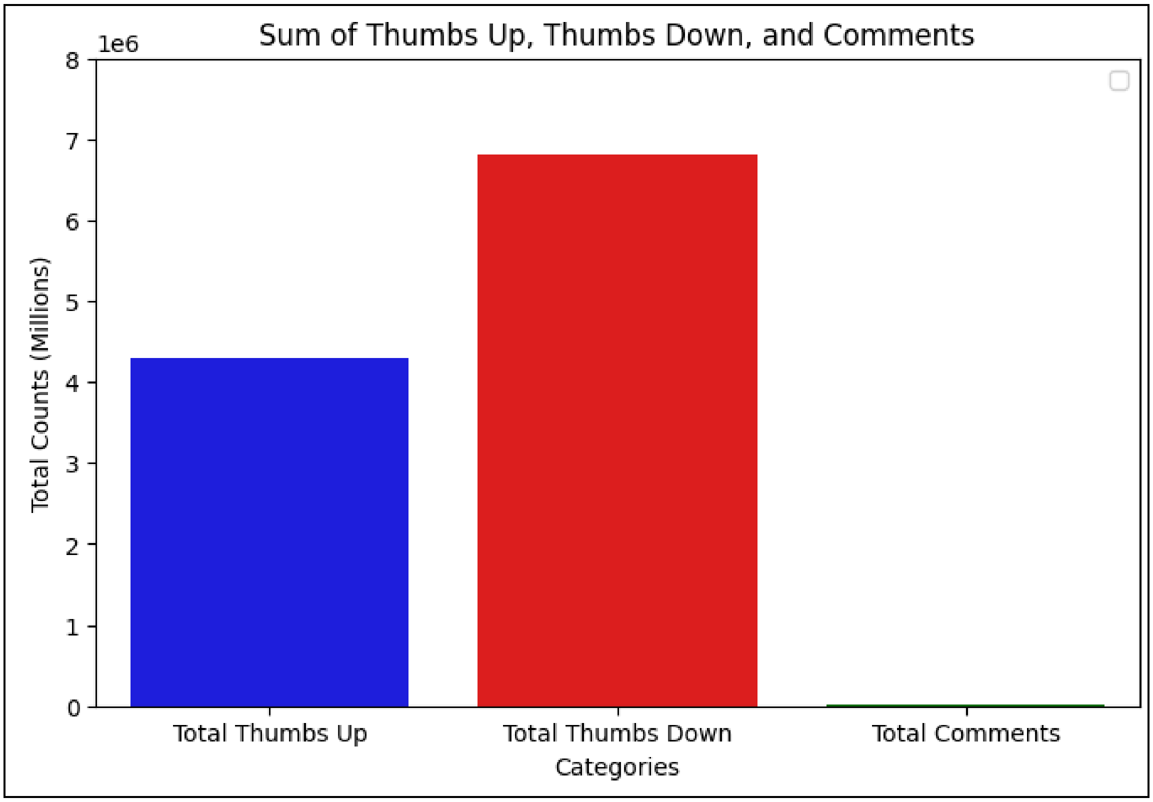


Figure 1 - Sun of Thumbs Up, Thumbs Down, and Comments.

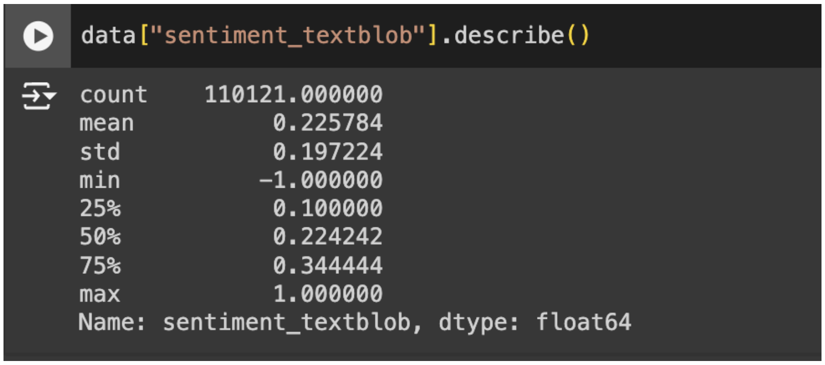
There were also engagement variables which represented the rate other users engaged with the review. The engagement variables were number of thumbs up, number of thumbs down, and the number of comments on the review. In Figure 1 pictured above, the number of comments is extremely small compared to the number of thumbs up or thumbs down. The goal of the project was to predict car review engagement which requires the creation of a special variable to measure a review's total engagement. I attempted to use a variety of different combinations of the engagement variables such as total engagement, percentage of thumbs up, thumbs up, and total thumbs up and thumbs down.

Data Preprocessing

I started data preprocessing by viewing the number of missing values in each column. The only two columns with significant missing values were asking the reviewer for their favorite feature and for any suggested improvements. It is possible that these two fields were optional due to the significant number of missing responses and the open-endedness of the question. Next, I tokenized the actual review text by removing punctuation, removing special characters, and removing any numerical values. I also performed stemming on the review text which reduces words down to their stem word.

Text Analysis

I applied three text analysis techniques in this report. The first technique used sentiment analysis using TextBlob to assign a polarity score to each car review. Figure 2 below shows the polarity score statistics after performing the polarity scoring.

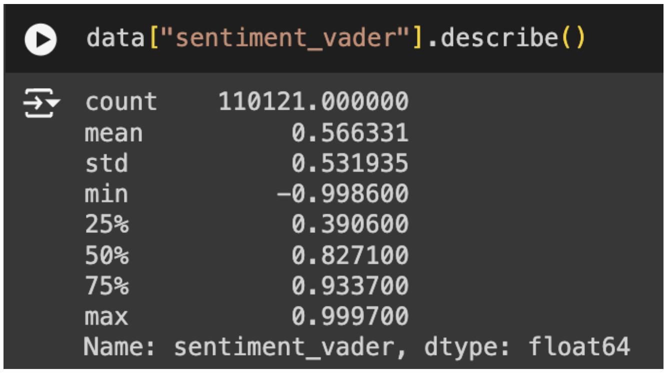


```
data["sentiment_textblob"].describe()

count    110121.000000
mean          0.225784
std           0.197224
min          -1.000000
25%           0.100000
50%           0.224242
75%           0.344444
max           1.000000
Name: sentiment_textblob, dtype: float64
```

Figure 2 - TextBlob polarity score metrics.

The second technique also used sentiment analysis using Valence Aware Dictionary and Sentiment Reasoner (VADER) which is tuned for social media text analysis. Given that the car reviews were more than likely posted on the internet, this sentiment analysis technique would be a good fit. Figure 3 below shows the polarity score statistics after performing the polarity scoring.

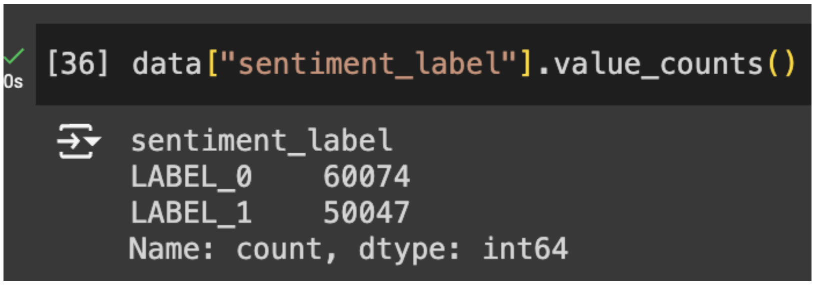


```
data["sentiment_vader"].describe()

count    110121.000000
mean          0.566331
std           0.531935
min          -0.998600
25%           0.390600
50%           0.827100
75%           0.933700
max           0.999700
Name: sentiment_vader, dtype: float64
```

Figure 3 - VADER polarity score metrics.

The third and final technique used sentiment analysis through a pre-trained transformer known as distilbert, which is provided by HuggingFace. This transformer analyzes the text and returns a positive or negative sentiment rating, as well as a confidence percentage of the sentiment rating. The execution time to run the transformer in Google Colab was quite long. I refactored the code to execute batch jobs which decreased the execution time significantly. Figure 4 below shows the output of the transformer after performing the sentiment analysis.



```
[36] data["sentiment_label"].value_counts()
```

```
sentiment_label
LABEL_0    60074
LABEL_1    50047
Name: count, dtype: int64
```

Figure 4 - Pre-trained transformer polarity score metrics.

Feature Selection

I began feature selection by performing some basic Exploratory Data Analysis (EDA) on different variables. The only features that I was able to find insightful were the rating features, the year of the car, and the make of the car. I collected linear coefficients and an R-squared score using these features to predict the target variable. The target variable chosen for this iteration was the total engagement, or sum of thumbs up, thumbs down, and comments, found on each review. The R-squared value was 0.0001. Next, I tried scaling the target variable, but the R-squared value was similar to the last iteration. I continued to experiment with different combinations of predictor variables, but there was no significant improvement in R-squared values. I also applied OneHotEncoding, which converts categorical variables into a data format compatible with regression models. The car make coefficients were somewhat insightful, so I saved them into an encoded dataframe.

Feature Engineering

One method of feature engineering I applied throughout the project was Lasso. Lasso is a regularization method that can also be applied as a feature engineering tool. Lasso stands for Least Absolute Shrinkage and Selection Operator. Lasso works by shrinking the coefficients used in the linear regression model to zero, which assigns a weight of zero, to the feature. The alpha value in Lasso is used to determine which features are affected. If the alpha value is greater than the absolute value of the coefficient, the feature will be removed from the prediction model.

Model Building and Model Evaluation

Throughout the model testing, I tried a series of different techniques to improve the test prediction. The neural network architecture consisted of Dense layers with a ReLU activation function. I used Adam as the optimizer, root squared error as the loss function, and mean absolute error as the performance metric. For each iteration, I collected the mean absolute error of the training set. After model training, a prediction was made using the test set. I compared the mean absolute error values of the training set and the testing set to judge the accuracy of the model.

The best performing model used total engagement as the target variable and the rating features as the prediction variables. The neural network model consisted of an input layer equal to the number of predictors, a hidden layer with 64 units, a hidden layer with 32 units, and an output layer with 1 unit. The training set measured a mean absolute error of 217. The testing set measured a mean absolute error of 217. This resulted in a decrease in performance of 0.64%.

I also tried scaling the target variable for total engagement, but there was no noticeable increase in model performance.

Figure 5 below shows a complete list of iterations and the outcomes.

Target Variable	**Predictor Variables**	**Neural Network Architecture**	**Other Info**	**Baseline MAE**	**Test MAE**	**Delta**
Total engagement	Sentiment Features	Dense 64 unit Dense 32 unit Dense 1 unit		216	219	(1.37%)
Total engagement	Rating Features	Dense 64 unit Dense 32 unit Dense 1 unit		216	217	(0.46%)
Total engagement scaled	Sentiment Features	Dense 64 unit Dense 32 unit Dense 1 unit		0.0508	0.0527	(3.61%)
Total engagement scaled	Rating Features and Sentiment Vader	Dense 16 unit Dropout (20%) Dense 16 unit Dense 1 unit		0.0508	0.0684	(25.73%)
Total engagement scaled	Rating Features, Sentiment Features, and Year	Dense 32 unit Dense 32 unit Dense 8 unit Dense 1 unit		0.0508	0.0690	(26.38%)
Total	Rating	Dense 16 unit		0.0508	0.0661	(23.15%)

engagement scaled	Features, Sentiment Features, and Year	Dense 8 unit Dense 1 unit				
Total engagement scaled	Rating Features, Sentiment Features, and Year	Dense 64 unit Dense 32 unit Dense 16 unit Dense 8 unit Dense 1 unit		0.0508	0.0796	(36.18%)
Total engagement scaled	Rating Features, Sentiment Vader, and Sentiment TextBlob	Dense 64 unit Dense 32 unit Dense 1 unit	Applied Lasso with alpha value of 0.002	0.0508	0.0550	(7.64%)
Total engagement scaled	Rating Features, Sentiment Vader, and Sentiment TextBlob	Dense 64 unit Dense 32 unit Dense 1 unit	Applied Lasso with alpha value of 0.001	0.0508	0.0550	(7.64%)
Total engagement scaled	Rating Features, Sentiment Features, and Year	Dense 64 unit Dense 32 unit Dense 1 unit	Applied Lasso with alpha value of 0.001	0.0508	0.0680	(25.29%)
Thumbs up percentage	Rating Features, Sentiment Features, and Year	Dense 16 unit Dense 8 unit Dense 1 unit		0.1374	0.1438	(4.45%)
Thumbs up percentage	Rating Features, Sentiment Features, and Year	Dense 8 unit Dense 4 unit Dense 1 unit		0.1374	0.1443	(4.78%)

Thumbs up percentage	Rating Features, Sentiment Features, and Year	Dense 16 unit Dense 16 unit Dense 1 unit	Applied Lasso with alpha value of 0.001	0.1374	0.1496	(8.16%)
Thumbs up percentage	Rating Features, Sentiment Features, and Year	Dense 16 unit Dense 16 unit Dense 1 unit	Applied Lasso with alpha value of 0.01	0.1374	0.1475	(6.85%)
Thumbs up percentage	Rating Features, Sentiment Features, Year, and Car Make	Dense 16 unit Dense 16 unit Dense 1 unit		0.1374	0.1484	(7.41%)
Thumbs up percentage	Rating Features, Sentiment Features, Year, and Car Make	Dense 16 unit Dense 16 unit Dense 1 unit	Applied Lasso with alpha value of 0.005	0.1374	0.1490	(7.79%)
Total engagement	Rating Features, Sentiment Features, Year, and Car Make	Dense 16 unit Dense 16 unit Dense 1 unit	Applied Lasso with alpha value of 10	216	265	(18.49%)
Total engagement scaled	Rating Features, Sentiment Features, Year, and Car Make	Dense 16 unit Dense 16 unit Dense 1 unit	Applied Lasso with alpha value of 0.005	0.0508	0.736	(30.98%)

Total engagement scaled	Rating Features, Sentiment Features, Year, and Car Make	Dense 16 unit Dense 16 unit Dense 1 unit	Applied Lasso with alpha value of 0.005, reduced batch size and increased epochs	0.0508	0.0569	(10.72%)

Figure 5 - Table of model performance outcomes.

Conclusion

Due to the difficulties I had deriving a target variable from the engagement variables, I was not able to develop a model that outperformed the baseline dataset. The predictor variables yielded extremely low R-squared values, indicating poor model performance. I applied various text analysis techniques such as pre-trained transformers, general sentiment analysis, and sentiment analysis tuned for social media. These text analysis techniques did not seem to improve model performance.