Product Brand Sentiment: A Multivariate Classification Prediction Analysis

Joseph Cruz
josephbcruz@lewisu.edu
DATA-51000-001, Summer 1 (2021)
Data Mining and Analytics
Lewis University

I. INTRODUCTION

Social media has gained significant popularity over the last decade and continues to be one of consumers most utilized websites/applications. On average, people spend about two hours and twenty-four minutes a day on social media, which means there are likely many things they can see pop up on their feed [1]. Consequentially, developing marketing strategies that take advantage of that time spent are ideal for any marketing team. There are many different marketing strategies out there, but one of the most successful are personalized ads. By understanding the wants of the consumer, marketing algorithms can be designed to accommodate for those desires and market more efficiently. For instance, suppose a consumer is investigating new brands of dog food. By understanding what they are looking for, the personalized ad algorithm can supply them with ads for different dog food brands rather than just providing them with food for a turtle. The utilization of personalized ad algorithms can make marketing much more efficient and, potentially, more effective.

Personalized ad algorithms that can account for consumer desires can do so by using predictive analysis. More specifically, the algorithm can learn from consumer data how to delineate between positive, neutral, and negative emotion associated with a specific brand or product. This can be accomplished by using a combination of classification models and supervised machine learning techniques. By training a classification model on consumer generated data, it is possible for the model to predict the sentiment or emotion associated with the data. These classification models could be used upon things like reviews, consumer surveys, or even tweets talking about a product. This learning association can lead to more effective personalized ad algorithms that may appeal to consumer views and opinions based upon data from the actual consumers themselves.

In this paper, three different classification models are applied and examined in the determination of consumer sentiment regarding different product brands. The three classification models used are as follows: Adaptive Boosting (AdaBoost), k-nearest neighbors (kNN), and random forest. These classification models will be applied to a dataset containing unique tweets and their respective emotion (either positive, neutral, negative, or unknown) towards multiple brands and products, if any [2]. Throughout the course of this report, a description of the dataset will be explained as well as an overview of the methodology utilized for the predictive analysis. Further, the results and discussion section will be reviewed and the conclusion from the data will soon follow. In Section II, the dataset will be fully described. In Section III, the methodology used for the predictive analysis will be explained. In Section IV, the results of the analysis will be reported and a discussion of the data will follow to clarify and explore the results. Finally, in Section V, the conclusions of the findings will be summarized and presented.

II. DATA DESCRIPTION

The data that will be used for the development of the predictive analysis model is a dataset that is composed of unique tweets regarding multiple brands and products as well as the emotion expressed in the tweet and the brand the emotion was targeted towards [2]. There are 9093 rows, or instances, in the dataset along with a total of three features. The features of the dataset can be seen in Table I. The dataset contains the feature "tweet_text", which contains the contents of the tweets gathered in their text form

TABLE I. ATTRIBUTES OF BRAND AND PRODUCT EMOTION DATASET

Attribute	Type	Example Value	Description
tweet_text	Nominal (string)	"@jessedee Know about @fludapp ? Awesome iPad/iPhone app…"	Text from tweets.
emotion_in_tweet_is_directed_at	Nominal (string)	"iPad or iPhone App"	Product brand that emotion in the tweet is directed at.
is_there_an_emotion_directed_at_a_brand_or_product	Numeric (string)	"Positive emotion"	Label of emotion detected in tweet (true label).

TABLE II. SAMPLE OF TWEET TEXT DATA AND ITS ASSOCIATED EMOTION LABEL

Tweet Text Data	Emotion Label
.@wesley83 I have a 3G iPhone. After 3 hrs tweeting at #RISE_Austin, it was dead! I need to upgrade. Plugin stations at #SXSW.	Negative emotion
@jessedee Know about @fludapp ? Awesome iPad/iPhone app that you'll likely appreciate for its design. Also, they're giving free Ts at #SXSW	Positive emotion

Fig. 1. Word cloud developed from the “tweet text” data.

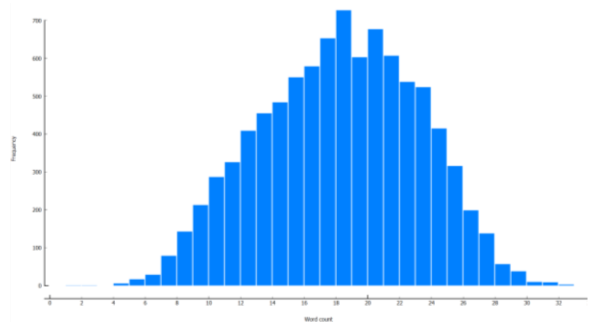


Fig. 2. Word count per tweet distribution.

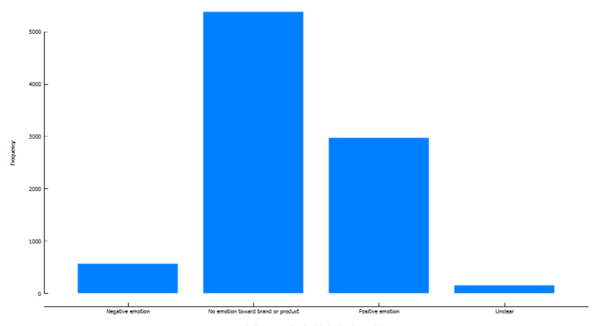


Fig. 3. Frequency distribution of “is_there_an_emotion_directed_at_a _brand_or_product” attribute

(including special symbols). The word cloud generated from the text data can be shown in Figure 1. Also, the word count for each tweet in the “tweet_text” can be shown in Figure 2. The next feature is “emotion_in_tweet_is_directed_at” which contains the brand that the emotion in the tweet was targeted towards. For instance, if the tweet were praising Samsung or Apple, these two companies would be recorded here. The final feature is “is_there_an_emotion_directed_at_a_brand_or_product”, which contains the emotion label for the tweet (“Negative emotion”, “No emotion toward brand or product”, “Positive emotion”, or “Unclear”). An example of some of the “tweet_text” data with its associated “is_there_an_emotion_directed_at_a_brand_or_product” emotional label can be shown in Table II. Of the dataset, the features “tweet_text” and “is_there_an_emotion_directed_at_a_brand_or_product” were the main features focused upon. The “tweet_text” was designated as a main feature because this data will be used to associate words to an emotion label and the emotion label feature will supply the learning models with the actual label to perform the learning associations. The frequency distribution of the emotion expressed in tweets can be shown in Figure 3.

III. METHODOLOGY

To perform the predictive analysis, the data mining toolkit Orange (version 3.26) was used. The flowchart shown in Figure 4 provides a summary of the steps utilized to perform the predictive analysis. First, the data was obtained and preprocessed first in Excel. The “I can’t tell” data in the “is_there_an_emotion_directed_at_a_brand_or_product” attribute was replaced with “Unclear”. Then, the data was split into metadata, target data, and feature data. The “tweet_text” feature was designated as metadata, the “emotion_in_tweet_is_directed_at” feature was designated as a feature, and the “is_there_an_emotion_directed_at_a_brand_or_product” was designated as the target. Then, the data was preprocessed in Orange. The data was then fed into the corpus widget to generate a corpus of the “tweet_text” feature. This corpus was then fed into the preprocess text widget, where the data was transformed into lowercase with accents removed, html parsed, and URLs removed. Further the corpus was tokenized using Regexp (pattern: \w+), and filtered out stop words from the English language and special symbols(\.|,|:|;|!|\?|\(|\)|\||\+|\'|\"|‘|’|“|”|\'|\’|…|\-|–|—|\$|&|*|>|<|\/|\[|\]). The new corpus was then fed into a bag of words widget to create a new corpus with a bag of words feature. The new corpus was created using a count term frequency, no document frequency, and a Euclidean (L2) regularization. At this point, the data was ready to be used for the classification models.

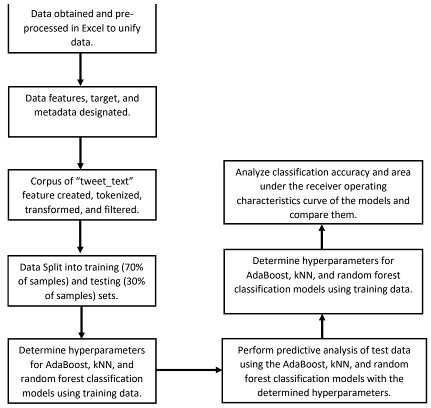


Fig. 4. Flow chart of predictive classification analysis.

The data was split into the training and testing sets. The training set contained 70% of the data and the test set contained 30%. The training set was utilized for tuning the hyperparameters of the models to develop models that performed the best. The training data was then fed into the three classification model widgets (AdaBoost, kNN, and random forest) as well as an additional constant classification widget. The constant classification serves only to provide a baseline as to predicting the most frequent class value from the training set and is only meant to help in understanding the results of the main three classification models. The training data and the resulting learners were then fed into the test and score widget. Here, the models were tested with the training data using a cross validation sampling technique. The results of these tests were then analyzed for area under the receiver operating characteristics (ROC) curve (AUC) and the classification accuracy (CA) with respect to the average over the classes. These previous steps were repeated until appropriate AUC and CA scores were obtained for each model.

After the hyperparameters of each classification model were established, the testing data was fed into the test and score widget along with the AdaBoost, kNN, and random forest learners with their respective hyperparameters in place. Within the test and score widget, the "train on test data" option was selected and the models were tested with the test data. The results were analyzed for AUC, CA, F1-scores, precision, and recall, with high focus on the AUC and CA scores, and the models were compared to one another. Furthermore, the results were fed into the confusion matrix widget and the resulting confusion matrix was analyzed for each classification type (i.e., positive, neutral, negative, unknown) for each model.

IV. Results and Discussion

A. Results

The results of the hyperparameter tuning for each classification model can be seen in Tables III-V. In Table III, the kNN model provided a maximum CA score of 0.88 with an AOC of 0.906, which was shown in trial 5. The hyperparameters that were able to provide the maximum CA score for the kNN model were 15 neighbors, using a Euclidean metric, and a distance for the weight. In Table IV, the random forest model provided a maximum CA score of 0.774 with an AOC of 0.872, seen in trial 6. The hyperparameters that were able to provide the maximum CA score for the random forest model were as follows: 25 trees, four attributes considered at each split, replicable training used, no limit of depth for individual trees, and a limitation to not split subsets that are smaller than 10. In Table V, the AdaBoost model provided a maximum CA score of 0.871 with an

TABLE III. Trial Parameters for the kNN Model Utilized During Hyperparameter Tuning. The Hyperparameters Used for the Test Data Are Highlighted in Yellow.

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Trial 7
Number of neighbors	5	10	10	15	15	20	20
Metric	Euclidean	Euclidean	Manhattan	Euclidean	Euclidean	Euclidean	Euclidean
Weight	Uniform	Uniform	Uniform	Uniform	Distance	Uniform	Distance
Area under the ROC curve	0.893	0.898	0.897	0.904	0.906	0.904	0.907
Classification Accuracy	0.875	0.88	0.86	0.879	0.88	0.879	0.88

TABLE IV. Trial Parameters for the Random Forest Model Utilized During Hyperparameter Tuning. The Hyperparameters Used for the Test Data Are Highlighted in Yellow.

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Trial 7
Number of trees	5	10	15	15	15	25	25
Number of attributes considered at each split	4	4	4	4	4	4	4
Replicable Training?	Y	Y	Y	Y	Y	Y	Y
Limit depth of individual trees	N	N	N	N	N	N	N
Do not split subsets smaller than	5	5	5	6	10	10	15
Area under the ROC curve	0.794	0.834	0.852	0.853	0.857	0.872	0.871
Classification Accuracy	0.72	0.752	0.758	0.76	0.763	0.774	0.765

TABLE V. TRIAL PARAMETERS FOR THE ADABOOST MODEL UTILIZED DURING HYPERPARAMETER TUNING. THE HYPERPARAMETERS USED FOR THE TEST DATA ARE HIGHLIGHTED IN YELLOW.

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Trial 7
Number of estimators	70	90	90	110	110	110	150
Learning rate	1	1	1	1	1	0.8	1
Classification algorithm	SAMME.R	SAMME.R	SAMME	SAMME.R	SAMME.R	SAMME.R	SAMME.R
Regression loss function	Exponential	Exponential	Exponential	Exponential	Square	Exponential	Exponential
Area under the ROC curve	0.905	0.906	0.896	0.906	0.906	0.905	0.907
Classification Accuracy	0.869	0.871	0.864	0.87	0.87	0.869	0.87

AOC of 0.90, shown in trial 2. The hyperparameters that could provide the maximum CA score were as follows: 90 estimators, a learning rate of 1, the Stagewise Additive Modeling Real (SAMME.R) classification algorithm, and the exponential regression loss function. These hyperparameters that could provide the maximum CA score, with respect to the average over the classes, were selected for the final testing hyperparameters.

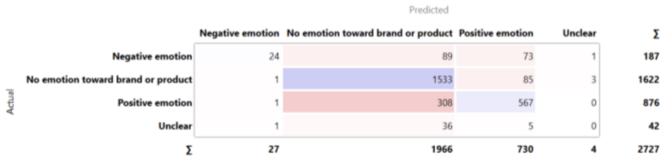
The results of the models on the testing data can be shown in Table VI and the resulting confusion matrices are in Figures 5-7. The constant classification widget produced a CA of 0.595 with an AUC of 0.5, F_1-score of 0.444, precision of 0.354, and recall of 0.595. The random forest model produced a CA score of 0.779 with an AUC of 0.880, F_1-score of 0.75, precision of 0.774, and recall of 0.779. The kNN model produced a CA score of 0.882, an AUC score of 0.908, an F_1-score of 0.856, precision of 0.862, and recall of 0.882. The AdaBoost model produced a CA score of 0.870 with an AUC of 0.909, F_1-score of 0.847, precision of 0.847, and recall of 0.870. From the confusion matrices, of the 2727 instances in the testing data, there were a total of 187 instances belonging to the "Negative emotion" class, 1622 instances belonging to the "No emotion toward brand or product", 876 instances belonging to the "Positive emotion" class, and 42 instances belonging to the "Unclear" class.

The confusion matrix for the random forest model (Figure 5) indicates that of the 2727 instances, there were a total of 27 instances predicted to be of the "Negative emotion" class, 1966 instances predicted to be part of the "No emotion toward brand or product" class, 730 instances predicted to be in the "Positive emotion" class and 4 instances to be part of the "Unclear" class. The model predicted 24 instances of the "Negative emotion" class properly, but misclassified the other 163 instances of the "Negative emotion" class. Of the 163 instances improperly classified, 89 instances were classified as part of the "No emotion toward brand or product" class, 73 instances were classified as the "Positive emotion" class, and 1 instance was classified as the "Unclear" class. The model predicted 1533 instances correctly of the "No emotion toward brand or product" class and misclassified 89 of the other instances of the "No emotion toward brand or product" class. Of the 89 instances improperly predicted, 1 instance was predicted as the "Negative emotion" class, 85 instances were classified as the "Positive emotion" class, and 3 instances were classified as the "Unclear" class. The model predicted 567 instances of the "Positive emotion" class properly and misclassified the other 309 instances of the "Positive emotion" class. Of the 309 incorrectly classified instances, 1 instance was classified as the "Negative emotion" class, 308 instances were classified as the "No emotion toward brand or product" class, and 0 were classified as the "Unclear" class. Finally, the model predicted 0 instances of the "Unclear" class correctly and misclassified all 42 instances. Of the 42 instances misclassified, 1 instance was classified as the "Negative emotion" class, 36 instances were classified as the "No emotion toward brand or product" class, and 5 instances were classified as the "Positive emotion" class.

The confusion matrix for the kNN model (Figure 6) indicates that of the 2727 instances, there were a total of 36 instances predicted as the "Negative emotion" class, 1741 instances predicted as the "No emotion toward brand or product", 950 instances predicted as the "Positive emotion" class, and 0 instances predicted as the "Unclear" class. The kNN model classified 28 instances of the "Negative emotion" emotion class correctly but misclassified the other 159 instances of the "Negative emotion" class. Of the 159 misclassified instances, 21 instances were classified as the "No emotion toward brand or product" class, 138 instances were classified as the "Positive emotion" class, and 0 instances were classified as the "Unclear" class. The model classified 1593 instances of the "No emotion toward brand or product" class properly and misclassified only 29 instances of the "No emotion toward brand or product" class. Of the 29 misclassified instances, 1 of them was predicted to be part of the "Negative emotion" class, 28 instances were classified as the "Positive emotion" class, and 0 instances were classified as the "Unclear" class. The model classified 783 instances of the "Positive emotion" class properly and misclassified the other 93 instances of the "Positive emotion class". Of the 93 misclassified instances, 7 instances were classified as the "Negative emotion" class, 86 instances were classified as the "No emotion toward brand or product" class, and 0 were classified as the "Unclear" class. The model classified 0 instances of the "Unclear" class correctly and misclassified all 42 instances. Of the 42 misclassified instances, 0 instances were classified as the "Negative emotion" class, 41 instances were classified as the "No emotion toward brand or product" class, and 1 instance was classified as the "Positive emotion" class.

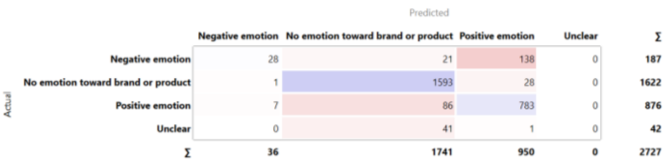
TABLE VI. RESULTS OF FINAL MODEL TESTING WITH THE TESTING DATA (AVERAGE OVER CLASSES)

Model	AUC	CA	F1	Precision	Recall
Constant	0.500	0.595	0.444	0.354	0.595
Random Forest	0.880	0.779	0.750	0.774	0.779
kNN	0.908	0.882	0.856	0.862	0.882
AdaBoost	0.909	0.870	0.847	0.847	0.870



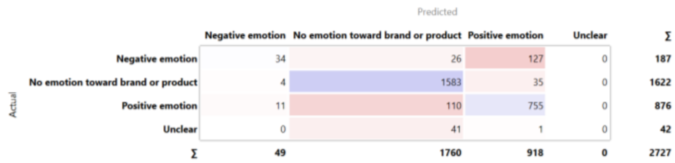
Actual \ Predicted	Negative emotion	No emotion toward brand or product	Positive emotion	Unclear	Σ
Negative emotion	24	89	73	1	187
No emotion toward brand or product	1	1533	85	3	1622
Positive emotion	1	308	567	0	876
Unclear	1	36	5	0	42
Σ	27	1966	730	4	2727

Fig. 5. Confusion matrix for random forest model.



Actual \ Predicted	Negative emotion	No emotion toward brand or product	Positive emotion	Unclear	Σ
Negative emotion	28	21	138	0	187
No emotion toward brand or product	1	1593	28	0	1622
Positive emotion	7	86	783	0	876
Unclear	0	41	1	0	42
Σ	36	1741	950	0	2727

Fig. 6. Confusion matrix for kNN



Actual \ Predicted	Negative emotion	No emotion toward brand or product	Positive emotion	Unclear	Σ
Negative emotion	34	26	127	0	187
No emotion toward brand or product	4	1583	35	0	1622
Positive emotion	11	110	755	0	876
Unclear	0	41	1	0	42
Σ	49	1760	918	0	2727

Fig. 7. Confusion matrix for AdaBoost

The confusion matrix for the Adaboost model (Figure 7) indicates that of the 2727 instances, there were a total of 49 instances predicted as the “Negative emotion” class, 1760 instances predicted as the “No emotion toward brand or product” class, 918 instances as the “Positive emotion” class, and 0 instances as the “Unclear” class. The AdaBoost model classified 34 instances of the “Negative emotion” class properly and misclassified the other 153 instances of the “Negative emotion” class. Of the 153 misclassified instances, 26 instances were classified as “No emotion toward brand or product”, 127 instances were classified as the “Positive emotion” class, and 0 instances were classified as the “Unclear” class. The model classified 1583 instances of the “No emotion toward brand or product” class properly but misclassified 39 instances of the “No emotion toward brand or product” class. Of the 39 misclassified instances, 4 instances were classified as the “Negative emotion” class, 35 instances were classified as the “Positive emotion” class, and 0 instances were classified as the “Unclear” class. The model classified 755 instances of the “Positive emotion” class properly and misclassified the other 121 instances of the “Positive emotion” class. Of the 121 instances misclassified, 11 instances were classified as the “Negative emotion” class, 110 instances were classified as the “No emotion toward brand or product” class, and 0 instances were classified as the “Unclear” class. Finally, the model misclassified all instances of the “Unclear” class. Of the misclassified instances, 0 instances were classified as the “Negative emotion” class, 41 instances were classified as the “No emotion toward brand or product” class, and 1 instance was classified as the “Positive emotion” class.

B. *Discussion*

The first result that needs to be discussed is the constant model to provide a baseline for the data that is being used. The constant model is used to predict the most frequent class from the training set for each instance. Per the data, the CA score was 0.595, which means that almost 60% of the sample data utilized is of the same class, which it is (1622/2727 = 0.595 samples of the “No emotion toward brand or product” class). Although this is not ideal, there is an indication that there is still variety in the sample data. Furthermore, the precision also indicates that there is variety, as selecting the most frequent class for all predictions yields a precision of 0.354 for the average of all the classes. If there was not a variety in the sample set, then the CA score and the precision scores would be much higher. Thus, the data is suitable to perform a predictive classification analysis upon.

Of the 3 classification models used, the order for them, from best CA to worst, is the kNN model (0.882 CA), the AdaBoost model (0.870 CA), and the random forest model (0.779 CA). All models retained a CA score above 0.7, indicating relatively strong classification accuracy, based on average of all the classes, for all models. Furthermore, the AUC scores for all three models were above 0.85 based on average of all the classes. The AUC scores can demonstrate the model’s ability to distinguish between the classes, where the higher the score, the more likely that the model can successfully distinguish between them. Since all models retained scores above 0.85, this means that all models were able to distinguish between the classes relatively well. Furthermore, the precision and recall for each model are approximately the same for that individual model. In other words, the precision and recall for the kNN model (0.862 and 0.882, respectively) are approximately the same and the same goes for the random forest and AdaBoost model.

Regarding the predictions made by these models, it seemed that the random forest model predicted more instances of the “No emotion toward brand or product” class than any of the other models with only 1533 instance of the total 1966 instances predicted being properly predicted. This may be a consequence of the fact that the “No emotion toward brand or product” class composed most of the testing sample and there was some slight overfitting with the model. Furthermore, it seemed as though the kNN model was able to properly classify more class instances than any of the other models, which can be seen in both the CA and the confusion matrices. Although the kNN and the AdaBoost models may have had the highest CA scores, the predictions for classification of the “Negative emotion” class was not the best. Of the 187 instances of the “Negative emotion” class, only 34 were properly classified by the AdaBoost method and only 28 instances of 187 were properly classified by the kNN method. This lack of classification accuracy may be attributed to the spread of the data in the dataset. Since there were not many instances of the “Negative emotion” class, the methods may not have been able to completely understand how to properly classify them. This can especially be seen for the instance of the “Unclear” class throughout all models, as there were no proper identifications of these class instances at all. However, for the “No emotion toward brand or product” and “Positive emotion” class there were many accurate classification predictions. In fact, many of the “Negative emotion” class instances were misclassified as the “Positive emotion” class for both the kNN and the AdaBoost models. This is likely due to the majority of samples belonging to these two classes.

To understand which of these models is best for the data at hand, understanding each model’s method is important. Each of these models work in different ways to generate class predictions for a given dataset. Because features were created for the tweet data with a bag of words model, the initial features used for prediction for each model were the words associated with the emotion label and their normalized word counts (vector length to sum of squares). To perform predictions, the kNN model finds the distance between a given query and the examples in the data and selects a specified quantity of examples (k-value) that are closest to the given query [3]. The resulting prediction is then based upon the most frequent label in the selected quantity of labels. The important feature of this model would be the distance in the sample space between its neighbors, since the prediction is based upon this feature. The AdaBoost model using the boosting ensemble method to create a strong classification method from multiple weak ones (SAMME.R in the case of this data). Basically, the AdaBoost method creates an initial model from the training data with predictions, creates a second model to attempt to fix the errors from the first model, and then continues creating models until the training set can be predicted completely or the maximum quantity of models have been reached [4]. When creating these models, higher scores are given to misclassified values thus causing the model to continually adjust more for those misclassified values. The final model uses the weighted average of the models to predict a class for the new data. The most important feature of this model would be the weights of the models created, as the larger the weights of the model, the more pull they have upon the final weighted average. The random forest model is another ensemble method that utilizes bagging, where many small decision trees create their own predictions and these predictions of each tree, or estimators, are then combined to predict the instances of a given class. The important feature for the random forest model is based upon the built-in feature importance called the Gini importance, which determines the feature importance through reduction in the Gini impurity (mean decrease impurity) [5]. So, basically, each predictor chooses features that reduce the overall Gini impurity.

Of the models utilized, the most effective models seemed to be between the AdaBoost and the kNN model. However, given the classification accuracies, the kNN model proved to be more effective in classifying this dataset. This makes sense because the kNN model utilizes distances between the sample and k neighbors which have the closest distance. When utilizing text visualized as feature vectors, the use of the distance measure seems to be the most natural form of multivariate classification. However, both the kNN model and the AdaBoost model have promise for use in sentiment classification predictive analysis with further tweaking and potential generation of more data to train for the minority classes.

V. Conclusions

Throughout the course of this paper, a predictive sentiment analysis was performed to determine which of three classification models performed the best on consumer generated text regarding sentiment of product or brands. It was determined that of the three models, the kNN model retained the best classification accuracy and was the best model for this dataset. Further, the kNN model is a suitable model to further pursue for sentiment analysis of these consumer generated product sentiment data and may ultimately be a suitable for utilization in future personalized advertisement models. However, given some of the limitations of the dataset, the kNN model could have easily been dethroned by the AdaBoost model. These dataset limitations were simply the lack of more instances of the lesser classes. In other words, given that more instances of the minority classes were present in the data, the models would have been able to perform better. To fully elucidate these findings, it would be ideal to obtain more data for the minority classes so that there are more opportunities for the models to learn and properly classify instances. Furthermore, some improvements of the current model can be pursued to make it better for this application.

References

[1] Kemp S, “DIGITAL 2020: 3.8 BILLION PEOPLE USE SOCIAL MEDIA”, We Are Social Inc., 30 January, 2020. [Online]. Available: https://wearesocial.com/blog/2020/01/digital-2020-3-8-billion-people-use-social-media#:~:text=Across%20mobile%20devices%20and%20computers,since%20this%20time%20last%20year. [Accessed: June 15, 2021].

[2] Crowdflower, *Brands and Product Emotions*, Crowdflower: data.world, inc. [Dataset]. Available: https://data.world/crowdflower/brands-and-product-emotions. [Accessed : 13 June, 2021].

[3] Harrison O, “Machine Learning Basics with the K-Nearest Neighbors Algorithm”, Medium, September 10, 2018. [Online]. Available: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761. [Accessed: June 16, 2021].

[4] Brownlee J, “Boosting and AdaBoost for Machine Learning”, Machine Learning Mastery Pty. Ltd., August 15, 2020. [Online]. Available: https://machinelearningmastery.com/boosting-and-adaboost-for-machine-learning/. [Accessed: 16 June, 2021].

[5] Płoński P, “Random Forest Feature Importance Computed in 3 Ways with Python”, MLJAR, Inc., June 29, 2020. [Online]. Available: https://mljar.com/blog/feature-importance-in-random-forest/. [Accessed: 18 June, 2021].