Analyzing Sentiment in Customer Reviews using Advanced Machine Learning Techniques and Feature Engineering

Group 5

Members: Harshaditya Kumar, Parinay Karande, Hemanth Mohan

Member's Roles and Responsibilities:

1. **Harshaditya Kumar:** Prepared initial project proposal document. Worked on Outliers/Anomaly detection and Advanced techniques using Keras.
2. **Parinay Karande:** Finding and finalizing dataset. Preparing final project report document. Implemented classifiers, clustering algorithms, and statistical tests for feature engineering.
3. **Hemanth Mohan:** Worked on Data cleaning, EDA and data preprocessing including tokenization, stop word removal, and stemming. Employed TF-IDF and word embeddings. Preparing the final report.

Abstract:

Our project aims to develop machine learning model to predict sentiment in customer reviews using historical consumer review Amazon product data and explore patterns and insights in the data using supervised and unsupervised methods. The motivation behind this study is to provide businesses with valuable insights to better understand their customers and enhance their products and services based on the analysis of customer reviews.

Table of Contents:

1. Introduction:

The primary significance of this dataset is that it provides a rich source of historical data on customer reviews, which can be used to develop machine learning models for predicting sentiment. The sentiment, derived from the rating and review text, is the key target variable in the dataset, which can be predicted using a variety of features such as previous ratings, review text, and product information. The goal of this analysis is to develop and test machine learning models that can predict the sentiments of reviews. A variety of types of models are explored in order to achieve this goal including logistic regression, gradient boosting and neural networks. Several preprocessing techniques are explored along with models for scaling and feature selection. The performance of the models is evaluated using evaluation metrics like Accuracy, Precision, Recall, and F1 score. This Analysis also contains few visualizations and statistical tests to identify and visualize the findings and differences observed between the variables.

Methodology:

1. Data Collection and Preprocessing: We used the customer reviews dataset for preprocessing, including tokenization, stop word removal, and stemming.

2. Feature Engineering: We employed TF-IDF and word embeddings for vector representation. We also implemented the Chi- Square statistical test to select the 100 most relevant features and performed scaling/transformation using MaxAbsScaler to normalize the data.

3. Supervised Models: We implemented Logistic Regression and Gradient Boosting classifiers for predictive modeling.

4.Unsupervised Model: We applied a K-means clustering algorithm for pattern discovery.

5.Outlier/Anomaly Detection: We then used an outlier/anomaly detection method called Isolation Forest to identify and handle anomalies in the data.

6. Model Evaluation: We performed the split on data into train-test sets using k-fold cross- validation and assessed the models performance using metrics such as accuracy, precision, recall, and F1 score.

7. Advanced Techniques: We incorporated an advanced method using deep learning with Keras to improve the results of the models.

8. Visualization: We finally used a visualization technique called word clouds to analyze and present our findings of most used words in the review text.

Results:

We Started our code analysis by importing the necessary libraries like Pandas, numpy, sklearn, nltk, seaborn, matplotlib, tensorflow, and keras. We also downloaded the necessary resources using the appropriate methods. After that, we loaded the dataset from the below URL.
URL: https://www.kaggle.com/datasets/datafiniti/consumer-reviews-of-amazon-products

As the dataset was loaded in a variable df, we went ahead to the data preprocessing and EDA. Firstly we converted all the text to lowercase, removed all the punctuation marks and replaced them with black spaces using the regular expression pattern. Then we went ahead and tokenized the words and performed word removal and stemming. We did this using the nltk library. In the next step we performed feature engineering by creating TF-IDF vectors using the TfidfVectorizer() method. We considered this as our X variable.

Next step was to identify and separate the target variable. Sentiment_Label was removed from the data set and stored as target variable y. Before being set as target variable, the column was made binary by giving values of 0 or 1. We then went ahead to split the data into testing and training data.

Then we went ahead to perform some feature selection, using ANOVA-F value as a scoring function. We obtain the top 10 features using SelectKBest with k = 10 as a parameter. These features have the highest score and are assumed to have a strong correlation with the target variable. We also applied scaling techniques like Standard scaler and MaxAbsScaler.

The first supervised classifier we used was Support Vector Classifier. Predictions were made on the test data and the metrics were obtained using the classification report. After looking at the observations, we got precision and recall as 0.94 and 1.00 respectively. This concludes that the sentiment was positive 94% of the time as 1 refers to high rating. Also, the accuracy observed here is 0.94.

Going ahead we performed a Logistic regression and Gradient boosting classifier. We got the same precision and recall score that we got with SVC. The accuracy was 0.94 and F1 was 0.97.

Then we wrote a code to perform KMeans Clustering on training data with 2 clusters. We applied PCA for reducing dimensionality and used a scatter plot to visualize the clusters.

The next part of our code refers to outliers and anomaly detection. We used the Isolation forest algorithm for identifying anomalies in the data. However after fitting the method, we did not get any anomalies. Then we used TF-IDF vectorizer with max 1000 features and set the contamination parameter to 0.1 before fitting the model. This time we got 3466 anomalies.

For the advanced techniques, we tried to train a neural network model using Keras library. When we evaluated the trained model on the test data we got an accuracy of 93.15 %. We also plotted graphs that illustrate the model's training and validation loss. These graphs can be found in the appendices section of this document.

In the end we wrote a code for WordCloud visualization which shows the most frequently occurring words in the reviews.

Actionable Insights:

1.A logistic regression model and a gradient boosting classifier are fitted to a dataset in the first code fragment, and their performance is assessed using the classification report. The report includes helpful measures for each class, including precision, recall, and F1 score, which may be used to identify the model's advantages and disadvantages.

2.The second code sample divides the data into two clusters using KMeans clustering and displays them using PCA. This can reveal information about the data's structure and point out any trends or anomalies

3.The third code snippet employs isolation forests and TF-IDF vectorization to find abnormalities in text data. The results can be used to spot probable outliers or odd data trends.

4.The fourth code snippet uses Keras to train a neural network and measures accuracy as a measure of performance. If the model is overfitting or underfitting, it can be determined by looking at the loss and accuracy plot over epochs.

5.The last piece of code creates a word cloud graphic from the text data, which can reveal information about the terms or subjects that appear most frequently in the dataset. This can aid in comprehending the text's primary ideas.

Discussion and Critics/Critiques:

1.Overfitting: The model may be overfit to the training data, depending on its complexity and the quantity of data provided. Poor performance on unknown data may come from this, necessitating the use of regularization techniques to avoid overfitting.

2.Many machine learning algorithms have hyperparameters that must be tuned in order to perform at their best. This can be a laborious process that necessitates a deep understanding of the method and the problem domain.

3.Data preprocessing: Text data needs to be processed in order to be transformed into a numerical representation that machine learning algorithms can use. Tokenization, stopword elimination, stemming or

lemmatization, and vectorization are a few examples of these processes. The effectiveness of a model can be significantly impacted by various preprocessing methods.

4.Uneven distribution of classes in a dataset is a common difficulty in classification problems. Due to this, it may be challenging for the model to learn from the minority class, necessitating the use of techniques like over- or undersampling to resolve the problem.

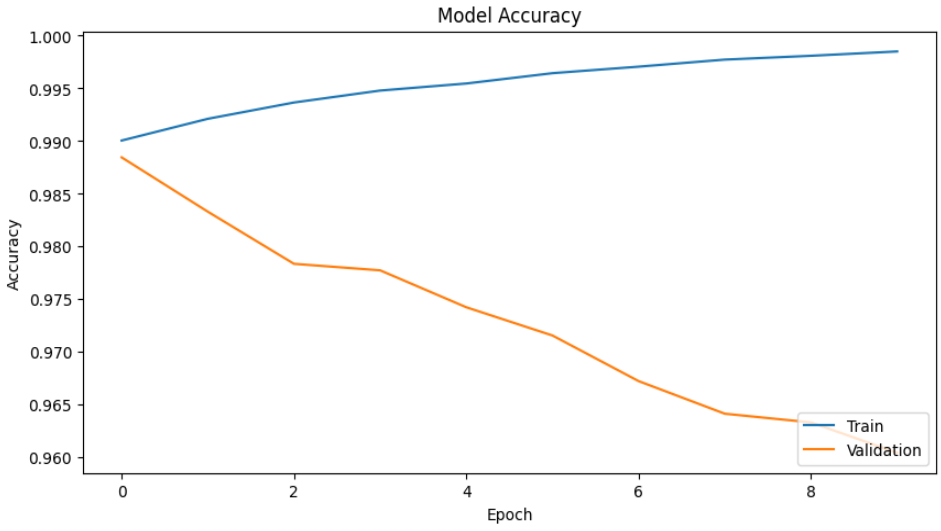
5.Interpretability: It might be difficult to grasp how a machine learning algorithm is making predictions when it comes to some algorithms, such as deep neural networks. This can be a concern in fields like healthcare or finance where interpretability is crucial.

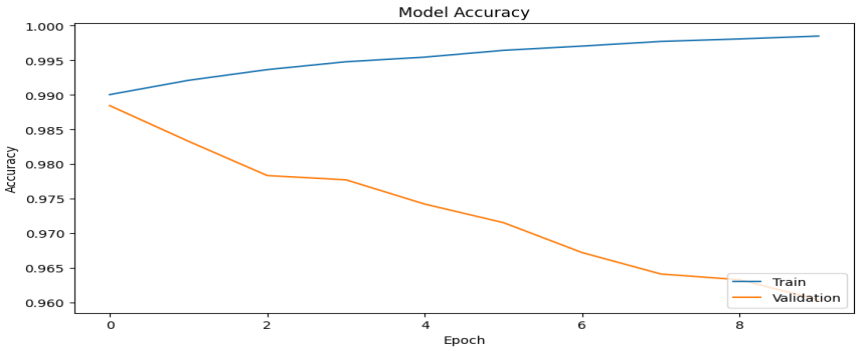
Conclusion:

- The scripts in the code show how several machine learning methods can be used to analyze text data. Classification and anomaly detection tasks were carried out using Logistic Regression, Gradient Boosting Classifier, Isolation Forest, and Neural Networks. For exploratory analysis of text data, K-Means clustering and Word Cloud were employed.
- These methods allow us to learn more about the properties of text data and spot trends and anomalies. It is crucial to remember that the caliber and kind of the data have a significant impact on how well these strategies operate. To get accurate and insightful findings, preprocessing, feature engineering, and hyperparameter tweaking are essential processes.
- Overall, these programs offer a solid foundation for text data analysis and can be improved upon and tailored for certain use cases.

Appendices:

The first graph illustrates the model's training and validation loss across epochs, showing how well the model is learning and generalizing over time, while the second graph demonstrates the model's training and validation accuracy across epochs, indicating how accurately the model is classifying the data throughout the training process.





This indicates how well the model is correctly classifying the data, with the 'Train' line representing accuracy on the training data and the 'Validation' line showing accuracy on the validation data. As epochs progress, we aim for the model's accuracy to increase and for both lines to converge, which would suggest the model is generalizing well and not overfitting to the training data.

The WordCloud visualization shows the most frequently occurring words in the reviews, with the size of each word indicating its frequency; the larger the word, the more frequently it appears in the review texts.