**Sentiment Analysis Based on Amazon Review**

Yi Ding, Pinqiao Wang

Instructor: Yufeng Liu

STOR565 Research Paper

30 November 2022

**Project Summary**

This project conducted sentiment analysis by performing binary classification on the polarity of Amazon user review. Our dataset was originally collected and shared by Dr. Julian McAuley at UCSD, which contains product reviews and metadata of Amazon, spanning May 1996 - July 2014. The predictor we focused on is the summary of the review, which is the shortened version of whole text, and we wanted to use that to predict whether the review is positive or negative.

We adopted different machine learning and deep learning techniques to perform the classification, including Naïve Bayes, Logistic Regression, Support Vector Machine, Random Forest, and Convolutional Neural Network. We ended up getting close results for the methods we took, and among all SVM returns the highest prediction accuracy.

Further improvements could be made to further increases the validity and accuracy of the classification, including introducing advanced word of bag model, using full text instead of summary to conduct the classification, and trying other more complicated deep learning methods such as Recurrent Neural Network.

1. **Overview**

Sentiment analysis belongs to the realm of Natural Language Processing, a popular field of Artificial Intelligence that receives increasing attention over time. It systematically identifies, extracts, and quantifies subjective information and use them to perform classification or prediction. It was first known as the General Inquirer, examining patient’s mental status based on their verbal behavior (Stone et al, 1966). Now, the technique is widely applied to reviews, survey responses, and social media.

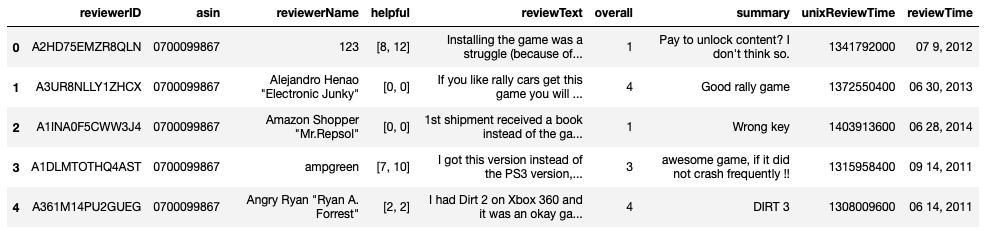
With decades of improvement in techniques, the accuracy of sentiment analysis has been relatively satisfying, however there are still major difficulties for any machine learning methods to overcome. Some of the common challenges are listed below (Wikipedia):

* + Negation: I do not dislike cabin cruisers.
  + Sarcasms: I'd really truly love going out in this weather!
  + Negative term used in positive sense: The movie is surprising with plenty of unsettling plot twists.
  + Qualified positive: I love my mobile but would not recommend it to any of my colleagues.

Frankly speaking, sometimes it is even hard for human beings to grasp the meaning of these complicated sentences, not to mention machine. Thus, it is worth trying different methods to see if any one of them can handle those difficulties exceptionally well.

1. **Data Description and Processing**

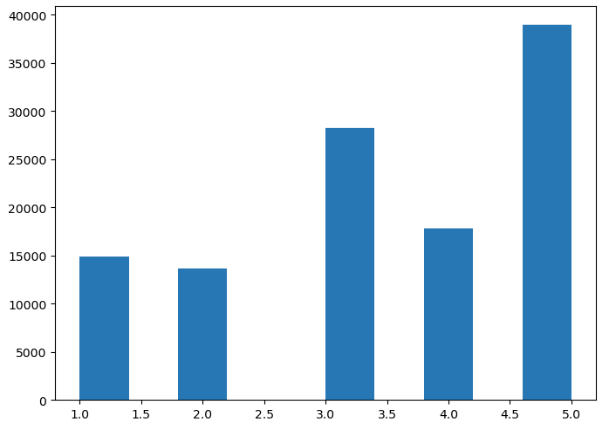
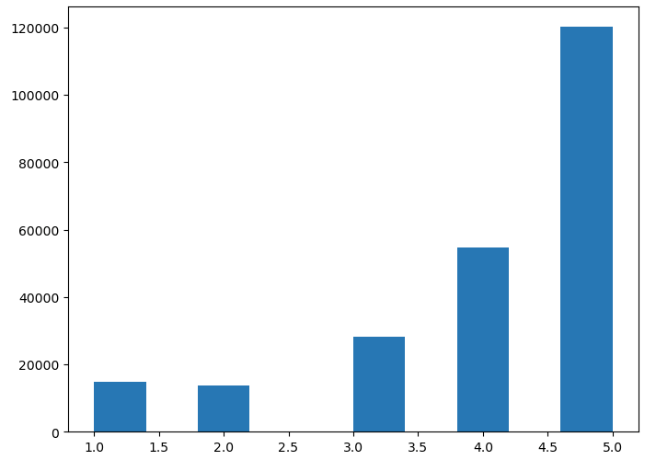
The dataset we used are originally collected by Dr. Julian McAuley at UCSD, which is a comprehensive dataset that contains 142.8 million reviews across different products of Amazon. Here, we chose the Video Games Dataset, whose information is quite comprehensive and the size is suitable for analysis. The following graph provides an outlook for what the data looks like:



**Figure 1**: Dataset

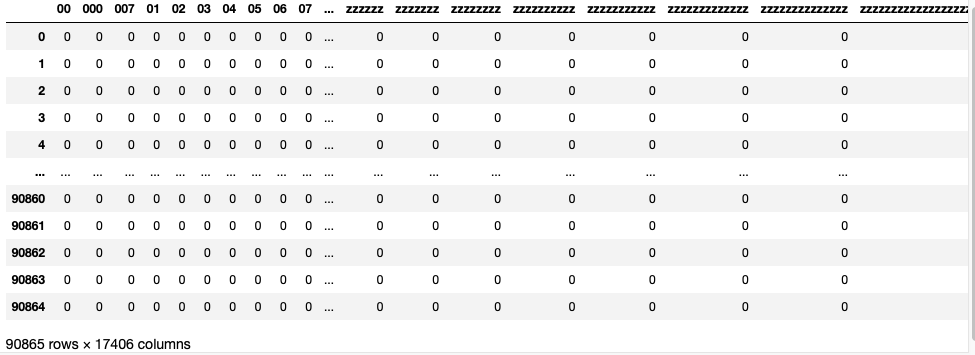
In total, there are 231780 observations. Since we are trying to use “summary” to classify

its sentiment, and use “overall” to test whether we have made right classification, we want to first make our dataset balanced in terms of the number of observations that belong to positive and negative respectively. We define reviews with rating <=3 as negative, >3 as positive. In this way, we end up having 56791 negative reviews and 174989 positive reviews, which is not very balanced. The following histogram gives an idea of the distribution of the ratings.

 **Figure 2**: Distribution of rating before/after adjustment

We randomly sampled 56791 positive reviews so that # of positive review = # of

negative review. In this way, we have created a balanced dataset! The distribution of ratings after adjustment is shown below, which looks more balanced.

Then, we divide out data into training and test set. Since we don’t have parameter to tune for the machine learning techniques chosen, we didn’t create a validation set. The training and test set is randomly chosen according to an 80:20 ratio. Next, we need to convert the words appeared in each review into matrices of numbers so that they can numerically enter the model. The bag-of-words model is introduced to count the number of words appeared in the review and their frequencies, so that the input is all the reviews, and the output is a huge sparse matrix. We used the package Sci-kit Learn with the function CountVectorizer to achieve this, which also provides other options such as tokenization and stop words. We ended up getting a matrix of 90865x17406 sparse matrix, which is shown below.

**Figure 3**: Sparse matrix used for classification

1. **ML and DL Techniques**

**3.1 Logistic Regression**

A logistic regression model was implemented to construct a linear model to predict the response “overall”, which is manually interpreted as “positive” comments when the value is greater than 3 and as “negative” comments otherwise, using the converted variable “summary”.

**A picture containing polygon

Description automatically generated**Chart, treemap chart

Description automatically generatedAfter fitting into logistic regression model, the computed confusion matrix shown below gives us 18049 correct predictions and 4668 incorrect predictions, which brings the model accuracy rate to 79.45%. This is a relatively high accuracy considering our large dataset, almost 80 percent of the outcomes were correctly predicted. However, model accuracy can be an insufficient concept when it comes to classification models because the metrics that are meaningful to the model will vary based on the purpose of the model. Even though we have manually reshaped our original dataset into a balanced distribution, but it is inevitable that we will encounter with imbalanced mixed reviews in the real world. Positive reviews occur more frequently than negative ones. Out of this concern, we introduced two new metrics by breaking down accuracy to discovery how our model truly predicted our outcomes under the challenges of the ambiguous nature of human language. This methodology will be using throughout all machine learning techniques implemented in this project.

**Figure 4:** Confusion Matrix for Logistic Regression model and ROC&AUC for Logistic Regression model

Sensitivity, the number of samples predicted correctly to be belonging to the positive class out of all the samples that in fact belongs to the positive class, and Specificity, the number of samples predicted correctly to be in the negative class out of all the samples in the dataset that in fact belongs to the negative class, are introduced to evaluate the model performance unbiased and comprehensively. Here, the sensitivity is 79.5% and the specificity is 79.4%, they represent that our data is balanced and relatively predicted well. ROC curves and AUC are introduced as well as they tell how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is. As shown below, the AUC value for logistic regression is 0.877 and approaches towards 1. It is finally safe to say that the logistic regression model has a relatively good performance. However, it is not the best.

**3.2 Naïve Bayes**

**A picture containing polygon

Description automatically generatedChart, treemap chart

Description automatically generated**Naïve Bayes usually doesn’t perform well on imbalanced data, but it becomes more valuable to try when we have manually solved this problem. After modelling and predicting, the confusion matrix of Naïve Bayes gives us 17723 correct predictions and 4994 incorrect predictions, resulting in an accuracy of 78.02%.

**Figure 5:**  Confusion Matrix for Naïve Bayes modelandROC&AUC for Naïve Bayes model

The sensitivity and specificity are 78.44% and 77.61% respectively, indicating that the correct prediction rate for positive reviews is slightly higher than for negative reviews. According to the ROC curve and AUC shown below, the performance of this model is also slightly worse than logistic regression by approximately 0.02.

One strongly possible reason of why Naïve Bayes performed worse logistic regression is

that in sentiment analysis, it is nearly impossible for, independence of variables, one of the most important assumptions of Naïve Bayes to stand. In our video games dataset, for example, the word “scary” and “dark” tend to depend on each other and are correlated to a certain extent with the word “graveyard” or “night”. Obviously, Naïve Bayes will not the best model as well.

**3.3 SVM (Linear and RBF)**

With all other things equal, support vector machine (SVM) usually performs the best when encounter imbalanced data, therefore we will examine this technique with linear kernel tricks and Gaussian Radial Basis (RBF) kernel given we have vectored the data.

Starting with linear SVM, we obtained a 79.5% accuracy. The confusion matrix shows 18061 correct predictions and 4656 incorrect predictions.

Chart, treemap chart

Description automatically generated

Figure 6:Confusion Matrix for Naïve Bayes model

The RBF kernel function for any two points X₁ and X₂ computes the similarity or how close they are to each other, which outperforms linear SVM who does not take influences of one’s neighbor into account. It has the advantages of K-NN and overcomes the space complexity problem as RBF Kernel SVM just needs to store the support vectors during training and not the entire dataset. Improved with switching kernel to Gaussian Radial Basis, our model accuracy raises to 81.09% with sensitivity and specificity are 81.43% and 80.77% respectively.

**3.4 Random Forest**

**Chart, treemap chart

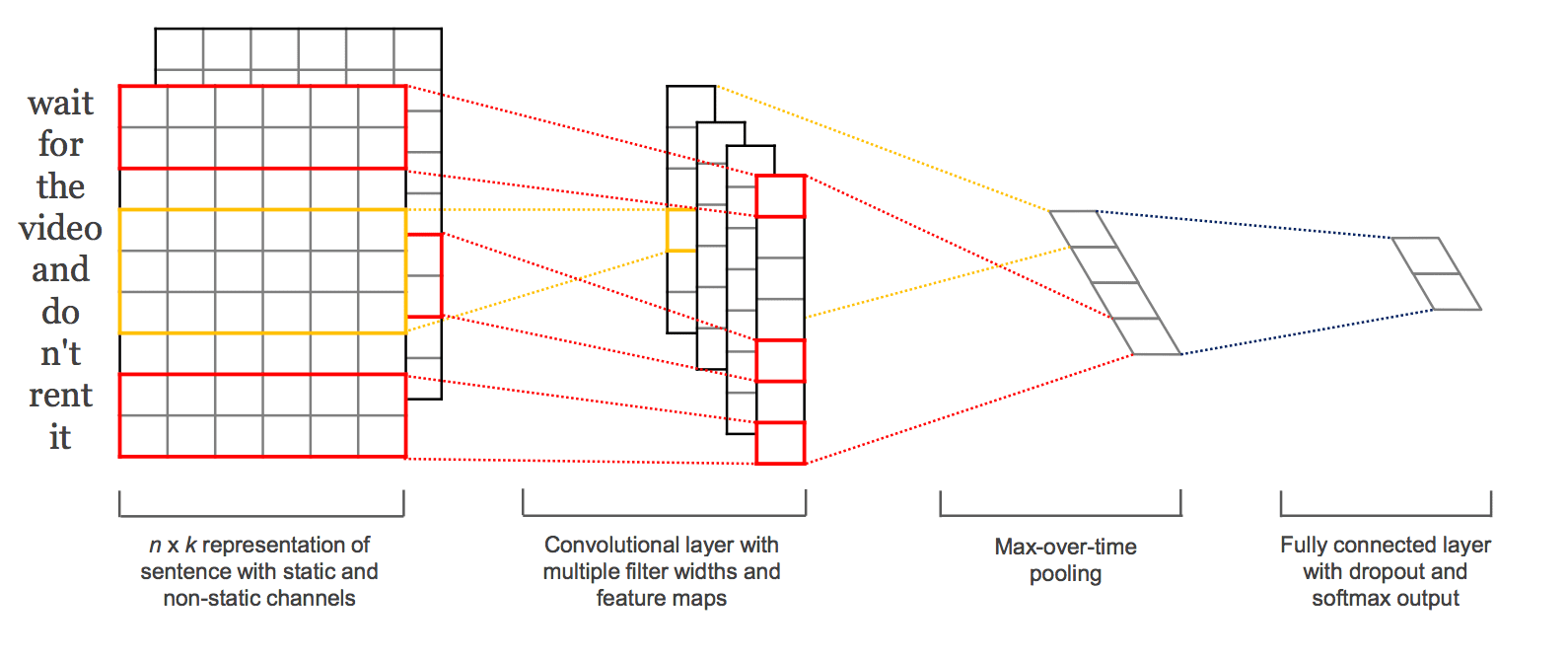
Description automatically generatedA picture containing polygon

Description automatically generated**Random Forest in another widely used machine learning algorithm since it not only it is a highly accurate and robust method because of the number of decision trees participating in the process, but also since it does not suffer from the over-fitting problem. The optimal number of trees used in random forest is usually the number of rows of dataset, however, we are unable to produce such prediction given that we have over 1 million rows of data. After a few times of tuning the parameter, we decide to use 1000 as our parameter. Based on the confusion matrix, we have 79.65% as accuracy rate, and 79.11% and 80.2% as sensitivity and specificity rates respectively. Out of concern about computational efficiency, we cannot build this model with more trees, yet the statistics obtained from 1000-tree model gives a good performance of AUC of 0.876.

Figure 7:Confusion Matrix for Naïve Bayes modelandROC&AUC for Random Forest model

**3.5 CNN**

In addition to the classical machine learning techniques we learned in class, we also applied deep learning methods that we acquired autonomously online in our project. The method we adopted is Convolutional Neural Network (CNN), which is an increasingly popular method that is widely used in Computer Vision, but it can also apply to simpler tasks such as binary classification of sentiment analysis in our case.

 Figure 8:Structure of CNN

The basic structure of CNN is shown in the above graph. The network takes the sparse matrix that we calculated before as the input. Convolutional layer is used to extract features, so that the most meaningful information will be able to pass while others being filtered. Consequently, the pooling layer serves to reduce dimension complexity and keep significant information. Paddings are optional, which guarantees that the feature map doesn’t shrink. Finally, we have the fully connected layer which contains the activation function to perform the classification. The following graph shows the result of the CNN model we built. Along the 5 epochs, we notice that the training error keeps decreasing, however the test error decreases from epoch 1 to epoch 2, then keep increasing. This indicates that we have encountered the problem of over-fitting, and epoch 2 is the best model we can get. Thus, using CNN, we reach a classification accuracy of 81.01%, which is almost as good as the best model from machine learning.

Table

Description automatically generated

Figure 9: CNN Training Results

1. **Conclusion**

Summing up, all machine learning and deep learning techniques performed rather closely

and adequately well. SVM model with RBF kernels has the best performance of 81.09%, followed by CNN model with an accuracy of 81.01%. SVM outperforms others due to its capacity to handle imbalanced in real-life practices and its emphasis on dependency of variables in this case. Naïve Bayes fitted worst since it assumes that there are no relationships between variables.

In the future, we will try to fit models with uncleaned and imbalanced thus to make them more practical in real-life scenarios noticing that sensitivities and specificities are closed to each other in this project. There are several aspects that can be improved: bag of words only analyzes the semantic meaning of words without context; therefore, it would be better if future studies can apply more advanced models such as n-gram to account for the sequence of word that appeared. Recurrent Neural Network could also provide great solutions; however, we are analyzing short-length text here, thus it may not perform very well. Finally,

future studies may be able to perform multinomial classification instead of binary one, but this also takes a higher computational cost.

1. **References**

Stone, Philip J., Dexter C. Dunphy, and Marshall S. Smith. "The general inquirer: A computer approach to content analysis." MIT Press, Cambridge, MA (1966)

R. He, J. McAuley, Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering, WWW, 2016

Kumar, Koushik; NLP: Bag of words and TF-IDF explained!, Jan 24 2021, <https://koushik1102.medium.com/nlp-bag-of-words-and-tf-idf-explained-fd1f49dce7c4>

Verma, Mayank, Binary Classification Using Convolution Neural Network (CNN) Model, May 8 2022, <https://medium.com/@mayankverma05032001/binary-classification-using-convolution-neural-network-cnn-model-6e35cdf5bdbb>