COMP30027 Assignment 2 Report

Anonymous
Word Count: 1833

1. Introduction

Machine learning techniques have proven to be powerful tools for a wide range of prediction tasks. In this report, I focus on the task of predicting book ratings based on text and numeric features. Text features include the name, author, publisher, language and description of the book, and the numeric features are the book's publication date and page number. By developing and comparing different machine learning models, I aim to find out the most effective features and the most suitable model, in order to achieve the most optimal predictions. By analysing the features, I manage to transform all features into text features. These features have been pre-processed and encoded using the CountVectorizer in the sklearn library, which transforms text documents into a "Bag of Words" representation. With multiple trials and errors during the modelling process, I find Multinomial Naïve Bayes to be the most effective model. Therefore, based on the machine learning techniques used in the insights of the book-rating task, this report aims to provide a comprehensive analysis of the predictions and explore the reasons behind the scene.

2. Approaches To Feature Optimizations

Often, how to approach complex Machine Learning problems is significant. However, there are some common strategies that Peter Hussami advocates. He suggests to "*set acceptance criteria*", "*clean data and maximise its information content*", "*choose the most optimal approach*" and indeed "*train, test and repeat*". (Peter Hussami, 2018)

2.1 Data Pre-processing

Firstly, I broadly view the training data set (see Figure 1). I came up with the following observations:

- There are nine features that we can use, including name, authors, PublishYear, Month and Day, Publisher, Language, pagesNumber and Description. Except for the publication date and page number, which are numeric features, others are all text features. Then relevant techniques that can deal with text features have crucial impacts on the quality of pre-processing.
- PublishYear, PublishMonth and PublishDay contain the information of the publication date. Therefore, it is intuitive to either to use one Year feature or three of them together, as a single month or day feature cannot provide useful information on separating rating.
- There are nan values in the entries belonging to Publisher and Language feature. Methods dealing with nan values will be effective.

```
RangeIndex: 23063 entries, 0 to 23062
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Name          23063 non-null  object
 1   Authors       23063 non-null  object
 2   PublishYear   23063 non-null  int64
 3   PublishMonth  23063 non-null  int64
 4   PublishDay    23063 non-null  int64
 5   Publisher     22915 non-null  object
 6   Language      5861 non-null   object
 7   pagesNumber   23063 non-null  int64
 8   Description   23063 non-null  object
 9   rating_label  23063 non-null  float64
dtypes: float64(1), int64(4), object(5)
memory usage: 1.8+ MB
```

Figure 1- Basic information of the training data frame

2.1.1 Impute Missing Data

It is normal that the real-world data contains a lot of missing values, which may be caused by data corruption or recording failures. Nevertheless, machine learning algorithms cannot proceed without implementing pre-processing on missing values.

Some feasible solutions include: "*leave the text as an empty string*", "*discard the instance with empty values*" and "*treating the instance with empty values specially by assigning it to the majority class*". (Erwan, 2021)

2.2 Text Features and Numeric Features

Text features represent textual information associated with the data. In order to be used as inputs to Machine Learning algorithms, text features are often transformed into numerical representations, by using techniques like Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF) and Word Embeddings (Word2Vec). In my work, BoW is chosen, which "*converts text into vector of numbers*". (高橋涉, 2019) Count Vectorizer from the sklearn package is used to achieve word counting purpose. The resulting feature vector represents the presence of words and will be trained by a selected model.

Numeric features represent quantitative or continuous variables. To make numeric features modelled by learning algorithms, they are converted to a string type, such that they can be integrated into the feature vector, output as a sparse matrix.

2.3 Learning Algorithms

Multinomial Naïve Bayes is a variant of the Naive Bayes algorithm commonly used for text classification tasks, where the features (input variables) are discrete and represent word counts or frequencies. It is based on the Bayes' theorem of probability and assumes that the features are conditionally independent given the class label. It is not hard to believe that text inputs are relatively independent, which can be tested by using cosine similarity. The numeric variables can be easily checked independence by using correlations (see Figure 3). Based on the type of my features, Multinomial Bayes algorithm is a reasonable approach.

3. Results Discussion

By splitting the training data set into train and test sections, predictions are made by my model and the results are compared with the given labels. The classification report (see Table 1) shows the performance for each rating prediction. This model predicts the rating of 4.0 better than other labels and the whole accuracy of the model is 0.70, which is reasonably acceptable.

	Precision	Recall	F1-Score	Support
3.0	0.49	0.26	0.34	1887
4.0	0.74	0.90	0.81	5404
5.0	0.26	0.08	0.12	320
Accuracy			0.70	7611
Macro Avg	0.50	0.41	0.43	7611
Weighted Avg	0.66	0.70	0.67	7611

Table 1- Classification report table of my model, testing on the training data

Below is a heatmap made by the Confusion Matrix between three labels. (see Figure 3) It can be seen that my model has a tendency to predict 4.0 rating, even though the true label is 3.0 or 5.0.

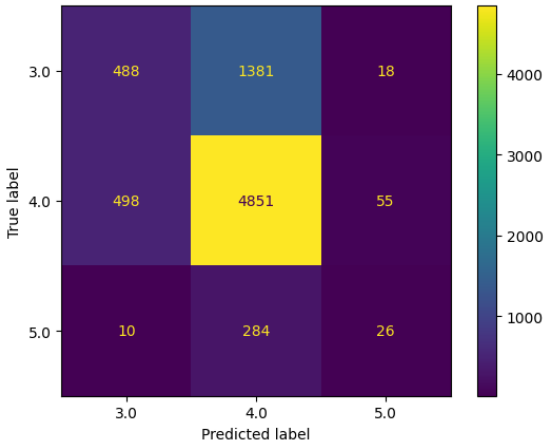

Figure 2- Confusion Matrix represented as a heatmap

4. Critical Analysis

I have briefly gone through how I approach this machine learning problem and the results based on the chosen algorithms. This section will illustrate a more detailed analysis to elaborate my overall idea.

4.1 Missing Values

The chosen method for dealing with missing values is the make each of them to be an empty string. In such a case, the count vectorizer will transform this empty string to be a zero matrix, which is exactly what it should be. Doing such a replacement, the model will tend **NOT** to select the corresponding feature, as it will produce a very low bayes probability.

4.2 Text Features

Count Vectorizer is used to handle text features. The CountVectorizer works by tokenizing the text into individual words or tokens and then counting the frequency of each token in each document. It creates a sparse matrix representation where rows represent documents, columns represent tokens, and the values in the matrix represent the count of each token in each document.

4.2.1 Publication date

To deduct whether to use just PublishYear attribute or combing them three together, I firstly checked the independence between three attributes (see Figure 3).

It is clear that they are not independent, which

is really practical. This fulfills the prerequisite for choosing any one of them. Otherwise, I must exclude some of them to ensure independence.

After checking the performance of my two ideas, it is found that using PublishYear attribute along has the highest performance on prediction, which gives an accuracy of 0.72403.

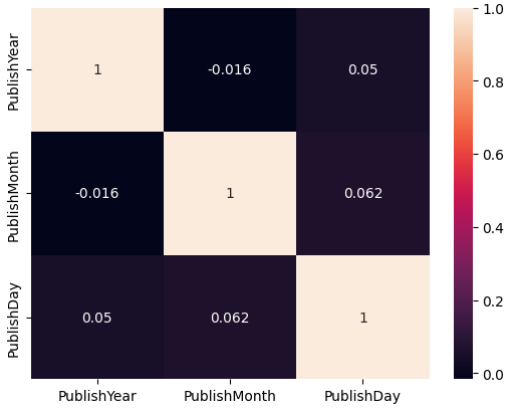

Figure 3- Heatmap of the three publish time variables

Figure 4- Prediction accuracies when using different features.

4.3 Multinomial Naïve Bayes

"*The Multinomial Naive Bayes algorithm is a Bayesian learning approach popular in Natural Language Processing (NLP). The program guesses the tag of a text, such as an email or a newspaper story, using the Bayes theorem. It calculates each tag's likelihood for a given sample and outputs the tag with the greatest chance.*" (Sriam, 2022)

4.3.1 Method Behaviour

The model is trained based on the count vector I manipulated. This high dimensionality matrix is used to calculate the probability density function for each feature. In this case, a multinomial Gaussian distribution function is used to find the probability. Then the most likely label is deduced based on the count vector (sparse matrix) manipulated from the features of the test data.

4.3.2 Theoretical Property and Error Analysis

Multinomial Naïve Bayes is capable for handling continuous data. The transformed count vector is continuous which suits this well. This method also has the capability to compute with very high dimensional datasets. In my trained sparse matrix, it has a size of (23063,137055). However, its disadvantage is when there are missing values, and the accuracy falls down. Then the methods for dealing with missing values help to make the method stable. According to the previous classification report, (see Table 1) the accuracy is 0.70, this means 30% of the predictions are wrong, which is the error rate of the method.

4.3.3 Flaw Data

This method predicts a 4.0 rating more often, which can be seen by a higher support of 4.0 and a higher recall value. A higher recall value (than precision) means this model does not need much evidence to identify a 4.0 rating. This could be resultant of a large percentile of the training data being made up of books with a rating of 4.0. This influences the accuracy of the training.

4.4 Comparison With Other Methods

During the trials and errors process, multiple modelling methods are attempted, but are not used. The following are all of them and reasons why they are not selected are mentioned.

- Support Vector Machine

"*The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points.*" (Gandhi, 2018) Since finding hyperplanes for thousands of features needs a huge amount of space and time and its accuracy is not higher than the Multinomial Naïve Bayes, I did not choose this method.

- Decision Trees

The decision trees implicitly perform feature selection. Since there are thousands of features in a dictionary that the Count Vectorizer produces, doing feature selections help to shorten the time running the algorithm. However, this may not guarantee an improvement on the performance, as losing some rare words may cause an incompleteness on the rating prediction. Also, "*Greedy algorithms cannot guarantee to return the globally optimal decision tree.*" (Gupta, 2017)

- Neural Network

Neural Network is "*a clustering and classification layer on top of the data you store and manage.*" (Nicholson, 2023) Such a method is very complicated and requires a substantial amount of data to train. However, a model trained with a lot of training data tends to overfit the data, which means that it is accurate to predict the seen data, but cannot guarantee an overall good performance on predicting unseen data.

5. Conclusions

In this task, I focused on the task of predicting the rating of a book based on its various features, including text and numeric attributes. Through my exploration of various features and comparison of different machine learning models, I aimed to identify effective features and develop accurate prediction models.

I explored various text encoding techniques, including the CountVectorizer in the sklearn library, which enabled me to transform textual data into numerical representations. This allowed me to utilize the power of machine learning algorithms that typically require numeric inputs.

After leveraging many Machine Learning algorithms, including Multinomial Naïve Bayes, SVM, Decision Trees and Neural Network to tackle this prediction problem, I discovered that each algorithm has its strengths and limitations. Choosing Multinomial Naïve Bayes as my solution predicts the rating well, by analysing factors such as accuracy, precision, recall and F1 score.

In conclusion, my project demonstrated the application of machine learning techniques for predicting book ratings based on various features. Going forward, further research and experimentation can be undertaken to enhance the accuracy and robustness of the models, potentially leading to valuable insights for the book industry and its stakeholders.

6. References

[1] Peter Hussami. 2008. How to Approach Machine Learning Problems. https://www.toptal.com/machine-learning/machine-learning-problems

[2] Erwan. 2021. How to impute missing text data? https://datascience.stackexchange.com/questions/99545/how-to-impute-missing-text-data

[3] 高橋涉. 2019. Name Classification with Naive Bayes. https://towardsdatascience.com/name-classification-with-naive-bayes-7c5e1415788a

[4] Sriam. 2022. Multinomial Naive Bayes Explained: Function, Advantages & Disadvantages, Applications in 2023. https://www.upgrad.com/blog/multinomial-naive-bayes-explained/#:~:text=The%20Multinomial%20Naive%20Bayes%20algorithm%20is%20a%20Bayesian%20learning%20approach,tag%20with%20the%20greatest%20chance.

[5] Rohith Gandhi. 2018. Support Vector Machine — Introduction to Machine Learning Algorithms. https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[6] Chris V. Nicholson. 2023. A Beginner's Guide to Neural Networks and Deep Learning. http://wiki.pathmind.com/neural-network