UNIVERSITY OF REGINA

FACULTY OF GRADUATE STUDIES AND RESEARCH

DEPARTMENT OF COMPUTER SCIENCE

PROJECT REPORT

Sentiment Classification on MOOC Provider Data

VINEEL GANNU (VGC361@UREGINA.CA)

KRISHNA PATEL(KAP788@UREGINA.CA)

;

Contents

1 Introduction

Nowadays, it has become normal and common to shop online and leave a review on a specific product. These reviews provide vital information by helping the other customers know about the product and helps the product providers to understand the product quality. Many customers provide different reviews on various products. Looking into every review and identifying its polarity is a tedious task. This is when the sentiment analysis comes into play. Sentiment Analysis combines different research areas such as Natural Language Processing, Data Mining, and Text Mining that helps to understand the opinions of consumer feedback and user-generated content by classifying them based on the polarity. In the later section, we will discuss how different algorithms are used in the sentiment classification and the evaluation of the results followed by the discussion on the related work. At last, the limitations of the project are discussed and we end our paper with a conclusion and appendices.

2 Problem Statement

A massive open online course (MOOC) is an online course platform where the users have free access to online courses. When compared to the traditional course materials, the MOOCs are interactive that allows the users to enhance their knowledge by participating in social media discussions. EdX, Coursera, and Udacity are few notable MOOC providers. One of the main challenges that MOOC providers face is to understand how well the users are satisfied with the courses and which courses have not satisfied the users. Although many users review the courses in the online course platform, it is tough to analyze each review. In this project, we use the Coursera user review data to build the classification models that can classify the user reviews based on the polarity. A web application is created where the members of MOOC providers input the customer review and the application outputs the polarity of the review.

2.1 Example

Coursera is an American MOOC provider. Many users review the courses in Coursera. The Coursera user review data is collected, cleaned, pre-processed, and transformed to build the model and perform sentiment analysis on the user reviews. This model can be used to mine the user reviews, thereby solving the business problem by allowing the operations research analyst to analyze the user reviews and make some informed decisions related to the courses that help in the business growth. Let us consider one of the examples of a user review.

'I like the course. Pretty concise and straight forward. Highly recommended.'

Firstly, all the characters in the user review are converted to lower-case and then the tokenization is applied where the user review is divided into tokens. Once the tokens are generated, all the stop words are identified and eliminated. Stop words are the commonly used words that do not affect the polarity of the review. At this point, the following tokens are available for the above example.

'like', 'course', 'pretty', 'concise', 'straight', 'forward', 'highly', 'recommended'

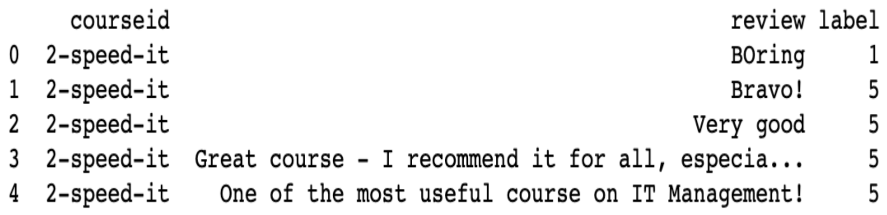
Now, stemming is applied to the tokens where the words are converted to their root form. The stemmed words are either transformed into a matrix or a sequence vector that is then fed into the machine learning model to train it.

3 Approach

The following steps are involved while implementing the project.

3.1 Data Selection

We worked on Coursera's Course Reviews Dataset that contains more than 100K records. The data source is Kaggle, which is an online community of data scientists and machine learning practitioners. The dataset has three different features - the courseid, review, and the label, of which, only review and



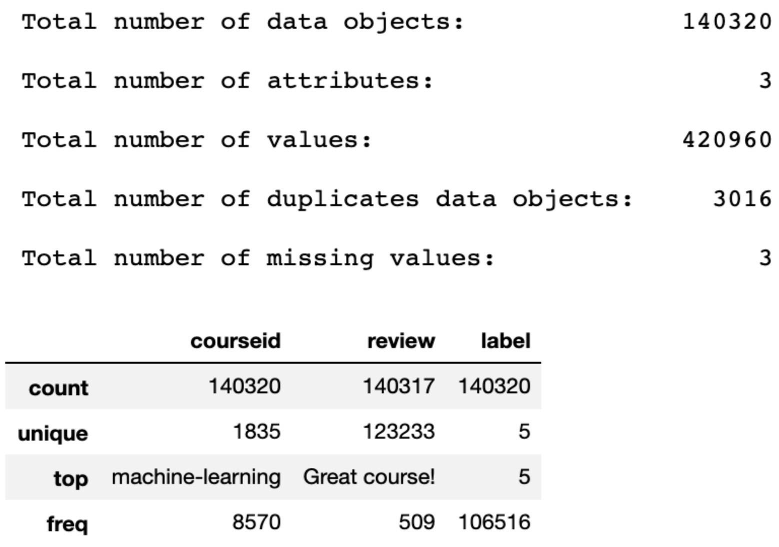
```
       courseid                                             review label
0  2-speed-it                                             BOring     1
1  2-speed-it                                             Bravo!     5
2  2-speed-it                                          Very good     5
3  2-speed-it  Great course - I recommend it for all, especia...     5
4  2-speed-it    One of the most useful course on IT Management!     5
```

Figure 1: Sample Coursera's Course Reviews Dataset

label are used in the project. There are five different class labels (1, 2, 3, 4, 5) that make our problem to be a multi-class classification problem. We saved the dataset in comma-separated values (CSV) format in the local computer whose file size is around 22MB. Figure 1 shows the sample dataset. Tools such as Jupyter Notebook, Google Colab, and Kaggle Kernel are used to load, pre-process, transform the data, and train the model.

3.2 Exploratory Data Analysis

The data that is currently available is the raw data. We need to pre-process the data to train the model. Before pre-processing the data, it is important to understand the important characteristics of the raw data. This is done in the exploratory data analysis (EDA) step. Figure 2 shows few aspects of the data. There are 140320 data objects with three different attributes of which there are 3016 duplicate data objects and 3 missing values.



```
Total number of data objects:              140320

Total number of attributes:                     3

Total number of values:                    420960

Total number of duplicates data objects:     3016

Total number of missing values:                 3
```

	courseid	review	label
count	140320	140317	140320
unique	1835	123233	5
top	machine-learning	Great course!	5
freq	8570	509	106516

Figure 2: Exploratory Data Analysis on Course Reviews Dataset

3.3 Data Cleaning

Although Exploratory Data Analysis and Data Cleaning come under Data preprocessing, they are considered to be different steps in this project. In this step, the data that falls in any of the below categories are eliminated.

- Data objects that are duplicate
- Data objects that contain missing values
- Data objects whose review is only numeric

3.4 Data Preprocessing

Once all the irrelevant data objects are removed in the above step, the data is now preprocessed. In this project, the following data preprocessing tasks are performed.

1. **Elimination of noisy data**

   In this task, all the punctuation marks, special characters are considered as noisy data and are eliminated from each review as these characters do not depict any information for sentiment classification.

2. **Elimination of Non-english reviews**

   Once the noisy data is eliminated, all the data objects whose reviews are not in the English language are discarded to keep the consistency among the reviews. When the punctuation marks and the special characters are eliminated in the above task, there are chances of reviews being blank. Such reviews are also eliminated in this task.

3. **Character case conversion**

   Once the data objects with non-english reviews are removed, the character case conversion is done where all the uppercase characters in all the reviews are converted to lowercase.

4. **Tokenization**

   In this task, all the reviews are divided into small chunks known as tokens. After tokenization, each word in the review is considered as a token.

5. **Elimination of stop words**

   Stop words are the most common words in the english language. The, Are, With, By are few of the stop words. The stop words are also eliminated as these words are not useful in classifying the sentiment of a review.

6. **Stemming**

   Once the stop words are removed, all the remaining words are stemmed, where the words are converted to their root-form. Once all the words for a given review are stemmed, they are joined back to form the review.

3.5 Data Transformation

Once the data preprocessing is done, we are left out with the reviews in which all the stop words are removed and the remaining words are stemmed. The reviews are in the form of text. Machine learning algorithms can only understand numeric data. Therefore, we need to transform the data that is in text format, into a numeric format which the machine learning algorithms can understand. The data transformation depends on the algorithm we use. In this project, the Naive Bayes Classification Algorithm and Recurrent Neural Network Algorithm are used. Although we planned to work on the Support Vector Machine (SVM) Algorithm, the higher training time for the algorithm made us choose a different algorithm.

3.5.1 Naive Bayes Classification Algorithm

Multinomial Naive Bayes Classifier is used to classify the class labels by identifying the sentiment. The preprocessed data that is obtained in the previous step is converted to a word count matrix. A word count matrix is a matrix where the data objects are in the rows and each word is considered as a column. The value in the word count matrix determines the count of a specific word that is specific to a data object. A tokenizer is used to build the internal vocabulary of the matrix based on the list of reviews. Although there is an imbalanced distribution of class labels in the dataset, we have not over-sampled or under-sampled the data as the accuracy of this data is less when compared to the original data. Thus, the data for this algorithm is neither over-sampled nor under-sampled.

3.5.2 Recurrent Neural Network Algorithm

A Recurrent neural network is an artificial neural network that can process variable length sequences of inputs. Although Recurrent neural network works with the sequential data, it can also be used to solve the classification prediction problems. The following data transformations are done when working with the recurrent neural network.

- As the neural networks cannot work on the categorical data, the one hot encoding is done on the labels so that the categorical data is converted to a form that neural networks can understand and predict better.

- The classes in the dataset are imbalanced, which means that there are majority class data objects that dominates the minority class data objects. To balance the data, an oversampling technique known as SMOTE (Synthetic Minority Oversampling Technique) is applied on the data. Figure 3 shows the distribution of class labels before and after oversampling.

- As discussed earlier, the machine learning and neural network algorithms cannot understand the text data. The text reviews or the preprocessed data is converted to the sequence of vectors.

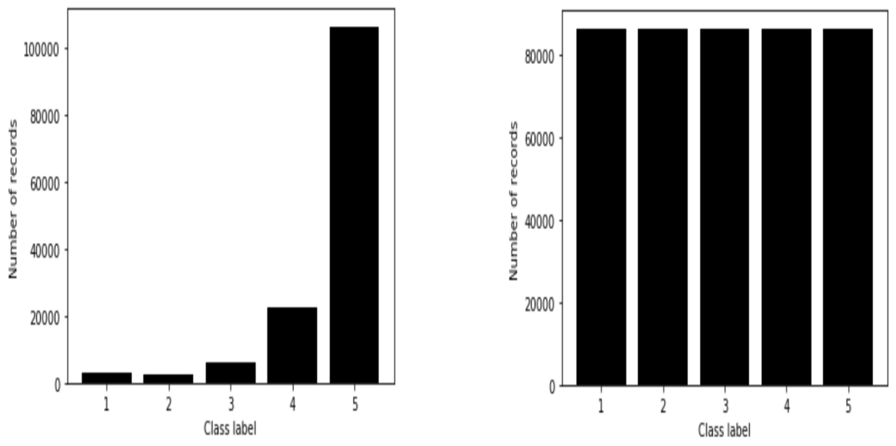


Figure 3: Distribution of class labels before and after oversampling

These vectors are padded with the maximum length so that the length of sequence vectors is consistent across different data objects.

3.6 Model Training

Once the transformed data is available, it is given as input to the model/algorithm. The transformed data is split into training and test data (in some cases, training, test and validation data) with 80% of training data and 20% of test data. The model is trained on the training data. Once the model is trained, the hyperparameters are tuned to improve the performance of the model.

3.6.1 Naive Bayes Classification Algorithm

The multinomial Naive Bayes classifier is trained on the training data. Various types of matrices such as the word count matrix, TFIDF matrix, binary matrix and frequency matrix are considered for training and at last the word count matrix is chosen. Figure 4 shows the performance metrics by matrix type where the accuracy, precision, recall and f1score for different matrices are compared. The word count matrix dominates all other matrices and thus the word count matrix was considered in the data transformation step.

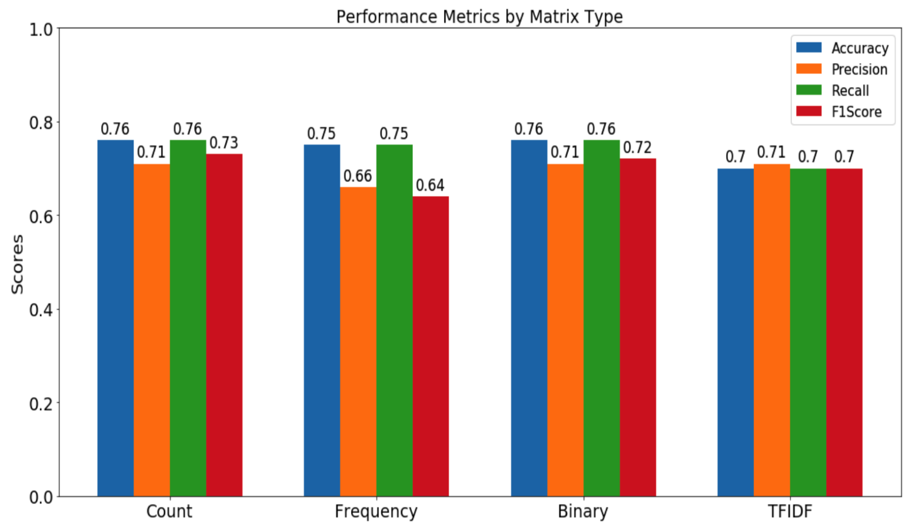


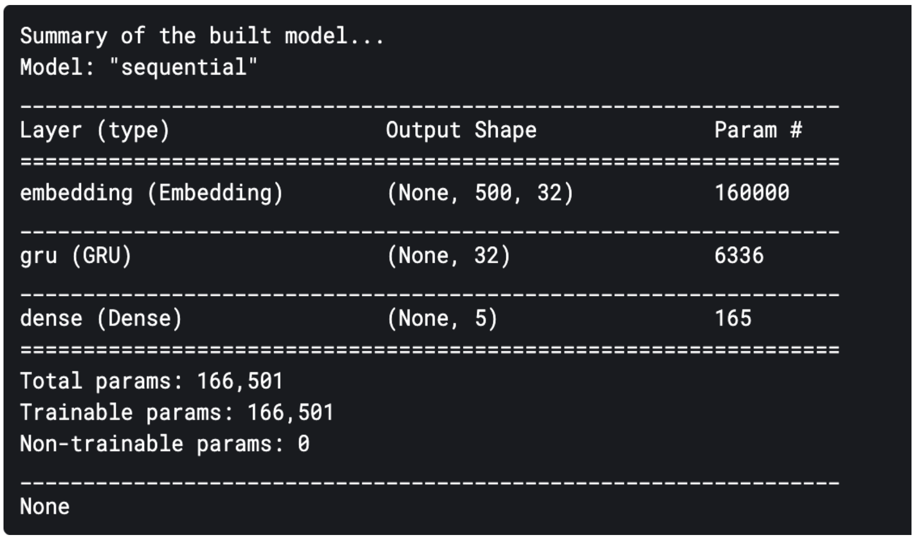
Figure 4: Performance Metrics by Matrix Type in Naive Bayes Classification

3.6.2 Recurrent Neural Network Algorithm

A Recurrent Neural Network (RNN) consists of three different layers - Embedding, Gated recurrent units, and Dense layer. The embedding layer is the first layer in the RNN. An input length is specified in the embedding layer. The gated recurrent unit (GRU) is the second layer in the RNN that has a gating mechanism. The dropout and recurrent dropout values are specified in the GRU layer that avoids overfitting. The last layer is the dense layer with a softmax activation function that normalizes the output of the network. Figure 5 shows the model summary. The hyperparameters such as the batch size, dropout percentage are tuned to improve the performance of RNN. A loss function is a function that is used to assess how good or bad the model performs the prediction. Since it is a multi-class classification problem, categorical crossentropy is used as a loss function. Adam optimizer is used to change the weights in the neural network. The model is trained on twelve epochs and figure 6 shows the output of the trained model with the accuracy and loss of both the training and validation data.

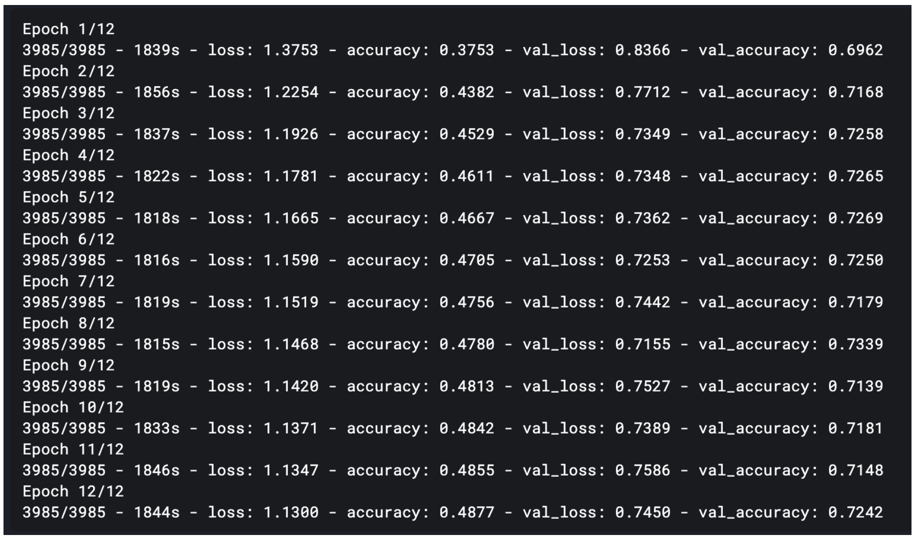
3.7 Model Evaluation

Once the models are trained using the training data, they are evaluated on the test data. The models are evaluated based on the performance metrics. The performance metrics that are suitable for the classification problem are the accuracy, precision, recall and F1score.



```
Summary of the built model...
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 500, 32)           160000
_________________________________________________________________
gru (GRU)                    (None, 32)                6336
_________________________________________________________________
dense (Dense)                (None, 5)                 165
=================================================================
Total params: 166,501
Trainable params: 166,501
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 5: Recurrent Neural Network Model Summary



```
Epoch 1/12
3985/3985 - 1839s - loss: 1.3753 - accuracy: 0.3753 - val_loss: 0.8366 - val_accuracy: 0.6962
Epoch 2/12
3985/3985 - 1856s - loss: 1.2254 - accuracy: 0.4382 - val_loss: 0.7712 - val_accuracy: 0.7168
Epoch 3/12
3985/3985 - 1837s - loss: 1.1926 - accuracy: 0.4529 - val_loss: 0.7349 - val_accuracy: 0.7258
Epoch 4/12
3985/3985 - 1822s - loss: 1.1781 - accuracy: 0.4611 - val_loss: 0.7348 - val_accuracy: 0.7265
Epoch 5/12
3985/3985 - 1818s - loss: 1.1665 - accuracy: 0.4667 - val_loss: 0.7362 - val_accuracy: 0.7269
Epoch 6/12
3985/3985 - 1816s - loss: 1.1590 - accuracy: 0.4705 - val_loss: 0.7253 - val_accuracy: 0.7250
Epoch 7/12
3985/3985 - 1819s - loss: 1.1519 - accuracy: 0.4756 - val_loss: 0.7442 - val_accuracy: 0.7179
Epoch 8/12
3985/3985 - 1815s - loss: 1.1468 - accuracy: 0.4780 - val_loss: 0.7155 - val_accuracy: 0.7339
Epoch 9/12
3985/3985 - 1819s - loss: 1.1420 - accuracy: 0.4813 - val_loss: 0.7527 - val_accuracy: 0.7139
Epoch 10/12
3985/3985 - 1833s - loss: 1.1371 - accuracy: 0.4842 - val_loss: 0.7389 - val_accuracy: 0.7181
Epoch 11/12
3985/3985 - 1846s - loss: 1.1347 - accuracy: 0.4855 - val_loss: 0.7586 - val_accuracy: 0.7148
Epoch 12/12
3985/3985 - 1844s - loss: 1.1300 - accuracy: 0.4877 - val_loss: 0.7450 - val_accuracy: 0.7242
```

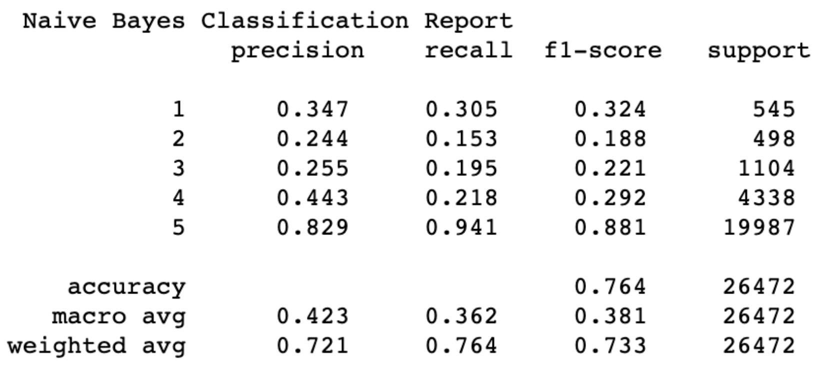
Figure 6: Recurrent Neural Network Trained Model Output

3.7.1 Naive Bayes Classification Algorithm

A classification report shows the main classification metrics for a given classification algorithm. Figure 7 shows the classification report of Naive Bayes Classification Algorithm where the precision, recall, f1score and support are calculated specific to a class label. The report also shows the macro and weighted average of different performance metrics.

3.7.2 Recurrent Neural Network Algorithm

Once the neural network is trained, it is evaluated on the test data and figure 8 shows the same where the accuracy and loss values for the test data is calculated.

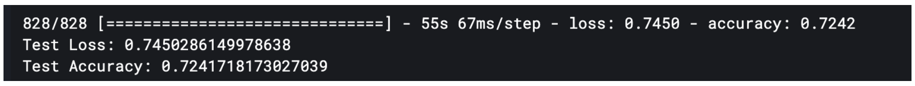


```
Naive Bayes Classification Report
              precision    recall  f1-score   support

           1      0.347     0.305     0.324       545
           2      0.244     0.153     0.188       498
           3      0.255     0.195     0.221      1104
           4      0.443     0.218     0.292      4338
           5      0.829     0.941     0.881     19987

    accuracy                          0.764     26472
   macro avg      0.423     0.362     0.381     26472
weighted avg      0.721     0.764     0.733     26472
```

Figure 7: Classification Report of the Naive Bayes Classification Algorithm



```
828/828 [==============================] - 55s 67ms/step - loss: 0.7450 - accuracy: 0.7242
Test Loss: 0.7450286149978638
Test Accuracy: 0.7241718173027039
```

Figure 8: Neural Network Model Evaluation on the Test Data

It is also important to compare the training and validation accuracy and loss for different epochs as it helps to understand how these metrics vary with respect to the number of epochs. Figure 9 shows the comparison of accuracy and loss on epoch in the Neural Network Algorithm.

3.8 Web Application

So far, two different algorithms - Naive Bayes Classification Algorithm and Recurrent Neural Network Algorithm are trained and evaluated. The web application takes a review as input, uses one of the trained algorithms on the back end to predict the probabilities of different class labels specific to review, and outputs the predicted values. The recurrent neural network algorithm is chosen over the

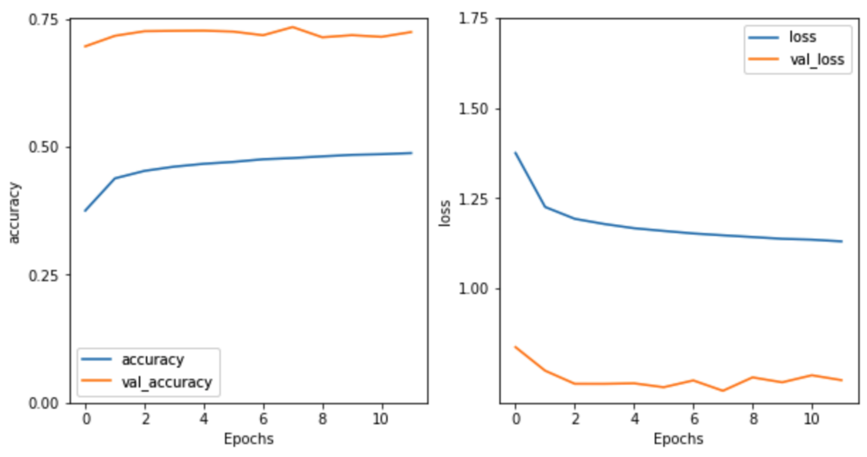


Figure 9: Comparison of Accuracy and Loss on Epoch in the Neural Network Algorithm

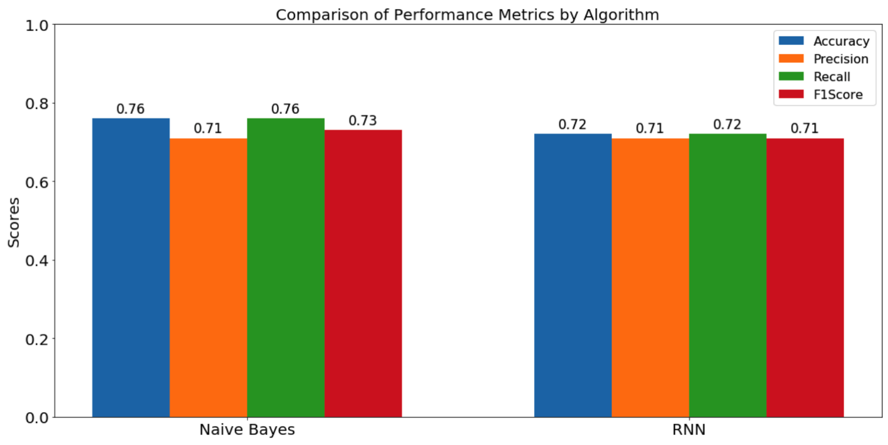


Figure 10: Comparison of performance metrics of Naive Bayes and Neural Network algorithms

Naive Bayes Algorithm because the recurrent neural network can predict the probability of different class labels specific to a review whereas the Naive Bayes Algorithm predicts the class label itself. A python web framework known as Flask is used to create a web application. Flask helps the HTML code to interact with the python code and runs the python code in the back end. HTML and CSS are used to build the user interface for the web application. The users of the web application are the members of Coursera. The web application is deployed on the Google Cloud Platform (GCP) and can be accessed with the below URL.

https://eng-voice-297302.nn.r.appspot.com/

4 Results

Once the models are trained, the performance metrics of the algorithms are compared. Figure 10 shows the comparison of different performance metrics of Naive Bayes and Neural Network algorithms. Although the output type of these algorithms vary(Naive Bayes outputs the class label whereas the Recurrent Neural Network outputs the probability of each class label), the performance metric values of Naive Bayes algorithm are slightly higher than the Neural network algorithm.

5 Discussion of Related Work

There are researchers who researched on the imbalanced class handling and multi-class classification. Yoga Pristyanto, Irfan Pratama, Anggit Ferdita Nugraha discussed about the imbalanced class handling on educational data mining multi-class Classification. They combined the Synthetic Majority Oversampling Technique (SMOTE) and One-sided Selection technique (OSS) to handle the imbalanced classes on multi-class on educational data mining (EDM). The results of their experiments show that the performance of the Support Vector Machine (SVM) classification method is enhanced. [1]

Houria Madani and team classified the ransomware into 9 different labels using artificial neural networks and Bayesian networks. These classification techniques are then compared with the old classification techniques which showed that there is an improvement in the results. [2]

Muhammad Zain Amin and Noman Nadeem used convolutional neural network text classifiers for Question Answering Systems (QAS). The neural network was developed on larger and two different datasets. Further, the implemented convolutional neural network was integrated with the question answering system. The neural network model was trained on the word embedding and softmax was used to calculate loss. [3]

Siwei Lai, Liheng Xu, Kang Liu and Jun Zhao performed text classification using recurrent convolutional neural networks which captures the contextual information and constructs the text representation using convolutional neural network. This model outperforms the start-of-the-art methods and works well on the document-level datasets. [4]

Vanaja and M. Belwal performed Sentiment Analysis on Amazon customer review data which focuses on tasks such as finding aspect terms from each review, identifying parts-of-speech, applying classification algorithm. Sentiment analysis are broadly classified into three different levels. They compared the Naive Bayes classification with the Support Vector Machine algorithm and found that the Naive Bayes Algorithm performed better. [5]

After looking into these previous works, we drew few ideas from various papers and came up with this approach.

6 Challenges

One of the challenges faced in this project is the unavailability of devices that have high processing capabilities. As an alternative, we used online platforms like Kaggle Kernel and Google Colab that can train the algorithms remotely. Although these platforms have access to Graphics Processing Unit(GPU) and Tensor Processing Unit(TPU), the algorithms can be trained only for a limited time.

7 Limitations and Possible Extensions

Since the systems with high processing GPU and TPU are not available, the Recurrent Neural Network Algorithm is trained on less number of epochs. Training the algorithm on a large number of epochs will increase its performance and can make accurate predictions.

As discussed, the Naive Bayes Algorithm and the Recurrent Neural Network algorithms are trained in this project. The project can be extended by considering the classification algorithms such as Random Forest, Support Vector Machine(SVM), or a combination of different algorithms, and the performance of these algorithms can be evaluated. The sentiment classification is done specific to the reviews. This can be extended by considering the courseids and the sentiment can be evaluated specific to a course.

8 Conclusions

To sum up, we have taken the Course reviews of Coursera and trained two different algorithms whose performance is evaluated and the overall sentiment of different reviews is identified. With the knowledge that is discovered from the trained models, the operations research analyst of Coursera can make informed decisions regarding the courses that are accessible to the Coursera users. It also gives an idea of how well the users are satisfied with the courses. A web application is created in which one of the algorithms is used to predict the probability of a review belonging to different class labels. This helps the members of MOOC to input the review and identify the predicted sentiment of the review.

9 References

[1] Y. Pristyanto, I. Pratama and A. F. Nugraha, "Data level approach for imbalanced class handling on educational data mining multiclass classification," 2018 International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, 2018, pp. 310-314, doi: 10.1109/ICOIACT.2018.8350792.

[2] H. Madani, N. Ouerdi, A. Palisse, J. -L. Lanet and A. Azizi, "Classification of ransomwaresusing Artificial Neural Networks and Bayesian Networks," 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS), Marrakech, Morocco, 2019, pp. 1-6, doi: 10.1109/ICDS47004.2019.8942294.

[3] Amin, Muhammad and Nadeem, Noman. (2018). Convolutional Neural Network: Text Classification Model for Open Domain Question Answering System.

[4] Recurrent convolutional neural networks for text classification –Siwei Lai, Liheng Xu, Kang Liu, Jun Zhao — 2015 — In Proc. Conference of the Association for the Advancement of Artificial Intelligence (AAAI)

[5] Vanaja and M. Belwal, "Aspect-Level Sentiment Analysis on E-Commerce Data," 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, 2018, pp. 1275-1279, doi: 10.1109/ICIRCA.2018.8597286.

[6] I. K. C. U. Perera and H. A. Caldera, "Aspect based opinion mining on restaurant reviews," 2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA), Beijing, 2017, pp. 542-546, doi: 10.1109/CIAPP.2017.8167276.

[7] Jan Charles Maghirang Adona, 2017 May, 100k Coursera's Course Reviews Dataset, Version 2. Retrieved in 2020 October from https://www.kaggle.com/septa97/100k-courseras-course-reviews-dataset.

[8] https://flask.palletsprojects.com/en/1.1.x/

[9] https://scikit-learn.org/stable/supervised_learning.html#supervised-learning

[10] https://www.kaggle.com/kernels

[11] https://colab.research.google.com/notebooks/

A User Manual

The members of Coursera MOOC are the users of the web application. The following steps guides the users on how to use the web application.

- Firstly, click the below link.

  https://eng-voice-297302.nn.r.appspot.com/

- Once the web application is loaded in the browser, enter a customer review.

- Once the review is entered, click the Predict button.

- Once the Predict button is clicked, a graph corresponding to the sentiment of the given review for different class labels is displayed along the probability of different class labels.

- A different review can be entered and the process is repeated.

- Once the task is done, close the browser window corresponding to the web application.

B Implementation Manual

After looking into the various business problems, we chose the problem of the Coursera MOOC provider and solved it using sentiment classification where two different algorithms were trained and evaluated. Before training the algorithms, the Coursera customer review data is cleaned, pre-processed, and transformed. The data transformation is dependent on the algorithm we use. Since we used two different algorithms, two different data transformation techniques (Word count matrix and Sequence vector) are used. One of the algorithms is used in the web application where the application takes the review as input and predicts the probability of class distribution of the input review. Figure 11 shows the project implementation overview. The link to the web application is available in the User Manual section that also includes the steps on how to use the web application.

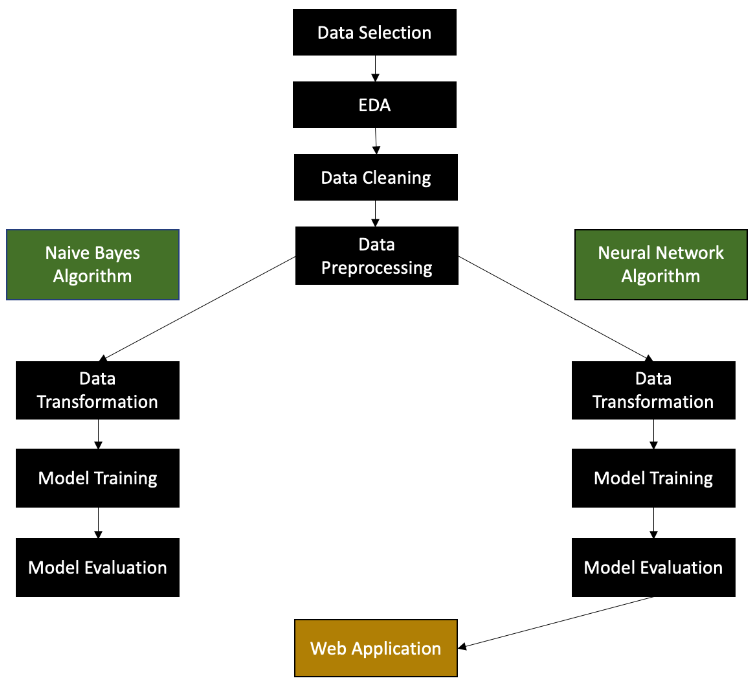
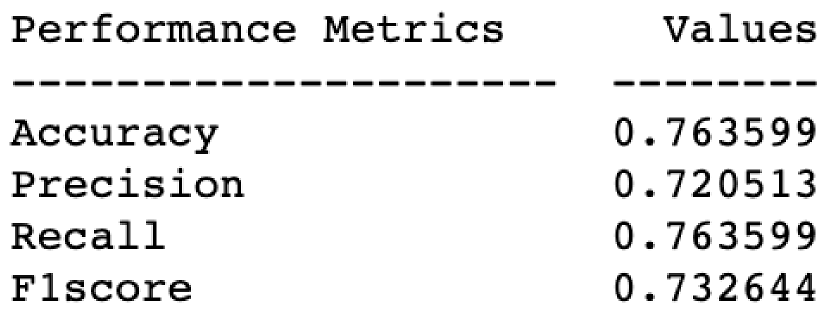


Figure 11: Implementation Overview

C Experimental Results

C.1 Naive Bayes Classification Algorithm

Figure 12 shows the performance metrics of the Naive Bayes Classification Algorithm. Although the accuracy is 76.3%, since the data is imbalanced, it is not suggested to evaluate the performance of the Naive Bayes Algorithm based on the accuracy alone. The confusion matrix describes the performance of a classifier based on the actual and predicted values. Figure 13 shows the normalized confusion matrix of the Naive Bayes Classification Algorithm.



```
Performance Metrics      Values
---------------------  --------
Accuracy               0.763599
Precision              0.720513
Recall                 0.763599
F1score                0.732644
```

Figure 12: Performance Metrics of the Naive Bayes Classification Algorithm

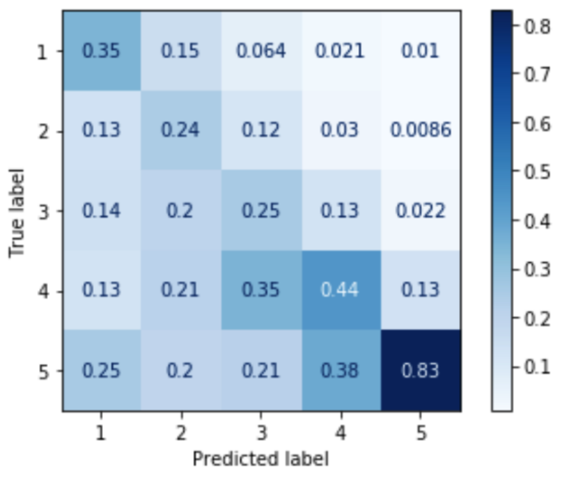
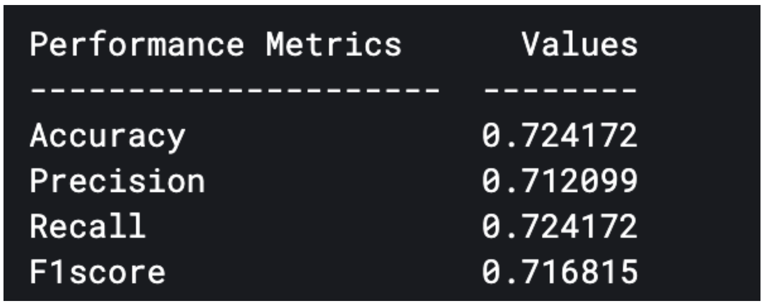


Figure 13: Normalized Confusion Matrix of the Naive Bayes Classification Algorithm

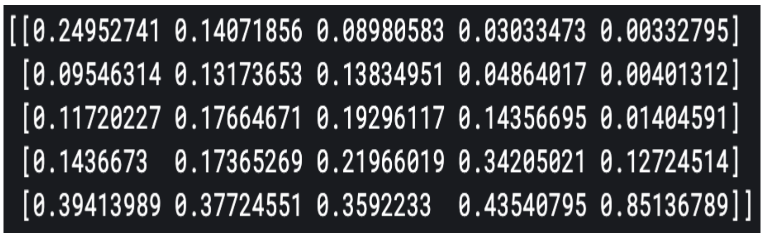
C.2 Recurrent Neural Network Algorithm

Figure 14 shows the performance metrics of the Recurrent Neural Network Algorithm. Metrics such as accuracy, precision, recall and F1score are used to assess the performance of the Recurrent Neural Network Algorithm. Figure 15 shows the normalized confusion matrix of the Recurrent Neural Network Algorithm.



```
Performance Metrics      Values
---------------------  --------
Accuracy               0.724172
Precision              0.712099
Recall                 0.724172
F1score                0.716815
```

Figure 14: Performance Metrics of the Recurrent Neural Network Algorithm



```
[[0.24952741 0.14071856 0.08980583 0.03033473 0.00332795]
 [0.09546314 0.13173653 0.13834951 0.04864017 0.00401312]
 [0.11720227 0.17664671 0.19296117 0.14356695 0.01404591]
 [0.1436673  0.17365269 0.21966019 0.34205021 0.12724514]
 [0.39413989 0.37724551 0.3592233  0.43540795 0.85136789]]
```

Figure 15: Normalized Confusion Matrix of the Recurrent Neural Network Algorithm