Machine Learning

Faculty: Science and Engineering

School: Computing and Information Science

Academic Year: 2020/21

Trimester: 2

[Word count excluding references and table of contents: 1690.]

Table of Contents

[1 Introduction 2](#_Toc100305765)

[2 Description of the Dataset 3](#_Toc100305766)

[3 Description of the Learning Task 4](#_Toc100305767)

[4 Preparing Data 5](#_Toc100305768)

[4.1 Data pre-processing 5](#_Toc100305769)

[4.2 Train-Test Split and Vectorization. 6](#_Toc100305770)

[5 Training and Evaluating Models 9](#_Toc100305771)

[5.1 Multinomial Naïve Bayes Model. 9](#_Toc100305772)

[5.2 Logistic Regression Model. 10](#_Toc100305773)

[5.3 LinearSVC Model. 11](#_Toc100305774)

[6 Evaluation Analysis 13](#_Toc100305775)

[7. References 14](#_Toc100305776)

# Introduction

The focus and purpose of this report is to implement an end-to-end machine learning pipeline. The project aims to fabricate a News Category Classifier that takes input of headline and short description and outputs the news category. Furthermore, the model reflects the working of various machine learning models and text vectorization techniques.

# Description of the Dataset

The given dataset is an excel sheet (Headlines.xslx) that comprises of more than 200,000 news records categorized onto 41 different categories. Each news record consists various attributes (columns) that are used for the prediction.

Graphical user interface, application

Description automatically generated Fig. 1. The head of the dataset representing the attributes.

From visual inspection of all the attributes (Fig. 1), we can draw a conclusion that the attributes ‘Headline’, ‘Short description’ can be given as an input to train the model and allow the model to predict ‘category’ of the news. The attribute of ‘authors’ hasn’t been used as it is not necessary for every news description to have the authors name specified.

Major composition of the dataset is comprised of ‘Politics’ category while ‘Arts’ has the least composition. The composition can clearly be seen from the pie chart (Fig.2).

Chart, pie chart

Description automatically generated

Fig.2 . Pie Chart representing the composition of categories.

# Description of the Learning Task

Classification and Regression are the two most common methods of Supervised Machine Learning Model. Classification is the task of approximating a mapping function(f) from input variables (X) to output variables (y), whereas on the other hand, Regression is the task of approximating function (f) from input variables (X) to continuous output variable (y) [1]. Our given Headlines dataset has an input variable (i.e. the headlines and short description) and also an output variable (i.e. the category of the dataset). We use a Machine Learning algorithm as the mapping function to our input and output variables. Hence, we can clearly state that our model is a classification and not a regression. Since our dataset has multiple labels, our classification is further sorted as Multilabel classification.

# 4 Preparing Data

## 4.1 Data pre-processing

Data pre-processing is the task of converting or making changes to the data from a given form to much more usable form [2]. The given dataset has been categorised into three different categories one with headlines, one just with the short description and the other one a concatenated set of both headline and short description ([All\_text] column) (Fig.3).

Graphical user interface, text, application

Description automatically generated

Fig.3. Showing the three different categories the dataset has been classified.

The headlines and short\_description have been concatenated and not zipped into a list so that they match the dimensions for fitting into the Machine Learning models. They have also been converted to string (using .astype(str)) so that we can apply the vectorization techniques. These datasets have few null values for which, if not removed, there might be an impact on the accuracy. Hence the rows containing the null values have been removed using dropna() function.

Text classification is mainly based on feature extraction. But before proceeding into the feature extraction, the process of text processing yields different performance results when applied correctly. The text processing includes various industry related techniques such as tokenizing, Stemming, Lemmatization, removing stop words and removing punctuation.

Table

Description automatically generated

Once all the pre-processing has been completed, our dataset is now ready to be split into training and testing datasets. All these functions can be applied using an lambda function so that they are applied to every row of the our datasets.

## 4.2 Train-Test Split and Vectorization.

One of the key aspects of supervised machine learning model is evaluation and validation [4]. When a model is evaluated, the predictive performance of the model should be unbiased [4]. train\_test\_split() from sci-kit learn allows to split the dataset to reduce potential bias. Our three datasets. headlines\_column, short\_description\_column and the All\_text can be split using the train\_test\_split with the training ratio being 0.67 (67 percent to train and 33 percent to test). Consequently, the labels (category column of our dataset) are also split, thus creating four variables per dataset x\_train, y\_train, x\_test and y\_test (Fig. 4). Moreover, the datasets gets split randomly into train and test sets so that accuracy of the classifiers can be determined.

Graphical user interface, text, application

Description automatically generated

Fig.4. splitting the datasets for training and testing.

After the text is cleaned, normalized, and split, the most trivial step, Feature Extraction comes into place. Feature extraction converts the normalized text into their respective features that are used for modelling. Some common feature extraction techniques used are TF-IDF vectorization, Count Vectorization, Hash Vectorization, Word2Vec, GloVe [5]. Out of these techniques, we use TF-IDF, count and hashing vectorizers as they work well with the supervised machine learning models that are to be used. Each technique uses different methods to obtain the features. The vectorization techniques are the core to extract the main features and play crucial role in prediction performance.

Table

Description automatically generated

# Training and Evaluating Models

Once the vectorization has been completed, the next main and the final step is to train our supervised machine learning models with the numerical data that we have. One main observation to be made is that when the vectorization has been done, the number of features has been reduced. This impacts the accuracy of various models that are trained. It is never possible to determine how many features are required for the model to give the maximum accuracy.

The models used for training are Multinomial Naïve Bayes, Logistic Regression and LinearSVC. All these models are supervised classification models that are known to give high accurate predictions and also other high industry metrics.

## 5.1 Multinomial Naïve Bayes Model.

Fig.5. showing the working of MultinomialNB from bayes rule.

Diagram

Description automatically generated

Originating from the roots of Bayes rule (Fig.5), MultinomialNB classifier is specially know for classification of discrete features [8]. Naïve bayes works on the assumption that every feature is independent of another. The highest accuracy obtained was by using a count vectorizer for the All\_text data. This can be justified by the fact that MultinomialNB is designed originally to take the input values of integer. Even though it accepts partial values of tf-idf vectorizer, the algorithm makes it difficult to process the partial values. Hence, we can clearly see that the least accuracy has been obtained when a tf-idf vectorizer has been used with the smallest dataset (headline\_column) (Fig.6).

Chart, bar chart

Description automatically generated

Fig.6. Showing the comparison of accuracy scores when MultinomialNB model is used with different datasets and vectorizers.

With top accuracy of around 60 percent, Multinomial naïve Bayes performs good, but few other models such as logistic regression tend to show much higher accuracy. The main reason why Multinomial is preferred is due to the fact that it takes much less time than other models to train and make predictions using this model.

## 5.2 Logistic Regression Model.

Unlike what the name suggests, Logistic Regression model can be used both for regression as well as classification. Logistic regression classifier uses the ‘Sigmoid’ function (Fig.10) to classify through categories.

Chart, line chart

Description automatically generated

Fig.10. Showing Logistic regression classifier working based on the sigmoid function.

Logistic regression is specially known for binary classification (i.e. classification into either one category or another), but when given with a special extension call ‘one-vs-rest’ (OvR), Logistic regression can be used for multiclass predictions [9]. Model accuracy scores show that Logistic regression works well both with tf-idf and count vectorizer (Fig.11). Logistic regression gives less accuracy when used with hash vectorizer loses more features when the negative values have been removed.

Chart, bar chart

Description automatically generated

Fig.11. Comparison of accuracy scores for logistic regression model, highest accuracy scores being tf-idf monogram vectorizer with All\_text data. Also Count vectorizer with same data gets nearly the same.

Feature extraction made from vectorizers using bigram shows less accuracy as the pair of unique two words makes it harder for logistic regression to apply multiclass classification using OvR.

Accuracies greater than 60 percent have been recorded, beating MultinomialNB by some margin. The major drawback with Logistic Regression is the time taken for it to make the classification. Logistic regression tries to convert the prediction into variable binary classification problems, hence, the runtime is more than other models.

## 5.3 LinearSVC Model.

The final model used for the classification is the LinearSVC model. LinearSVC model works based on hyperplanes. Based upon the data that we input, it returns a best fitting hyperplane that categorises our data. It uses supporting vectors and find the hyperplane with maximum margin width (Fig.12).

Chart, scatter chart

Description automatically generated

Fig.12. Showing Support vector classification by finding the optimal hyperplane.

Support vectors are the points in our dataset that are close to the hyperplane and have impact and influence the hyperplane [10].

Chart

Description automatically generated

Fig.13. Showing the comparison of accuracy scores by different models using LinearSVC

Comparing to the previous two models, LinearSVC has performed better with all types of vectorizers and datasets. Like Logistic Regression, even LinearSVC tries to convert multiclass prediction into multiple binary class predictions, but it has surpassed the accuracy obtained by Logistic Regression and also the time taken for the prediction is comparatively much better than Logistic Regression model.

# Evaluation Analysis

Multiclass prediction is always a difficult task, especially when it comes to large datasets (likes the headlines dataset given). Even if the features are low, the diversity of the classes (41 classes) makes it even more difficult for Machine Learning algorithms to have good prediction performance.

Coming to the comparison of the vectorization techniques, we can clearly justify that tf-idf monogram vectorization has done a better job than the other three vectorizations used. Predictions made using bi-grams always tend to show less accuracy, this can be explained based on the number of features obtained after the vectorization process. The features obtained for Tf-idf monogram, Count Uni and bi-gram and even the hashing vectorizer have been significantly more than those obtained using the Tf-idf bigram vectorizer.

Of the three models that were used to make the classification, LinearSVC performed better than the other two models. The dataset used also made an impact on their predictive performances. Models trained with vectorised All\_text data (the concatenated set of headlines and short\_description) showed relatively better performances than the other two datasets used. All the top accuracies obtained from various models were with tf-idf vectorizer using the All\_text data. This allows us to conclude that tf-idf vectorizer is better in feature extraction when used with supervised machine learning models.

Although various models have given different accuracy scores, it hard to come to a conclusion for determining the best machine learning model. It all depends on the feature extraction techniques and mainly due to the structure of the given dataset. Therefore, using the specific model based on the feature extraction technique used yields more accuracy rather electing one model to be the best. [11]

Fig.14. Showing the comparison of accuracy scores of all the models.

Fig.15. Showing the comparison of accuracy scores of top models (models with highest accuracy)

A picture containing chart

Description automatically generated

Fig.14.

Chart, bar chart

Description automatically generated

Fig.15.

# 7. References

[1] Jason Brownlee, 2017, Difference between Classification and Regression.

<https://machinelearningmastery.com/classification-versus-regression-in-machine-learning/>

[2] geeksforgeeks, 2021, Understanding data pre-processing.

<https://www.geeksforgeeks.org/ml-understanding-data-processing/>

[3] Olga Fugs, 2018, Classification of News Dataset. Stanford Education.

<https://cs229.stanford.edu/proj2018/report/183.pdf>

[4] Mirko Stojiljkovic, 2018, Split you data using train\_test\_split(). Real Python.

<https://realpython.com/train-test-split-python-data/>

[5] Wikipedia, 2010, Feature Extraction.

<https://en.wikipedia.org/wiki/Feature_extraction>

[6] Koushik Kumar, 2021, NLP- Bag of words and TF-IDF explained!

<https://koushik1102.medium.com/nlp-bag-of-words-and-tf-idf-explained-fd1f49dce7c4>

[7] scikit learn, 2021, sklearn.feature\_extraction.text.HashingVectorizer.

<https://scikitlearn.org/stable/modules/generated/sklearn.feature_extraction.text.HashingVectorizer.html>

[8] Synced, 2017, Applying Multinomial Naïve bayes to NLP problems.

<https://medium.com/syncedreview/applying-multinomial-naive-bayes-to-nlp-problems-a-practical-explanation-4f5271768ebf>

[9] Jason Brownlee, 2021, Multinomial Logistic Regression with NLP.

<https://machinelearningmastery.com/multinomial-logistic-regression-with-python/>

[10] Rohith Gandhi, 2018, Support Vector Machine – An introduction to ML Algorithms.

<https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>

[11] U Suleymanov, 2018, Automated news categorization using ML techniques.

<https://aegeanconference.org>