ASSIGNMENT 3 ON "DATA SCIENCE GROUP WORK"		
Student's Code	AIMS African Institute for Mathematical Sciences CAMEROON	Deadline
Group2: Gibbs NWEMADJI		**03.04.21, 8:30 am**
November 14, 2021		Ac. Year: 2020 - 2021
Lecturer(s): "Dr. Bubacarr Bah"		

1 Introduction

Drugs usage have either a positive or negative impact on patients. Patients ratings and reviews of prescribed drugs are very important in the pharmaceutical field and to all health practitioners to help ease their workload.Patients ratings to drugs are essential in predicting if a drug is good or not.In a clinical trial,drugs which are clinically significant are rated to be good and are brought to the market for sale. When choosing drugs for a patient's conditions,medical practitioners use experience and clinical trials in administering drugs to their patients.These experiences are from the feedbacks or ratings of patients on the prescribed drugs based on their condition.

2 Problem Definition and Algorithm

This project is a text classification problem ,where we are predicting a three multi-class classification.

Aims

1 To predict how good a prescribed drug is on the scale of 1 to 3 in relation to a particular health condition given "drugname,health condition,reviews and useful-count" where 0 is rated as a bad,2 as good and 3 is very good.

2 To build a machine learning algorithm to do this prediction using deep learning algorithm(Artificial Neural Network)

2 To apply an appropriate traditional ML algorithm for this prediction.

3 To compare the traditional and deep learning algorithm(Artificial Neural Network) based on the predictions.

2.1 Datasets

The dataset is about drugs prescriptions.The data basically talks about some drug prescriptions medical practitioners prescribed for their patients based on their health conditions or complaints about their health and their reviews and 10 star patients' ratings after the usage of the drug.The data is called Drug Review Dataset (Drugs.com) uploaded by Surya Kallumadi from the Kansas State University,USA.It is a multivariate and text data from the *UCI Machine Learning Repository*. Our dataset consist of **161297 rows and 7 columns** which contain 3 numerical variables and 4 categorical variables. Our features are 'drugName', 'condition' 'review', 'usefulCount' and our target is **'rating'**.

Description

1. drugName (categorical): name of drug
2. condition (categorical): name of condition
3. review (text): patient review
4. rating (numerical): 10 star patient rating
5. date (date): date of review entry
6. usefulCount (numerical): number of users who found review useful.

2.2 Algorithm Definition

Generally in Machine Learning problem, we do not know the exact algorithm or model which can make a good prediction of our outputs for given features. As data scientist, we are faced with the problem of automatically knowing which algorithm is the best off-head.We have to try different algorithms based on the problem being it classification or regression and fine-tuned some hyper-parameters to get good models.Diving deep on the understanding of writing good models for predictions brought the concept of **Deep Learning.**
There are numerous classical models that are used in predictions but here are some classical models used in predicting our multi-class classification problem.
K-Nearest Neighbour algorithm The K Nearest Neighbour algorithm allows us to solve multi-class classification problems in a simple and very efficient way. K-NN is a **supervised learning algorithms** meaning learning a function that maps an input to an output based on example input-output pairs. Our goal is to discover a function $h : X \to Y$ so that having an unknown observation $\mathbf{x}$, $\mathbf{h(x)}$ can positively predict the identical output $\mathbf{y}$.
K-nearest neighbor algorithm works as, for a given value of K,the algorithm will find the K nearest neighbor of unseen data point and then it will assign the class to unseen data point by having the class which has the highest number of data points out of all classes of **K neighbors**. Since we are dealing the neighborhood of the points, this can be well characterized by metric distance , by using Euclidean metric.
The Euclidean metric is given by:

$$d(x, x') = \sqrt{(x_1 - x_1')^2 + (x_2 - x_2')^2 + ... + (x_n - x_n')^2}$$

Finally, the input x gets assigned to the class with the largest probability defined as:

$$P\left(y = j | X = x\right) = \frac{1}{K} \sum_{i \in \mathcal{A}} I\left(y^{(i)} = j\right) \tag{1}$$

3 Experimental Evaluation

3.1 Methodology

3.2 Data pre-processing

- Data importation:
  All the necessary python libraries such pandas,english_words were all imported.We then imported the dataset.

- Exploring data:
  To know more about the data we explore the our data by using commands like **info() , head(), hist()...** to get a feel of the data set. Our data contained a total of 3436 different

drug names, 884 different 'conditions', $1-10$ 'ratings' and 112329 different 'reviews' and 899 missing values found at the feature 'condition', **birth control**is the predominant condition, **levolorgestrel** has the highest frequency.From exploring the data,we saw as shown below that there is 1% missing data on condition and no missing values for the other features.

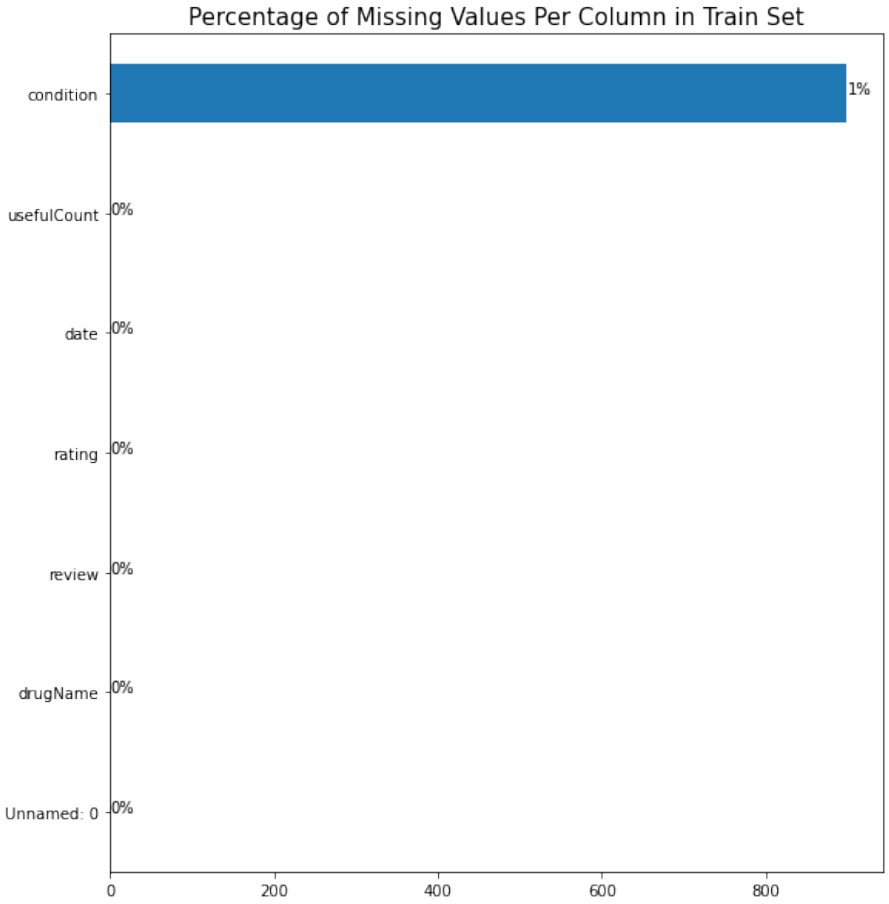

Figure 1: missing data

- Feature engineering:
  From domain knowledge we know features like Unnamed and date are not really important in predicting if a drug is very good,good or bad so we dropped them.On the issue of missing values we dropped 889 rows which counts for 1% condition since drooping would not affect our results. –Next is on the reviews.The reviews were comments of people.We extracted the vocabulary from all the reviews. After that we deleted all the words with numerical character like '54mg45'. The next step was to delete in that list all the word which are not the English words. We do this by using a list of all english word inside the library of python. With the new list of word we download from **Github** a list of word which contains all the English adjectives. That new operation reduce our list of word from around 9000 to 703. After do that we wanted to give a mark of each word then we also found on **Github** two list of word which express the positive sentiment and the negative sentiment. Using our precedent list of word we create 703 hundreds new with initially '1' in the word express a positive sentiment , -1 if it express a negative sentiment and zero it is not between the precedent list. Because of limited time we didn't use all the 700 words but only 300 of them. Using those 300 features for a particular reviews the mark of a particular adjective will be the same initially if this article contains that word and it was be replace by zero if this review don't contains that word. Now to have the

mark of that review, we had add all the 300 columns features . At the end I had cancel those 300 hundred features and also the feature containing all the review it remain only the features name 'mark' which contains the mark of each review.

We then extracted and filtered the words contained in 'review'. using the **github** vocabulary contained on-line After filtering, these words were grouped by assigning 1 to positive adjectives, -1 to negative adjectives, and 0 to others. This allowed us to create a new 'Mark' column.

–Finally, we proceeded to the encoding using *Binary-encoder* (categorical variables) which was 'Drugname' and 'condition'. We use the standardization in the variable name 'usefulcount' and 'mark' by using the formula $x = \frac{x - mean}{standart\ \ deviation}$. Also, the target variable 'rating' has been transformed into three ranks: To the numbers from 0 to 4 we have assigned 0 which means bad, to the numbers from 5 to 7 we have assigned 1 which means good, and to those from 8 to 10 we have assigned 2 which means very good.

3.3 Algorithm Building

Random Forest

Traditional Machine Learning Algorithm: The classical algorithm used is the multinomial naives Bayes since we are dealing with a classification problem.Before we dive in to build the model,we import the MultinomialNB from sklearn, train the model using "MNB.fit()",predict the test data and compute the accuracy and confusion matrix of the model.
Another classical model is built using k-NN algorithm.We import KNeighborsClassifier from sklearn.neighbors and we train the model with our datasets, make predictions and calculate our accuracy and confusion matrix.

Artificial Neural Network: Before we build our model we have to know our input dimension of the data.The optimizers,loss functions,activation functions are all important in the model building.We are using the "softmax" function for the output layer because of our multi-class classification problem.The optimizer is used to determine the error or loss between the computed and desired output.The loss function is used to compute the error and it is reduced by the optimizer.
We then import models and layers from Keras to help build the model.We then instantiate the model using "models.Sequential " to make the model ready to be trained.Our model has 7 layers with 64 nodes in the 6 first layers and 3 nodes in the output layers. After that we compile our model with 'adam' for optimizer, 'categorical_crossentropy' for loss and our metrics is 'accuracy'. The trained data is furthered split into train and validation data sets in order to increase the accuracy of the model.The number of epochs=20,batch size=100 with the validation test being 20%. our model and then the test the on the test datasets.we compute the loss and the accuracy of the model.Some hyper-parameters like epochs,batch sizes ,optimizer, activation function, loss function are tuned to during the training to get a good model.

3.4 Results

Random Forest Model Precision Interpretation: 44%, 19% and 71% for 0,1,2 ratings respectively are the correctly predicted cases which turns to be positive.

Recall Interpretation: 32%, 8% and 72% for 0,1,2 ratings are the actual positive cases that the model predicted correctly.

	precision	recall	f1-score	support
0	0.44	0.32	0.37	7870
1	0.19	0.08	0.12	4795
2	0.71	0.72	0.72	19415
micro avg	0.61	0.53	0.57	32080
macro avg	0.45	0.37	0.40	32080
weighted avg	0.57	0.53	0.54	32080
samples avg	0.53	0.53	0.53	32080

Figure 2

KNN Model

```
[[ 3929   640  3301]
 [ 1847   414  2534]
 [ 4392  1011 14012]]
              precision    recall  f1-score   support

           0       0.39      0.50      0.44      7870
           1       0.20      0.09      0.12      4795
           2       0.71      0.72      0.71     19415

    accuracy                           0.57     32080
   macro avg       0.43      0.44      0.42     32080
weighted avg       0.55      0.57      0.56     32080
```

Figure 3

Accuracy Interpretation : The accuracy is 57% for the model.
Precision Interpretation: 39%, 20% and 71% for 0,1,2 ratings respectively are the correctly predicted cases which turns to be positive.
Recall Interpretation: 50%, 9% and 71% for 0,1,2 ratings are the actual positive cases that the model predicted correctly.

Artificial Neural Network

```
              precision    recall  f1-score   support

           0       0.46      0.44      0.45      7870
           1       0.11      0.00      0.00      4795
           2       0.68      0.87      0.77     19415

    accuracy                           0.63     32080
   macro avg       0.42      0.44      0.41     32080
weighted avg       0.55      0.63      0.57     32080

[[ 3470     5  4395]
 [ 1431     1  3363]
 [ 2564     3 16848]]
```

Figure 4

Accuracy Interpretation : The accuracy is 63% for the model.
Precision Interpretation: 46%, 11% and 68% for 0,1,2 ratings respectively are the correctly predicted cases which turns to be positive.
Recall Interpretation: 44%, 0% and 87% for 0,1,2 ratings are the actual positive cases that the model predicted correctly.

3.5 Discussion

The metrics computed by both k-NN and random Forest are almost the same ,however the f1-score for the k-NN is better than that of random forest.Therefore we opt for k-NN alogorithm. The correct model prediction for deeep learning algorithm is 77% whilst the correct model

prediction for k-NN is 71% .By this comparison we conclude the deep learning algorithm is better than K-NN. The accuracy of model by using deep learning is 63% whilst the accuracy for k-NN is 57% therefore based on their accuracies we say the deep learning model has high performance as compared to k-NN.

4 Future Work

Future works could be using the features to recommend drugs to patients by sentimental analysis.

5 Conclusion

We can conclude that the Artificial Neural Network (ANN)model is making correct predictions of 63%. Time is a very big challenge ,the time-frame given for this work is very small.Our knowledge on text classification problem is very limited ,however by the endless effort our tutors we learned how to process text data.

6 Reference

·https://github.com/shekhargulati/sentiment-analysis-python/blame/master/opinion-lexicon-English/pos
words.txt

·https://github.com/shekhargulati/sentiment-analysis-python/blob/master/opinion-lexicon-English/negative-words.txt

·https://github.com/shekhargulati/sentiment-analysis-python/blob/master/opinion-lexicon-English/adjective-words.txt