Sentimental Analysis of Drug Reviews and Prediction of Rating for Drug

LAXMAN KUMAR, Syracuse University, USA
HARSHITA ASNANI, Syracuse University, USA
HITESH THADANI, Syracuse University, USA[i]

Additional Keywords: Regression, Drugs, Conditions, Sentimental Analysis

ABSTRACT

The purpose of the project described below was to attempt predict rating for drugs and sentimental analysis of reviews. We trained various classifiers and compare their accuracy in order to find the best model for a task. We have used accuracy, precision and recall comparing the models. In addition to adjusting vectorizer and classifier parameters, I incorporated several lexical and other non-word features such as review length. We have also removed rows based on drug and conditions whose reviews count is less than 2. Doing so will help us in focusing on the higher frequency drugs and conditions and at the same time will increase our accuracy. After testing different combination of vectorizers and its hyperparameter, model's hyperparameter we were able to define best model for each of the task. We have also built a deep neural network for predicting rating based on the reviews. For each of the model, a confusion matrix is created in order to find the false negatives and false positive of a task.

1. Introduction

Common illnesses have thousands of alternative drugs available worldwide. Each drug has similar chemical property with slight change in its composition. Same drug can react differently on different person based on the sensitivity of the person's body. With the hundreds of options available is difficult to choose. Making a generalized system which can dynamically update the rating of drugs based on condition will people choosing better drugs for their condition. For building some analysis and predictive models we have used dataset available on UCI repository. We will generate some analysis and build different models to target 3 different problem system. Our first problem will target about predicting rating of drugs based on their reviews. Generating sentiments about the drugs based on the reviews will also help people getting a 3- scale rating about the drug Positive, Negative and Neutral. Therefore, our second task is sentimental analysis of reviews. Our third task is to predict condition based on the reviews. This task is all about clustering and we will only focus on top 10 conditions in the dataset.

2. Dataset

Dataset consists of reviews of different drugs with their condition and a 10-start patient rating which reflects the overall efficiency of drug. This data is collected by crawling online pharmaceutical reviews sites and download from UCI repository [1]. The dataset also contains two additional variables named useful count and date which denotes the count of people who founds the review useful and the date on which the review was submitted. Dataset contains more than 160k reviews.

2.1 Data Pre-processing

Since the data is raw and fetched from online review sites, it must be cleaned before any kind of analysis. Since we are also dealing with text data, cleaning process may include cleaning the reviews using regex and removing html tags. One of the first and important step of data cleaning is handling null values. We have less 0.05% null values. Since the proportion of null values in comparison with the dataset is very less, removing the rows with null values would not affect out analysis in any way. When traversing through the data, we found out the there are 900 rows which contains some unprocessed and garbage values in condition column. These rows were removed from the dataset. Also as part of data-cleaning process we have renamed all the condition with 'Not Listed' or 'Other' label from 'Others'. This will decrease the number of unique group formed by condition variable. All the drugs and condition with their total review count less than 2 were also removed in order to better predict rating and clustering of drugs based on condition. Finally, all the reviews were filtered from a custom regex pattern for cleaning any sort of non-alphanumeric character such as emoticons or html tags. A new CSV file was generated after the cleaning process.

2.2 Insights from data

We had performed an extensive exploratory and explanatory data analysis to find the relation between data and hidden insights. After evaluating thoroughly and making some plots we notice that the data is highly left skewed w.r.t rating column. From below figure 1, we have more than 50,000 reviews from the group of "rating 10" followed by "rating 9" with 27219 count. "rating 1" group is the contains the third most count with 21391. Rating group from 2 - 7 contains less than 10,000 reviews count. Also, from box plot, we can notice that 50% of the data lies between 8 to 10 since the median of the plot is at 8 and the third quartile and max are at 10. From figure 2, dataset is balance w.r.t to month but slight left-skewed w.r.t to year. All the months contain approximately equal amount of reviews. But as the year increases the reviews also increases with highest peak in 2016 and again dropped in 2017.

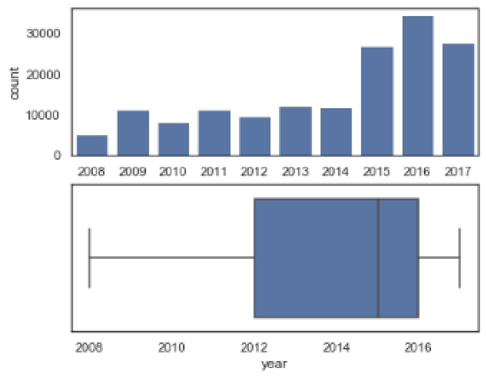


Figure 1 Distribution of Reviews by Year

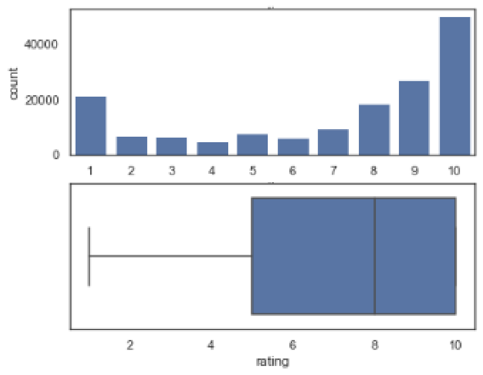


Figure 2 Distribution of Rating

From the below figure 3, condition "Birth Control" contains the highest number of reviews which is about 18% of the data followed by "Depression" approx. 5% and "Pain" approx. 3%.The below figure 4, gives us the information about the number of different drugs available for each of the condition mentioned. We have plotted the top 20 condition in the decreasing order of their drugs count. The condition mentioned as "others" have the highest number of drugs available, it is since the specific condition isn't mentioned in this case. The second highest is the "Pain", it has 200 alternate drugs available. Painkillers drugs are widely sold drugs since they are used in all cases irrespective of age, gender and the critical conditions. The third most number of alternate drugs available are for the condition "Birth Control". According to Wikipedia, in the United States 98% of sexually active women have used birth control at some point in time, and 62% of those of reproductive age are currently using birth control. The later condition in the series also makes sense in real world, since blood pressure, acne, ADHD, depression, anxiety, insomnia is some of the common health condition and whose prescription people take without consulting a doctor. These conditions are some of the most common causes in the world and there are thousands of alterative drugs available worldwide.

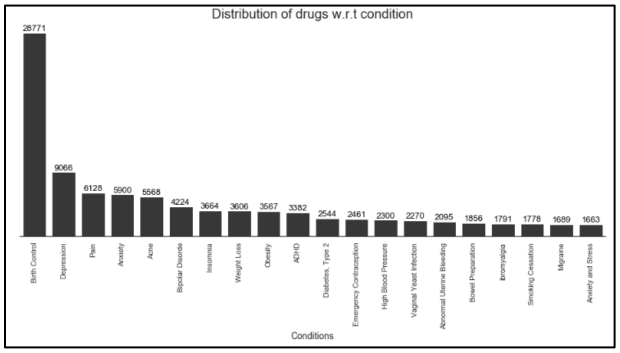


Figure 3 Drugs Distribution

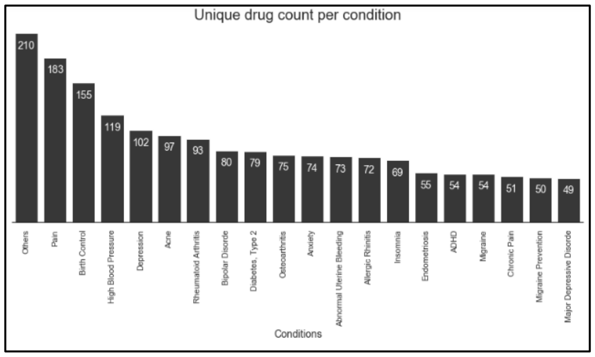


Figure 4 Unique Drug Count

From the below figure 5, we can notice that the word "Side Effect" is the largest followed by 'birth control' and 'started taking'. The word 'Side effect' is quite contradictory since we have more than 50% of reviews with 8 or greater than 8 rating (30% revies with 10 rating). Despite the data with higher number of positive rating, the word 'side effect' is the largest in the worcloud which means side effect has higher frequency than any other word. The word 'birth control' seems to be in proper place since birth control has about 18% of total reviews and third highest unique drug available. Words like 'started taking', 'mood swing', 'two week' are also relevant to the drugs reviews.

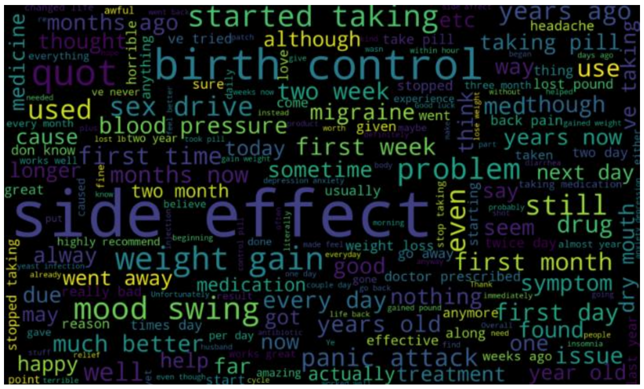


Figure 5 Word Cloud of Review

3. Experiment

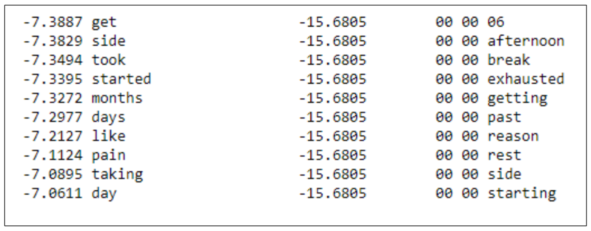
To get further understanding of the data and the problem statement we have implemented multiple models for each of three tasks.

3.1 Prediction of rating

One of the important task of this project is predicting the 10 scale rating of the drug based on the customer review. Since this is a text mining problem we will use text mining based pipeline in order to complete the task. We have used hold out method for all the models in this task. We have divided the dataset into train set and test set in a ratio of 80:20. For this task, three traditional machine learning algorithm had been used Linear SVM, Bernoulli Naïve Bayes and Multinomial Naïve Bayes. In addition with the traditional ML algorithm, we have also build a deep neural network using Keras with TensorFlow backend. For building model, different combination of vectorizer with hyperparameter, ngram range and models had been used. Hyperparameter tuning of model is one of the important task, tuning hyperparameter according to the data is essential and accuracy of the model can increase or decrease based on the tuned hyperparameters. After different combinations Count Vectorizer with ngram range (1,3) gives the best possible result. Therefore all the evaluation are made on these parameters.

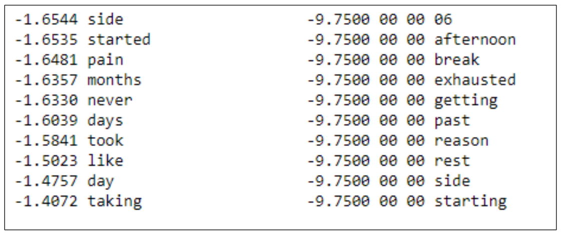
3.1.1 Traditional Machine Learning Algorithm

We ran all the three traditional algorithms with their default parameters and recorded the result. Bernoulli Naïve Bayes gives the worst accuracy of 38.7%, Multinomial Naïve Bayes gives the accuracy of 52.2% and the best accuracy is generated by Linear SVM with accuracy of 67.7%. To evaluate the reason why SVM gives such high accuracy and why the other two model fails, we generated the top 10 most important words and least 10 words of all the three generated models. Both Bayesian algorithm fails to calculate the important words correctly. As we can see in the below two figures, both Bernoulli and Multinomial naïve bayes have similar least important and most important words but their weight values have been changed. In the most important words, both the algorithm have generated the only unigrams and also words like get and took were given important preference. But on the other hand SVM was able to give both unigram and bigram higher weights. It has pulled out some of the words like positive effect, sickening, worked first which are heavinly related to the effects and side effects of taking any drug. After the extensive study and from the result, we choose SVM to the best model and started investigating the features learned by svm in deep.



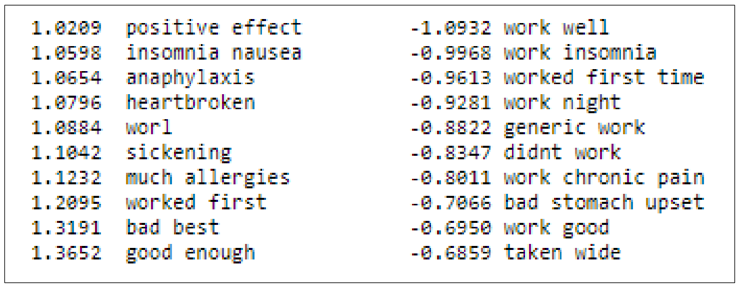
```
-7.3887 get                        -15.6805        00 00 06
-7.3829 side                       -15.6805        00 00 afternoon
-7.3494 took                       -15.6805        00 00 break
-7.3395 started                    -15.6805        00 00 exhausted
-7.3272 months                     -15.6805        00 00 getting
-7.2977 days                       -15.6805        00 00 past
-7.2127 like                       -15.6805        00 00 reason
-7.1124 pain                       -15.6805        00 00 rest
-7.0895 taking                     -15.6805        00 00 side
-7.0611 day                        -15.6805        00 00 starting
```

Figure 6 Multinomial Naive Bayes



```
-1.6544 side                       -9.7500 00 00 06
-1.6535 started                    -9.7500 00 00 afternoon
-1.6481 pain                       -9.7500 00 00 break
-1.6357 months                     -9.7500 00 00 exhausted
-1.6330 never                      -9.7500 00 00 getting
-1.6039 days                       -9.7500 00 00 past
-1.5841 took                       -9.7500 00 00 reason
-1.5023 like                       -9.7500 00 00 rest
-1.4757 day                        -9.7500 00 00 side
-1.4072 taking                     -9.7500 00 00 starting
```

Figure 7 Bernoulli Naive Bayes

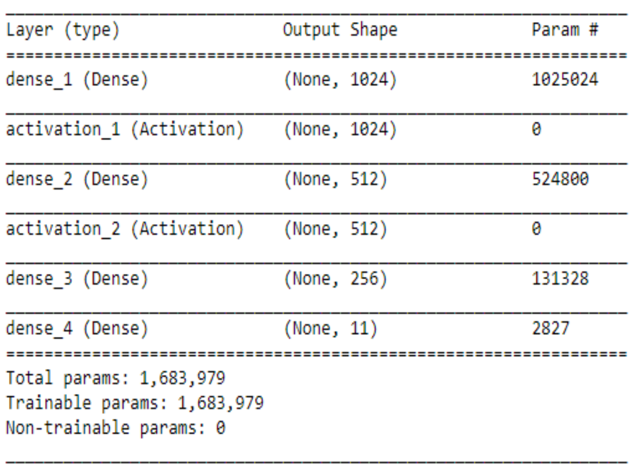


```
1.0209  positive effect         -1.0932 work well
1.0598  insomnia nausea         -0.9968 work insomnia
1.0654  anaphylaxis             -0.9613 worked first time
1.0796  heartbroken             -0.9281 work night
1.0884  worl                    -0.8822 generic work
1.1042  sickening               -0.8347 didnt work
1.1232  much allergies          -0.8011 work chronic pain
1.2095  worked first            -0.7066 bad stomach upset
1.3191  bad best                -0.6950 work good
1.3652  good enough             -0.6859 taken wide
```

Figure 8 Linear SVM

3.1.2 Deep Neural Network

Modern machine learning algorithms are well known to provide very good accuracy irrespective of the data. We have used Keras with TensorFlow backend for building deep neural net for predicting 10-scale rating of the drug based on the reviews. The model summary is mentioned in figure 11 below. We have one input layer, two hidden layers and one output layer. Input layer have 1000 input dimension and 1024 neurons, hidden layers have 512 and 256 neurons respectively and are fully connected layers. Our output layer has shape of 11, since we have 10 classes and converting those into One hot encoded vectors will result in 11 columns.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 1024)              1025024
_________________________________________________________________
activation_1 (Activation)    (None, 1024)              0
_________________________________________________________________
dense_2 (Dense)              (None, 512)               524800
_________________________________________________________________
activation_2 (Activation)    (None, 512)               0
_________________________________________________________________
dense_3 (Dense)              (None, 256)               131328
_________________________________________________________________
dense_4 (Dense)              (None, 11)                2827
=================================================================
Total params: 1,683,979
Trainable params: 1,683,979
Non-trainable params: 0
_________________________________________________________________
```

Figure 10 Neural Net Model Summary

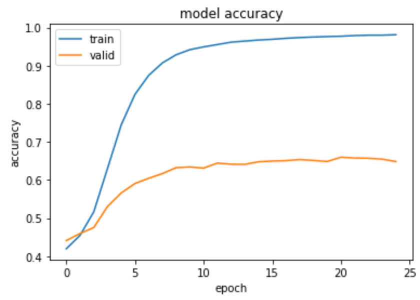


Figure 9 Model Accuracy line plot

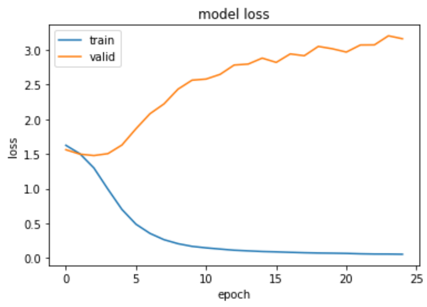
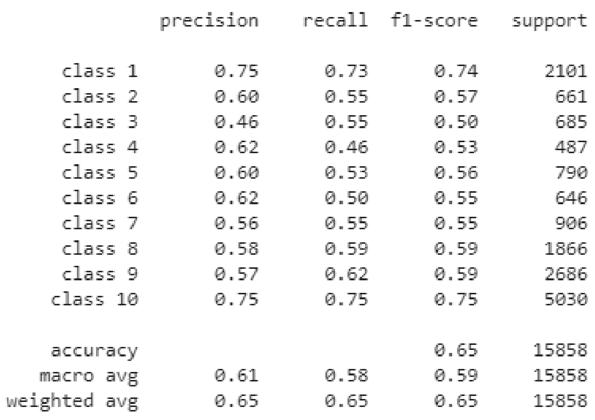


Figure 11 Model Loss line plot

From figure 9 and 10, we can observe that train accuracy began to increase rapidly after the 4th epoch and train loass began to decrease at the same time. We got train accuracy of 99% in 25 epoch. The validation set accuracy increases slowly during the training and reached max at 65%. From figure 12, we can observe that we have very high predicion for class 1 and class 10 incomparison to other classes. Also we have low precision for class 9 inspite of we have very high number of reviews belong to this class. We can infer that it was difficult for neural net for predicting the class 9. From figure 13, confusion matrix we can observe that there are high number of false positive for class 9 in class 10. Neural net face difficulty in predicting the class 9 and class 10 since these two classes are belong to positive family and they are close enough to make wrong predictions.

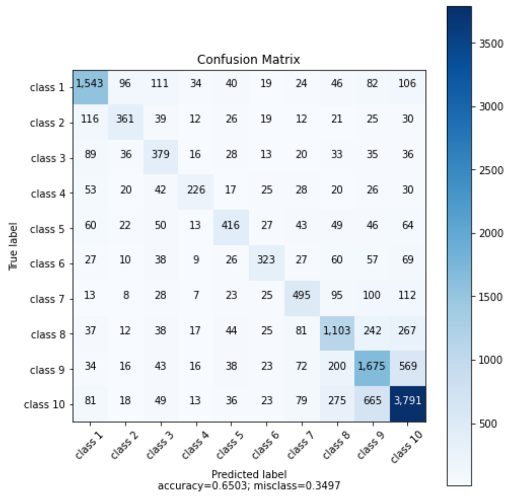


```
              precision    recall  f1-score   support

     class 1       0.75      0.73      0.74      2101
     class 2       0.60      0.55      0.57       661
     class 3       0.46      0.55      0.50       685
     class 4       0.62      0.46      0.53       487
     class 5       0.60      0.53      0.56       790
     class 6       0.62      0.50      0.55       646
     class 7       0.56      0.55      0.55       906
     class 8       0.58      0.59      0.59      1866
     class 9       0.57      0.62      0.59      2686
    class 10       0.75      0.75      0.75      5030

    accuracy                           0.65     15858
   macro avg       0.61      0.58      0.59     15858
weighted avg       0.65      0.65      0.65     15858
```

Figure 12 Evaluation Metrics of Neural Net



	class 1	class 2	class 3	class 4	class 5	class 6	class 7	class 8	class 9	class 10
class 1	1,543	96	111	34	40	19	24	46	82	106
class 2	116	361	39	12	26	19	12	21	25	30
class 3	89	36	379	16	28	13	20	33	35	36
class 4	53	20	42	226	17	25	28	20	26	30
class 5	60	22	50	13	416	27	43	49	46	64
class 6	27	10	38	9	26	323	27	60	57	69
class 7	13	8	28	7	23	25	495	95	100	112
class 8	37	12	38	17	44	25	81	1,103	242	267
class 9	34	16	43	16	38	23	72	200	1,675	569
class 10	81	18	49	13	36	23	79	275	665	3,791

Figure 13 Confusion Matrix

3.2 Sentimental Analysis

Sentiment Analysis is the process of 'computationally' determining whether a piece of writing is positive, negative or neutral. It's also known as opinion mining, deriving the opinion or attitude of a speaker. Local language slangs, word contractions etc. are common while analyzing sentiments and sometimes effect the accuracy of the model built to detect the same. The dataset consisted of 160k consumer reviews of more than 3k drugs for various conditions and also a rating column denoting the rating for the drug as provided by the consumer on a scale of 10. It is really important to establish the ground truth i.e. assign correct sentiment labels to the reviews to further use them as an input (training and validation data) in supervised machine learning algorithms build to detect the sentiment of a review. The rating column in this case was used to establish the ground truth, all the consumers who gave 1-5 rating their reviews were labelled as negative, for 6-7 it was neutral and above 8 were labeled as positive. The below image shows the distribution of the three classes

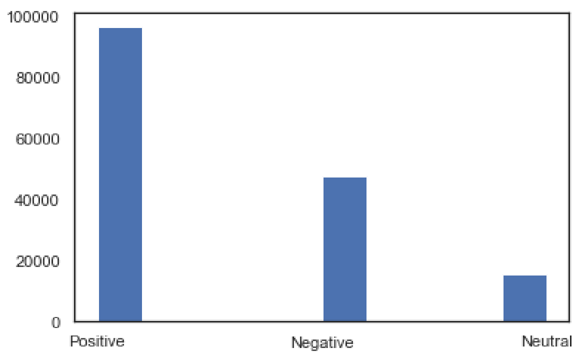
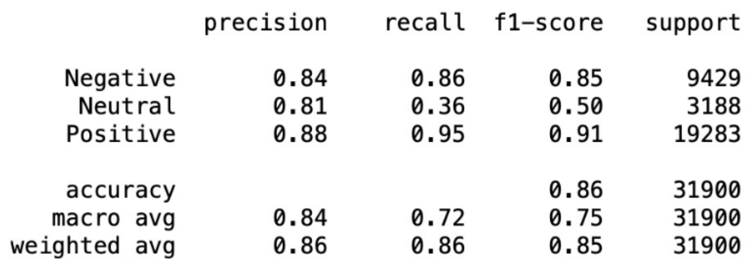


Figure 14 Distribution of Review by Sentiment

As we can see the amount of reviews tagged as positive is almost twice as compared to reviews tagged as negative and almost four times as compared to the reviews tagged as neutral. Furthermore to resolve the problem of language contractions python’s pycontractions library was used. All the contractions such as I’m are expanded to their original form (I am). For contractions that can be resolved in multiple ways (she’d can be she had or she would) pycontractions uses contextual placing of words to determine the correct form. The foremost important task while analyzing sentiments is tokenization. Thus, for tokenization tokenizer from spacy library and sikit learn’s default tokenizer is used. The tokenization of 10k reviews using spacy took around 3hrs while sikit learn’s default tokenizer took just a few seconds. Sikit learn’s tokenizer uses delimiters to perform tokenization while spacy’s uses context trees to correctly tokenize the sentences. As the time consumed by sikit learn’s tokenizer was way less than the time consumed by spacy sikit learn’s tokenizer was used for further analysis.

The next step is to decide the vectorization techniques and for this task we decided to compare the performance of CountVectorizer(Frequency Count vectorizer) and TfidfVectorizer on three models – Multinomial Naïve Bayes, Bernouli Naïve Bayes and LinearSVC(with C=0.1). The accuracy for the three models using count vectorizer were 69.8,70.4,76.02 respectively. As Bernoulli Naïve Bayes accepts only Boolean vectorizer the binary attribute in sikit learn’s CountVectorizer was set to True. The accuracies for MNB and SVC using Tfidf Vectorizer were 71.7 and 76.4. As the accuracy from Tfidf vectorizer was a little higher than Count Vectorier we chose to work with Tfidf Vectorizer for the purpose of Vectorization and SVC gave the best accuracy so it was best for further analysis of sentiments.



```
              precision    recall  f1-score   support

    Negative       0.84      0.86      0.85      9429
     Neutral       0.81      0.36      0.50      3188
    Positive       0.88      0.95      0.91     19283

    accuracy                           0.86     31900
   macro avg       0.84      0.72      0.75     31900
weighted avg       0.86      0.86      0.85     31900
```

Figure 15 Evaluation Metric of Linear SVM

3.2.1 Effect of stop words and other parameters:

The above default models removed stop words from the tokens and uses a minimum document frequency of 10 for Tfidf Vectorizer. Min document frequency corresponds to a value which denotes the least number of documents in our case reviews in which the token should have been present to consider it as a feature while analyzing the sentiment. The above method reduced the feature size to 10741 tokens after not removing stop words and using min document frequency as 1, TfidfVectorizer and LinearSVC(C=0.1) the feature size increased to 41931 tokens which is almost thrice of what earlier and the accuracy increased to 77.8%. The execution time increased by just a few seconds which is acceptable if we have higher accuracy.

3.2.1 Effect of including bigrams and trigrams to features

Unigrams + Bigrams : The number of features increases to 934345 after changing ngram_range=(1,2) and the accuracy increases to 81.06%. The execution time increases by a few seconds

Unigrams + Bigrams + Trigrams : The number of features increases to 3891377 after changing ngram_range=(1,3) and the accuracy decreases to 80.89%. The execution time increases by a few minutes

Bigrams: The number of features increases to 2957032 after changing ngram_range=(2,2) and the accuracy decreases to 76.39%. The execution time again decreases to a few seconds but removing unigrams decreases the accuracy.

Tuning LinearSVC regularization parameter: C=1 as regularization parameter gives best accuracy. The classification report for the same is as follows:

As we can see the accuracy for this model is almost 86%. It is difficult for the model to predict Neutral sentiments accurately this could be due to the bias in number of data points available

3.3 Prediction of Condition based on Reviews

The Drug Review Dataset consists of various conditions for which patients have taken a specific drug and have given review for that drug being used for a specific condition. Dataset had 885 unique conditions, to predict these many conditions is almost impossible. So, the strategy used was to create a subset dataset of top 10 most occurring conditions along with their reviews. This dataset was used for solving the problem of predicting the condition based on the reviews accurately. There are 3 algorithms used for this purpose. Naïve Bayes (Probabilistic approach), Support Vector Machines (Hyperplane approach) and Random Forest. For Naïve Bayes, both Multinomial and Bernoulli models were used. The models were initially fed reviews as input without stemming and then to improve the model learning and performance, Snowball stemmer from NLTK was used for stemming. Stop words were removed for all the models. Both Hold out test and Cross Validation using 5-fold cross validation was carried out for all the models. For Hold out test, the data was split into 80:20 ratio as train and test data. Data is imbalanced for the target variable. The various parameters that were tuned were Minimum Document Frequency (5, 10, 15, 20), Ngram_range(Unigrams, Unigrams & Bigrams, Unigrams, Bigrams & Trigrams, only Bigrams, only Trigrams). For Multinomial Naïve Bayes, only Count Vectorizer with TF as input was used, Bernoulli NB Count Vectorizer with only Boolean Input. For SVM, both TFIDF Vectorizer and Count Vectorizer with Boolean, TF and TFIDF inputs were used. Random Forest was given the same input as the SVM above. For SVM, additionally the value of parameter C was tuned. For Random Forest, parameters n_estimators (Number of Trees) and maxDepth for the trees was tuned.

For Multinomial NB, the variation of min_df parameter from 5 to 20 in the interval of 5 was tested. As the min_df increased, the size of the vocabulary decreased. The words less frequent in the review for a condition would be missed keeping min_df high. For example, words like ovarian cancer, psychiatrist came only 5 times so if we were using higher min_df we would end up losing these important words. Also, the weights were reduced when higher df was used for the common most important words. Unigrams did not help as using them gave words like make, use, think, doctor, sleep along with a limited vocab. Bigrams along with Unigrams made it better for the model to learn features like depress anxieti, suicid, antidepress. Only Bigrams usage did not contribute to the model as words such as ve use, year old were more. Unigrams & Bigrams along with min_df 5(Vocab Size 56619) gave both the best features along with best performance for MNB. It was able to achieve precision of 0.95, recall of 0.79 and f1 score of 0.86 for ADHD condition where the row count was the least which is very good considering the Majority Vote Baseline of around 0.36 data being imbalanced.

For Bernoulli NB, where the input is Boolean, using Unigram along with Bigrams produced almost the same features but reduced model performance. Also, the model performance was increased by using min_df of 5(Vocab Size 56619) capturing words with higher weights to contribute to the features model has learnt. The model outperforms the majority vote baseline for all the conditions to be predicted. The some of the most important features learnt by model were dermatologist, cystic, skin, acn.

For SVM, Boolean Input even though performed overall well in prediction the features learnt for model were parnat, Latuda, mania, can for Depression and for not Depression bipolar, ambien. So, using Boolean input is not a good technique for the model. TFIDF gave best features over TF input. TFIDF had most important features as diet, appetit, pound, lost, weigh for Weight Loss and for not Weight Loss condition depress, gain, hcg as for SVM we have One vs All classification. TF gave features like food energi, qysmia, 25lbs for Weight Loss and tenuat, can for not Weight Loss. So, TFIDF is the best input using TFIDF Vectorizer. Min_df=5(Vocab Size 72390) gave the words and performance better than higher min_df. For the choice of Ngrams_range = (1,3) did make the most useful words. The performance was better where the model predicted conditions accurately with less errors in 7 out of 10 conditions. The value of Regularization Parameter C was tuned which changed the prediction score as well had effect on important features for a condition. The penalty was set to L1 for all models. Best value of C came out to be 0.2. The performance was better than majority vote and better than MNB.

	5-fold CV	Hold Out	Features
C=0.2	0.857	0.853	Weigh,diet,lost
C=0.5	0.856	0.848	Lost, fastin
C=0.9	0.854	0.842	dri mouth, adipex, belviq
C=1.5	0.847	0.840	Naltrexone, 13lbs, qysmia

Table 1 SVM Parameter C tuning comparison

Random Forest where the input was provided to be Boolean, TF, TFIDF gave the best score for Boolean Input while for TF and TFIDF it was almost less by 4%. Only for Random Forest, the best accuracy score was achieved by using highest min_df = 20 instead of lower values. Unigrams and Bigrams gave the best overall performing model for Random Forest. The parameters specific to Random Forest algorithm which were tuned were n_estimators 80 and maxDepth 10. Random Forest model did not perform even half as better as the least performing NB or SVM models. Random Forest showed a very low Recall and F1 score. Also, for some conditions the model was not able to predict the condition.

Model	Hold Out Test	Cross Validation (5 -fold)
MNB	0.84	0.84
BNB	0.83	0.80
SVM	0.85	0.85
Random Forest	0.42	0.40

Table 2 Model Comparison for Hold Out and 5-fold CV

Error Analysis on all the algorithms was done to see which conditions were mostly misclassified. Weight Loss was being wrongly predicted as Obesity and vice-versa for all models. This was because there were common words in both such as lose, weight, diet which are used for when a person wants to lose weight and for person who is suffering from obesity being linked to each other. The models were not able to distinguish this subtle difference between the two conditions. Log ratio difference between conditional probabilities for these 2 conditions gave distinct words

4. Result

After extensive study for each of the three task we were able to achieve very good accuracy and precision. We have also explained hidden insight for each for the build model. For prediction of rating task, accuracy of 67% was achieved using Linear SVM classifier. For sentimental analysis 86% accuracy was achieved using Linear SVM and for the third task predicting the condition based on the reviews, we were able to generate 85% accuracy using SVM.

5. Conclusion

All the results shown in the experiment section are real result generated using python and its supporting library. Code for each of process in available via GitHub link. We have tried to implement all the concepts learned during the course. We have learned not only how to implement different models, but also how to handle data in real word. How real world is sometimes messy and need extensive cleaning and pre-processing before modelling. Exploratory data analysis is also an important part of any project completion. All the relation between the data and hidden insights can be uncovered using exploratory and explanatory data analysis. Our goal of learning how to implement models and decoding the model to find the features learned by the model has been successfully completed. We have also calculation different evaluation metrics for model comparison and confusion matrix to determine the number of false positives and false negatives with their true positive and negative. By combining the evaluation metric and confusion matrix we can infer that which classes were easy to learn for the model and which were the most confusion class for the model. While learning the traditional machine learning models, we have also extended ourselves in building a modern deep neural network by using Keras. We were able to build a fine model with 2 hidden layers and accuracy of 65%. At the end, we have leaned a vast amount of different concepts throughout this class and were successful in implementing those on real world data. We also like to thank Professor Yingya Li to provide timely positive feedback for the project and the area of improvement for each of the assignment and project.

References

1. https://archive.ics.uci.edu/ml/datasets/Drug+Review+Dataset+%28Drugs.com%29
2. https://www.kaggle.com/jessicali9530/kuc-hackathon-winter-2018
3. https://scikit-learn.org/
4. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html
5. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html#sklearn.metrics.classification_report
6. https://keras.io/

The project is an academic project created for class IST 736 instructed and guided by Professor Yingya Li. All the group members are full time student at School of Information Studies, Syracuse University.