# **INFO 5505 Applied Machine Learning for Data Scientists**

## Assignment 5

Name: Ravi Varma Kumar Bevara

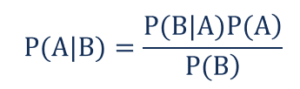
### **PART I: Model and Dataset Description**

Naive Bayes Classifier

It's a classification method based on Bayes' Theorem and the assumption of predictor independence. A Naive Bayes classifier, in simple terms, posits that the existence of one feature in a class is unrelated to the presence of any other feature.

For example, if some fruit is red, round, and roughly 3 inches in diameter, it is termed an apple. Even if these characteristics are reliant on one another or on the presence of other characteristics, they all add to the likelihood that this fruit is an apple, which is why it is called 'Naive.'

Bayes Theorem - It describes the likelihood of an event based on prior knowledge of relevant conditions.



Advantages of Naïve Bayes

* It is simple and quick to forecast the test data set's class. It's also good at multi-class prediction.
* When the assumption of independence is met, a Naive Bayes classifier outperforms alternative models such as logistic regression and requires less training data.
* In comparison to numerical input variables, it performs well with categorical input variables.
* A normal distribution is assumed for numerical variables (bell curve, which is a strong assumption).

Disadvantages of Naïve Bayes

* If a categorical variable in the test data set has a category that was not included in the training data set, the model will assign a probability of 0 (zero) and will be unable to generate a prediction. This is commonly referred to as "Zero Frequency." We can utilize the smoothing approach to remedy this. Laplace estimation is one of the most basic smoothing techniques.
* On the other hand, because naive Bayes is a bad estimator, the probability outputs from anticipated probabilities should be regarded with caution.
* The assumption of independent predictors is another restriction of Naive Bayes. In actual life, getting a collection of predictors that are totally independent is nearly impossible.

The data that is used for this assignment is about the reviews of the six topics form the research paper provided in the assignment description. The dataset collection is described in the papers as the positive reviews are collected from Amazon and negative reviews are contributed from the IMDB reviews for the movies. So, when it is provided to me, it is in the rtf format which is the Rich Text Format. When I tried to import that file into the python, it interpreted some metadata and unnecessary information that is causing issues to the read documents function provided to read the data. So, I have converted the rtf file into a plain text file using inbuilt macOS facility. We can also achieve this with the help of striprtf module in python by converting rtf file into a text file.

In the given data, it is formatted as category, sentiment label, review file name, contents in the document. As we feel that the file name is not related to any analysis, I decided to remove the category, file name from the data and move the labels along with the contents into a separate array.

The data set includes two variants of the sentiments which are positive and negative about the reviews of the six categories including music, books, DVD, camera, health, and software. It is having in total 11914 reviews.

The bigger picture in this assignment is predict the sentiment of review based on the words in the content provided by the end user or customer.

### **PART II: Exploratory Data Analysis (Data Pre-Processing, Data Visualizations)**

To start, I decided to use the “Google Colab” as my programming IDE for the tasks. Also, I decided to store my dataset on the google drive. So, the first step is to import various python modules which can be used to better understand the data. Then, I mounted the google drive for accessing the dataset.

Graphical user interface, application, Teams

Description automatically generated

To read the contents of the reviews, I have followed the same code block provided by the professor to read the labels and contents of all the reviews into separate arrays.

Graphical user interface, text, application

Description automatically generated

For exploratory data analysis purpose, I created a data frame using the two arrays using the series functionality in pandas. After creating that, I checked for the class imbalance problem as it can cause serious problems in the process of building a model.

Graphical user interface

Description automatically generated with medium confidence

Now, we will go through the shape of the data using .shape function and we will check for the null values using isnull() function and also using the seaborn visualization.

Graphical user interface, application

Description automatically generated

Also, we can see that there is no null values in the data, there will be no issues or have any impact in model training and prediction.

Graphical user interface, application

Description automatically generated

Then, I started to make a word cloud for both positive and negative values. For that, I converted the list of tokens in the reviews to a sentence using a custom function. It adds all the indexes in the list to a string to replace that index in the data frame. Once that is done, using the word cloud module from the python I created two-word clouds for both the positive words and negative words.

Graphical user interface, application

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated with medium confidence

Text

Description automatically generated

As we can see that from the word clouds of the both the labels, major contents are present in both the labels which will cause a problem in attainting a better probability. So, it confirms that we are required to perform some sort of smoothing technique to handle that problem.

### **PART III: Model Training (Splitting the Data, Applying the Model)**

As provided in the assignment description, I did not use any python modules for building a machine learning model, but written custom function for training, splitting, accuracy, evaluation, and cross validation. After looking at the data in the preprocessing phase, I have decided to build only one model and provide the accuracy of the model by using accuracy and tuning that model using cross validation methods. To start off with, I will split the data into 80,20 proportions respectively for training and testing phases of the model.

Graphical user interface, text, application

Description automatically generated

For model training function, it uses the training set of documents and labels to estimate the probabilities in the naïve bayes model,

Then I will train a base model using random forest classifier with no real setting of the parameter that will affect the model performance. So, it is looping through the set of labels of both the polarities and counting the words for each label. I have defined some stop words which are general in the language. So, I will be avoiding the stop words in the counter phenomena performed using Counter module in python. If the tokens are some sort of symbols, I have assigned a polarity of ‘1’ before counting the total number of values for each label or polarity and then using the log transformation to ease the computation by converting the multiplication in the bayes theorem formula to the additions according to the property of the logarithmic expressions.

Graphical user interface, text, application, email

Description automatically generated

According to the provided document for the sanity checks, I created a function for returning the logarithm values of the observing words in the documents and the sentiment polarity for that document. For checking that scoring ability in the function, I checked with small sentences which worked out well according to the frequencies of the words.

Graphical user interface, text, application, email

Description automatically generated

Now, it is the time to create the training function which is the actual classifier which will return the maximum of the probabilities generated in the above functions for the corresponding polarities which are the keys for the dictionary created in the training function.

Graphical user interface, text, application, email

Description automatically generated

**PART IV: Model Evaluation Metrics (Prediction Results, Test Scores, and Metrics)**

Machine learning model performance assessment is just like assessing this course, how are used to evaluate our schools in high schools and colleges for the meeting the eligibility criteria for getting the best courses or getting selected into the campus interviews for companies, etc. so apparently the good school recognizes the fact that the candidate is always good. The same is being expected in the machine learning model and that should achieve the expected result in predictions or forecasting or any other required automation in the problem statements.

Accuracy is just a number, for getting a better understanding of a prediction-based problem that corrects the predictions which are made by the model built by the team with the available number of records. So, we need to train the model across different combinations of data to get better accuracy.

For this aspect, I have created a function and using the NumPy sum function, I have returned the percentage of correction predictions in the total number of predictions.

Graphical user interface, text, application, email

Description automatically generated

Although the model achieved 82% accuracy with the test data, for betting understanding the incorrect predictions, I decided to perform some error analysis. I collected the indexes of the incorrect predictions and created a data frame with those entries, then using NumPy stack function, I added them to a data frame long with the predicted labels and the contents of the review.

As per analysis, I created the logarithmic scores for both positive and negative polarities for that content and examined the absolute difference to view the gap between the actual values and the predictions. And for the results, we can see maximum of the contents are having both negative keywords and positive keywords almost equally, so the absolute difference for majority of the bad predictions is less than 1. Also, out of 428 wrong predictions, almost half of them are less than 1 which is better precision in deciding for the polarity.

Table

Description automatically generated

So, the evaluation method that I choose in this assignment to evaluate the naïve bayes classification algorithm is K Fold cross-validation. It means in each set of training and testing that would be performed precisely once during the entire process of building one decision tree we will be dividing the whole data set into K samples in which one sample will behold for testing that phase of training.

Text

Description automatically generated with medium confidence

From the above picture, we can see that the model has been trained with 10 possible samplings of training data and the testing data and the overall average of all the models is almost the same as 82% which we achieved in using the whole data set. So, it is almost the same accuracy in testing the model with the small test set, so that is inferring that the model is not being overfed with any one polarity which is causing that accuracy.

Also, we have produced the estimate interval using the beta distribution from the stats module in python, which is taking in the argument of the setting value for the confidence interval which is 95% in this use case. It is giving the better estimates for the lower end and higher end in the accuracy metric for the classifier which are 80% and 82%.

### **PART V: Summary and Conclusion**

In this assignment, I have not used any of the python libraries or modules for the training the machine learning models. I have used all the code blocks from the assignment description by the professor and completed them using the normal python libraries and other data structures.

We have converted the dataset into tokens which are the list of words in this case, which is equivalent to the lemmatization, tokenization in the text classification or sentimental analysis. Then using the count of the words in the respective polarities in the dataset, and then using the mathematical functions to calculate the scores for each word in the corresponding polarities or sentiments.

Then, using logarithmic transformation on those scores eases the computation part of the naïve bayes classifier by converting the multiplication in the bayes theorem formula to the additions. Also, we can convert them back to the normal using exponential transformation to check for the small sentences. For the classification, I counted on the maximum values among the count of keys with the log values of probabilities for the reviews to result in the best polarity.

For creating the accuracy, I have divided the correct predictions to the total predictions which is the basic rule of metric for accuracy ion machine learning. And then for evaluating the model, I have used the cross-validation technique with the option of folding mechanism in which I have chosen 10-fold validation out of which 9 will be used for training the model and the rest one for the testing purpose. Even then, the model is achieving almost 82% accuracy in both the cases. For the estimated interval or confidence interval I’m using the beta distribution from the stats module with the 95% interval leaving out 5% for the tails of the distribution.

By comparing the accuracy to a given target value, the results (the number of tested samples (2383) and the right results (1936)) are got from previous experiment. The results with the hypothesis classifier's accuracy

0.82 gives a p-value 0.13 indicating that the hypothesis accuracy is accepted at the 5% level of significance because the returned p-value is greater than the critical value of 5%.

Also, this can be reproduced by using both scikit learn and nltk modules to compare all the models with the custom written function for training and evaluating the model.