Text Classification using Naive Bayes Classifier and Logistic Regression

Mohammad R. Yousefi & Dheeman Saha, University of New Mexico 04/07/2020

Abstract

The Naive Bayes Classifier and Logistic Regression algorithms are some of the well-known Machine Learning algorithms for text classification. In this project, we have made use of these algorithms to categorize the text data which are grouped into 20 different categories. The sample dataset is compromised of 12000 text documents which are grouped into 20 different newsgroups with the vocabulary size of 61188. The classifiers are trained using this dataset and then using unlabelled 6774 datasets the accuracy of the classifier is measured. The maximum accuracy we got using the Naive Bayes Classifier is 88.07%. The maximum accuracy we got using the Logistic Regression Classifier is 86.57%. Thus, we can see the performance of both the classifiers are somewhat similar.

Introduction

The Naive Bayes Classifier and Logistic Regression are some of the popular techniques for the classification of the discrete dataset. In these sorts of classification models, the patterns from the training dataset are identified using some statistical techniques and then the trained models are used to make predictions using the testing dataset which is unlabelled.

Data Analysis

The given training dataset is composed of documents that consist of a list of **vocabulary V** and the amount of words in the **vocabulary** is 61188 and the number of instances in the training dataset is 12000. Moreover, these documents are compromised by 20 different classes known as **newsgroups**.

In the test dataset, there are in-total 6774 instances and all of them are unlabelled.

The purpose of this project is to identify the different **newsgroups** classes of the unlabelled test dataset. In order to do that at first, we need to train the two different types of classifiers: Naive Bayes and Logistic Regression. Then the trained classifiers will be used to determine the accuracy of the test dataset. Moreover, different parameters of both models are tuned to achieve the best results.

Classification Models

In this section we will share the overview of the models that are used for the classification purpose:

Naive Bayes

The Naive Bayes algorithm is based on the Bayes Rule and it is always assumed that for all i and $j \neq i$. Thus, X_i and X_j are **Conditional Independent**. The following equation represents the Bayes Rule:

$$P(Y = y_k|X = x_i) = \frac{P(X = x_i|Y = y_k)P(Y = y_k)}{\sum_j P(X = x_k|Y = y_j)P(Y = y_j)} \tag{1}$$

Equation 1 states that we are trying to approximate the unknown target function $f : X \rightarrow Y$ which states the outcome is the probability of Y given X. In this equation, Y is the Boolean valued variable and there are n of them. The X is the vector which is represented as $X =< X_1, X_2, ..., X_n >$ where X_i represents the ith attribute. As previously stated the Naive Bayes classifier assumes that each of the estimated of the vectors of X is independent of one another given Y. The assumption helps to simplify the representation of $P(X|Y)$ and the problem of estimating it from the training data. Moreover, the computation power of the model P(Y|X) also reduces as the number of parameters to be estimated decreases from $2(2^{n-1})$ to $2n$.

In this project we have considered the **Maximum Likelihood Estimation (MLE)** to estimate $P(Y)$ and $P(X|Y)$ is estimated using the **Maximum A Posteriori Probability (MAP)**. Furthermore the Dirichlet $(1+\beta, ..., 1+\beta)$ distribution is considered as the prior. The equations for MLE and MAP is represented in Equation 2 and 3.

MLE for P(Y)

$$P(Y_k) = \frac{\#\, of\ the\ docs\ labeled\ Y_k}{\#\, number\ of\ docs} \tag{2}$$

MAP for P(X|Y)

$$P(X_i|Y_k) = \frac{(\ count\ of\ X_i\ in\ Y_k) + (\alpha - 1)}{(total\ words\ in\ Y_k) + ((\alpha - 1) * (length\ of\ vocab\ list))} \tag{3}$$

where

$\alpha : 1 + \beta$

$\beta : 1/|V|$

k : number of classes i.e 20

i : number of attributes i.e 61188

After finding the probability values of P(Y) and P(X|Y), the prediction value of the **newsgroups** is done using Equation 4.

Classifier

$$Y^{new} = argmax[\log_2(P(Y_k)) + \sum_i (\#\ of\ X_i^{new}) \log_2(P(X_i|Y_k))] \tag{4}$$

Logisitc Regression

The Logistic Regression classification learning function is similar to that of the Naive Bayes which is in the form $f : X \rightarrow Y$ or $P(Y|X)$. This classifier is used to estimate the probability of a certain class and this is done using the Equations 5 and 6, where Y is the discrete-valued and X is the vector of continuous or discrete values represented as $X =< X_1, X_2, ..., X_n >$ where X_i is the ith attribute. The Logistic Regression assumes a parametric form for the distribution P(Y|X), then directly estimates its parameters from the training dataset. Moreover, the sum of the probabilities is always equal to 1.

$$P(Y = 1|X) = \frac{1}{1 + \exp\left(w_0 + \sum_{i=1}^{n} w_i X_i\right)} \tag{5}$$

$$P(Y = 0|X) = \frac{\exp\left(w_0 + \sum_{i=1}^{n} w_i X_i\right)}{1 + \exp\left(w_0 + \sum_{i=1}^{n} w_i X_i\right)} \tag{6}$$

Equations 5 and 6 can be expressed as a simple linear expression for classification. That is, we can maximize the outcome $P(Y = y_k|X)$ for any given X. For example, if we want to determine the value of $Y = 0$ then we represent Equation 7 by taking the ratio of Equation 5 and 6. After that, taking the natural log both sides we get a linear equation which is a classification function for label $Y = 0$ represented in Equation 9 otherwise the label is $Y = 1$

$$1 < \frac{P(Y = 0|X)}{P(Y = 1|X)} \tag{7}$$

$$1 < \exp\left(w_0 + \sum_{i=1}^{n} w_i X_i\right) \tag{8}$$

$$0 < w_0 + \sum_{i=1}^{n} w_i X_i \tag{9}$$

The parametric form of $P(Y|X)$ is similar to that of the Naive Bayes classifier and this implies that Logistics Regression holds all the assumptions of the Naive Bayes classifier.

For our purpose we the above theoretical idea is represented in Equation 10 which is an approximation of $P(Y|X, W) \sim \exp(WX^T)$. This approximation is used for classification purposes.

$$\begin{aligned} ln\ P(D_Y|D_X, W) &= \sum_{j=1}^{N} ln\ P(y^j|x^j, w) \\ &= \sum {}_j y^j (w_0 + \sum {}_i{}^n w_i x_i^j) - ln(1 + \exp\left(w_0 + \sum_{i=1}^{n} w_i X_i\right)) \end{aligned} \tag{10}$$

Another important equation that we made use of the Gradient Descent rule with Regularization.

$$W^{t+1} = W^t + \eta(((\Delta - P(Y|X, W))X - \lambda W^t) \tag{11}$$

where,

m : the number of examples

k : the number of classes

n : the number of attributes in each examples

η : the learning rate

λ : the penalty term

Δ : (*k* x *m*) matrix where $\Delta_{ji} = \delta(Y^i = y_j)$

X : (*m* x *(n+1)*) matrix of examples

Y : (*m* x *1*) vector of true classification for each example

W : (*k* x *(n+1)*) weight matrix

$P(Y|X, W) \sim \exp(WX^T)$, a *k* x *m* matrix of probability values

Answer 1

In any Naive Bayes classifier, it is always assumed that for any given pair of a document the $P(X_i|Y_i)$ and $P(X_j|Y_j)$ will be completely different. As the Naive Bayes classifier is used to estimate P(X|Y) and P(Y), then it is reasonable to ask how much training data will be required to obtain a reliable estimate of the distributions. If the given dataset is composed of 1000 documents, every 1000 words long where each word comes from a 50, 000-word vocabulary and then we need to consider the positions of the word from the document. Thus, the probability estimation of $P(X|Y)$ will be extremely large. Therefore, to the optimal value of the parameters, we will need to observe each of the distinct instances multiple times, which is unrealistic as the computation time will be extremely large. As for a given 1000 documents, 1000 words per document and 50, 000-word vocabulary for a given sample problem will need a probability matrix of $1000^{50000} - 1$ parameters, which is extremely hard to estimate.

Answer 2

Figure 1 represents the accuracy of the Naive Bayes classifier over a range of β values. From the outcome, we can see there is a drop in accuracy for both small and large values of β. This is because for small values of β the smoothing is nearly negligible and does not have any prominent effect. But for large values of β, the smoothing factor becomes prominent and dominates the given dataset and gradually the accuracy drops.

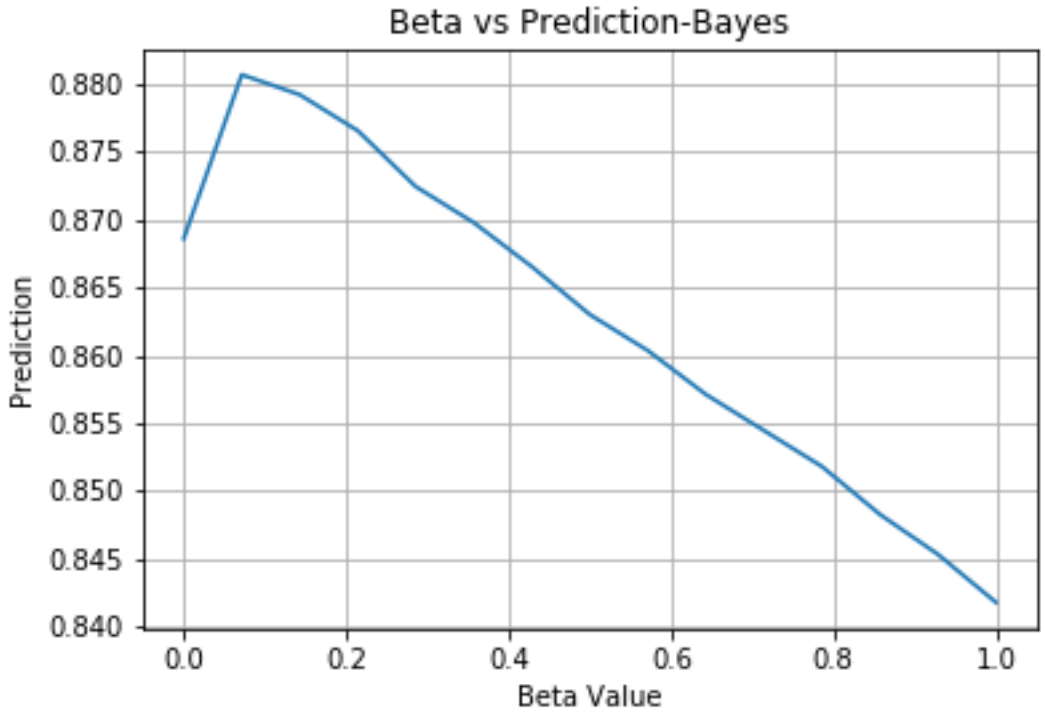

Figure 1: β against Prediction Bayes Classifier

Answer 3

Figure 2 represents the accuracy of the Logistic Regression classifier over a range of η and λ values. Here the range of η and λ is varied between 0.01 and 0.001. The maximum number of iteration is set to be 20, 000 and we have selected the best weight matrix that provided the minimum loss value. The graph is plotted using different combination values of the η and λ. From the outcome we can, we can see that maximum accuracy is achieved for a smaller value of those parameters. This is because for smaller values of those parameters the search is more precise and follows the gradient a better way. However, the number of iterations needs to be much higher as in our case the best weight matrix is generated on the 20, 000th iteration. Probably, increasing the iteration could have increased the prediction of the classifier and the "sweet spot" would have been much more accurate. Moreover, for combinations of η and λ values of [(0.001, 0.001), (0.001, 0.004), (0.001, 0.007)] our prediction of 86.566% remained same and probably we have reached the maxima.

Moreover, for larger values of those parameters, the prediction varied between 65% and 75% and the best weight matrix was found during 100 iterations. This can be stated that the accuracy of the classifier drops significantly for larger values of η and λ parameters.

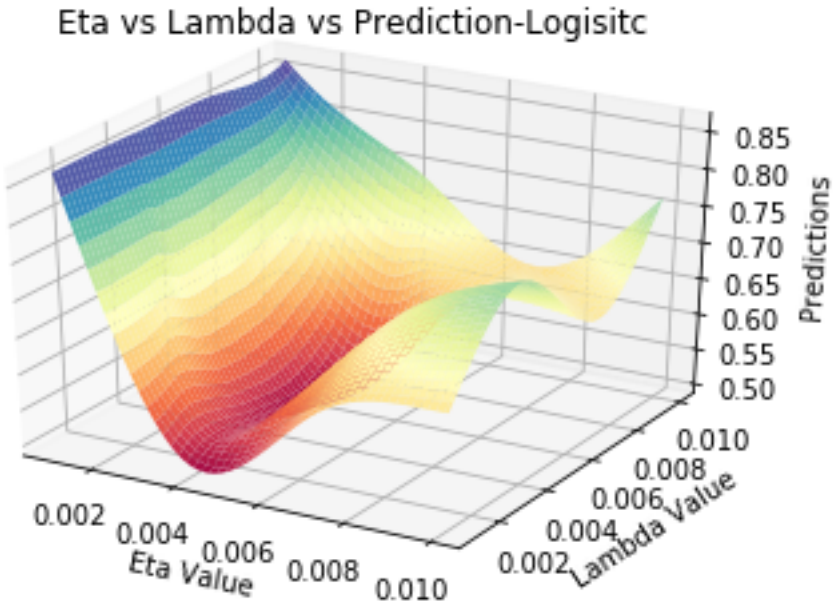

Figure 2: η, λ against Prediction Logistic Regression Classifier

Answer 4

In Figures 3 and 4 we have displayed the outcome of the confusion matrix of both the classifiers. The split ratio between the training and validation sets are 80:20, where the first 2400 rows are considered for the evaluation purpose and the remains for the training purpose. On *Kaggle* the testing accuracy of the Naive Bayes varied between 84.174% - 88.072% and for Logistic Regression the prediction ranged between 51.668% - 86.654%. The best prediction for the Naive Bayes came for the β value 0.07. Moreover, the best prediction outcome for the Logistic Regression came for the η and λ values 0.001 and 0.01 with 20000 iterations.

The confusion matrices of both the classifiers are represented in a 20 x 20 dimensional matrix, where the rows represent the actual values of Y and columns represent the predictions. The numbers displayed in each of the row and column names are the document numbers mentioned in the **newsgrouplabels** file.

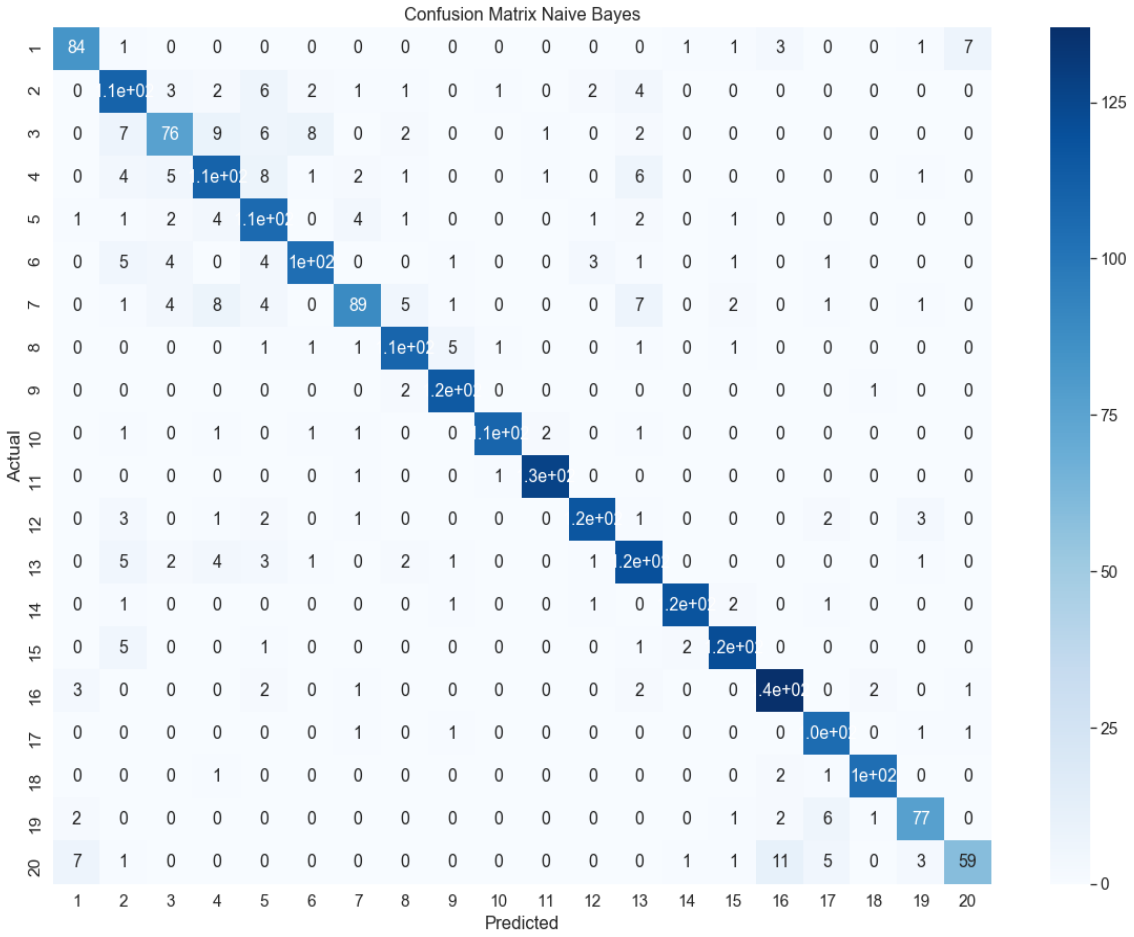

Figure 3: Confusion Matrix Naive Bayes Classifier

The Confusion Matrix for the Naive Bayes classifier has an accuracy of 87.50% using the β value of 0.05. Where else the Confusion Matrix for the Logistic Regression classifier has an accuracy of 87.00% using the previously mentioned values of η and λ.

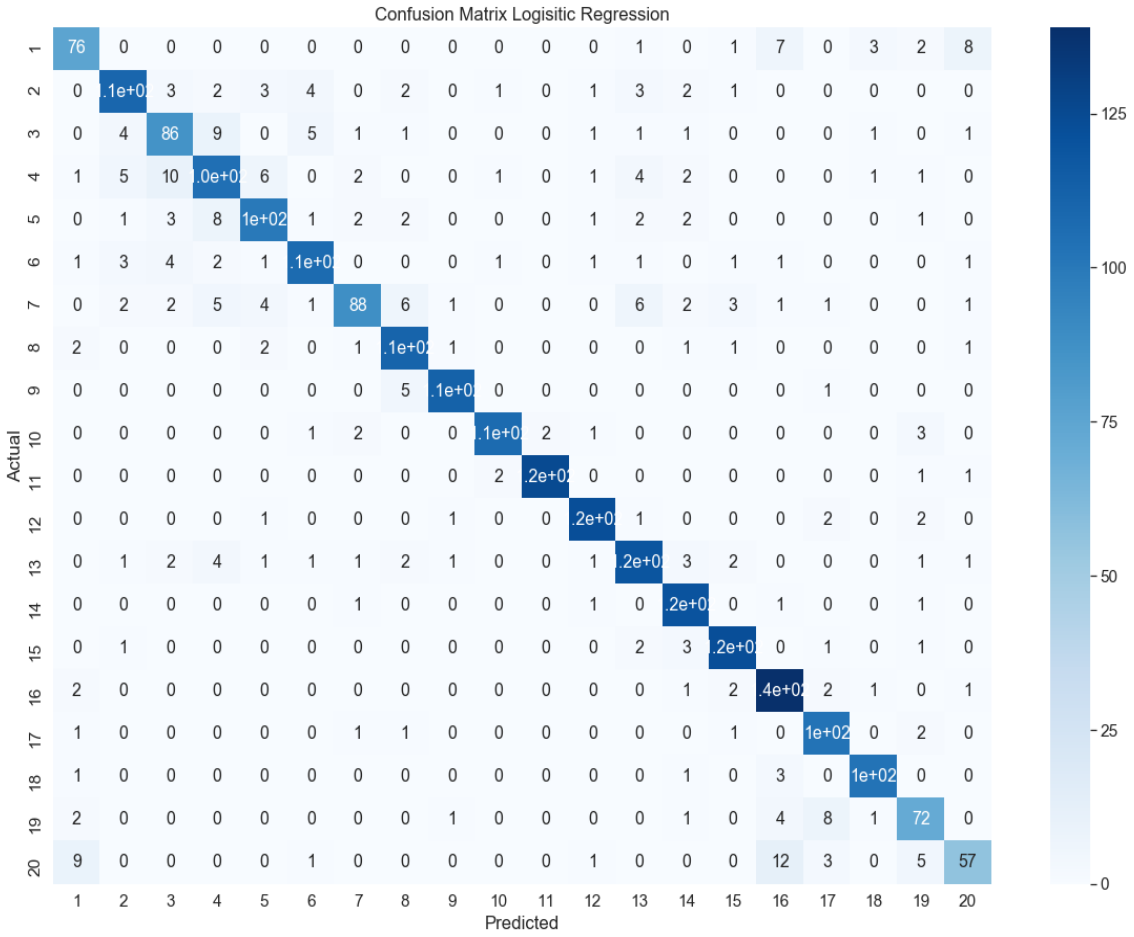

Figure 4: Confusion Matrix Logistic Regression Classifier

Answer 5

From the outcome of both of the confusion matrix, we can see that the category religion has some sort of confusion among the class labels. The class labels 1, 16 and 20 are composed of religion category. From the outcome, we can see that the class 16 (soc.religion.christian) has the highest number of class labels among all the classes. Moreover, we can see that there are a high amount of misclassification in class 1 (alt.atheism) and 20 (talk.religion.misc) with class 16, which clearly there are some level of confusion among the common classes.

Answer 6

Let Θ be the set of parameters values in the Maximum Aposteriori Estimator where Θ_i^j denotes the coefficient value for feature j and class i. We define the importance value of a feature as the weighted average of the coefficients for the features across all classes. For this we define

$I(j)$ as

$$I(j) = \frac{1}{k}\sum_{i=1}^{k} \Theta_i^j M_i = \frac{1}{k}\Theta^j \cdot M \tag{12}$$

where k is the total number of classes and M estimator for the classes.

Answer 7

The list of top 100 words using **beta value 1/|V|**:

drporter, suvm, photosynthesis, mccarthyite, adherent, phobe, skyblu, retracting, umbrage, dispassionate, fantastical, hindparts, centralization, danb, babcock, mellish, pigidinnsot, predominately, paradoxes, personified, flibble, glop, groink, predicate, blanketing, cfj, skyscrapers, afghans, mutable, survivability, introns, exons, intron, oversimplify, revisited, impropriety, implanting, contradictive, ednclark, kraken, salameh, harming, misinterpretations, humorist, jxd, jeesus, jed, mainstreaming, prevost, ommited, instate, xmas, jcwx, jzn, headpiece, gillow, unhelpful, walla, youngster, bifurcation, bvickers, mentiopning, pave, overpowering, hangover, verry, prego, manifesto, chrismas, compeling, seachg, condusive, madhabs, sadiq, floggings, extramarital, borrower, borrows, relabeling, woodlice, arguer, ignorantium, equivalently, searle, causa, equivocation, amphiboly, antiquitam, crumenam, reification, kck, netoprwa, ncsuvm, chpetk, jkp, kuoppala, torkel, franzen, ajr, hri

The list of top 100 words using **beta value 0**:

archive, atheism, alt, modified, december, atheist, addresses, organizations, freedom, religion, foundation, darwin, fish, bumper, stickers, assorted, paraphernalia, ffrf, madison, wi, telephone, evolution, designs, symbol, christians, cars, deluxe, moulded, plastic, postpaid, laurel, canyon, hollywood, bay, lynn, gold, mailing, figmo, aap, publish, critiques, bible, biblical, contradictions, handbook, foote, pp, isbn, edition, absurdities, atrocities, immoralities, contradicts, king, james, austin, tx, cameron, prometheus, haught, holy, horrors, east, amherst, buffalo, alternate, newer, glenn, african, americans, humanism, promoting, secular, uncovering, history, freethought, quarterly, newsletter, aah, examiner, norm, allen, jr, kingdom, rationalist, association, society, islington, holloway, london, ew, nl, british, humanist, ethical, lamb, conduit, passage, conway, wc

Answer 8

There seems to be a bias towards religious topics. To see this we first compute our top 100 words using beta value 0. Next for each of these words we find all samples which have the word at least once and count the labels for those samples. In our training data, we see that categories soc.religion.christian, alt.atheism, talk.religion.misc with rank 1st, 2nd, 4th in label count respectively constitute 0.32 of the labels using the top 100 words. This is in contrast with the 0.15 label count ratio we expect to see in a uniform distribution across the top 100 words.