Name – Kanad Das

Roll No. – B18733

University – Ramakrishna Mission
Vivekananda Educational & Research Institute

Program Course – Big Data Analytics (BDA)

Program Release Date – 12 February 2019

Date of Submission – 22 February 2019

SUPERVISED LEARNING ALGORITHMS:

K-NEAREST NEIGHBOUR CLASSIFIER

vs

NAÏVE BAYES CLASSIFIER

Introduction

Aim: - This is an experiment on data classification using two different classifiers, namely the naive-Bayes classifier and the k-nearest neighbour classifier, conducted on three different datasets. The aim is to evaluate the best performance for each of the classifiers regarding each dataset. by properly tuning the parameters of each classifier so that the least error is recorded during the classification.

Procedure: - From the given datasets, we create a data feature matrix or a data frame. We then divide each dataset into training set (formed randomly by choosing 80% data from each classes of a dataset) and the test set (formed by the remaining 20% data of the class).

We train our model for a particular classifier using the training sets and obtain a optimum set of parameters in order to achieve the best performance. We then use our model to categorize the data points in the test set of each dataset and evaluate the performances for each dataset. Finally, we analyse and compare our results with the actual class labels of the test samples and come up with appropriate conclusions based on our comparisons.

We give a short description of the three datasets to be used in our experiment as follows:

1. **Breast Cancer (Wisconsin) data:** This breast cancer database was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg and the samples were donated by Olvi Mangasarian and received by David W. Aha.[1] There are 11 attributes in the dataset out of which the 11^{th} attribute is the class label (2 for benign, 4 for malignant). The dataset contains 699 instances of each attribute, where 16 instances contain a single missing attribute value.
2. **Wine recognition data:** These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.[2] There are 14 attributes in the data set where the 1^{st} attribute is the class label (class 1, class 2, class 3). The dataset contains 178 instances for each attribute with no missing attribute values.
3. **SPAM E-mail Database:** The database generated in June-July 1999 from the following sources:
    a. Creators: Mark Hopkins, Erik Reeber, George Forman, Jaap Suermondt, Hewlett-Packard Labs, 1501 Page Mill Rd., Palo Alto, CA 94304.
    b. Donor: George Forman (gforman at nospamhpl.hp.com) 650-857-7835.

    There is a total of 58 attributes out of which 57 attributes are continuous data-type and the last attribute is the nominal class label (spam (1), non-spam (0)). The dataset contains 4601 instances for each attribute with no missing attribute values.

[1] (Wolberg, September 1990)

[2] (UCI Machine Learning Repository- Wine Data Set)

Methods to be Used

For the data classification, we use the following classifiers:

1. **Naive Bayes Classifier:** Let us assume a training set of m samples $S = \{S_1, S_2, \dots, S_m\}$ where every sample S_i is represented as an n −dimensional vector $\{x_1, x_2, \dots, x_n\}$ corresponding to attributes $A_1, A_2, \dots, A_n$ respectively and let there be k classes $C_1, C_2, \dots, C_k$ and each sample belongs to one of these classes. The Naïve Bayes classifier predicts the class of a additional data sample X using the highest conditional probability $P\left(\frac{C_j}{X}\right)$, where $j = 1,2, \dots, k$. These probabilities are computed using Bayes theorem:

$$P(C_j/X) = [P(X/C_j).P(C_j)]/P(X)$$

We maximise the numerator term in the above equation since $P(X)$ is constant for all classes.[3] The naive Bayes classifier finds the best class for the data sample X as the most likely or maximum a posteriori (MAP) class C_{map} as follows

$$C_{map} = \max_{j \in \{1,2,\dots,k\}} P(C_j) \prod_{i=1}^{m} P(X_i/C_j)$$

The maximization that is done in most of the naive Bayes models is as follows:

$$C_{map} = \max_{j \in \{1,2,\dots,k\}} \left[\log P(C_j) + \sum_{i=1}^{m} \log P(X_i/C_j)\right]$$

where the prior probabilities $P(C_j), \forall j$ can be computed from the training set as,

$P(C_j) =$ (Number of samples inC_j)/ (Total number of samples in the data), $\forall j$

There exist different naive Bayes classifiers based on different assumptions on the distribution of $P(X_i/C_j)$ $\forall i, j$. For the continuous features, we use the following classifier:

a. **Gaussian Naive Bayes Classifier:** In this case, the typical assumption is that the continuous values associated with each class are distributed according to Gaussian distribution.
We compute the mean and variance of a feature $x_i, \forall i$ in each class from the training data and then following the Gaussian distribution we have

$$P(x_i/C_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} exp\left\{-\frac{(x_i - \mu_{ij})^2}{2\sigma_{ij}^2}\right\}, \forall i, j$$

Hereμ_{ij}, σ_{ij} are respectively be the mean and variance of the values of $x_i$belong to C_j in the training data. This classifier is known as *Gaussian naive Bayes classifier*.

b. **Multinomial Naïve Bayes Classifier:** For the discrete case, we estimate the conditional probability $P(x_i/C_j), \forall i, j$ as the relative frequency of feature x_i belonging to class C_j as

$$P(x_i/C_j) = \frac{F_{ij}}{\sum_{i=1}^{m} F_{ij}}$$

where F_{ij} is the frequency of feature x_i in class C_j.

[3] (Kantardzic, 2011)

c. **Bernoulli Model:** The naive Bayes classifier using this model estimates the conditional probability $P(x_i/C_j), \forall i,j$ as the fraction of documents of class C_j that contain the term x_i.

2. **K-nearest neighbour Classifier:** The k-nearest neighbour decision rule puts a test data point into a particular class, if the class has the maximum number of members among the k-nearest neighbours of the set of training samples Let $(\overrightarrow{x_i}, y_i), i = 1,2, \dots, n$ be given, where $x_i \in \mathbb{R}^m$ be a m dimensional feature vector and $y_i \in \{1,2, \dots, c\}$ denotes the class label of $\overrightarrow{x_i}, \forall i$, where n be the number of samples in the training set and $c\ (\geq 2)$ be the number of classes. Let us consider $\overrightarrow{x_0}$ be a test data point for which the class label is to be determined. The steps of the k-NN algorithm is as follows:
   a. Calculate $\rho(\overrightarrow{x_0}, \overrightarrow{x_i}), \forall i = 1,2, \dots, n$, where ρ is the suitable distance function.
   b. Arrange n number of ρ values in non-decreasing order.
   c. Consider the first k (say) ρ values and find the k data points corresponding to these k distances.
   d. Let L_i denotes the number of points belonging to y_i among the k points, for $i = 1, 2, 3, \dots, c$. Now assign $\overrightarrow{x_0}$ to y_i if $L_i > L_j, \forall j \neq i$.

   For ensuring the best performance of the k-NN classifier, we need to choose a good k value and a suitable distance function. Choosing the optimal value for k is best done by first inspecting the data. Another way to determine a good k value is the method of c-fold cross validation.

   c-fold Cross Validation: The procedure has a single parameter called c that refers to the number of groups that a given data sample is to be split into. The general procedure is as follows:
   1. Shuffle the dataset randomly.
   2. Split the dataset into c groups.
   3. For each unique group:
      a) Take the group as a hold out or test data set
      b) Take the remaining groups as a training data set
      c) Fit a model on the training set and evaluate it on the test set
      d) Retain the evaluation score and discard the model
   4. Summarize the skill of the model using the sample of model evaluation scores.

   Choose that value of k for which the c-fold cross validation model has the highest score.

   As for the suitable distance function, there are several distance-measuring functions based on the type of variables given below:
   a) Continuous variables:

   $$\text{Euclidean Distance} = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

   $$\text{Manhattan Distance} = \sum_{i=1}^{k}|x_i - y_i|$$

   $$\text{Minkowski Distance} = \left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{\frac{1}{q}}$$

   b) Categorical variables:

   $$\text{Hamming Distance, } D_H = \sum_{i=1}^{k}|x_i - y_i|$$
   $$x = y \Rightarrow D = 0$$
   $$x \neq y \Rightarrow D = 1$$

For proper tuning of the parameters, we use some specific modules of the scikit learn (sklearn)[4] package of python library. For our experiment, we use the gridsearchCV for choosing the best combination of parameters depending on the training sets of each dataset. For the k-NN classifier, we define a parameter grid by giving a range for k-value and the metrics set i.e., the various distance functions.

[4] (Pedregosa, 2011)

Evaluation Criteria

There are different types of evaluation criteria to evaluate the performance of a classifier using the ground truths of the given data set. Let us assume there are two classes in the data set, say, positive and negative. A classifier makes error when a positive sample is predicted as negative and a negative sample is predicted as positive. The same may be observed from the table below:

		Predicted Class	
		Positive	Negative
Actual Class	Positive	TP	FN
	Negative	FP	TN

where,
TP (*true positive*) = number of data points correctly predicted to the positive class.
FP (*false positive*) = number of data points that originally belong to the negative class, but predicted as positive (i.e., *falsely predicted as positive*).
FN (*false negative*) = number of data points that originally belong to the positive class, but predicted as negative (i.e., *falsely predicted as negative*).
TN (*true negative*) = number of data points correctly predicted to the negative class.
The above contingency table is known as the *confusion matrix*.
To measure the effectiveness of a classifier, we compute the following measures:

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

Again, there are two conventional methods to evaluate the performance of a classifier aggregated over all classes, namely *macro-averaging* and *micro-averaging*.
The macro averaged precision and recall for a set of m classes are computed as follow:

$$\text{Macro-averaged Precision} = \frac{1}{m}\sum_{i=1}^{m} \frac{TP_i}{TP_i+FP_i}$$

$$\text{Macro-averaged Recall} = \frac{1}{m}\sum_{i=1}^{m} \frac{TP_i}{TP_i+FN_i}$$

Here TP_i stands for true positive of the i^{th} class, FP_i stands for the false positive of the i^{th} class, FN_i stands for the false negative of the i^{th} class and TN_i stands for the true negative of the i^{th} class.
The micro averaged precision and recall for a set of m classes are computed as follow:

$$\text{Micro-averaged Precision} = \frac{\sum_{i=1}^{m} TP_i}{\sum_{i=1}^{m} TP_i + \sum_{i=1}^{m} FP_i}$$

$$\text{Micro-averaged Recall} = \frac{\sum_{i=1}^{m} TP_i}{\sum_{i=1}^{m} TP_i + \sum_{i=1}^{m} FN_i}$$

The f-measure combines recall and precision with an equal weight in the following form:

$$\text{F-measure} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}$$

The closer the values of precision and recall, the higher is the f-measure. A high f-measure value is desirable for good classification.

Analysis and Results

The results obtained from running the different types of classifiers for the three datasets are as follows.

1. **Breast Cancer Dataset:**

   a. **K-NN Classifier:** The optimum performance of the classifier is obtained for **k = 3** and **distance function = hamming distance**. The confusion matrix is given by

$$\mathbf{CM} = \begin{bmatrix} \mathbf{93} & \mathbf{0} \\ \mathbf{1} & \mathbf{43} \end{bmatrix}$$

   b. **Naïve Bayes Classifier:** The confusion matrices for the different variations of NB classifier are:

Classifiers	**Method of prediction of class labels**	
	Fitting	**Cross Validation**
Bernoulli NB	$\begin{bmatrix} \mathbf{99} & \mathbf{0} \\ \mathbf{38} & \mathbf{0} \end{bmatrix}$	$\begin{bmatrix} \mathbf{97} & \mathbf{0} \\ \mathbf{40} & \mathbf{0} \end{bmatrix}$
Gaussian NB	$\begin{bmatrix} \mathbf{84} & \mathbf{0} \\ \mathbf{0} & \mathbf{53} \end{bmatrix}$	$\begin{bmatrix} \mathbf{82} & \mathbf{0} \\ \mathbf{0} & \mathbf{55} \end{bmatrix}$
Multinomial NB	$\begin{bmatrix} \mathbf{97} & \mathbf{3} \\ \mathbf{4} & \mathbf{33} \end{bmatrix}$	$\begin{bmatrix} \mathbf{92} & \mathbf{1} \\ \mathbf{5} & \mathbf{39} \end{bmatrix}$

The summary of the evaluation criteria is presented in the following table:

		Classifiers and the Method of Prediction of class labels						
Evaluation Criteria		**Bernoulli Naïve Bayes Classifier**		**Gaussian Naïve Bayes Classifier**		**Multinomial Naïve Bayes Classifier**		**K-NN Classifier (k = 3)**
		Fit	**CV**	**Fit**	**CV**	**Fit**	**CV**	
Micro-averaged	Precision	0.722628	0.708029	1.00	1.00	0.948905	0.956204	0.992701
	Recall	0.722628	0.708029	1.00	1.00	0.948905	0.956204	0.992701
	F-measure	0.722628	0.708029	1.00	1.00	0.948905	0.956204	0.992701
Macro-averaged	Precision	0.361314	0.354015	1.00	1.00	0.938531	0.961727	0.994681
	Recall	0.5	0.5	1.00	1.00	0.930946	0.937805	0.988636
	F-measure	0.419492	0.41453	1.00	1.00	0.934642	0.948496	0.991579

Analysis: From the above table, we can see that the micro-averaged as well as the macro-averaged precision and recall for the **Gaussian Naïve Bayes Classifier**, for both the methods of prediction i.e., method of fit and method of cross validation (**c = 10**) are the highest (= 1.00) as compared to the other classifiers. Consequently, the f-measures (= 1.00) are maximum too. As for the **k-NN Classifier** for **k = 3**, the measures are all close to 0.99, making it the second most optimal classifier after Gaussian NB Classifier.

2. **Wine Dataset:**

   a. **K-NN Classifier:** The optimum performance of the classifier is obtained for **k = 8** and **distance function = Manhattan distance**. The confusion matrix is given by

$$\mathbf{CM} = \begin{bmatrix} 12 & 0 & 0 \\ 2 & 9 & 4 \\ 0 & 1 & 8 \end{bmatrix}$$

   b. **Naïve Bayes Classifier:** The confusion matrices for the different variations of NB classifier are:

Classifiers	Method of prediction of class labels	
	Fitting	Cross Validation
Bernoulli NB	$\begin{bmatrix} 0 & 10 & 0 \\ 0 & 18 & 0 \\ 0 & 8 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 7 & 0 \\ 0 & 19 & 0 \\ 0 & 10 & 0 \end{bmatrix}$
Gaussian NB	$\begin{bmatrix} 12 & 0 & 0 \\ 0 & 15 & 0 \\ 0 & 0 & 9 \end{bmatrix}$	$\begin{bmatrix} 10 & 0 & 0 \\ 0 & 15 & 0 \\ 0 & 0 & 11 \end{bmatrix}$
Multinomial NB	$\begin{bmatrix} 11 & 0 & 1 \\ 0 & 15 & 0 \\ 0 & 0 & 9 \end{bmatrix}$	$\begin{bmatrix} 9 & 1 & 0 \\ 0 & 15 & 0 \\ 0 & 0 & 11 \end{bmatrix}$

The summary of the evaluation criteria is presented in the following table:

Evaluation Criteria		Classifiers and the Method of Prediction of class labels						
		Bernoulli Naïve Bayes Classifier		Gaussian Naïve Bayes Classifier		Multinomial Naïve Bayes Classifier		K-NN Classifier (k = 8)
		Fit	CV	Fit	CV	Fit	CV	
Micro-averaged	Precision	0.555556	0.527778	1.00	1.00	0.972222	0.972222	0.80556
	Recall	0.555556	0.527778	1.00	1.00	0.972222	0.972222	0.80556
	F-measure	0.555556	0.527778	1.00	1.00	0.972222	0.972222	0.80556
Macro-averaged	Precision	0.185185	0.175926	1.00	1.00	0.979167	0.97619	0.80794
	Recall	0.333333	0.333333	1.00	1.00	0.966667	0.944444	0.82963
	F-measure	0.238095	0.230303	1.00	1.00	0.971703	0.957351	0.80166

Analysis: From the above table, we see that the micro-averaged as well as the macro-averaged precision and recall are both equal to 1.00 (hence the f-measures being equal to 1.00) for the **Gaussian Naive Bayes Classifier** over both the methods of prediction (method of fit and method of cross validation (**c = 10**)) making it the most optimal classifier. Among the remaining classifiers, we see that the **Multinomial Naïve Bayes Classifier** using the method of fit for prediction, has the values of precision,

recall and f-measure, above 0.95 for both the methods of aggregation and hence occupies the position for the second most optimal classifier.

3. **Spam E-mail Dataset:**

   a. **K-NN Classifier:** The optimum performance of the classifier is obtained for **k = 5** and **distance function = hamming distance**. The confusion matrix is given by

$$\mathbf{CM} = \begin{bmatrix} 574 & 0 \\ 4 & 343 \end{bmatrix}$$

   b. **Naïve Bayes Classifier:** The confusion matrices for the different variations of NB classifier are:

Classifiers	**Method of prediction of class labels**	
	Fitting	**Cross Validation**
Bernoulli NB	$\begin{bmatrix} 538 & 4 \\ 6 & 373 \end{bmatrix}$	$\begin{bmatrix} 561 & 6 \\ 14 & 340 \end{bmatrix}$
Gaussian NB	$\begin{bmatrix} 575 & 0 \\ 1 & 345 \end{bmatrix}$	$\begin{bmatrix} 547 & 0 \\ 1 & 373 \end{bmatrix}$
Multinomial NB	$\begin{bmatrix} 449 & 93 \\ 40 & 339 \end{bmatrix}$	$\begin{bmatrix} 456 & 85 \\ 25 & 355 \end{bmatrix}$

The summary of the evaluation criteria is presented in the following table:

		Classifiers and the Method of Prediction of class labels						
Evaluation Criteria		**Bernoulli Naïve Bayes Classifier**		**Gaussian Naïve Bayes Classifier**		**Multinomial Naïve Bayes Classifier**		**K-NN Classifier (k = 5)**
		Fit	**CV**	**Fit**	**CV**	**Fit**	**CV**	
Micro-averaged	Precision	0.989142	0.978284	0.998914	0.998914	0.855592	0.880565	0.995657
	Recall	0.989142	0.978284	0.998914	0.998914	0.855592	0.880565	0.995657
	F-measure	0.989142	0.978284	0.998914	0.998914	0.855592	0.880565	0.995657
Macro-averaged	Precision	0.98918	0.979156	0.999132	0.999088	0.851461	0.877422	0.99661
	Recall	0.988394	0.974935	0.998555	0.998663	0.861436	0.888547	0.99403
	F-measure	0.988782	0.976958	0.998842	0.998874	0.853502	0.879111	0.995296

Analysis: From the above table, we see that the micro-averaged as well as the macro-averaged precision, recall and f-measure are all close to 0.999, for the **Gaussian Naïve Bayes Classifier** for both the methods of prediction i.e., method of fit and method of cross validation (**c = 10**). Hence, it can be chosen as the most optimal classifier among the others. As for the **k-NN Classifier** for **k = 5**, the precision, recall and f-measures are all close to 0.996 for both the methods of aggregation, thus making it the second most optimal classifier.

Conclusion

From the above observations, we see that the most optimal classifiers for the classification of the labels for the three given datasets seem to revolve among a single classifier, i.e. the Gaussian Naïve Bayes Classifier.

For the breast-cancer dataset, the most optimal classifier is the Gaussian Naïve Bayes Classifier according to the different measures of performance. Although it is better to use the k-NN classifier for k = 3 to classify the dataset as it is non-parametric in nature, i.e., it makes no assumption about the data distribution, whereas the Gaussian NB Classifier assumes the attributes come from some continuous distribution, particularly the gaussian distribution.

For the wine dataset, again we have the Gaussian Naïve Bayes Classifier as the most optimal classifier, according to the various measures of performance. The classifier takes into consideration the relationship between the attributes and the classes, proving to be the desired classifier for the respective dataset, along with the fact that the data are continuous in nature.

For the spam e-mail dataset, the Gaussian Naïve Bayes Classifier is again the most optimal classifier contrary to the fact that the Bernoulli Naïve Bayes Classifier provides better results for text classification in most cases. The Gaussian NB Classifier takes less time to predict the class labels in comparison to k-NN Classifier, considering the size of the dataset. Also, it is highly efficient compared to the remaining classifiers for this dataset since it takes into consideration the evolution of spam -related terms on a regular basis.