Data Mining Lab on KNN and Naive Bayes

Jagdish Khadka(073/BEX/313) Krishna Bahadur Ojha(073/BEX/317) Rajad Shakya(073/BEX/332)

Abstract— **This report describes about the how KNN algorithm can implemented using sklearn library and its time and space complexity are analyzed and value of best hypermeter k is found by plotting a graph and selecting value of k which minimizes error and different process was found in order reduce the space complexity of algorithm. Naive Bayes is another algorithm of classification based on the Bayes theorem with an assumption of independence among predictors. Here naive bayes algorithm is derived and implemented in a text data which is scraped from the sklearn dataset to predict the accuracy of the model.**

Index Terms—**KNN, Naive Bayes, Precision, Recall**

INTRODUCTION

This document is lab report on the KNN algorithm and Naive Bayes and its practical implementation. KNN algorithm is widely used supervised learning algorithm for the classification and regression predictive function. The best part of KNN is easy to interpretation and low calculation time. This algorithm is also called a lazy algorithm as it doesn't need to be trained and its classification or regression is done on the go by calculating the distance between a checkpoint and all other points at the start of prediction and then, the checkpoint is classified to respective class according the hyperparameter value k. The Naive Bayes is another classification-based algorithm. It is best algorithm for the multiclass prediction problem. When assumption of independent holds this algorithm like logistic regression and it requires less training data. This algorithm is generally used for real time prediction, recommendation system and text classification / sentiment analysis.

I. KNN

KNN is mainly used for classification predictive problems in industry. It is also a non-parametric learning algorithm because it doesn't assume anything about the underlying data uses 'feature similarity' to predict the values of new datapoints which further means that the new data point will be assigned a value based on how closely it matches the points in the training set. The KNN algorithm can be explained in following steps:

1. Load the data
2. Initialize the hyperparameter k to your chosen number of neighbors
3. For each example in the data
   - Calculate the distance between the query example and the current example from the data.
   - Add the distance and the index of the example to an ordered collection
4. Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances.
5. Pick the first K entries from the sorted collection.
6. Get the labels of the selected K entries.
7. If regression, return the mean of the K labels.
8. If classification, return the mode of the K labels.

A. Selecting training and testing dataset:

Training dataset are those selective set of original datasets which fits across the target value and which after fitting create the module and on the basis of that module test set of data are fed to predict the target value to check the accuracy of the module. Therefore, the training dataset are always greater than the test dataset. The training data of 70% is used to learn the KNN module and remaining 30% is used to test the module. In making of dataset we take positive data and negative dataset. Here 1 is taken as the positive data and 0 is taken as the negative dataset.

B. KNeighborsClassifier

This class used to create KNN classifier. Which uses the n_neighbors as parameter to make prediction on the basis of number of neighbors. Here if the neighbor is 3 then the label of the two nearest neighboring data is selecting as the output. It contains eight parameters and all are optional with algorithm = 'auto', n_neighbor = 5, metric = 'minkowski' as a default and 4 attributes.

C. Knn.fit and knn.predict:

Knn.fit () function call fits the all attributes of train set across the label (Target class) so that all attributes classified according to the label. It contains different algorithm of classification such as 'auto', 'ball_tree','kd_tree','brute'. where auto algorithm is default for the fit function. Here 700 samples with 10 attributes are fit as a train set and the Knn.predict function is used to predict the test set of 300 random samples with 10 columns. which means all 300 test samples are predicted across the label and return the label for each data in the binary form, because the label of a given dataset is binary that is label is either a 0 or 1.

D. Confusion Matrix:

A confusion matrix is a table which is often used to describe the performance of the classification model on a set of test data where true values are known. Generally, it contains 2x2 matrix and gives the following information:

True negative (TN) It is the 11 positional value of the matrix, which means number of correct predictions of negative data.
False positive (FP) It is the 12 positional value of the matrix; it is the number of incorrect predictions of positive value.
False negative (FN) It is the 21 positional value of the matrix which gives the number of false predicted value of the negative label.
True positives (TP) It is the 22 positional value of the matrix, which means the cases in which model predicted yes.
The sum of first row and second row gives the actual number of negative label and positive label respectively. where first column and the second column provide the number of correct predictions of the negative label and positive label respectively.

E. Space and time complexity

During the training set, the time complexity is O(nd) where n is number of rows in dataset and k is number of features. This complexity is for brute force method where distance between a query example and features the distances are calculated and they are summed. And the space complexity is O(nd). This complexity is greatly reduced when we use kd_tree or ball-tree which have time complexity is same as binary search i.e. the time complexity becomes O(logn*d).

F. Calculation of best K

K is the number of neighbors in KNN algorithm with which algorithm deals in case of prediction of label. K is chosen so that the value of k provides the least error percentage. Here the number of twenty loops are taken to predict the least error for the selection of best value of K. Each value of neighbors from 1 to 20 are fed to the classifier and error is calculated for each value of k and finally all 20 error of respected value of K is plotted and analyzed to get best value of K for least error.

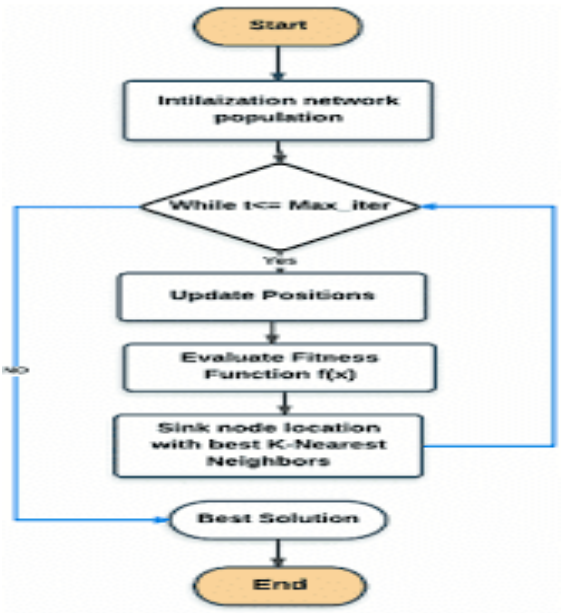


Figure: Flowchart of the KNN Algorithm

II. NAIVE BAYES

Naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong independence assumptions between the features. Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. It assumes that the value of a particular feature is independent of the value of any other feature, given the class variable. Naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods. There are different probabilities which are used according to the different events some of them are as follows.

A. Conditional probability

The conditional probability of an event B is the probability that the event will occur given the knowledge that an event A has already occurred. This probability is written P(B|A). If events A and B are independent (where event A has no effect on the probability of event B), the conditional probability of event B given event A is simply the probability of event B, that is P(B).

B. Joint probability

Joint probability is a measure of two events happening at the same time, and can only be applied to situations where more than one observation can occur at the same time. Joint probability should not be confused with conditional probability, which is the probability that one event will happen given that another action or event happens.

C. Bayes theorem

Bayes' theorem relies on incorporating prior probability distributions in order to generate posterior probabilities. Prior probability is the probability of an event before new data is collected. Posterior probability is the revised probability of an event occurring after taking into consideration new information. Posterior probability is calculated by updating the prior probability by using Bayes' theorem. In statistical terms, the posterior probability is the probability of event. It is a simple mathematical formula used for calculating conditional probability that describes how to update probabilities of hypothesis when evidence is given. It, named after 18th-century British mathematician Thomas Bayes, is a mathematical formula for determining conditional probability.

D. Laplace Smoothing

It is a technique for parameter estimation which accounts for unobserved events. It is more robust and will not fail completely when data that has never been observed in training shows up. It is used to smooth the categorical data. Laplace produce this technique to estimate the chance that the sun will rise tomorrow.

III. MATH

A. Mathematical expressions used for the KNN

The mainly used distance calculation methods are

Euclidean distance = $\sqrt{(x2-x1)2+(y2-y1)2}$

Manhattan distance = $|x_1 - x_2| + |y_1 - y_2|$

Minchowski distance = D (X, Y) = I = $(\sum_{i=1}^{n} |X_i - Y_i|)^{1/p}$

precision formula:
Precision = TPTP+FP

Recall = TPTP+FN

F1_Score = 2*Recall*precision Recall + Precision

B. Mathematical expressions used for Naïve Bayes

The Formula for Bayes' Theorem is
P(A|B) = P(A∩B)/P(B)
= P(A)·P(B|A)/P(B)

where:
P(A) = The probability of A occurring
P(B)= The probability of B occurring
P(A|B) =The probability of A given B
P(B|A) = The probability of B given A
P(A∩B) = The probability of both A and B occurring
Also,
P(B) = P (A ∩ B) + P (A' ∩ B)
= P(B|A) P(A) + P (B|A') P (A')

P(A|B) = P (B|A) P (A)P (B|A) P (A)+P (B|A') P (A')
P(Ai|B) = P(B|Ai) P (Ai)P (B|Aj) P (Aj)

Proof:
P(A ∩ B) = P(A)P(B|A)
P(A ∩ B) = P(B)P(A|B)
these two equations give
P(B)P(A|B) = P(A)P(B|A)
thus,
P(A|B) = P(A) P(B|A)P(B)

This equation, known as Bayes Theorem is the basis of statistical inference.

Given a problem instance to be classified, represented by a vector x=(x1,...,xn) representing some n features (independent variables), it assigns to this instance probabilities
p(Ck| x1,...,xn) for each of K possible outcomes or classes Ck.
Using Bayes' theorem, the conditional probability can be decomposed as

p(Ck|x) = p(Ck) p(x|Ck)p(x)

In plain English, using Bayesian probability terminology, the above equation can be written as
posterior =prior * likelihood evidence
The numerator is equivalent to the joint probability model is p(Ck,x1,...,xn).
=p(x1, ..., xn, Ck)
=p(x1|x2, ..., xn, Ck)p(x2, ..., xn, Ck)
=p x1|x2, ..., xn, Ck)p(x2|x3, ..., xn, Ck)p(x3, ..., xn, Ck)
= ...
=p(x1|x2, ..., xn, Ck)p(x2|x3, ...,xn, Ck) ... p(xn-1| xn, Ck) p(xn|Ck) p(Ck)
assume that all features in x are mutually independent, conditional on the category Ck. Under this assumption,
p(xi|xi+, ..., xn, Ck)=p(xi|Ck)
This is the condition for naive criteria which is used to simplify the bayes theorem.

Thus, the joint model can be expressed as
p(Ck| x1, ..., xn) p(Ck, x1, ..., xn)
= p(Ck)p(x1|Ck)p(x2|Ck)p(x3|Ck) ... p(xn|Ck)
= p(Ck)i = 1np(xi|Ck)
Which denotes proportionality.

This means that under the above independence assumptions, the conditional distribution over the class variable C is:
P(Ck| x1, ..., xn) = 1Z p(Ck)I = 1np(xi|Ck)
where Z = p(x) = kp(x|Ck)

IV. DATA

The dataset used in this report for KNN part contains 11 columns where last column name is TARGET CLASS whose value is either a 0 or 1 (i.e. binary classification). This data contains 1000 rows or total of 1000 data samples. The main other 10 columns are WTT (), PTI (), EQW (), SBI (LQE (), QWG (), FDJ (), PJF (), HQE (), and NXJ () columns. Now, the last column of the dataset is dropped. First the all, the values in the data is stored in the data frame using pandas library, last column is dropped using DataFrame.drop() function and then, standardized (i.e. all 10,000) data such that their mean is zero and standard- deviation is zero. This step of standardization is done in order to give equal importance to all the features, as during dealing with numeric data some feature value may be in range of 100s and other might be in range of 0.1s which causes features of range of 100s suppresses the importance of range of 0.1s. To avoid this issue and to provide equal importance to all the features we use standardization.

The dataset used in this report for Naive Bayes part contains 20 classes of data. It comprises around 18000 newsgroups posts on 20 topics they are alt.atheism, comp.graphics , comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp sys.mac.hardware, comp.windows.x, misc.forsale, rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey, sci.crypt ,sci.electronics, sci.med, sci.space, soc.religion.christian, talk. politics.guns, talk.politics.mideast, talk.politics.misc, talk. religion.misc. This dataset consists of text data mainly in

paragraph which is input to the model and the labels are genre to which this paragraph is related to.

V. PERFORMANCE OF MODELS

- Confusion matrix

A confusion matrix is a table that is often used to describe the performance of a classification model or classifier on a set of test data for which the true values are known. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing.

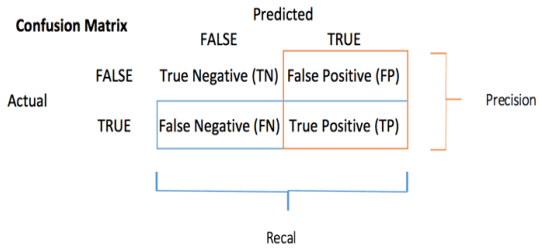


Figure : Confusion Matrix

calculating TN, FP, FN, TP from lab data are following:

TN(True Negative)=132
FP(False Positive)=11
FN(False Negative)=14
TP(True Positive)=143

- Precision

Precision is the ratio of the correctly +be labeled by our program to all +be labeled.

- Recall

Recall is the ratio of the correctly +be labeled by our program to all who are diabetic in reality.it is also called ask sensitivity.

- F1 score

F1 Score considers both precision and recall. It is the harmonic mean(average) of the precision and recall. F1 Score is best if there is some sort of balance between precision (p) & recall (r) in the system. Oppositely F1 Score isn't so high if one measure is improved at the expense of the other.
For example, if Precision is 1 & Recall is 0, F1 score is 0.

VI. LIBRARIES AND FUNCTIONS

A. Libraries

pandas: It is an open source library for high-performance data analysis and provides an array of data into user readable format.

NumPy: It is a fundamental package for scientific computing with python. It is used as a multidimensional container of generic data.

scikit-learn: It is an open source python library. which is used for traditional machine learning, cross-validation, visualization and preprocessing.

matplotlib.pyplot: It is a 2D plotting library of python programming language for visualization of arrays.

IPython.display: It provides different constructions to control over audio, video and images.

Pydotplus: PyDotPlus is an improved version of the old pydot project that provides a Python Interface to Graphviz's Dot language.

time: Time library provides the time related control. It contains time related functions like time(), sleep(), localtime(), gmtime() etc.

B. Functions

scaler.fit(): It is a normalizing method. Which calculate mean and standard deviation used for later scaling along the feature axis.

scaler.transform(): It performs standardization by centering and scaling. It uses to parameter to scale along the feature axis.

df.drop(): This function remove the feature given to the parameter of this function.

pd.DataFrame(): It converts the array of the data into user readable format.

df_feat.head(): It provides the top five sample of the dataset.
knn.fit(): The train and test data of dataset separated by using the train_test_split is fed to this function to classify the trained data using auto fitting algorithm.

knn.predict(): This function return the predicted value of the test set.

confusion_matrix (): It provides the matrix of TP, TN, FP, FN. Which helps to analyze the accuracy of the model.

classification_report (): It returns the precision, recall, f1-score and support value of the predicted value and the label of the test set.

append (): It append the parameter value to the array.

accuracy_score (): It returns the accuracy of the trained model by analyzing the test label and predicted label of the dataset.

plot(): It plots the graph across the x, y value. It is the function of matplotlib.pyplot.

show(): It shows the graph plotted by the plot function. It is also the function of matplotlib.pyplot.

min(): It returns the minimum value of the array.

fetch_20newsgroups(): This function return the fetch_20newsgroup dataset.

make_pipeline(): This is a shorthand for the Pipeline constructor and helps to create model of the multinomial naive bayes.

TfidfVectorizer(): It Convert a collection of raw documents to a matrix of TF-IDF features.

MultinomialNB(): It is used for the classification of discrete features.

sns.heatmap(): This function helps to plot heat map between predicted and true value.

xlabel(): This function is used to give label name to x-axis.

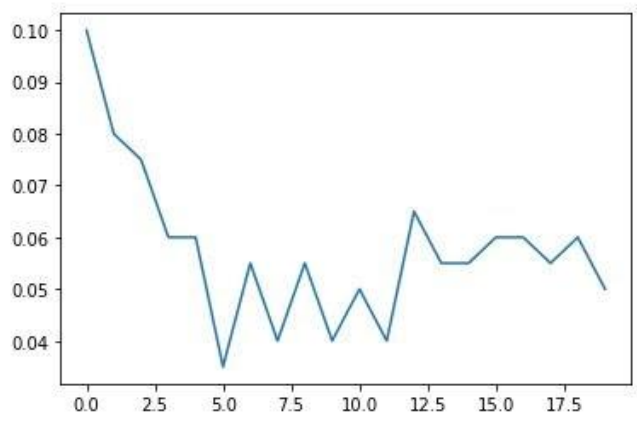
ylabel(): This function is used to give label name to y-axis.

savefig(): This function saves the image or graph to the given file name.

predicting_category(): It returns the categories of the data from the dataset.

VII. RESULTS

In KNN implementation, to calculate the best value of hyperparameter k, which gives maximum accuracy and minimum accuracy, the graph is plotted between error vs value of for best value of k, we put the value of k from 1 to 20 and plot error value for each case and appending them to a list and plot them in a graph using a for loop and matplotlib.pyplot.plot is use to plot the graph. Then, after that the minimum value of list of error is selected and the corresponding value of k is found by finding index of that minimum error value in list. This graph is plotted across value of k and error of the model. Here the value of error=0.035 is minimum for the value of k=5. Which means the module is best fitting for the neighbor number of 5. But for better evaluation of model we can do k fold cross validation where the dataset is divided into training set, validating set and testing set where, the training and validating set is used to find the best value of hyperparameter k. And the testing set is used to evaluate the model (i.e. to calculate accuracy, precision and recall).




```
0.03500000000000003
```

Figure: Finding of smallest value of K using the plot and the minimum error

In order to evaluate the performance of the model we use various performance evaluation metrics like confusion matrix, precision, recall and f1-score which are provided by sklearn.metrics class functions confusion_matrix and classification_report .Here confusion matrix shows that there are 180 correct prediction and 20 incorrect prediction. The accuracy of the KNN module with k=1 is 90% for the testing of 200 data.

```
[[85 13]
 [ 7 95]]
              precision    recall  f1-score   support

           0       0.92      0.87      0.89        98
           1       0.88      0.93      0.90       102

    accuracy                           0.90       200
   macro avg       0.90      0.90      0.90       200
weighted avg       0.90      0.90      0.90       200
```

Figure: Confusion matrix and classification report

A heatmap is a graphical representation of data that uses a system of color-coding to represent different values. Heatmaps are used in various forms of analytics but are most commonly used to show user behaviour on specific webpages or webpage templates. The term heatmap was first trademarked in the early 1990s, when software designer Cormac Kinney created a tool to graphically display real-time financial market information. Nowadays, heatmaps can still be created by hand, using Excel spreadsheets. Heat maps display numeric tabular data where the cells are colored depending upon the contained value. Heat maps are great for making trends in this kind of data more readily apparent, particularly when the data is ordered and there is clustering. To create a heatmap in Python, we can use the seaborn library. The seaborn library is built on

top of Matplotlib. Seaborn library provides a high-level data visualization interface where we can draw our matrix. We use seaborn. Heatmap function to plot the heatmap whose parameters are data which is 2D dataset, square is set true, and not to true, fmt is set to d to format string, cbar to false to draw a color bar. The color changes as value of data changes other than 0. Following is a prediction of fetch_20newsgroups data is made using Naive Bayes and heat map is made over the training set of the data.

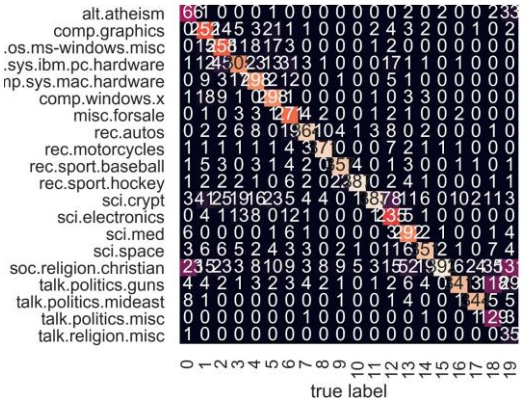


Figure: Heat map across the true value and predicted value.

VIII. DISCUSSION AND CONCLUSION

The two major classification algorithms like knn and naive bayes theorem where knn can be used to classify the query example into class labels without training the model and graph is plotted between error and value of hyperparameter k where value of k is varied and error is plotted and one with minimum error is taken and best value of k is selected. And performance evaluation metrics like confusion matrix, accuracy, recall and f1 score is used to evaluate the algorithm. Also, naive bayes algorithm is implemented using sklearn module and differences between bayes theorem and naive bayes is found. It is found naive bayes works very well on the text data.

APPENDIX

https://github.com/rj7shakya/datamining/blob/master/lab/lab3.ipynb

REFERENCES

[1] Geron Aurelien. *Hands-On Machine Learning with Scikit-Learn and TensorFlow. o'reilly*, New York,2017.

[2] R. O. Duda, P. E. Hart , and D. G. Stock. *Pattern Classification*. John Wiley & Sons,Inc., New York,2nd edition, 2001.

[3] Pang-Ning Tan, Michael Steinbach, Vipin Kumar Introduction to D

[4] W. Zhang and F. Gao, ―Procedia Engineering An Improvement to Naive Bayes for Text Classification,‖ vol. 15, pp. 2160–2164, 2011.

[5] I. Rish, "An Empirical Study of the Naïve Bayes Classifier," no. January 2014.