Comparison Study of kNN, SVM and Naive Bayes Classification Algorithms On the Iris Dataset

M G Christopher
Indian Institute of Information
Technology, Kottayam
christophermg18bcs@iiitkottayam.ac.in

1. Introduction to the problem

Classification problem is one of the oldest and common problems for humans as well as the computers. Basically, the classification problem is to predict a class label based on the features we give as input. A few examples of classification problems are predicting ”spam” or ”not-spam” emails, classifying cats and dogs from a set of images, predicting who wins the cricket match and a lot more. So what we do in making a classification model is, we train or model by providing some known labelled data as input. The model learns how the features and classes are mapped as per the algorithm used. Now when a new set of unlabelled features are provided to the model, it predicts which class does it belong based on the training data. If there are only 2 class labels it is called binary classification and if there are more labels, then it is called multi-label classification. Whether the data is text based or is in the form of images, we can come up to some interesting patterns and conclusions if we could classify the data. Because of its wide spectrum of use-cases and applications in the real world, researchers have developed numerous classification algorithms and are still researching on new ones for different scenarios. Some of the commonly used classifications algorithms are kNN, SVM, Decision Trees, Naive Bayes etc.

2. Algorithms Used

As we discussed in the introduction, there are many classification algorithms available. In this article, we will do a comparative study of 3 of the most widespread algorithms which are k-NN(k-Nearest Neighbours Algorithm), SVM(Support Vector Machine Algorithm) and Naive Bayes(Bayesian Classifier). The detailed discussion of these algorithms follows.

2.1. k-NN Algorithm

The k-NN algorithm is one of the simplest and widely used classification Algorithm in machine learning and deep learning. The algorithm is easy to understand and implement. We will now see how this algorithm works.

```
k-Nearest Neighbor
Classify ($\mathbf{X}, \mathbf{Y}, x$) // $\mathbf{X}$: training data, $\mathbf{Y}$: class labels of $\mathbf{X}$, $x$: unknown sample
for $i = 1$ to $m$ do
  Compute distance $d(\mathbf{X}_i, x)$
end for
Compute set $I$ containing indices for the $k$ smallest distances $d(\mathbf{X}_i, x)$.
return majority label for $\{\mathbf{Y}_i$ where $i \in I\}$
```

As we see in the pseudocode above, to classify an unlabelled set of features, the knn first computes the distance(Manhattan Distance) from the other set of classified entries in the dataset.

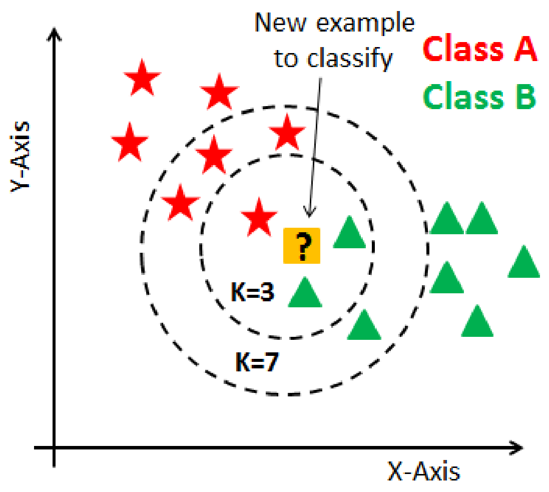


Figure 1. Simple Visualization of kNN

After the distances are computed, the distances are sorted in the ascending order and the first k distances or nearest neighbours are considered. After that majority voting is done within this k nearest neighbors, so the label of this unlabelled feature set will be the labels which are most in number under this set of k nearest neighbours. The value of k might vary based on the dataset. We just have to ensure that there is a clear majority for the deciding of the labels. The idea will be clear from the simple visualization of the algorithm 1 above.

2.2. SVM (Support Vector Machine)

Support Vector Machine or SVM is a supervised machine learning Algorithm. It can be used in both classification and the regression challenges.The algorithm is easy to understand but the implementation can be tricky. The algorithm is mostly used in classification problems. As we

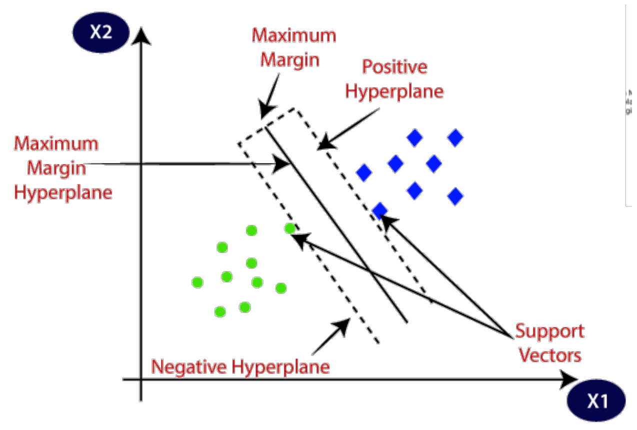


Figure 2. Simple Visualization of SVM

see in the above figure 2, each of the data-item is marked as a point in n-dimensional space where n is the no.of features we have, with the value of each feature being the value of a particular coordinate. We then find the best hyper plane that separates the different classes. To find the best hyperplane, we use the support vectors(nearest data points from either set). The distance between the hyperplane and the support vectors is known as the margin. The goal is to choose a hyperplane with the greatest possible margin thereby giving a greater chance of new data being classified correctly.

2.3. Naive Bayes Classifier

Naive Bayes Classifier or Gaussian Classifier is one of the most easiest, efficient and accurate classification algorithms. It works on the principle of Bayes theorem. Lets say, if a bird walks like a duck and quacks like a duck, then its probably a duck. Same principle is used in Bayes theorem (1).

$$P(\mathbf{A}|\mathbf{B}) = \frac{P(\mathbf{A})\,.\,P(\mathbf{B}|\mathbf{A})}{P(\mathbf{B})} \tag{1}$$

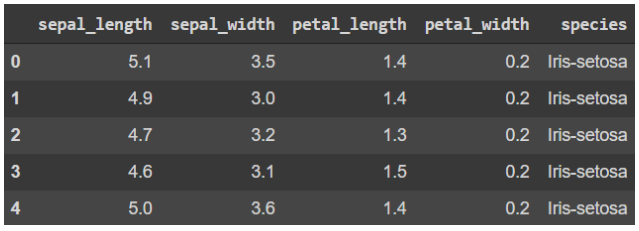
There are mainly two types of Naive Bayes classifiers. The first one is Multinomial Naive Bayes Classifier. This classifier is used when the values of the features are discrete. For example, suppose we have a feature named season, the values of this feature are spring, autumn, summer. That is where we use Multinomial Naive Bayes Classifier. Suppose we have been given set of features to classify as 'A' or 'B'. The probability that the given set of features will be 'A', would be the product of prior probability that it is 'A' multiplied by the probabilities of each of the features to exist in the class 'A'. Similarly, The probability that the given set of features will be 'B', would be the product of prior probability that it is 'B' multiplied by the probabilities of each of the features to exist in the class 'B'. The class we get the highest probability is the correct class.

But if the values of the feature are continuous, like in the iris dataset 3, we take the length and width of sepals and petals, we cannot use the Multinomial Naive Bayes. Instead we use the Gaussian Naive Bayes Classifier. The Gaussian Classifier makes the Gaussian curve with the values of the features in each of the classes. Refer to 4 for the Gaussian curves in the Iris dataset. So in the Gaussian Classifier, to classify it as 'A' or 'B', the probability that the given set of features is 'A' is the product of prior probability of 'A' and the corresponding likelihood values form the Gaussian of Class 'A', and similarly for class 'B'. Since these products can be really small, we often take the log(product) so that its easy to compute.The correct classification is the class with highest probability. The Naive Bayes Classifier is highly scalable and are often used in Sentiment Analysis.

3. Experimental Environment

The experimental environment used here is Google Colab. Google Colab is a free cloud based Jupyter Notebook environment developed as a part of Google Research. This requires no setup to use and it also provides free access to computing resources including GPUs.The colab environment for this project had a 13GB RAM with 103GB ROM Space.

4. Iris Dataset



	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Figure 3. Subset of the Iris dataset

The Iris dataset is one of the oldest benchmark dataset by Sir Ronald Fisher. The Dataset consists of the observa-

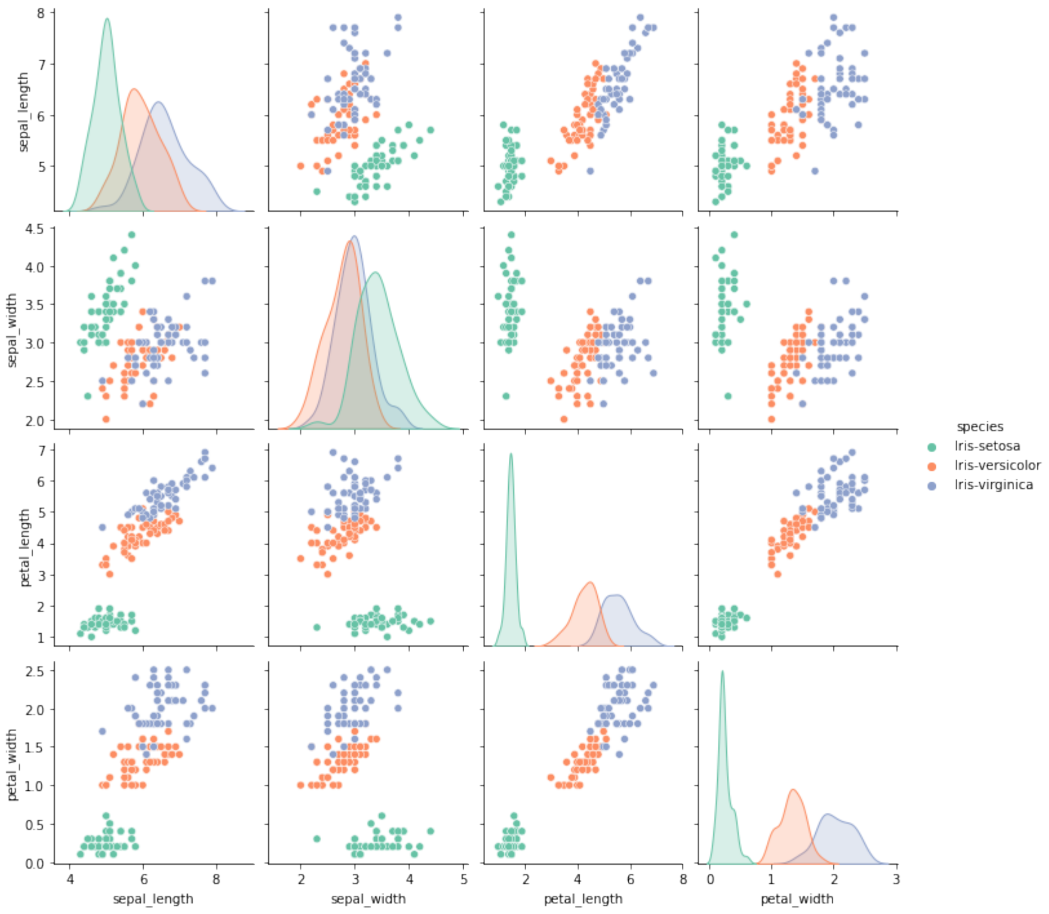


Figure 4. Visualization of the similarities and differences between the species in the Iris Dataset

tions of three Iris flower species nameley Iris-setosa, Iris-virginica and Iris-versicolor. The Dataset contains 50 entries for each of these classes mapped with sepal-length, sepal-width, petal-length and petal-width, making a total of 150 entries. The subset of the iris dataset is shown in 3. The scatter plot and Gaussian plots are shown in 4

5. Performance Metrics

There are mainly four performance metrics used for the comparison of classification algorithms. They are Accuracy, Precision, Recall and F1 score. These metrics give an idea for choosing the right model with the right dataset. Lets try to understand what each of these means.

Accuracy
It is the most basic performance metric which is the ratio of correct predictions to the total number of predictions.

$$\textbf{Accuracy} = \frac{\textbf{No.ofCorrectPredictions}}{\textbf{Size.of.the.TestSet}} \tag{2}$$

Precision
It can be considered as the ratio of actual no,of positives to the no.of tuples that are classified as positives by the classifer.

$$\textbf{Precision} = \frac{\textbf{TruePositives}}{\textbf{TruePositives} + \textbf{FalsePositives}} \tag{3}$$

Recall
It is the ratio of the classified positives to the actual no.of positives.

$$\textbf{Recall} = \frac{\textbf{TruePositives}}{\textbf{TruePositives} + \textbf{FalseNegatives}} \tag{4}$$

F1 Score
It is the harmonic mean of precision and recall.

$$\textbf{F1score} = \frac{\textbf{2} * \textbf{Precision} * \textbf{Recall}}{\textbf{Precision} + \textbf{Recall}} \tag{5}$$

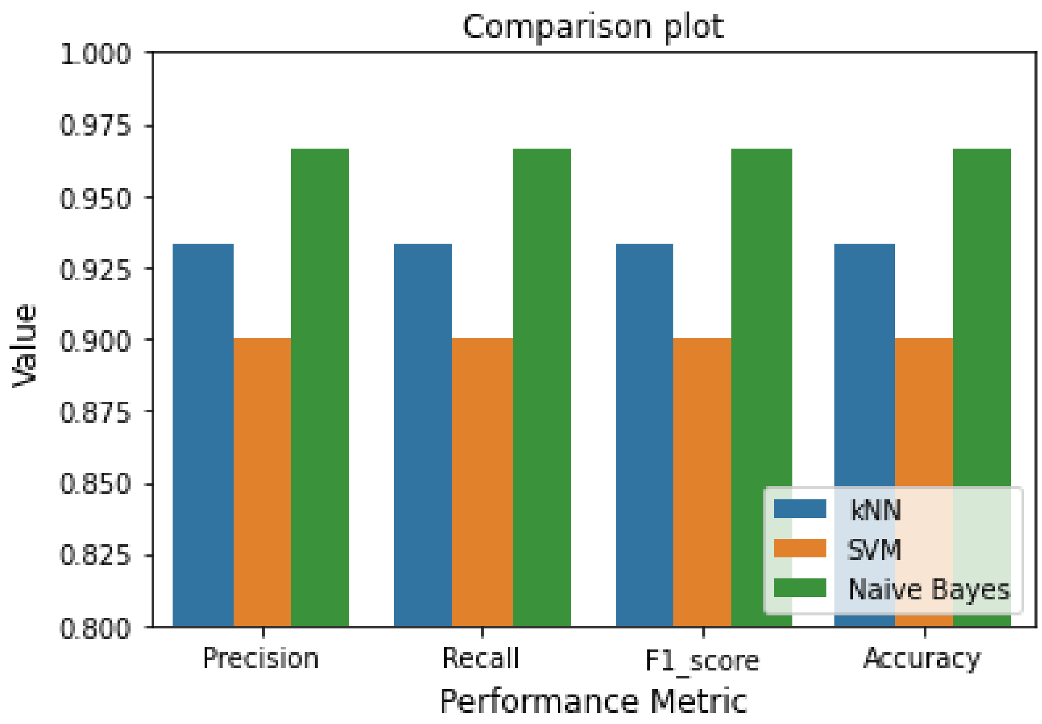


Figure 5. Visualization of the similarities and differences between the species in the Iris Dataset

6. Comparison Charts and Observations

The above figure 5 helps us to compare the classification models based on different performance metrics like accuracy, precision, recall and F1 score. We can observe that Naive Bayes has the highest accuracy,precision, recall and F1 score followed by kNN and SVM. Therefore Naive Bayes is best suited for the classification of the Iris Dataset.

7. Implementation and Image Source

Implementation in Google Colab
https://colab.research.google.com/drive/1sfTlW4y76WEHGDkoZHFNTm49CBlWR2xX?usp=sharing

Image source 1 2
https://www.kdnuggets.com/wp-content/uploads/popular-knn-metrics-0.png

https://static.javatpoint.com/tutorial/machine-learning/images/support-vector-machine-algorithm.png