

# Supervised Classifiers Comparison

Kenneth Hansen- cph-kh415@cphbusiness.dk  
Martin Høigaard Cupello - cph-mr221@cphbusiness.dk

May 5, 2021

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Supervised learning</b>	<b>2</b>
2.1	KNN . . . . .	2
2.2	Gaussian Naive Bayes and Multinomial Naive Bayes . . . . .	2
2.3	Decision Tree . . . . .	2
<b>3</b>	<b>Iris Dataset</b>	<b>3</b>
3.1	Classifiers comparison . . . . .	3
3.2	Results . . . . .	4
<b>4</b>	<b>Penguin Dataset</b>	<b>4</b>
4.1	Classifiers comparison . . . . .	4
4.2	Results . . . . .	5
<b>5</b>	<b>Wine Quality Dataset</b>	<b>5</b>
5.1	Classifiers comparison . . . . .	5
5.2	Results . . . . .	6
<b>6</b>	<b>Results comparison</b>	<b>6</b>
<b>7</b>	<b>Conclusion</b>	<b>6</b>
<b>8</b>	<b>Bibliography</b>	<b>6</b>

# 1 Introduction

In this paper we will compare different machine learning classifiers, and try to determine which one is the most accurate for supervised classification. We will use the popular iris dataset, to test different supervised machine learning algorithms, collect metrics, and finally test our findings against two other datasets to see if we can confirm our results and compare accuracy scores to find the best model.

## 2 Supervised learning

Supervised learning is a type of artificial intelligence and also a machine learning algorithm. The algorithm takes an input variable  $x$  and an output  $y$ , to learn how the input gets the desired output  $Y = f(X)$ . This is normally done with a training dataset to produce an accurate prediction. It can be divided into two groups: Classification, where the model assigns a category based on the given input, and regression, which is used to understand the relationship between dependent and independent variables.

In this paper, we will be using classification models to make predictions.

### 2.1 KNN

KNN (K Nearest Neighbor) tries to determine which group the data belongs to, based on proximity to the nearest neighbors. KNN does not make any assumptions; it only uses input data which makes an educated guess of the output based on its closest neighbors. How to choose K: A good estimate to choose K is to take the square root of  $n$ , where  $n$  is the size of your training data. However, depending on the size of the dataset, it might not be the best.

### 2.2 Gaussian Naive Bayes and Multinomial Naive Bayes

These classifiers are called “probabilistic classifiers” and use statistics by adding the probability of each input parameter to determine which category the data is most likely to belong to. They are called “naive” since they all treat variables as independent of each other, but they use different algorithms to predict the result. The gaussian naive bayes uses a gaussian distribution, where the probability is calculated using the standard deviation and mean for each input. The multinomial naive bayes assumes that all distributions are multinomial, which means more than 2 input variables.

### 2.3 Decision Tree

The decision tree is a tree-like structure consisting of nodes and branches. The root node is the starting point which is the entire dataset. Then the node is split into decision nodes, based on the labels attributes. At the end of the tree we have terminal nodes which are our labels, and the classifications which we

```
graph TD; A[Has feathers?] -- True --> B[Can fly?]; A -- False --> C[Has finns?]; B -- True --> D[Hawk]; B -- False --> E[Penguin]; C -- True --> F[Dolphin]; C -- False --> G[Bear];
```

### 3 Iris Dataset

### 3.1 Classifiers comparison

```
success_rate = []  
for i in range(3,40):  
    knn = KNeighborsClassifier(n_neighbors=i)  
    knn.fit(X_train,y_train)  
    pred_i = knn.predict(X_test)  
    success_rate.append(accuracy_score(y_test, pred_i))  
  
print(success_rate)  
success_rate.index(max(success_rate))+3
```

[0.9, 0.9333333333333333, 0.9, 0.8666666666666667, 0.8666666666666667, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0.93333333333333  
33, 0.9, 0.9, 0.9, 0.9, 0.9, 0.8666666666666667, 0.8666666666666667, 0.8666666666666667, 0.8333333333333334, 0.833333  
3333333334, 0.8, 0.8, 0.8333333333333334, 0.9, 0.9, 0.8333333333333334, 0.8666666666666667, 0.8333333333333334, 0.833  
333333333334, 0.8333333333333334, 0.8333333333333334, 0.8333333333333334, 0.8333333333333334, 0.8333333333333334, 0.  
8333333333333334]

Above picture shows that 4 is actually the best value for K as it gives a higher success rate with a train/test split of 80

Figure 3: KNN Accuracy Score  
0.9333333333333333

Figure 4: Gaussian Accuracy Score  
0.8333333333333334

Figure 5: Multinomial Accuracy Score  
0.8333333333333334

Figure 6: Decision Tree Accuracy Score  
0.9

### 3.2 Results

After optimizing and testing each classifier on the dataset, the results suggest that knn is the best classifier for a data set with few attributes and a size of 150 rows. To test this new hypothesis we will take a similar sized dataset and apply the same classifiers to this data.

## 4 Penguin Dataset

We are using the penguins data set for this experiment. On this dataset we will try to classify which species a penguin is, based on a penguins features. Since the penguins dataset are about double the size of the iris dataset (a little longer actually), we first take a sample of 45% of the full penguins dataset. This leaves us with 150 entries in the sample, close enough to say that it is similar to the length of the iris dataset. We can now begin applying the models.

### 4.1 Classifiers comparison

Figure 7: KNN Accuracy Score  
0.8333333333333334

Figure 8: Gaussian Accuracy Score  
0.9333333333333333

Figure 9: Multinomial Accuracy Score  
0.8

Figure 10: Decision Tree Accuracy Score  
0.9333333333333333

## 4.2 Results

After testing each classifier on the penguin data, we can see that our hypotheses were incorrect. We assumed that a similar sized dataset would give us the same results, but the results clearly show KNN is not the best one in this case. Gaussian Naive Bayes and Decision Tree were more accurate on the penguin dataset.

## 5 Wine Quality Dataset

Our last test will be on the wine-quality dataset. It is larger than the iris and penguins dataset combined. On this dataset we will try to classify if the wine is red or white, based on the wines physico-chemical content and its quality. Our assumption on this dataset is that since Decision Tree has performed consistently well on the two previous datasets (93,3% on penguins and 90% on the iris dataset), that it will also perform well on the wine-quality dataset, even though it is larger.

### 5.1 Classifiers comparison

Figure 11: KNN Accuracy Score  
0.9492307692307692

Figure 12: Gaussian Accuracy Score

0.9807692307692307

Figure 13: Multinomial Accuracy Score

0.9292307692307692

Figure 14: Decision Tree Accuracy Score

0.9907692307692307

## 5.2 Results

Our results on this dataset shows that the decision tree is the best classifier in our case, while Multinomial Naive Bayes continues to be the worst of them all.

## 6 Results comparison

KNN (93,3%, 83,3%, 94,9%)	≈ 90,5%
Gaussian Naive Bayes (83,3%, 93,3%, 98%)	≈ 91,5%
Multinomial Naive Bayes (83,3%, 80%, 93%)	≈ 85,4%
Decision Tree (90,0%, 93,3%, 99,1%)	≈ 94,1%

From our results we can see that Decision Tree performs well in most cases, though it is not the most accurate one in all cases. Worst of all in MultiNomial Naive Bayes which got the lowest score in all cases.

## 7 Conclusion

In the end we can debunk our original hypothesis that there is a best classifier. We can conclude this because given our results there was no classifier that had the highest score on all of the datasets. It is impossible to decide beforehand which one is the best. We can then conclude that to find the best classifier for your dataset, you have to try different models and see which one provides the best result.

## 8 Bibliography

### References

- [1] Amey Band. How to find the optimal value of k in knn?  
<https://towardsdatascience.com/how-to-find-the-optimal-value->

of-k-in-knn-35d936e554eb. 23 May 2020.

- [2] Jason Brownlee. Naive bayes for machine learning. <https://machinelearningmastery.com/naive-bayes-for-machine-learning>. 15 August 2020.
- [3] Jason Brownlee. Supervised and unsupervised machine learning algorithms. <https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms>. 20 August 2020.
- [4] Nagesh Singh Chauhan. Decision tree algorithm, explained. <https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html>. January 2020.
- [5] Davuluri Hemanth Chowdary. Decision trees explained with a practical example. <https://towardsai.net/p/programming/decision-trees-explained-with-a-practical-example-fe47872d3b53>. 28 May 2020.
- [6] P. Cortez. Wine quality data set. <https://archive.ics.uci.edu/ml/datasets/wine+quality>. 04 May 2021.
- [7] IBM Cloud Education. Supervised learning. <https://www.ibm.com/cloud/learn/supervised-learning>. 9 August 2020.
- [8] Onel Harrison. Machine learning basics with the k-nearest neighbors algorithm. <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>. 10 September 2018.
- [9] jlund3. Naive bayes for machine learning. <https://stats.stackexchange.com/questions/33185/difference-between-naive-bayes-multinomial-naive-bayes>. 22 july 2014.
- [10] Prateek Majumder. Gaussian naive bayes. <https://iq.opengenus.org/gaussian-naive-bayes>. 5 May 2021.
- [11] Madison Schott. K-nearest neighbors (knn) algorithm for machine learning? <https://medium.com/capital-one-tech/k-nearest-neighbors-knn-algorithm-for-machine-learning-e883219c8f26>. 22 April 2019.