UiO : Department of Informatics
University of Oslo

Spesialpensum FYS-STK 4155

Espen Lønes

Abstract

Kernel methods and nearest neighbour methods where used to make a classifier for three types of iris plant. All models achieved excellent results with mean accuracy scores of 0.983 and 0.966 for K-nearest-neighbour and support vector machine respectively. The Kernel density estimator had a mean log-likelihood score of -2.6.

The .py files contains all code used to produce the results and plots in this report.
load_data.py contains a function used to read and structure the data in the iris.data file.
parameter_search.py does a grid search for the best hyperparameters for the different methods.
TraiML.py trains the models and gives the scores using the beat parameters found by parameter_search.py.
run.py contains auxiliary functions preforming some minor tasks.

The special curriculum agreement can be found in the spesialpensum_signed pdf and relevant literature is in the literature folder.

Contents

List of Figures

CHAPTER 1

Introduction

The goal of this project is to use the techniques and methods described in Bishop 2006 chapter 2, 6 and 7. As-well as Trevor Hastie 2009 chapters 5 and 6

A dataset containing 3 classes, each with 50 instances. Each class is a type of iris plant. One class is linearly separable from the other two. But those are not linearly separable from each other. Each sample has 4 features; sepal length, sepal width, petal length and petal width. The data is taken from the UCI machine learning repository; https://archive.ics.uci.edu/ml/datasets/Iris.

CHAPTER 2

Method

Parametric vs. non-parametric methods

Parametric methods use probability distributions that have distinct forms determined by a set of parameters. The values of these parameters are then determined by a data set. A drawback of this approach is that the chosen probability distribution may bee a poor fit for the distribution that generated the data. This may result in poor performance. E.g. a data set arising from a multimodal distribution may never be perfectly captured by a Gaussian, since a Gaussian is inherently unimodal. On the other hand non-parametric methods make very few assumptions about the shape of the distribution. I will focus on two of these types of method, Kernel density estimators and Nearest neighbour methods.

Kernel density estimators

Suppose the data is drawn from a probability density $p(x)$ which we want to estimate. Considering a region R containing x, the probability mass of this region is:

$$P = \int_R p(x)dx$$

Bishop 2006 eq. 2.242

Having N data points drawn from $p(x)$. Then each data point has a probability P of beeing in R. The K number of points that lie in R is decided by the binomial distribution.

$$Bin(K|N,P) = \frac{N!}{K!(N-K)!}P^K(1-P)^{1-K}$$

Bishop 2006 eq. 2.243

The mean fraction of the data falling inside R is $E[K/N] = P$. We also see that the variance is $VAR[K/N] = P(1-P)/N$. For large values of N this distribution will be sharply peaked arround the mean.

$$K \simeq NP$$

Bishop 2006 eq. 2.244

If we also assume R small enough that $p(x)$ is approximately constant over the region. We get:

$$P \simeq p(x)V$$

Bishop 2006 eq. 2.245

With V being the volume of R. Using these equations we get:

$$p(x) = \frac{K}{NV}$$

Bishop 2006 eq. 2.246

Which is a function for $p(x)$ as we wanted. We must however be aware of the opposing assumptions. R needs to be small enough that the density is approx. constant over the region, while large enough that the K number of points inside the region is sufficient for the distribution to be sharply peaked.

This equation can be used in two ways. Either by fixing the value of K and use the data to set V, resulting in the K-nearest-neighbour method. Or we can fix V and set K using the data, giving the kernel method. And as shown in Duda, Hart et al. 1973 both the K-nearest-neighbour and kernel methods approach the true probability density as N tends to infinity and V shrinks with N and K grows with N.

Kernel method

Using the eq. 2.250 from Bishop 2006 where K was fixed using a Gaussian kernel and combined with eq. 2.246 we get:

$$p(x) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{(2\pi h^2)^{1/2}} exp\left\{-\frac{||x-x_n||^2}{2h^2}\right\}$$

Where h is the standard deviation of the Gaussian components giving the connection to the fixed V value. This equation gives the density model by

placing a Gaussian on each data point. Then adding them all together. (And dividing by N to normalize the density). Other kernels may be used as long as they satisfy the two conditions:

$$k(u) \geq 0$$
$$\int k(u) = 1$$

Nearest-Neighbour (NN) methods

One problem with the kernel method that having a fixed h value gives problems is the data is to dense in some regions. A to high h value in data dense regions may result in over smoothing of the model. Conversely lowering h may result in increased noise in other parts of the region. The NN methods address this by instead fixing the K value and getting $V(h)$ from the data.

We again use eq. 2.246 as a starting point. We sett a value for K, then use a sphere centered on a point x and expend this sphere until it contains K points. We then get $p(x)$ by using the volume of the final sphere as V. This in known as the K-nearest-neighbour (KNN) method. This method has an equal problem as the kernel method, where there is a sweet spot for the fixed value (not to big not to small). Using the KNN method for classification is done by doing the estimation for each class separately then using Bayes' theorem we calculate the posterior probability of class membership for each class. Then chose the class with highest posterior probability

CHAPTER 3

Results

Analysing the data

Below I plotted the 6 combinations that can be made from the 4 features. From this we can see that the setosa class is the one that is linearly separable from the others. Whereas vesicolor and virginica are harder to separate.

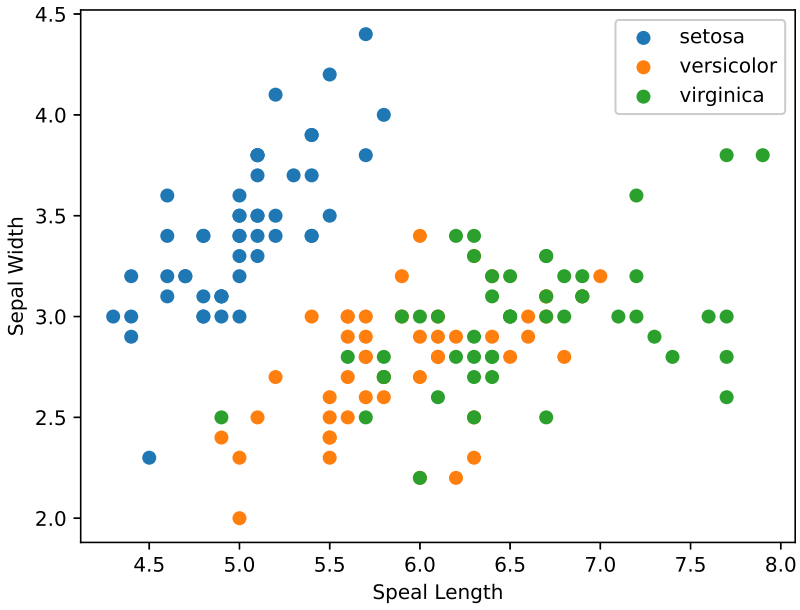

Figure 3.1: Sepal length v. Sepal width

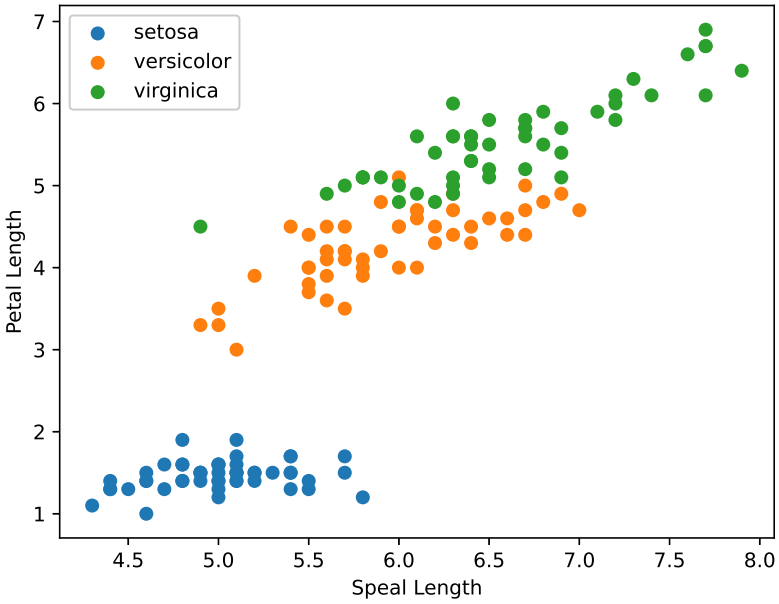

Figure 3.2: Sepal length v. Petal length

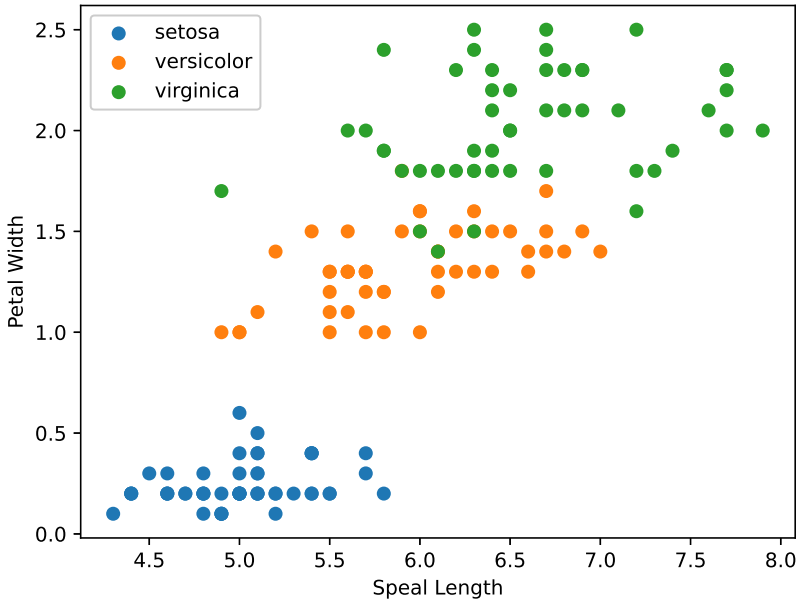

Figure 3.3: Sepal length v. Petal width

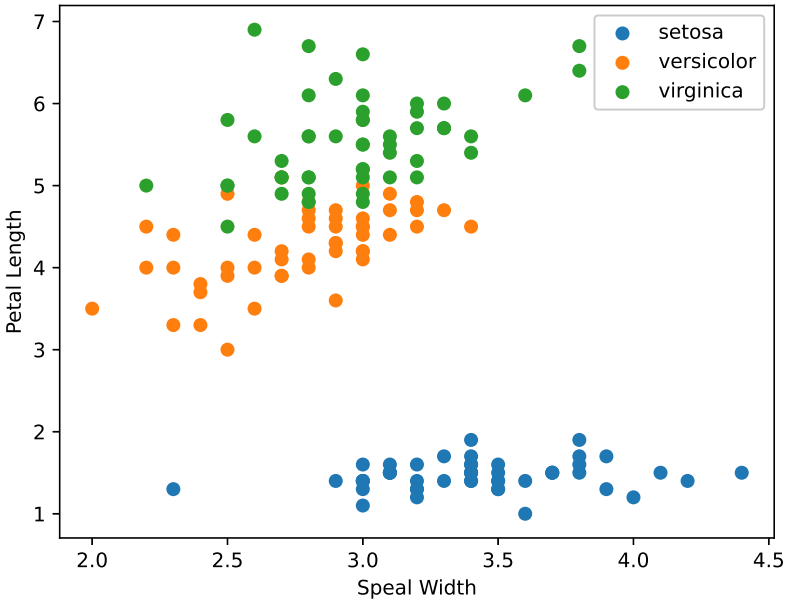

Figure 3.4: Sepal width v. Petal length

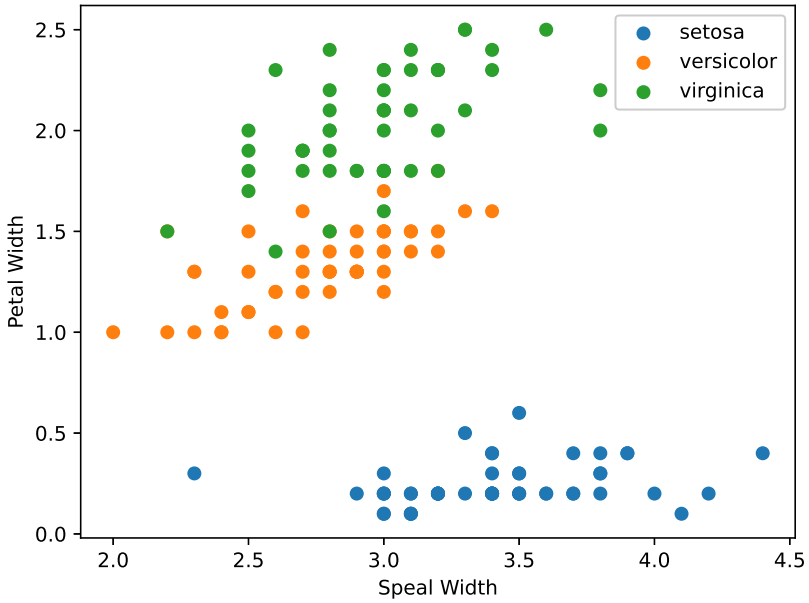

Figure 3.5: Sepal width v. Petal width

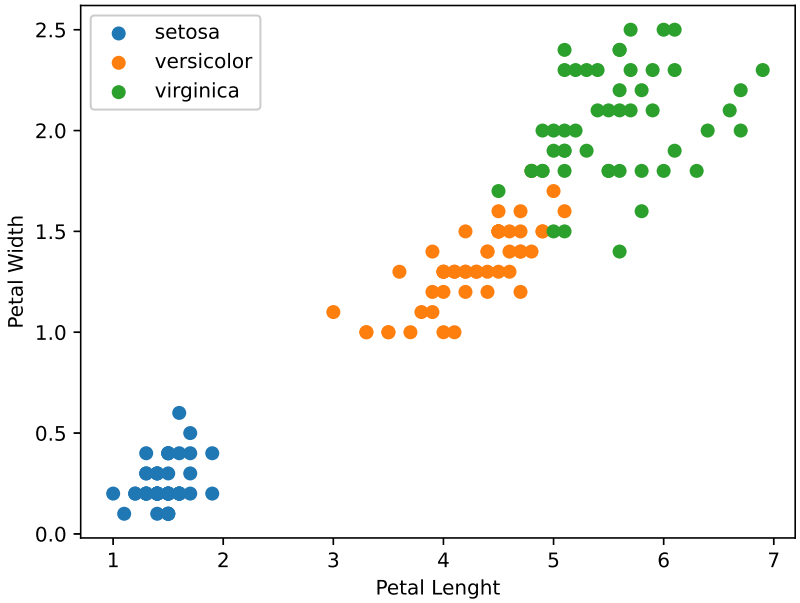

Figure 3.6: Petal length v. Petal width

There was not a large difference in scale between the features so scikits standard scalar was used.

feature	min	max	mean
1	4.3	7.9	5.843333
2	2.0	4.4	3.054
3	1.0	6.9	3.758667
4	0.1	2.5	1.198667

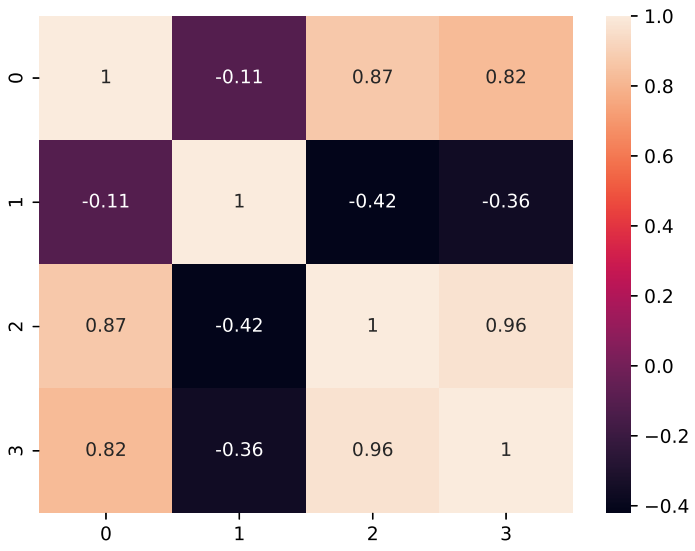

Figure 3.7: Correlation matrix of the data

Parameter search

I tested three methods, scikit learns' kernel density (KD) estimator, KNN and support vector classifier (SVC).

For the KD i tested with Gaussian and exponential kernels with h values in a range from 0.1 to 1. The other types of kernel in the image below were also tested but were unstable and worked poorly.

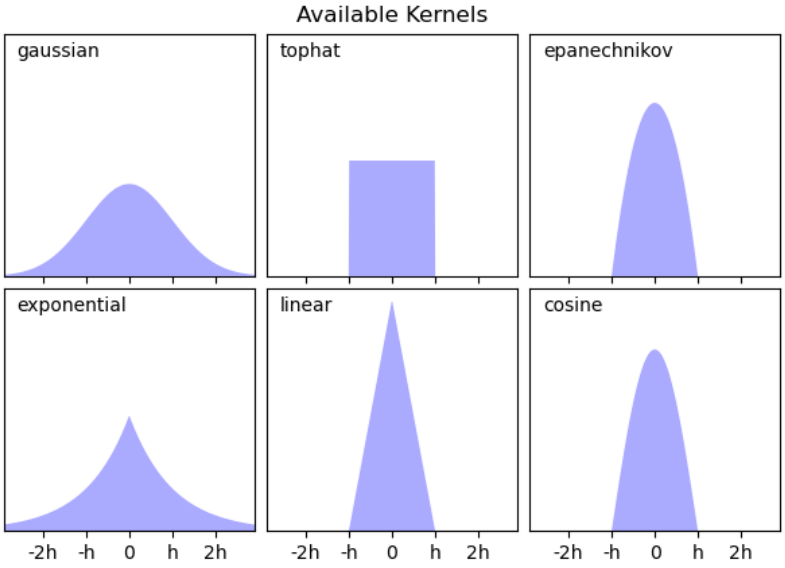

Figure 3.8: Kernel types available in scikit

The best test result for the KD estimator was achieved with the Gaussian kernel and a h value of 0.1918918918918919.

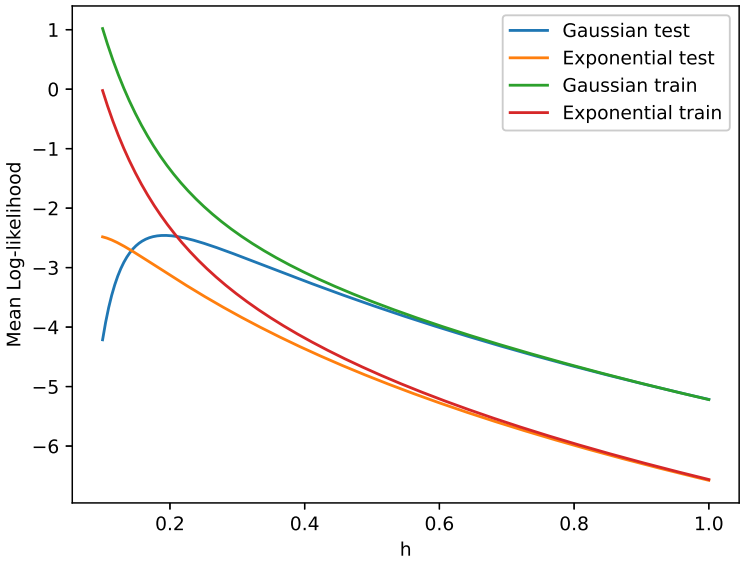

Figure 3.9: KDE scores

The best test result for the KNN classifier was achieved with a k value of 13.

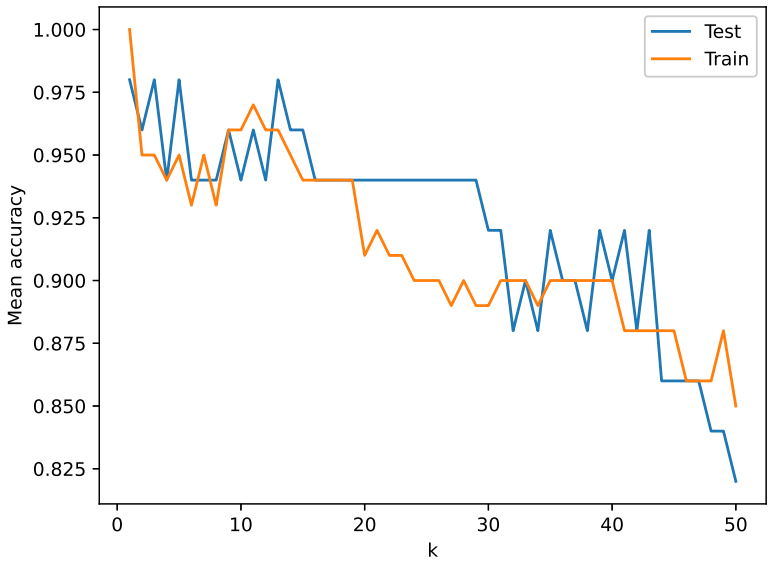

Figure 3.10: KNN scores

The best test result for the SVC was achieved with the polynomial kernel of degree 1 making it a linear model. And a gamma value of 0.12396396396396396.

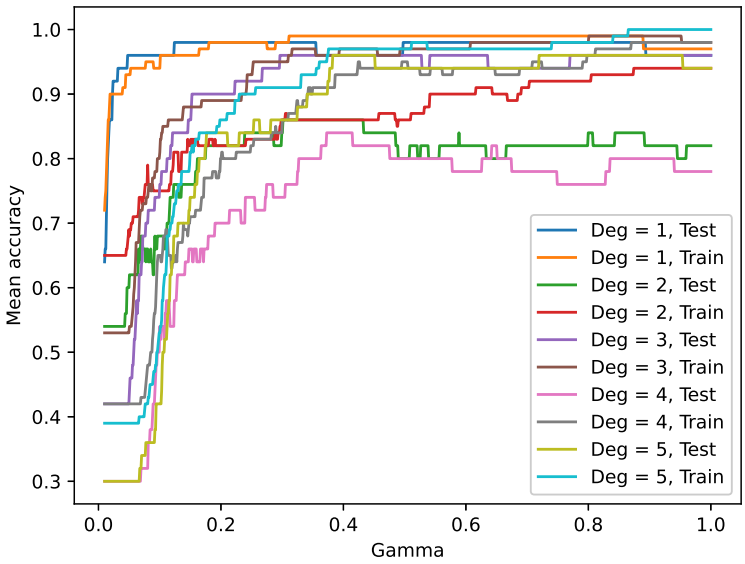

Figure 3.11: SVC scores

CHAPTER 4

Conclusion

KNN achieved a best test mean accuracy score of 0.983
SVC achieved a best test mean accuracy score of 0.966
KDE achieved a best test mean log-likelihood score of -2.61

All the models achieved extremely good results. This is not so surprising as the data sett is not very complex. The low complexity is also shown by the best fitting polynomials are the low degree polynomials with the best being the first degree polynomial.

Bibliography

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer New York, NY.

Duda, R. O., Hart, P. E. et al. (1973). *Pattern classification and scene analysis*. Vol. 3. Wiley New York.

Trevor Hastie Robert Tibshirani, J. F. (2009). *The Elements of Statistical Learning*. Springer New York, NY.