# Assignment 2 - Regression and classification

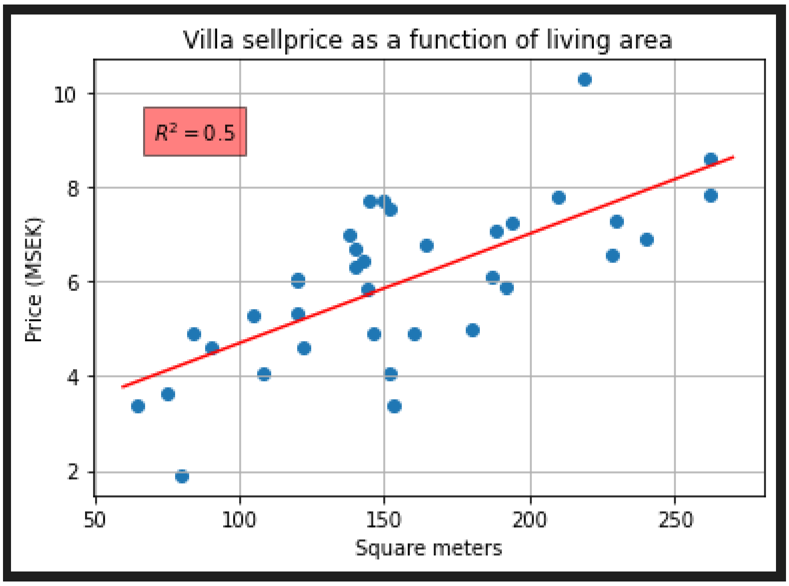
Group 62
Fredrik Lilliecreutz & Erling Hjermstad

Time spent on the module:
Fredrik Lilliecreutz: 6 hours
Erling Hjermstad: 6 hours

**1.**

**The following web page lists the selling prices of villas in Landvetter that were sold in the past 6 months. Find a linear regression model that relates the living area to the selling price. (You may transcribe the values from the web page into your program or into a data file by hand, or you can write a program to do this, but don't spend too much time doing this because "web scraping" is not the main objective of this assignment!)**
**https://www.hemnet.se/salda/bostader?location_ids%5B%5D=940808&item_types%5B%5D=villa&sold_age=6m**

**i.**
**What are the values of the slope and intercept of the regression line?**

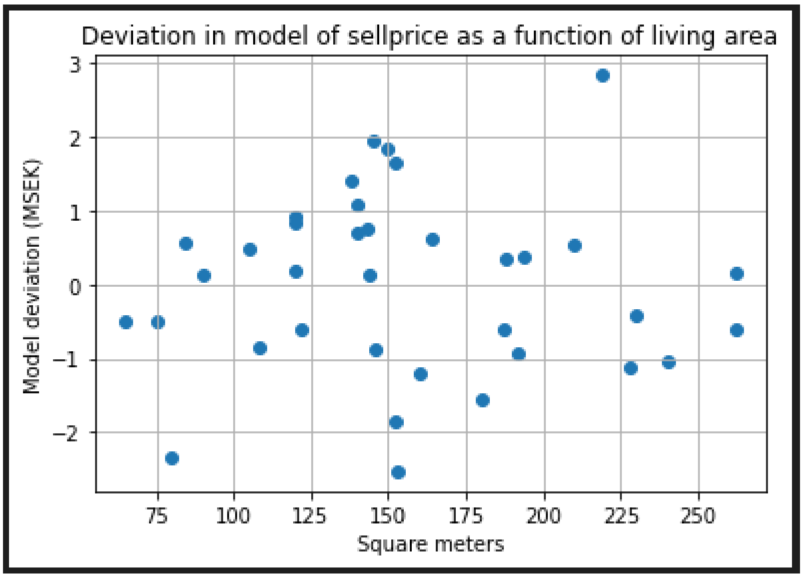


Slope = 0.023132744168234725
Intercept = 2.3907936306816584

**ii.**
**Use this model to predict the selling prices of houses which have living area 100 m^2, 150 m^2 and 200 m^2.**

Cost of 100sqm villa in Landvetter: 4.7 million SEK
Cost of 150sqm villa in Landvetter: 5.86 million SEK
Cost of 200sqm villa in Landvetter: 7.02 million SEK

**iii.**
**Draw a residual plot.**

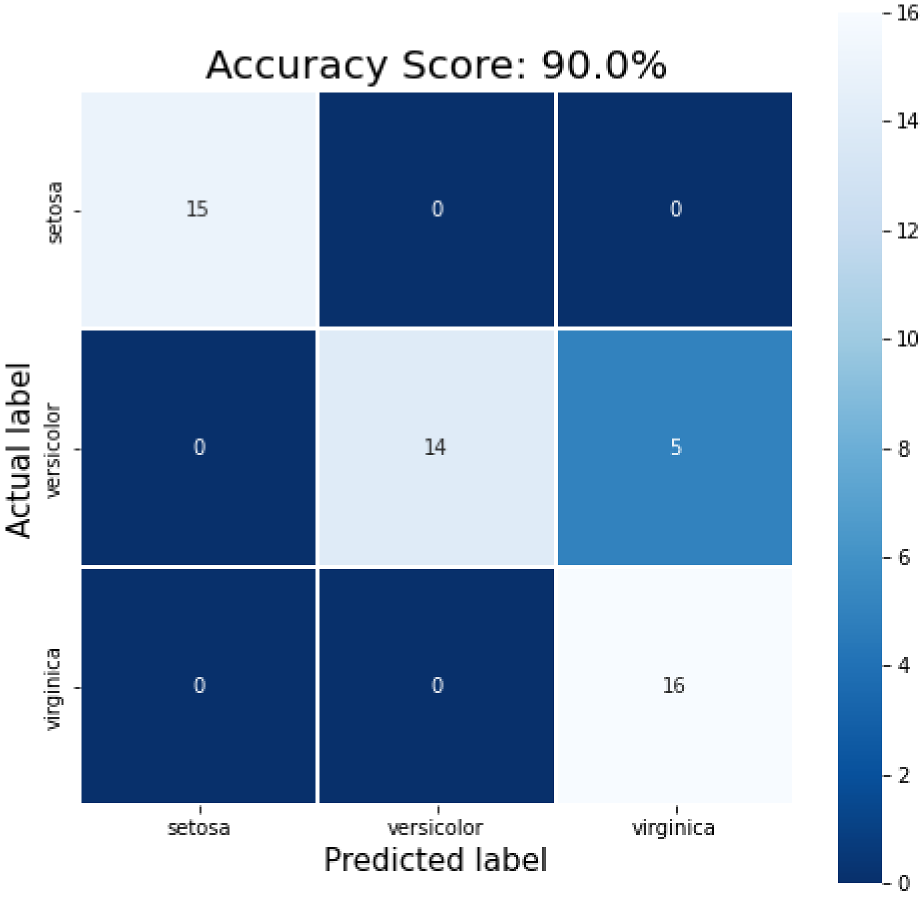


**iv.**
**Discuss the results, and how the model could be improved.**

Sum of residuals: -0.003
Mean of residuals: -0.0001
Variance of residuals: 1.4128
SD of residuals: 1.1886

The residuals are seemingly randomly distributed around zero. This indicates a good fit with linear regression. However the variance is big, which indicates that more factors should be included for more precise estimates. In this case we are just using one factor when predicting the price, which most likely is insufficient for precise estimates on an individual house. There are multiple ways of improving predictability. One factor could be to classify the selling houses to different geographic locations. The Landvetter area is quite big and just 100 meters could be the difference between living right next to the highway or having a house looking over the nearby lake. This could be implemented by clustering k-means. How big the gardening areas are could also have a significant impact on the housing prices, ranging from barely any garden to extraordinary ones.

**2.**
**Use a confusion matrix to evaluate the use of logistic regression to classify the iris data set. Use the one-vs-rest option to use the same setup as in the lectures for multiclass regression.**



This was one of many possible confusion matrices depending on the partition of training and test data. Rerunning the code, creating a new partition, leads to different outcomes. This was one of the worse observed results. The Setosa flowers were almost always categorized 100 % correctly, while the Viriginica and Versicolor were mixed to varying degrees. The test set was ⅓ of the entire set.

**3.**
**Use k-nearest neighbors to classify the iris data set with some different values for k, and with uniform and distance-based weights. What will happen when k grows larger for the different cases? Why?**

| | Weighting | |
|---|---|---|
| k = | Uniform | Distance |
| 1 | 0.9 | 0.9 |
| 2 | 0.8 | 0.9 |
| 3 | 0.92 | 0.92 |
| 4 | 0.84 | 0.88 |
| 5 | 0.92 | 0.9 |
| 6 | 0.88 | 0.92 |
| 7 | 0.94 | 0.92 |
| 8 | 0.9 | 0.92 |
| 9 | 0.92 | 0.94 |
| 100 | 0.3 | 0.82 |

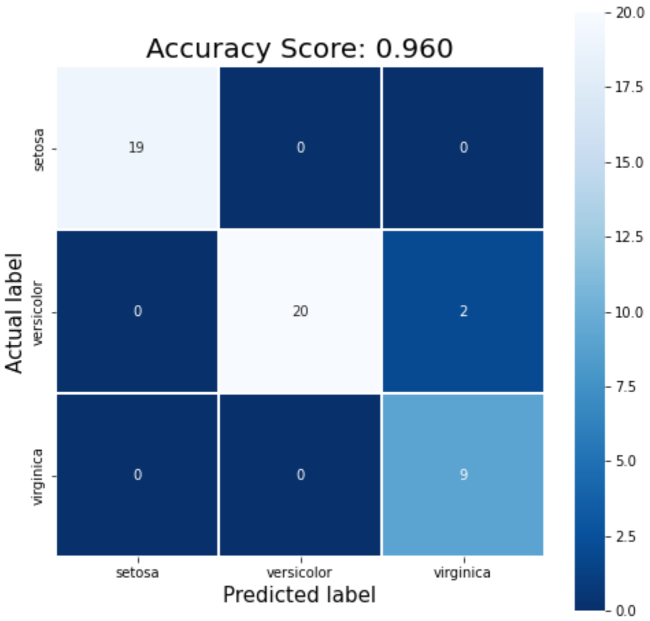
The table above shows the prediction score for a particular partition of the dataset. One can see that increasing k does not necessarily lead to better precision. When k-grows, more data points will be considered when predicting the value of a new datapoint. Too large k might lead to data points far away influencing the result. This means the model may become underfitted, not sensitive enough to differences. Too low k and the model might be overfitted, too sensitive.

We can see that the different weighting methods perform similarly for low ks, but the uniform-method's performance degrades far more when k grows large. This is because of the underfitting discussed above. The distance-method is more resilient to this, since the closer points still will have a more important vote when queried compared to the more distant ones.

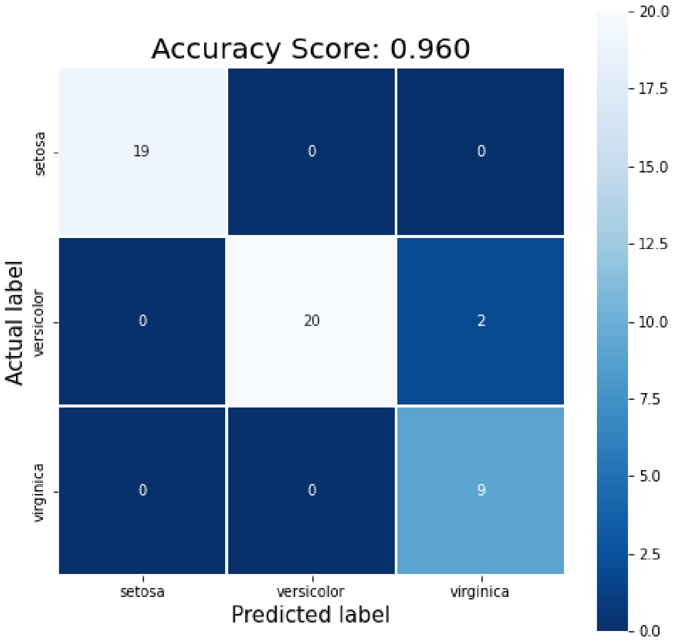
NB. The data in the table is from a different partition of the dataset than the confusion matrices were generated from.

**4.**
**Compare the classification models for the iris data set that are generated by k-nearest neighbors (for the different settings from question 3) and by logistic regression. Calculate confusion matrices for these models and discuss the performance of the various models.**

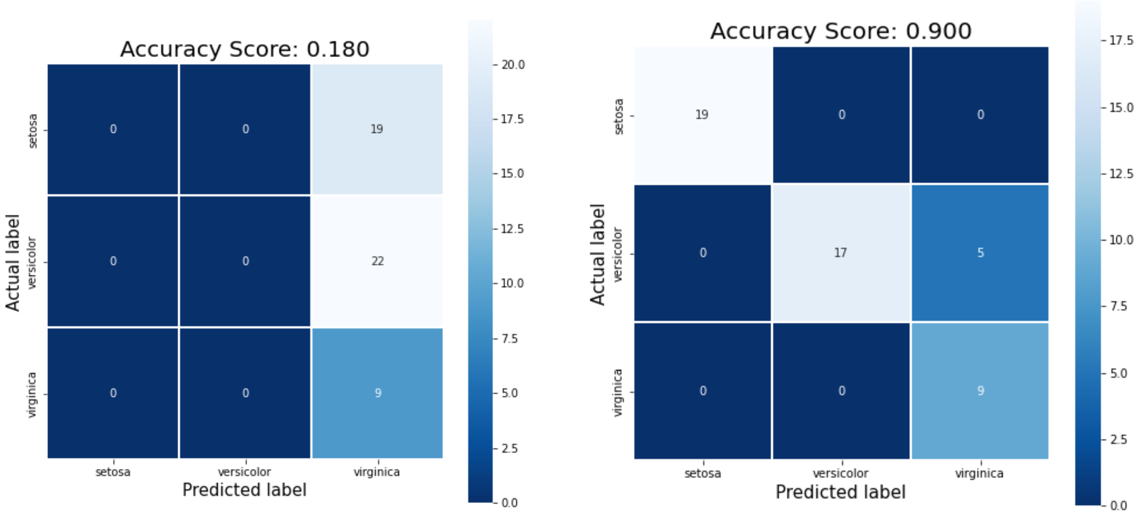
We made a total of 21 different models. 1 using logistic regression and 20 variants of k-nearest-neighbor. The variants are listed in the table of task 3 along with their performance.
The confusion matrix for the logistic regression is shown in task 2. Below are the confusion matrices for some of the knn-models.



K = 3. Same result for uniform and distance weighting.



K = 5. Same result for uniform and distance weighting.



K = 100. Model with uniform weighting to the left and distance weighting to the right.

When analyzing the knn-model results it can be seen that the performance does not increase after k = 3. As seen from the confusion matrices above, k = 5 yields the exact same result on the test-set. The performance is better than that of the logistic regression, but only for some values of k. For k = 100 the uniform knn-model predicts all elements of the test set to belong to the same category. This is to be expected due to the huge underfitting. K = 100 distance knn-model does ok, but as expected does not outperform the k = 3. This illustrates the importance of choosing a good k. If the model's training is computationally demanding, you might not want to develop multiple models and compare their performance. Then a logistic regression may be a better option, if a good k is difficult to predict.

**5.**
**Explain why it is important to use a separate test (and sometimes validation) set.**

The training set is the data used to train the model on. The separate test set is used to see how the model is performing on new data. If the test would be used on the same training data set, there is a chance that the model would have memorized some of the data and therefore give an unrealistic result. The model could be overfitted to the training set and therefore perform poorly when it is used on new data. This is evaluated by using a separate test, so called test set, to see how the model performs on a new dataset. This is meant to reflect how the model should perform on real world data. If the model performs poorly on the test set, one would need to look over the model again and make adjustments. Sometimes a third set is used, a validation set. In machine learning, training of a model can be a continuous loop of incremental adjustments to the model. It can be hard to determine when the model is good enough. This can be decided by letting the model predict on a validation set and determine from the result if the model is trained enough. The validation set can also be used to choose between different models.

When splitting the dataset into the two or three subsets there is no definitive rule as to how big each part should be. Generally the training set is dominating. For this task we used two sets, ⅔ of our data was put in the training set and ⅓ in the test set. The knn-model does not actually train. It uses the given datapoints to interpolate and categorize the new datapoints. Therefore it can be smart to keep the “training set” as large as possible to have a big enough knn-population to interpolate on. Leave-one-out cross-validation can be a good method to test and give the model a score.