# ~ Comparison between kernelized and non-kernelized methods for classification of diabetes patients ~

Silje Marie Anfindsen & Jonathan Stålberg

5/5/2021

**Abstract**

Summary of the what the report will be about

# Contents

# Introduction

https://www.sciencedirect.com/science/article/pii/S2352914819300176

use this one for the introduction and abstract.

Is it possible to build a machine learning model to accurately predict whether or not women have diabetes or not based on their health condition?

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

# Previous Work

What to write here ? about the data set? maybe history of the methods.

https://www.kaggle.com/sridm2007/svm-with-principal-component-analysis https://www.kaggle.com/uciml/pima-indians-diabetes-database

# Theory

Theory behind SVM and kernelized methods.

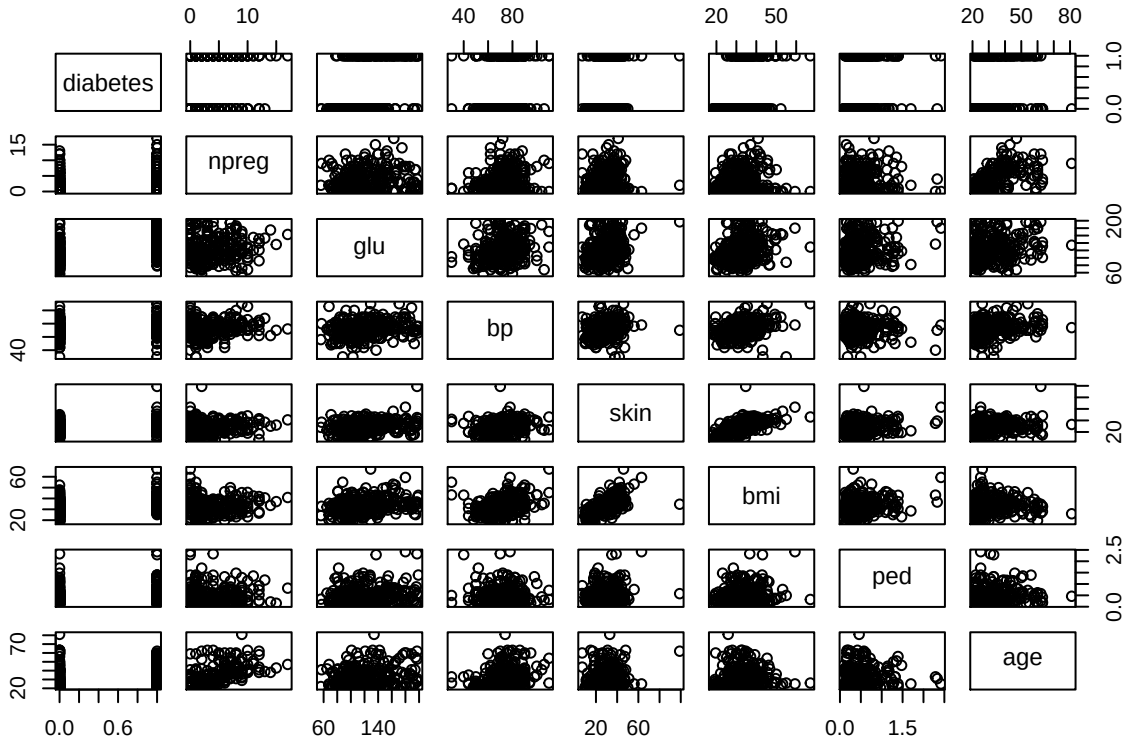
about SVM radial boundary:
https://rstudio-pubs-static.s3.amazonaws.com/296261_b5acc05b2b0e41879c917033b1497543.html

# Experiments & Discussion

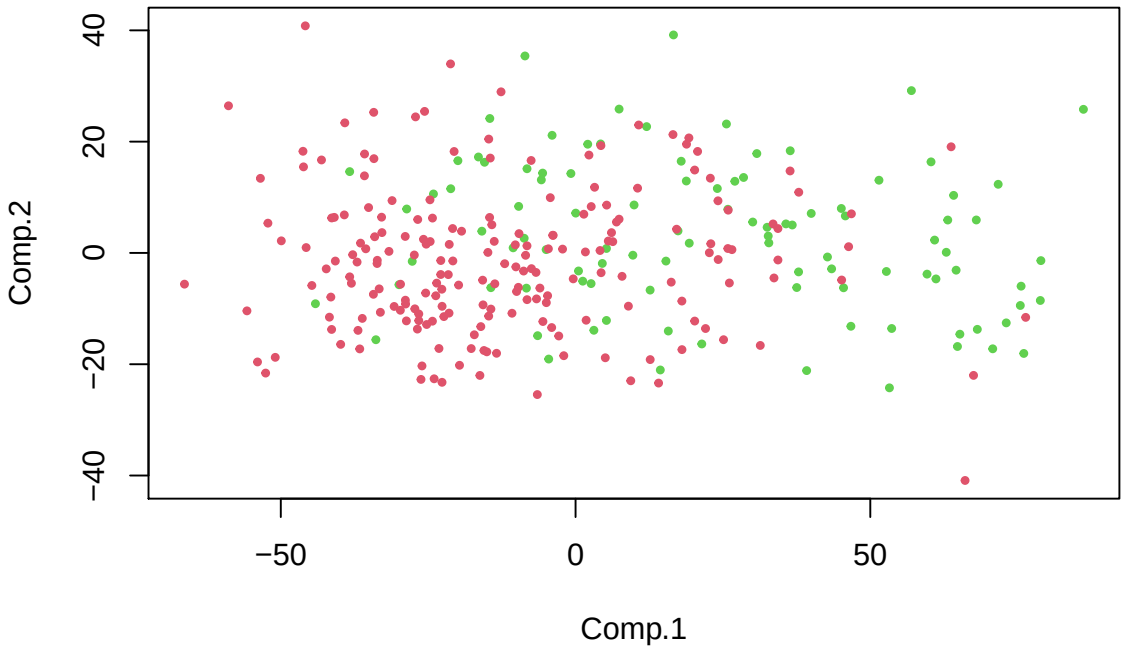
We will first take a look at the diabetes data that we have chosen to analyse. The datasets consists of several medical predictor variables and one target variable, diabetes which is a categorical variable (0,1) for whether a woman has diabetes or not. The predictor variables includes the number of pregnancies the patient has had, their BMI, diagnostic blood pressure, skin thickness insulin level, age, and so on.

Below is a print-out of the first observations of the data set and a plot displaying the relationship between each variable combination telling us something more about the covariance structure in the data set.

```
##     diabetes npreg glu bp skin  bmi   ped age
## 339        1     7 109 80   31 35.9 1.127  43
## 270        1     7 152 88   44 50.0 0.337  36
## 210        1     9 119 80   35 29.0 0.263  29
## 117        1     9 112 82   32 34.2 0.260  36
## 390        1     8 151 78   32 42.9 0.516  36
## 217        1     2  90 68   42 38.2 0.503  27
```




We will now calculate the principal components of the data set in order to visualize the data better.

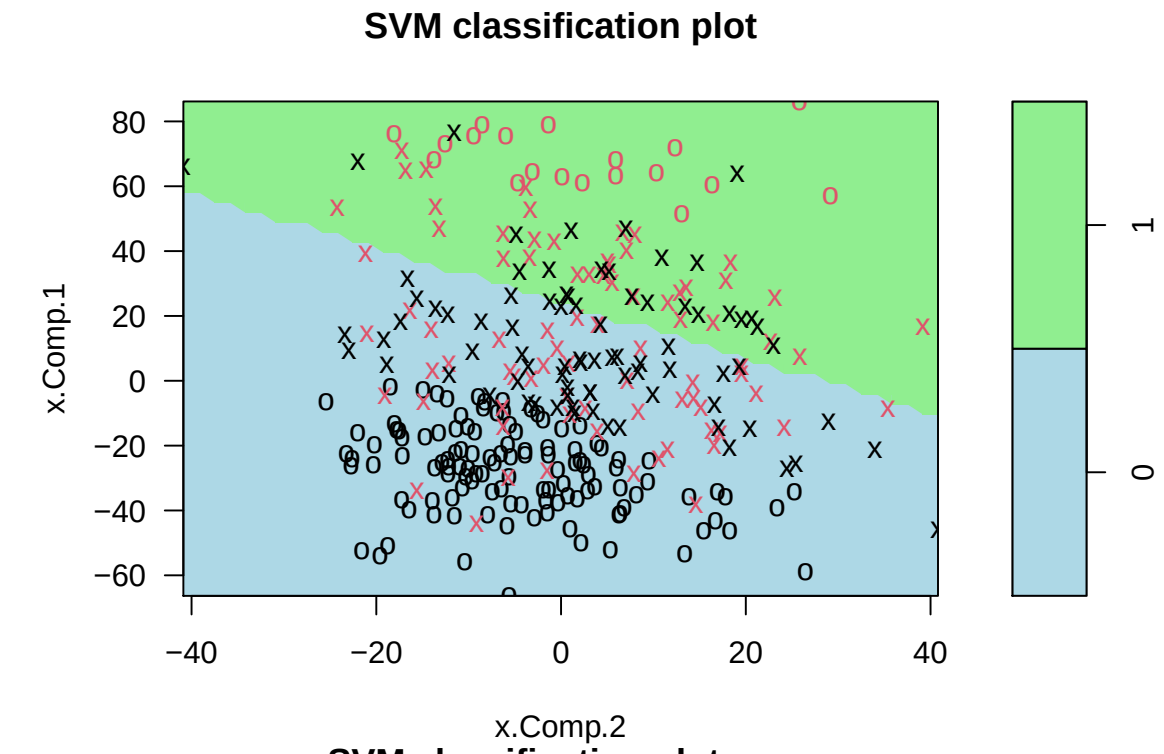


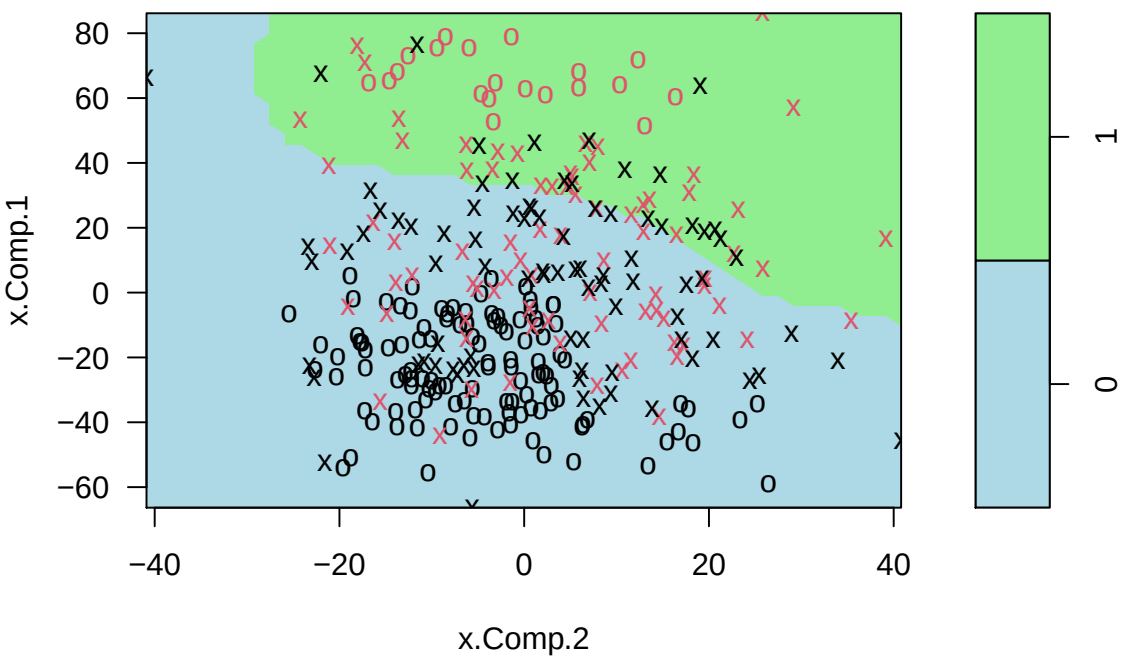

We notice that the observations in the training set can be separated quite well using the first two PCs.

| | Proportion of variance explained (%) |
|---|---|
| Comp.1 | 72.4724938 |
| Comp.2 | 11.6243865 |
| Comp.3 | 7.8304696 |
| Comp.4 | 6.1733269 |
| Comp.5 | 1.4638584 |
| Comp.6 | 0.4274691 |
| Comp.7 | 0.0079957 |

Now, in order to learn if the two first PCs manage to explain enough of the information in the model we look at the proportion of variance explained by all the eight PCs. Notice that the first two components together explain approxiatly 84% of the variance in the data set which we can be quite satisfied with at least for the next task. We will try to fit a suitable margin to the data to try if we can classify the diabetes status of a patient based on the information we have from the two first PCs.

First find a linear margin to separate the data points using SVM with linear kernel function and then a non-linear margin using SVM with radial kernel function.



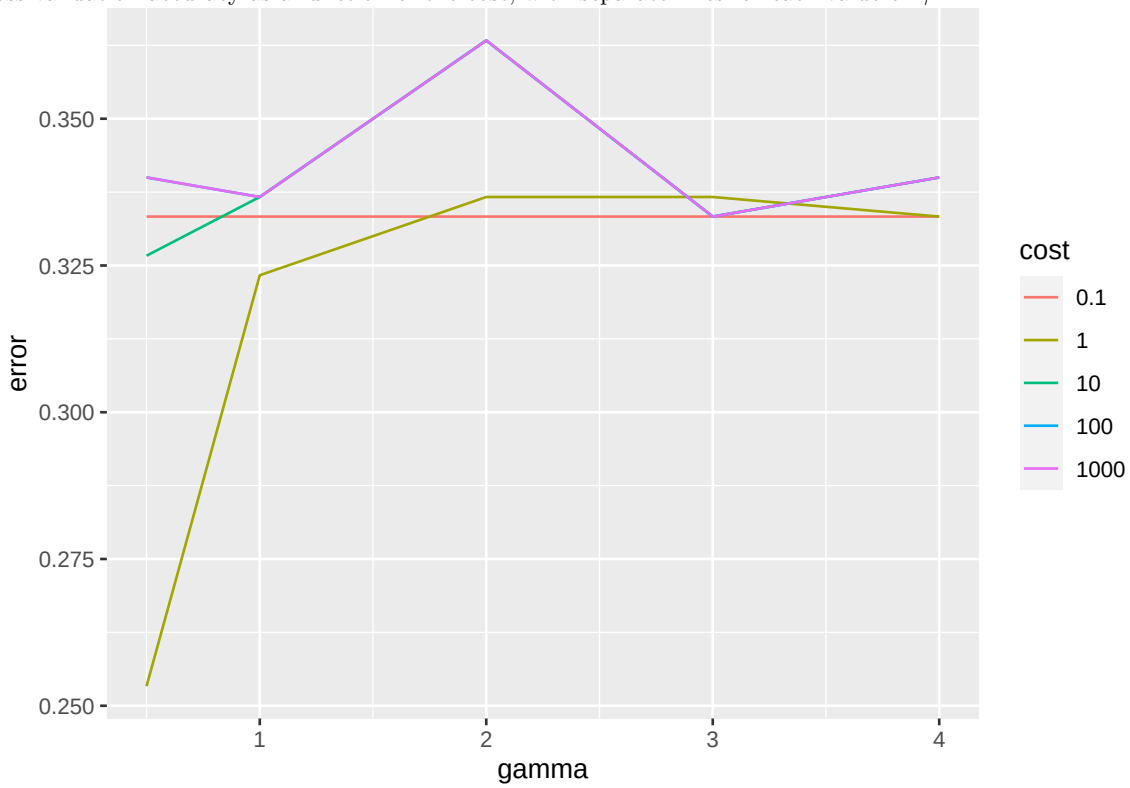

The plots above show all the observations we have from the 300 patients in the training set. The crosses are the support vectors and the boundary between the green and blue colors is the linear boundary from the classifier. We notice that both margins manages to separate most of the training observations quite well.

Now let us go back to using the full data set and not only the first two principal components as we now are going to compare the SVM with radial boundary with other known machine learning methods.

Recall that the SVM with radial boundary has two parameters. The cost which controls the trade-off between margin maximization and error minimization. The other parameter is often called a kernel

parameter, $\gamma$ which is a tuning parameter accounting for the smoothness of the decision boundary as well as the variance of the model. With a large $\gamma$ we get a wiggly decision boundary giving us high variance and possibly overfitting while for too small $\gamma$ the boundary is smoother but with lower variance.

We start by choosing the parameters for the SVM with radial kernel function by using 10-fold cross validation. To get an idea on how the two parameters influence the training set accuracy, we plot the cross-validation accuracy as a function of the cost, with separate lines for each value of $\gamma$.




From the plot above we notice that the most optimal cost is clearly 1 as it gives a small error for small $\gamma$ approx. 0.5. Thus we choose these values for our hyper parameters when fitting the model with radial boundary.

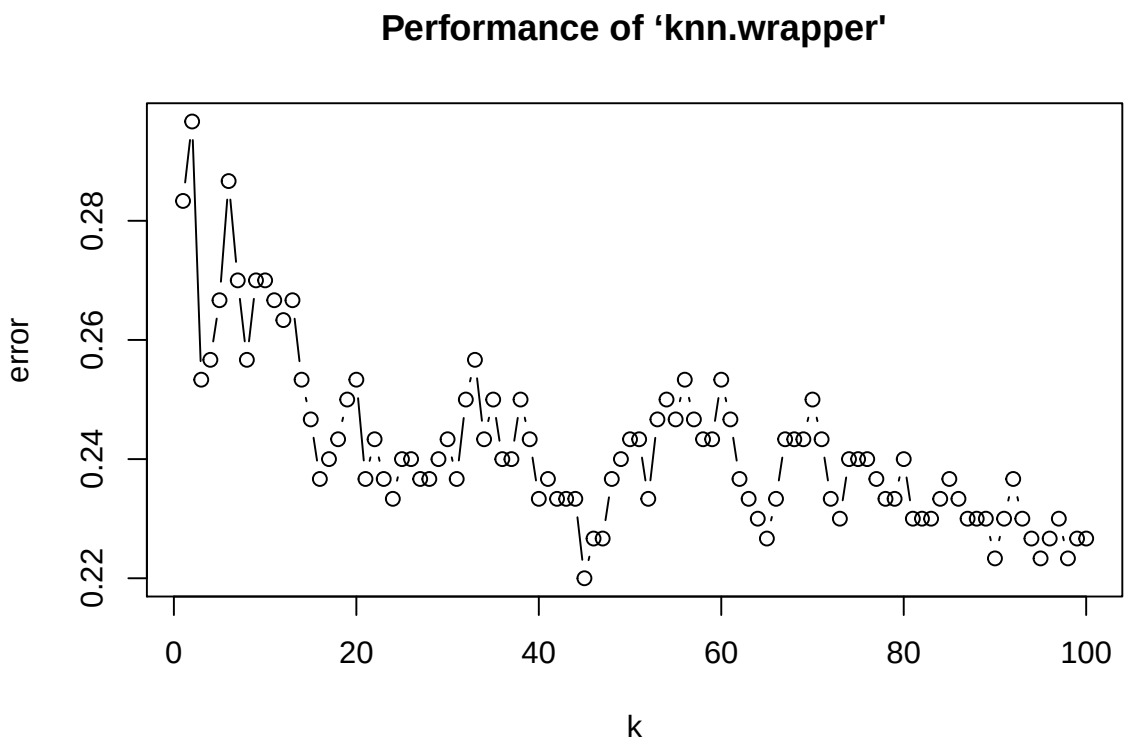
Next we will also fit a support vector with linear kernel function using CV to find the optimal hyper parmeters as we did above. In addition to the two SVMs we will fit a more statistical classification method, logistic regression and at last k-nearest neighbours.

```
## Scale for 'fill' is already present. Adding another scale for 'fill', which
## will replace the existing scale.
```

```
## Scale for 'fill' is already present. Adding another scale for 'fill', which
## will replace the existing scale.
```

```
## Scale for 'fill' is already present. Adding another scale for 'fill', which
## will replace the existing scale.
```

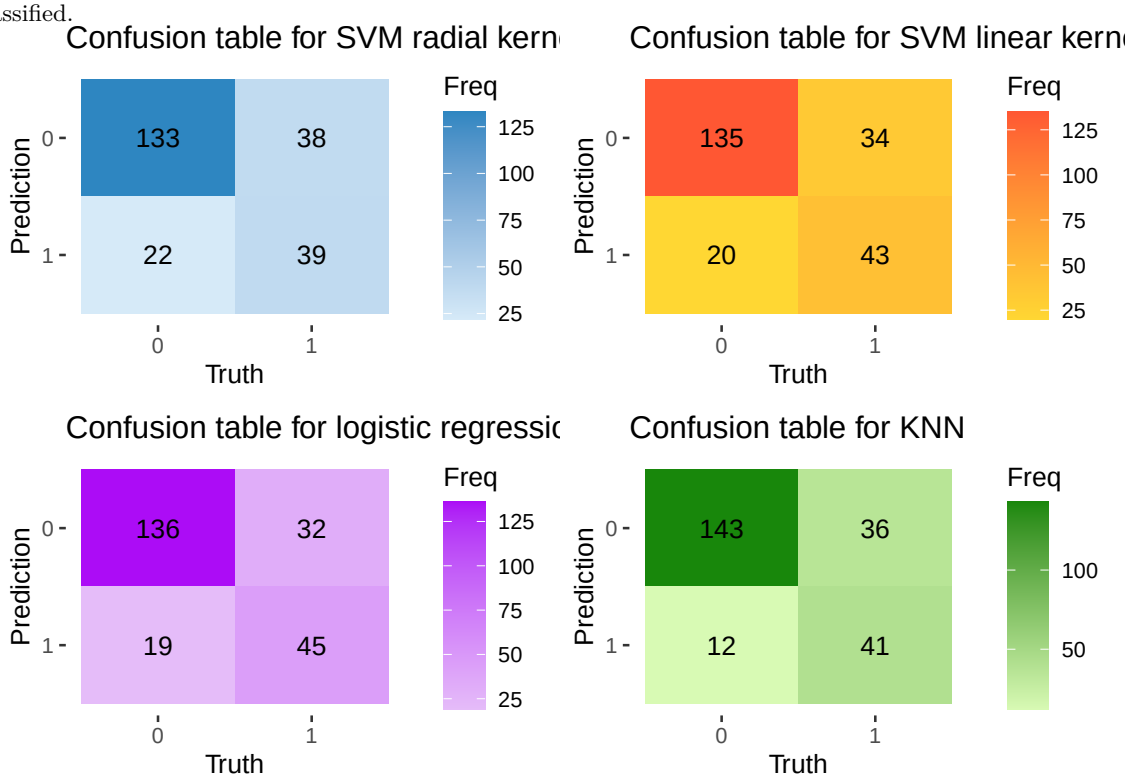
Some explanation for why we choose K here....




```
##     k
## 45 45
```

```
## Scale for 'fill' is already present. Adding another scale for 'fill', which
## will replace the existing scale.
```

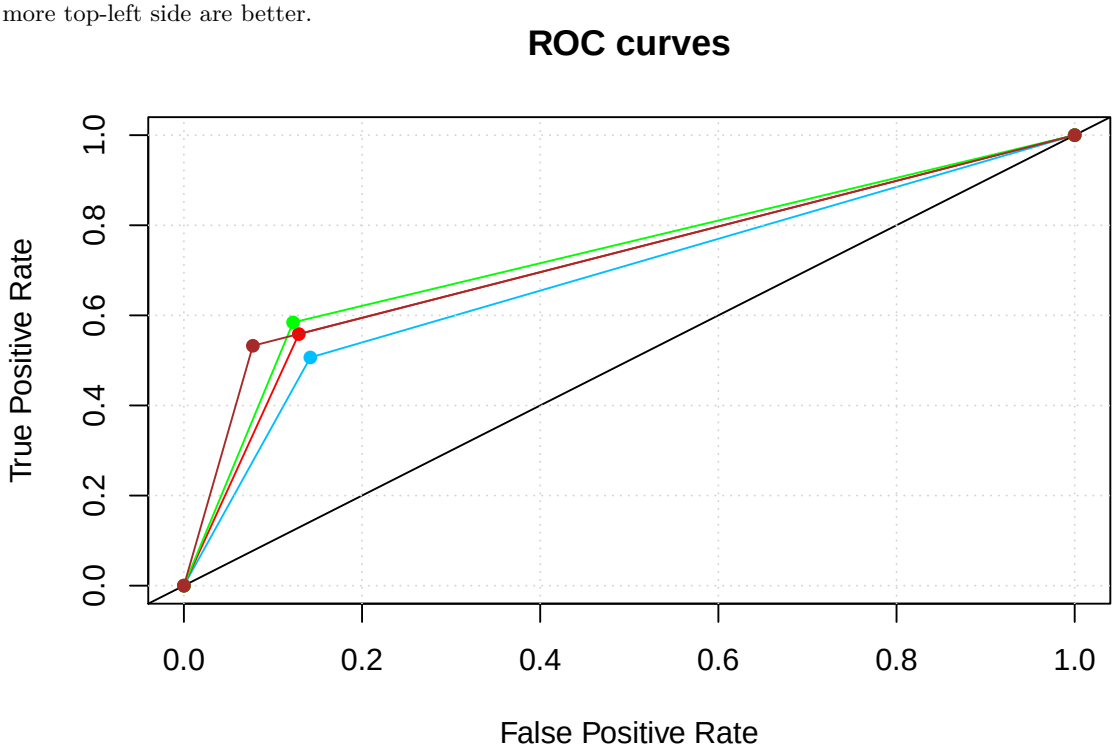
We will first look at the different confusion tables and accuracy measures for the four chosen models used. The accuracy measures how many observations, both positive and negative that were correctly classified.



| | Accuracy |
|---|---|
| svm.rad.acc | 74.13793 |
| svm.lin.acc | 74.13793 |
| glm.acc | 78.01724 |
| knn.acc | 79.31034 |

The confusion matrices show us that KNN seems to do a fairly better job classifying both true positive and true negative observations with logistic regression right behind. The two SVMs are managing to catch the true negative quite good, meaning that patients which does not have diabetes are more easily concluded to not not have diabetes compared to patients who actually have diabetes.

We will now take a closer look at the relationship between the True and false positive rate from the confusion matrices using a ROC curve. The ROC curve is a chart that visualizes the trade-off between true positive rate (TPR) and false positive rate (FPR). Here the green line is the logistic regression model, the brown is KNN while the red line is the SVM with linear kernel and the blue line is the SVM with green kernel. The higher TPR and the lower FPR is the better and so classifiers that have curves that are more top-left side are better.




We notice here that the SVM with radial boundary actually is beaten by the two linear approaches. Meaning, in this case the kernelized method is actually doing a worse job. All over we can also say that the curve is a bit far away from being in the top-left corner giving a high True Positive Rate and low False Positive Rate for any of the methods.

We can also take a look at the times spent for the four methods. Notice that the SVM with radial boundary actually spends remarkable more time to do the CV and fit the best model with optimal parameters compared to the other non-kernelized approaches.

| | Time spent |
|---|---|
| svm.rad.time | 3.3949609 |
| svm.lin.time | 0.1873572 |
| glm.time | 0.0262740 |
| knn.time | 1.8282759 |

In logistic regression all observations contribute to the decision boundary, while for SVMs, only the support vectors (the points closest to the decision boundary) contribute to the margin. A consequence of

this is that LR is more sensitive to outliers than SVM. For classes that are well separeted SVM tend to perform better than LR, while in more overlapping regimes we usually prefer LR. We also know that logistic regression produces probabilistic values, while SVM produces binary values. This can be an advantage if we want an estimation rather than just the resulting class for each observation.

We will now compare the performance of the support vector classifier and support vector machine with a new method: logistic regression. As we have observed above, our data seems to performe well when being classified timeith a linear decision boundary. Logistic regression models the probability that the response belongs to one of the two classes, producing a linear decision boundary. Therefore this method seems to be a good choice for our data.

# Conclusion and future work