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

Silje Marie Anfindsen & Jonathan Stålberg

5/5/2021

Abstract

In this report we will focus on the comparison of different kernelized and non-kernelized methods. We will do the comparison by using a diabetes data set with the label 0 if the women didn't have diabetes and label 1 if she had it. In this report we will focus on the accuracy and the computational time for k-nearest neighbours, SVM with linear kernel, SVM with radial kernel and logistic regression. For this data set we found out that KNN had the highest accuracy and the general logistic regression had the best accuracy combined with computational time.

Contents

Introduction

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?

Diabetes is a common, chronic disease caused by the fact that the body does not have enough insulin. The symptoms are not always easy to detect and connect to the disease. Approximately 5 million of the 18 million people with diabetes in the U.S. do not know they have it. With proper diet and medication the danger of the disease can be managed. But then the patient has to know if it has diabetes.

Early detection and treatment of diabetes is an important step toward keeping people with diabetes healthy. It can help to reduce the risk of serious complications such as premature heart disease and stroke, blindness, limb amputations, and kidney failure. Thus prediction of diabetes at an early stage can lead to improved treatment

Theory

We will focus on 3 different methods and they will all be described below.

K-nearest neighbours (KNN) is a classification method that calculates the distance to a fix parameter k number of neighbours. When the new point have identified which are the k nearest neighbour it checks which of the two classes that are most represented by the neighbours and then classifies the new point to that class.

Logistic regression is also a classification method that is used to calculate the probability of a class, usually a 0/1-relationship, given several parameters. When a new observation is given, the model calculates the probability for the observation to belonging to each of the classes and from that we can classify the new observation.

SVM is a supervised learning model that aims to find a optimal hyperplane which can be used to classify new data points into the correct class. This is done by measuring the similarities between points from the two classes. The most similar examples between the classes are chosen to be the support vectors which decide the position of the margin that separates the classes. In order to make the classifier more predictable for new data points the SVM can allow margin violations, this is done through the cost parameter. SVM can be used for both non-linear and linear data sets. For the non-linear sets we need to map the data into a higher dimensional feature space until the classes are linearly separable, thus we need to make us of the kernel trick to avoid alot of work and storage need for our model.

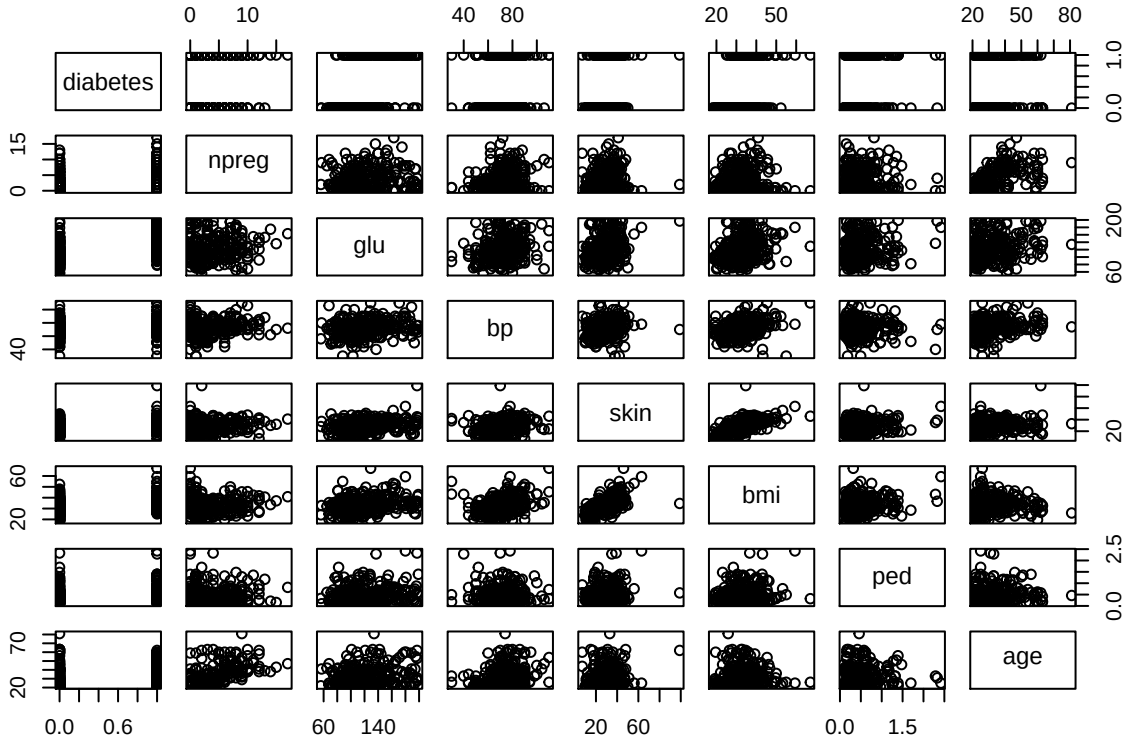
There are several reasons to choose wisely which kernel to use. A linear kernel gives us the linear SVM which is a parametric model. Compared to for example an RBF kernel SVM the latter is more expensive to train as it has to keep the kernel matrix around and the projection into this higher dimensional space. There are also more hyperparameters to tune, for example gamma which measures how far the influence of a single training example reaches. Thus it is important to know which kernel function to choose to save possible work.

Experiments & Discussion

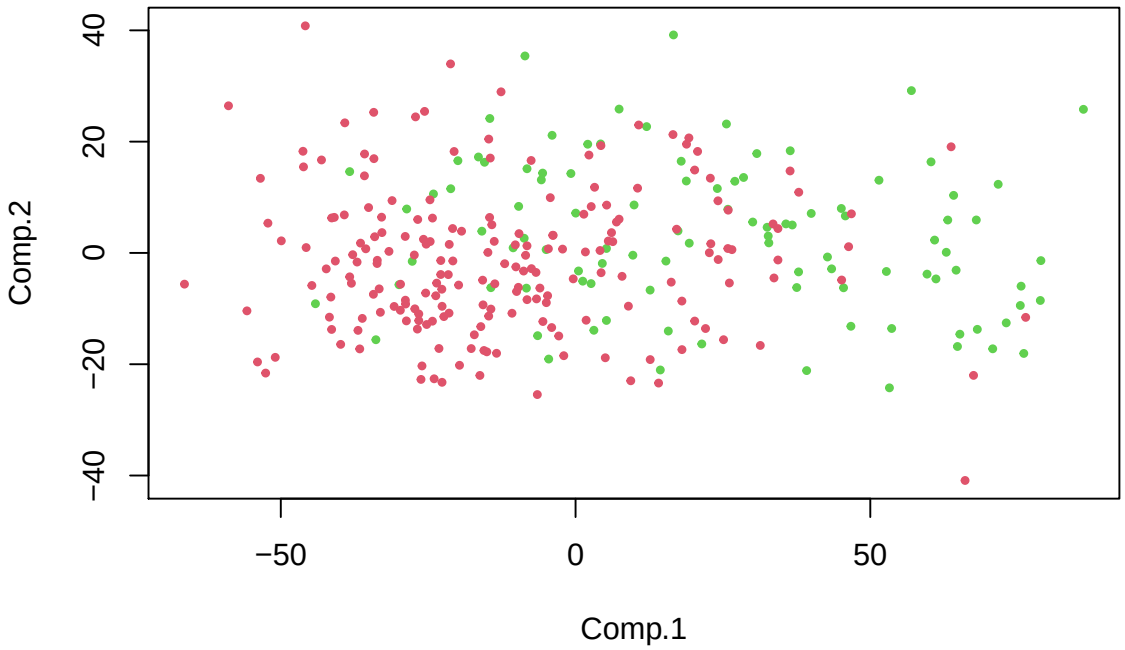
This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes (0 or 1) based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. 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.

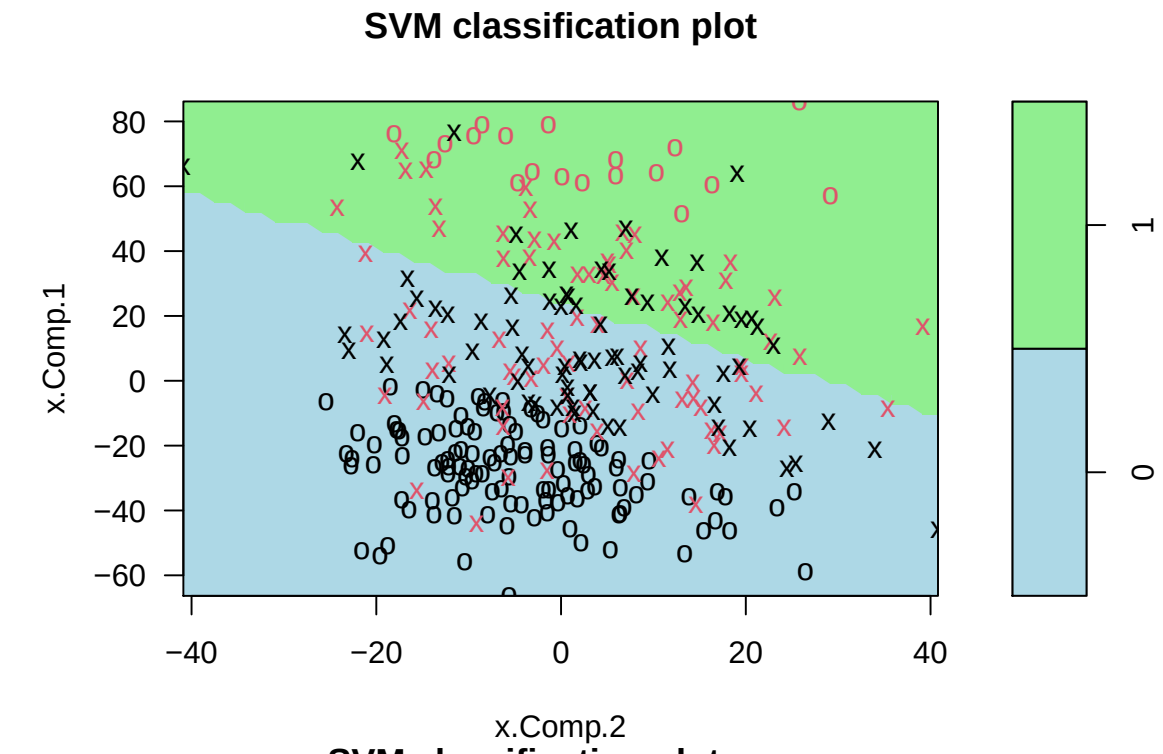


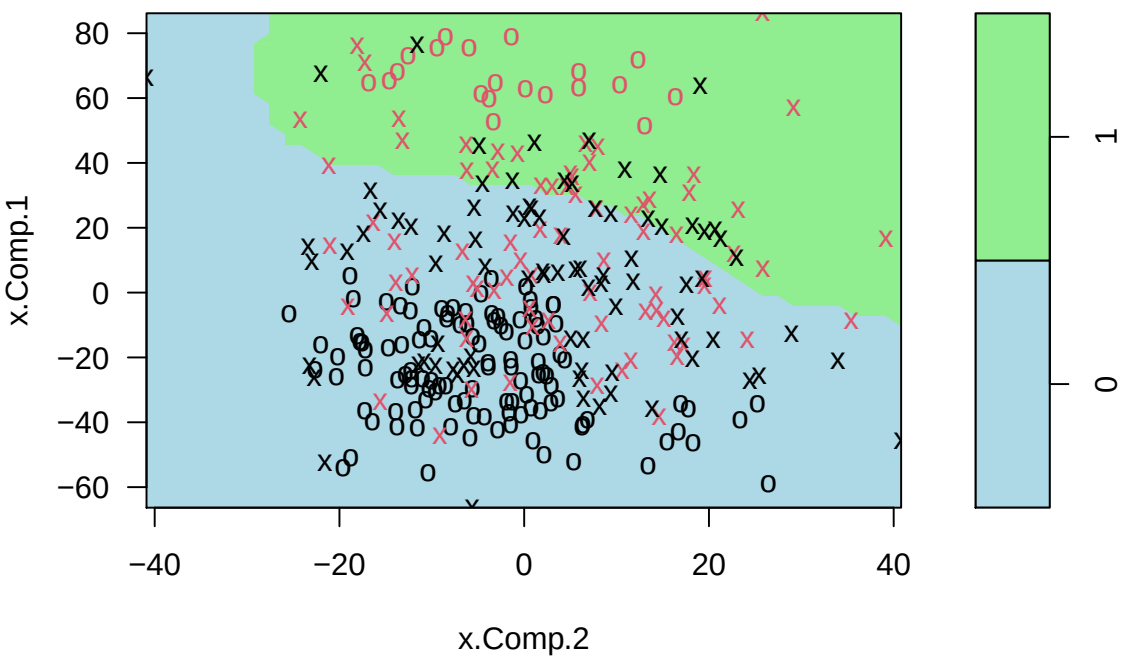
We notice that the observations in the training set can be separated quite well using the first two PCs.

	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 approximately 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.





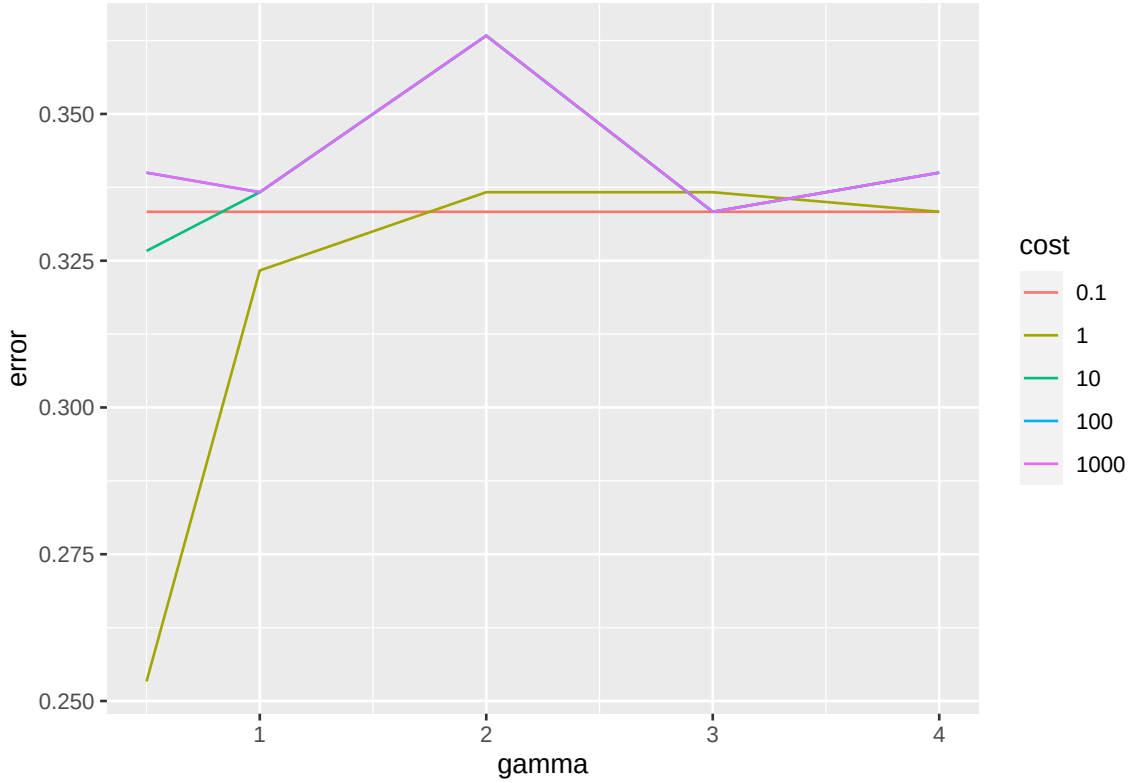
The plots above show all the observations we have from the 300 patients in the training set. The crosses are the support vectors and the boundary between the green and blue colors is the linear boundary from the classifier. We notice that both margins manages to separate most of the training observations 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 linear and 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 γ is a tuning parameter accounting for the smoothness of the decision boundary as well as the variance of the model. With a large γ we get a wiggly decision boundary giving us high variance and possibly overfitting while for too small γ 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 γ.

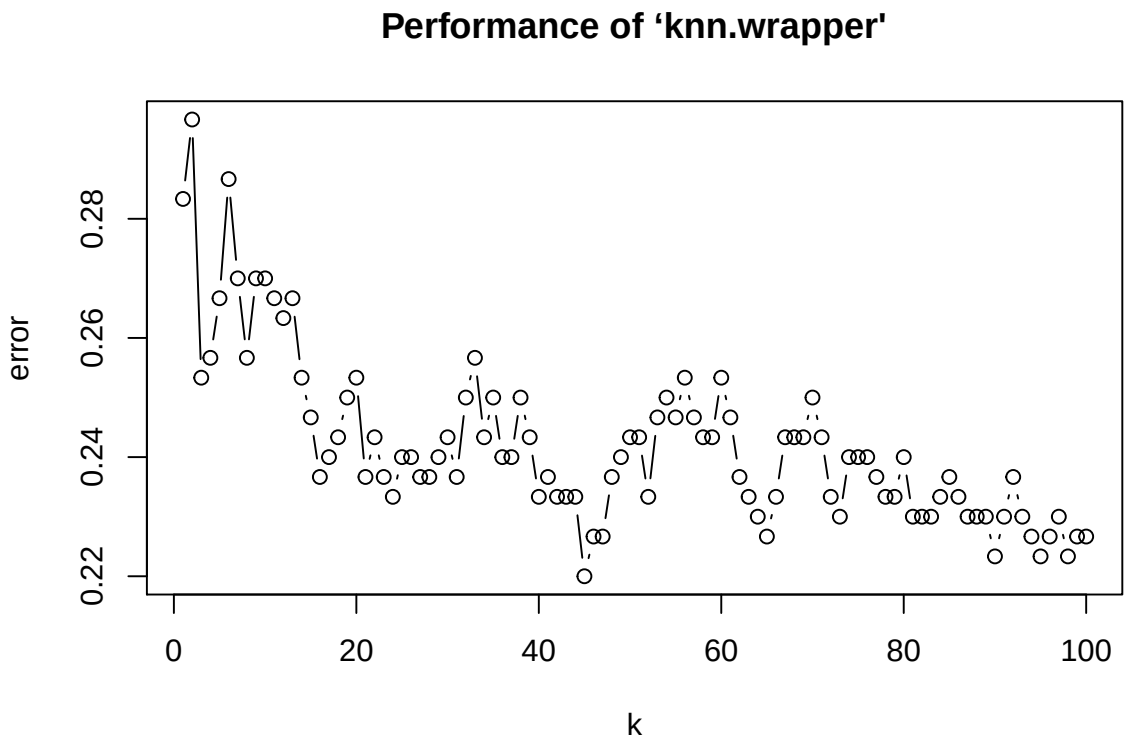


From the plot above we notice that the most optimal cost is clearly 1 as it gives a small error for small γ approx. 0.5. Thus we choose these values for our hyper parameters when fitting the model with radial boundary.

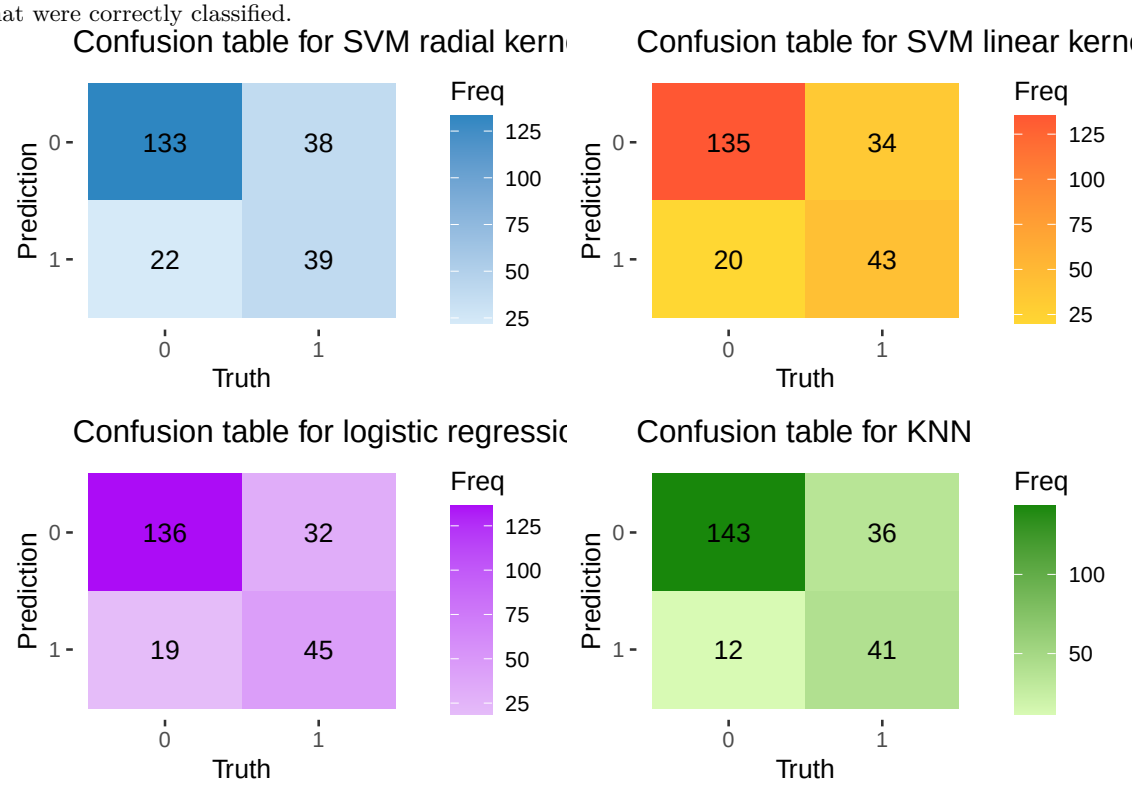
Next we will also fit a support vector with linear kernel function using CV to find the optimal hyper parameters as we did above. In addition to the two SVMs we will fit a more statistical classification method, logistic regression and at last k-nearest neighbours.

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

In order to choose the optimal number of k neighbours to fit the classifier with we use cross validation and plot the error for different k. We notice that the optimal k is 45.



To compare the four models 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.

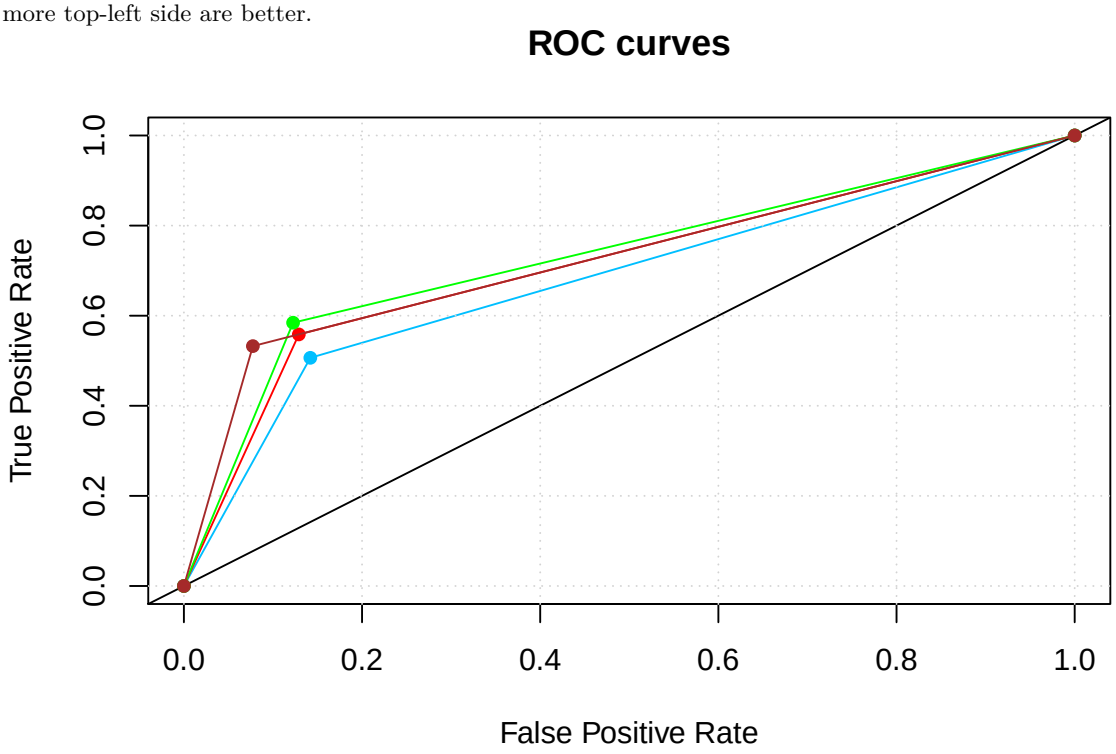


	Accuracy
svm.rad.acc	74.13793
svm.lin.acc	74.13793
glm.acc	78.01724

	Accuracy
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 radial 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. Also KNN use a lot more time then the fastest algorithm logistic regression.

	Time spent
svm.rad.time	2.9994931
svm.lin.time	0.1009009
glm.time	0.0060909
knn.time	1.5048981

Based on both the accuracy of prediction and time spent logistic regression seems to be a good choice as a classifier for this data set. In logistic regression all observations contribute to the decision boundary, while for SVMs, only the support vectors 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.

Conclusion

Our conclusion for this report is that knn is the best model to fit to this particular dataset and that logistic regression is the best if you want to consider both time and accuracy. This can seem surprising since generally you find kernelized methods that have better accuracy than non-kernelized. This conclusion may be from two reasons, one is basically that the kernelized methods we choose are not the best fit to this dataset and the second one is that sometimes LR performes better than SVM when the data is not so easily seperable.