Open your mind. LUT.
Lappeenranta University of Technology
Spring 2019

# Fuzzy data analysis

# Practical assignment with Bank dataset

Joni Kettunen

# Contents

# Introduction

This paper is a documentation for the practical assignment done for Fuzzy data analysis course in Lappeenranta university of technology. The first part of the paper is data wrangling where the data pre-processing methods are gone through. The same processed dataset is used for every model used in this data, but normalization methods are slightly modified between models.

Second part of the paper consists results and methods which were used to make classification based on the data. The used classification methods include Fuzzy K-nearest neighbours (FKNN), normal K-nearest neighbours (KNN), Similarity based classification, Decision tree-based classification, logistic regression, SVM classification and simple linear regression classification. Classification parameter optimization process was done only for FKNN, KNN and similarity-based classification methods. Other methods were quick to implement and were attached as comparison. Decision tree method was added since it is easy to understand and analyse.

Last part of this paper consists discussion about the results and suggestions for further study. As many classification models were applied with limited timeframe in this study, the main goal was to get known to these classification methods instead of trying to find best method in terms of classification performance. The used classification methods may contain coding errors and mistakes and should not be used as template for something else.

FKNN, KNN and similarity-based classifications are done with only-normalized data, PCA processed data and FPCA processed data. The goal of using dimensionality reduction methods was to learn about the effect of dimensionality reduction on classification model performance.

**The classification problem**

The dataset contains bank customer data with 16 independent variables which are properly introduced in the bank-names.txt file attached with this paper. The goal of the classification problem is to predict whether the customer is going to subscribe a term deposit (binary independent variable).

By accessing this problem from practical perspective, it is easy to see that finding the clients who are willing to subscribe is more important than accurately classifying the not interested customers. The cost of not finding a potential subscriber from the potential client pool is higher than the customer acquisition cost of spending time on phone with a client who is not interested to subscribe a term deposit. For this reason, classification sensitivity (TP/ (TP+FN)) is preferred

over specificity. Higher sensitivity means higher proportion of the customers who are willing to subscribe is found.

The parameter optimization in this study is done by finding the parameters that maximise the sensitivity. This approach might not have been wise, since high sensitivity and low accuracy may lead to too high customer acquisition costs. However, by defining different goal for the parameter optimization would have made the progress too complicated and time demanding for the writer.

## Data wrangling

The bank dataset included many categorical variables. One of them , education, was decided to be ordinal and other were decided to be categorical without clear ranking. The education variable goes on scale 1 to 3 with 1 being primary education and 3 being higher education. The missing education data were replaced by NaN and later filled in with imputation method.

Dummy variables were created for the variables: Job, Marital, Contact, Last contact month and poutcome. Since the current month (month when dataset was created) was not known, the last contact month was defined as ordinary. If the variable consisted missing values, a dummy variable for missing job etc. was created. The unknown job can be seen to provide extra information value (the person might not give job information for a reason). However, if any other non-ordinal categorical value included missing data, the ‘missing variable data dummy column’ was not created due to unlikely achieved extra information.

Since all binary variables were in ‘yes’, ‘no’ form in the data, these had to be replaced with binary number (1, 0) values. The independent variable y was later modified to have values (1,2) depending on whether the customer had subscribed (2) or not. This was done because the template classifying models expected independent class variable to have positive class values.

Pdays variable was problematic since it included -1 values if the client was not previously contacted. Creating a dummy variable and changing the rows to 0 were considered, but these methods would have problematic since the zeros would not be compatible in terms of ordinality. As a solution fuzzy sets were used.

neverContactedF = [-100 -100 -1 -0.1]; % Taking the -1 never contacted values to this crisp set

recentlyContactedF = [0 50 100 150]; % trapezoidal fuzzy set with core 50…100 and support 0...150.

sometimeAgoF = [100 150 200 250];

aWhileAgoF = [200 250 300 350];

longTimeAgoF = [300 350 1000 1000];

For each row of Pdays variable a degree of membership was calculated for each of these trapezoidal fuzzy sets. These fuzzy sets were added as columns to the data and the original Pdays variable was removed.

Normalization using minmax method to scale variables to 0..1 scale was implemented in wrangling part, and this scaling method was used for every other than KNN, FKNN and Similarity methods. These methods used normalization using scale.m file, which scales the variables using standard deviation and variable center. This normalization method was used on course exercises and was compatible with PCA method.

The missing education values were filled using knnimpute function on MATLAB. If all of the rows which included missing values were to be removed, too much information would have been lost in the process.

The relevant m files for wrangling process are:

Wrangling.m | The main wrangling m file

Minmaxnorm.m | normalization method

removeNaN.m | removal of NaN values

to_categorical.m | Categorization mfile

# Classification models

This documentation does not go into technical process of the classification model. More specified process can be found from the code comments. Studying Implementation and performance of the classification methods taught in the Fuzzy data analysis course are the objectives in this part.

All the models use the whole bank.csv dataset, which has 45212 rows. 70% of these rows are used on training and validation sets, and 30% are used on testing set. To make the results comparable between classification models the split ratio is kept same. During the cross-validation process 50% of the data is always split into validation set. The number of cross validation splits for KNN and FKNN methods is 30 and the similarity classifier training data is split into train and validation sets 100 times.

FPCA parameters are same over models where the data pre-processing method is used. It is interesting to see if PCA processed data gains higher sensitivity in classification results than only-normalized data and if FPCA provides better results than PCA processed data.

## Normal K-NN

FPCA with K-NN provided the highest sensitivity as can be seen from table 1. However, the differences are small and higher sensitivity was achieved with lower specificity.

**Table 1.** K-NN performance table

| K-NN | | Training set | | | Test set | | |
|---|---|---|---|---|---|---|---|
| | Parameters | Acc | Sen | Spe | Acc | Sen | Spe |
| BASELINE | | | | | 0.9419 | 0 | 1 |
| K-NN | K-NN neighbours: 1<br>Independent variables: 25 | 0.925 | 0.200 | 0.970 | 0.746 | 0.170 | 0.943 |
| PCA & K-NN | K-NN neighbours: 1<br>Independent variables: 28 | 0.937 | 0.333 | 0.971 | 0.750 | 0.213 | 0.932 |
| Fuzzy PCA & K-NN | K-NN neighbours: 1<br>Independent variables: 30 | 0.934 | 0.323 | 0.971 | 0.724 | 0.251 | 0.885 |

The number of K-NN neighbours and independent variables were optimized w.r.t training set sensitivity. Since the optimization loop took over 2 hours to complete for each method, the

loop only went through including 25 to 33 variables. Smaller number of independent variables should have been tested as well, but when checking with reduced dataset the results did not seem to improve. The computations were mostly done on LUT virtual machine, which is available online for students.

From figure 1 it can be seen that the number of independent variables does not have high effect on classification sensitivity. A higher range of variables to include should have been tested. However, the sensitivity seems to diminish as the number of k-nearest neighbours is increasing. The figures for KNN with PCA and FPCA look pretty much the same, and figures for sensitivity, accuracy and specificity can be found from the plots folder.

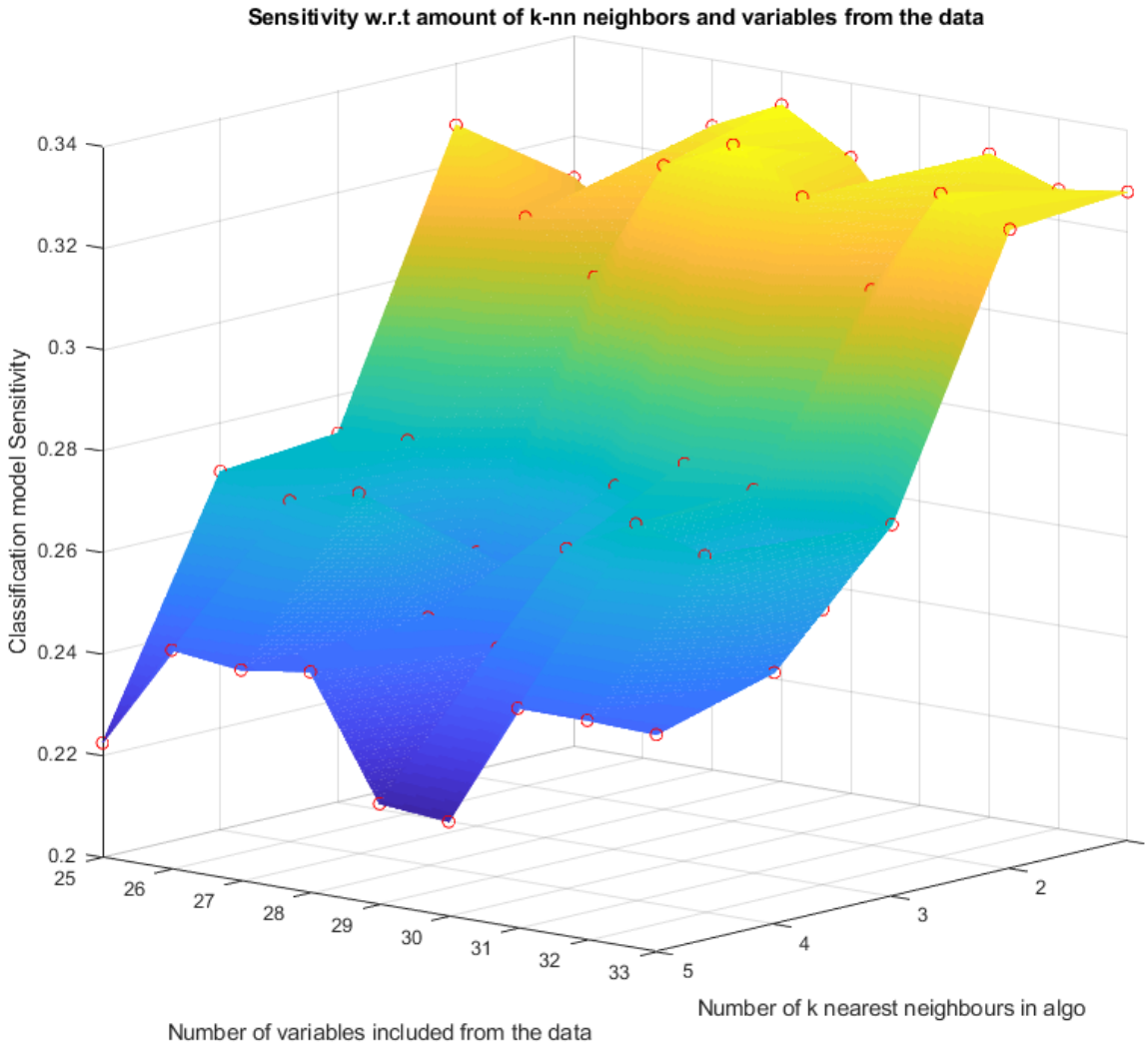


**Figure 1.** K-NN PCA Senstivity

## Fuzzy K-NN

Fuzzy K-NN algrorithm provides class-memberships values for each data row (observation). In this classification problem two categories exists, and classification sensitivity/specificity could be easily increased by increasing the treshold of observation belonging to either class. Such experiment was done by inreasing the needed membershipdegree for belonging to class 2 (subscriber), which increased model sensitivity while decresing model accuracy and specificity. This tradeoff made the model more complicated and made the modelling less comparable to other methods, so it was not included in the results.

Since each observation has a membership degrees for each class, FKNN could provide improved results when predicting classes for new observations. FKNN does not consider each known observation to have equal weight in classification progress.

The FKNN method seems to behave similarly to KNN method in terms of parameter optimization, as can be seen from figure 2.

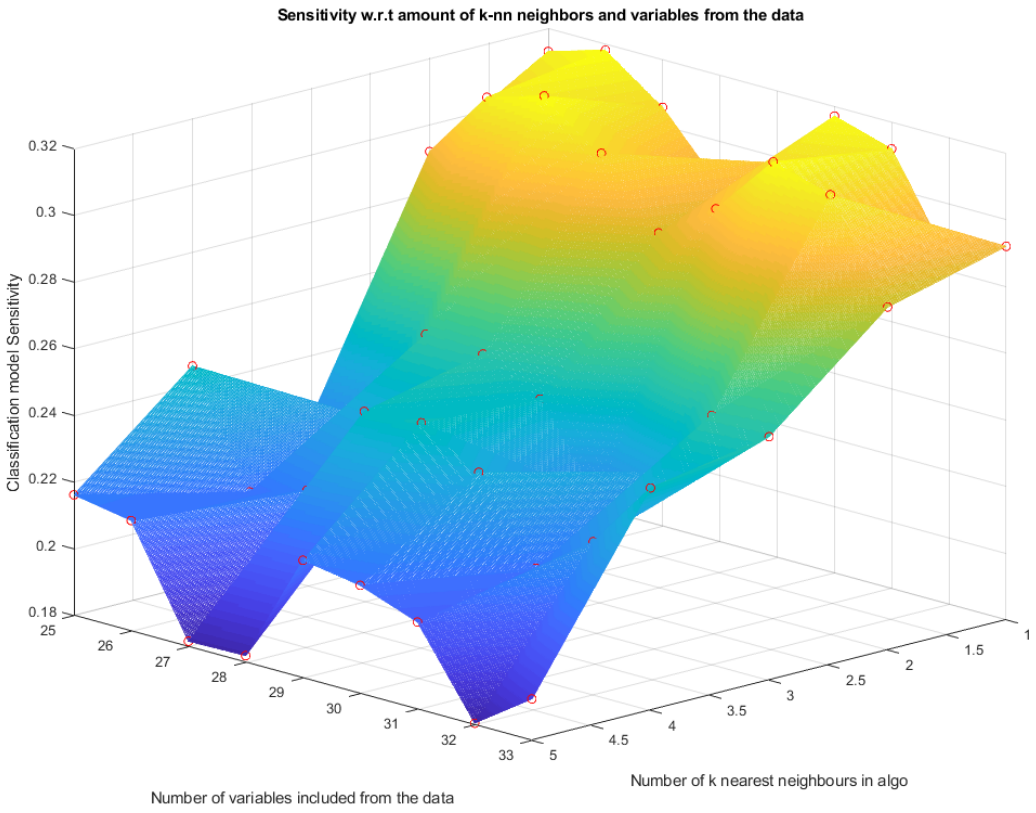


**Figure 2.** FKNN FPCA Sensitivity

Classification performance on FKNN is also very similar to KNN. However, FPCA & FKNN provided the highest sensitivity in both fuzzy and non-fuzzy k-nn methods, which might suggest that the FPCA is able to recognize some outliers in the data.

It seems that the computed columns and mostly dummy variables only add noise to the data, and even fewer independent variables should have been included in the classification process. Table 2 consists the results of FKNN classification performance.

**Table 2.** Fuzzy K-NN performance table

| FUZZY K-NN | | Training set | | | Test set | | |
|---|---|---|---|---|---|---|---|
| | Parameters | Acc | Sen | Spe | Acc | Sen | Spe |
| BASELINE | | | | | 0.9419 | 0 | 1 |
| Fuzzy K-NN | K-NN neighbours: 1<br>Independent variables: 25 | 0.923 | 0.201 | 0.970 | 0.746 | 0.169 | 0.943 |
| PCA &<br>Fuzzy K-NN | K-NN neighbours: 1<br>Independent variables: 32 | 0.933 | 0.335 | 0.970 | 0.750 | 0.202 | 0.937 |
| Fuzzy PCA &<br>Fuzzy K-NN | K-NN neighbours: 1<br>Independent variables: 26 | 0.933 | 0.318 | 0.971 | 0.718 | 0.253 | 0.876 |

## Testing FKNN optimal parameter values with reduced dataset

This part was added after the results from large dataset were achieved. Since the number of variables did not have as significant effect on model sensitivity, a larger range of parameters was tested with reduced dataset. A smaller dataset was used due to limited computational power.

The whole possible range of features to include (2:47) and a range of 1:15 k nearest neighbours were used to study optimal parameters with reduced dataset. The reduced dataset "data_whole_S" has 10% of the rows (4521) of the whole dataset. Also, the amount of cross validation splits was increased to 50. The results can be seen from figure 3.

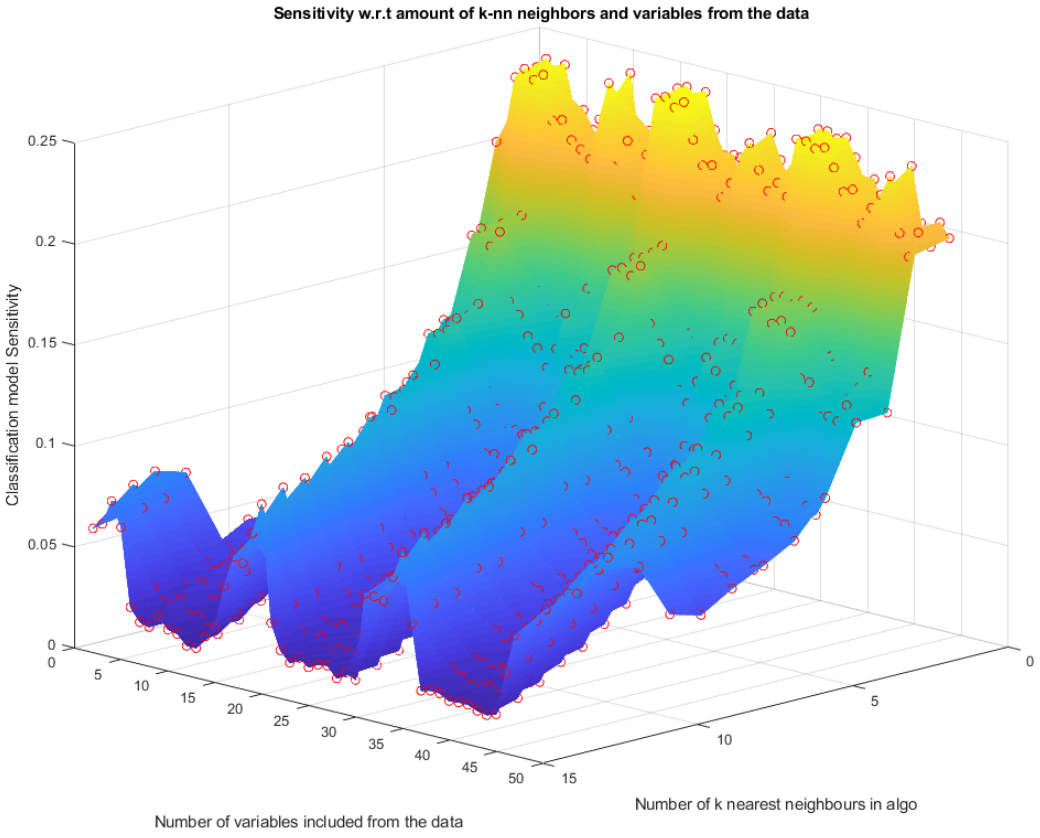


**Figure 3.** FKNN FPCA with wider range of parameters and smaller dataset

It seems that overall the number of variables (columns) included in FKNN has very little effect overall. The dataset was pre-processed with FPCA, which may explain why good results were achieved with only using few columns of variables in classification. Also, the sensitivity seems to have 'wave' like pattern.

## Similarity classifier

The similarity based classification was made using the similarity toolbox introduced on exercices as a template. This method provided very stange results, and odd optimal parameter values. Even with a loop dividing training set to train and validation sets 100 times the optimal parameter values and classification performances changed after each run. The classifiaction models are optimized w.r.t parameter in generalized Likasiewics similarity (p) and generalized mean parameter in arithmetic mean (m).

The main oddity encountered was the optimal parameter value. Since the optimal parameter values (pp and mm) are picked inside the loop that divides data into training and validation sets N times, the parameter values may not represent values that achieve maximum sensitivity on average. However these parameters are used to calculate the classification performance on the testing set. ***We discussed this subject on the last lecture.***

On figures 4 and 5 the blue dots represent the p and m parameter values which had the highest sensitivity in the optimization loop. Z axis value on blue dot represents the actual sensitivity which is achieved on the train set with those parameters on average.

The red dot in figures 4 and 5 represents the best parameter values that produce highest sensitivity on average (when train set is split N times). Perhaps due to outliers blue and red dot do not overlap on figure 4. When fewer independent variables are included in the classifiaction model, the red and blue dots do overlap in figure 5.

Overall the tradeoff between accuracy, sensitivity and specificity can be well seen from the figures. Unlike K-nn in similairy classifier changing the parameter values have drastic effect on classification sensitivity. All possible parameters are not optimized (for exapmle distance calculation method or number of independent variables to include in similarity classifier). As the classification model for similiarity method was ran, question arise whether the wrong parameters to optimize were picked for K-nn methods since the number of independent variable to include had little effect on classification sensitivity.

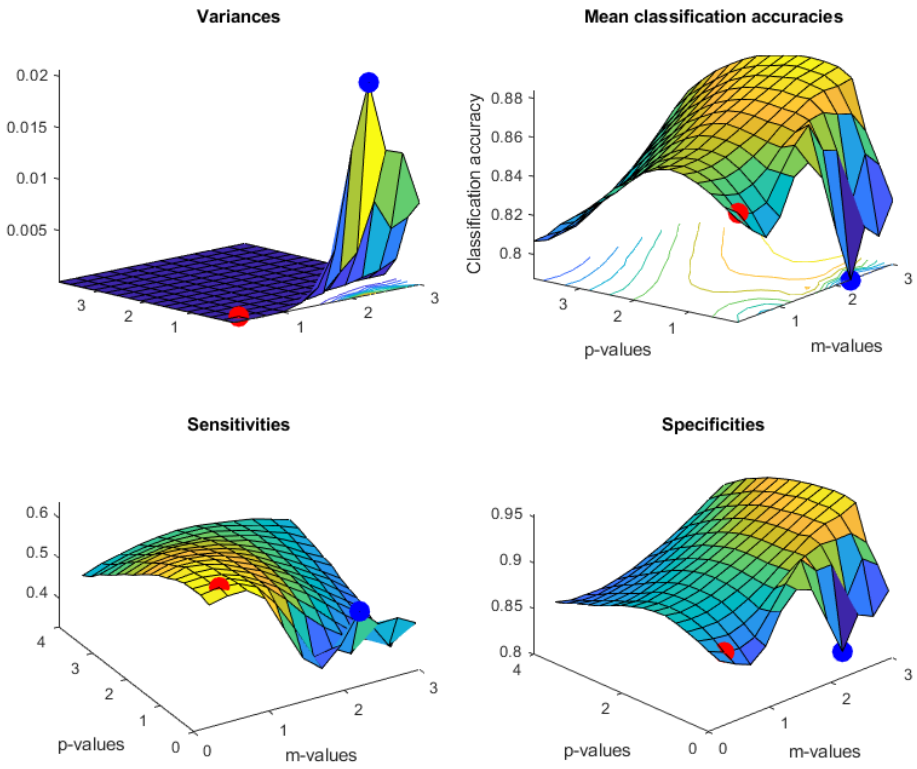


**Figure 4**. PCA & Similarity classifier with all independent variables

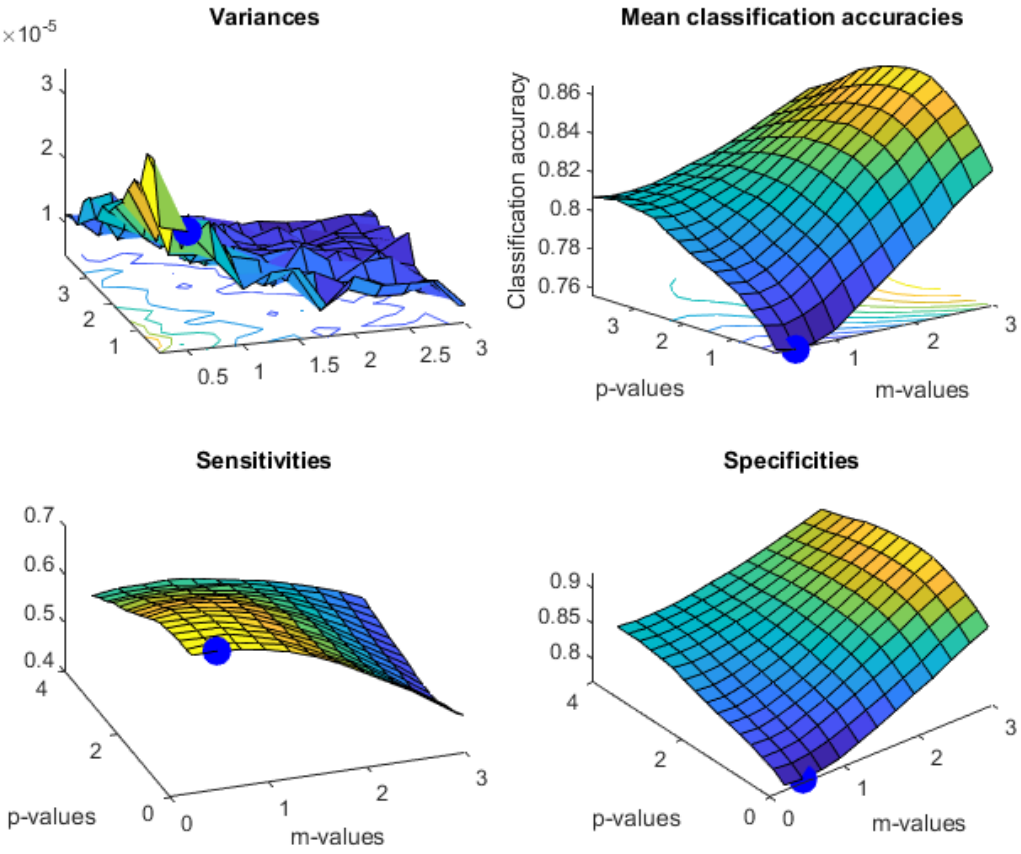


**Figure 5.** PCA & Similarity classifier with 25 independent variables (red and blue dots overlap)

Table 3 consists results from the similarity classifier results with different data pre-processing technics (PCA, FPCA , only-normalized) and with different number of features included. Overall these results are significantly better than KNN or FKNN methods in terms of maximising sensitivity of the results. With only background in this course it is hard to tell if KNN or FKNN classifications were done poorly or if this similarity classifier method provides better results in this specific case.

FPCA data pre-processing seems to improve classification performance significantly. The highest sensitivity is very good, and similar results were achieved upon multiple runs. It seems that the FPCA & Similarity with 25 columns of data can maximise sensitivity well while classification accuracy and specificity is diminished. In this case the accuracy might even be too low when considering the original classification problem, the customer acquisition costs might end up too high since the model classifies many people incorrectly to be potential subscribers.

**Table 3.** Similarity classifier performance table

| Fuzzy similarity classifier | Parameters | | Training set | | | Test set | | |
|---|---|---|---|---|---|---|---|---|
| | Blue dot | Red dot | Acc | Sen | Spe | Acc | Sen | Spe |
| BASELINE | | | | | | 0.9419 | 0 | 1 |
| Similarity classifier | P: 0.35<br>M: 3.00 | P: 0.85<br>M: 0.50 | 0.721 | 0.636 | 0.732 | 0.762 | 0.562 | 0.789 |
| PCA & Similarity classifier | P: 0.1<br>M: 2.25 | P: 0.35<br>M: 0.50 | 0.840 | 0.640 | 0.866 | 0.808 | 0.533 | 0.845 |
| PCA & Similarity classifier WITH 25 columns of data | P: 0.10<br>M: 0.50 | P: 0.10<br>M: 0.50 | 0.756 | 0.679 | 0.766 | 0.799 | 0.537 | 0.835 |
| FPCA & Similarity classifier | P: 3.85<br>M: 3.00 | P: 3.85<br>M: 2.75 | 0.826 | 0.460 | 0.874 | 0.649 | 0.626 | 0.653 |
| FPCA & Similarity classifier WITH 25 columns of data | P: 0.10<br>M: 0.25 | P: 0.10<br>M: 0.25 | 0.813 | 0.600 | 0.842 | 0.442 | **0.827** | 0.391 |

## Other methods and summary results

Other classification methods were implemented as well, but no parameter optimization was done for them. Decision tree classification method decision trees are included in appendices, which provide interesting results. Out of all variables (dummies and other since the whole wrangled dataset was included) 2 were necessary to get nearly same accuracy, sensitivity and specificity when compared to larger decision tree classification methods. Duration (the last contact duration in seconds) seems to be very effective variable alone when determining whether the customer is going to subscribe.

Overall Similarity based classifier with 25 columns of data and data pre-processed with FPCA to find outliers looks to be the method which provides the highest sensitivity in test set which can be seen from table 4**.** The trade-off between accuracy, sensitivity and specificity can be well seen from the table. However, the FPCA & Similarity classifier with 25 columns of features method provides larger sensitivity in test-set than in train-set, which tells that the sensitivity with other new data would probably not be as high. The performance features could be made more trustworthy by making cross validation on training set and test set (as was already done for training set and validation set).

The main goals for this assignment for myself were learning more about these classification methods. The results show that PCA improves classification accuracy when compared to using only normalized data and FPCA further improves the accuracy by being more robust (disregards outliers). This goes as expected from the theory of these methods. Further improvisation for classification accuracy could be easily done by tinkering with the parameters and distance calculation method, but due to limitations in time and resources the achieved sensitivity is good enough.

The bank could profit from the classification results by knowing beforehand if the customer is likely to subscribe the product (whatever the product/service is). The highest sensitivity acquired is likely to be outlier in the results as the division between train and test set was always kept the same, but according to results in table 4 a sensitivity of 60% is realistic to be achieved.

**Table 4.** Summary results of classification models

| | Training set | | | Test set | | |
|---|---|---|---|---|---|---|
| **Classification method** | **Accuracy** | **Sensitivity** | **Specificity** | **Accuracy** | **Sensitivity** | **Specificity** |
| FPCA & Similarity classifier WITH 25 columns of data | 0.813 | 0.6 | 0.842 | 0.442 | 0.827 | 0.391 |
| FPCA & Similarity classifier | 0.826 | 0.46 | 0.874 | 0.649 | 0.626 | 0.653 |
| Similarity classifier | 0.721 | 0.636 | 0.732 | 0.762 | 0.562 | 0.789 |
| PCA & Similarity classifier WITH 25 columns of data | 0.756 | 0.679 | 0.766 | 0.799 | 0.537 | 0.835 |
| PCA & Similarity classifier | 0.84 | 0.64 | 0.866 | 0.808 | 0.533 | 0.845 |
| Fuzzy PCA & Fuzzy K-NN | 0.933 | 0.318 | 0.971 | 0.718 | 0.253 | 0.876 |
| Fuzzy PCA & K-NN | 0.934 | 0.323 | 0.971 | 0.724 | 0.251 | 0.885 |
| Logistic Regression | 0.948 | 0.307 | 0.988 | 0.768 | 0.222 | 0.954 |
| PCA & K-NN | 0.937 | 0.333 | 0.971 | 0.75 | 0.213 | 0.932 |
| PCA & Fuzzy K-NN | 0.933 | 0.335 | 0.97 | 0.75 | 0.202 | 0.937 |
| K-NN | 0.925 | 0.2 | 0.97 | 0.746 | 0.17 | 0.943 |
| Fuzzy K-NN | 0.923 | 0.201 | 0.97 | 0.746 | 0.169 | 0.943 |
| Simple linear regression | 0.946 | 0.159 | 0.995 | 0.751 | 0.118 | 0.967 |
| Decision tree classification | | | | 0.757 | 0.092 | 0.984 |
| Kernel SVM | 0.946 | 0.139 | 0.995 | 0.748 | 0.025 | 0.992 |
| BASELINE | | | | 0.9419 | 0 | 1 |

Prior to this course I had not done any courses related to classification or machine learning stuff so by doing this practical assignment in a way that I explored many different classification methods I was able to learn more than if I would have focused on making one model that would have the best classification performance.

# Appendices

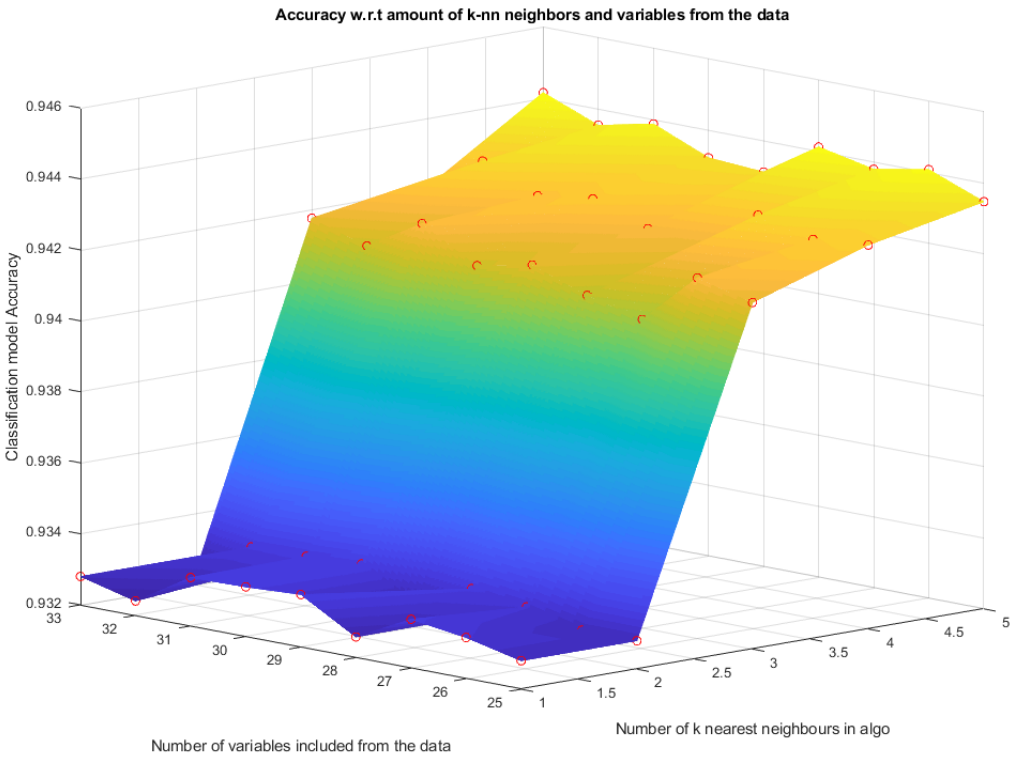


**Figure 6**. Accuracy on FKNN FPCA

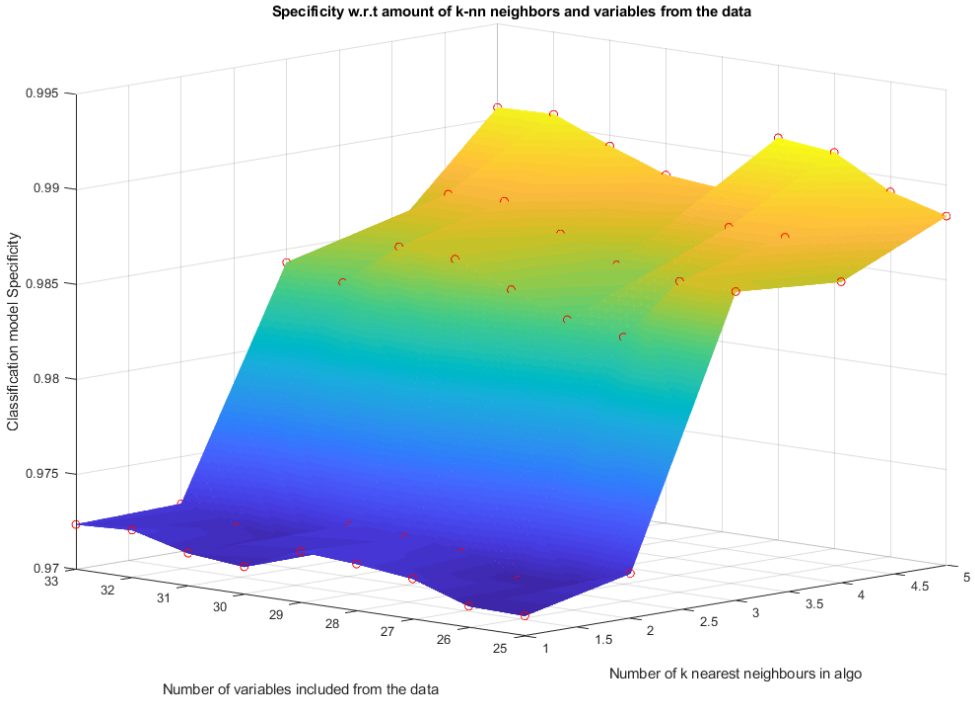


**Figure 7.** Specificity on FKNN FPCA

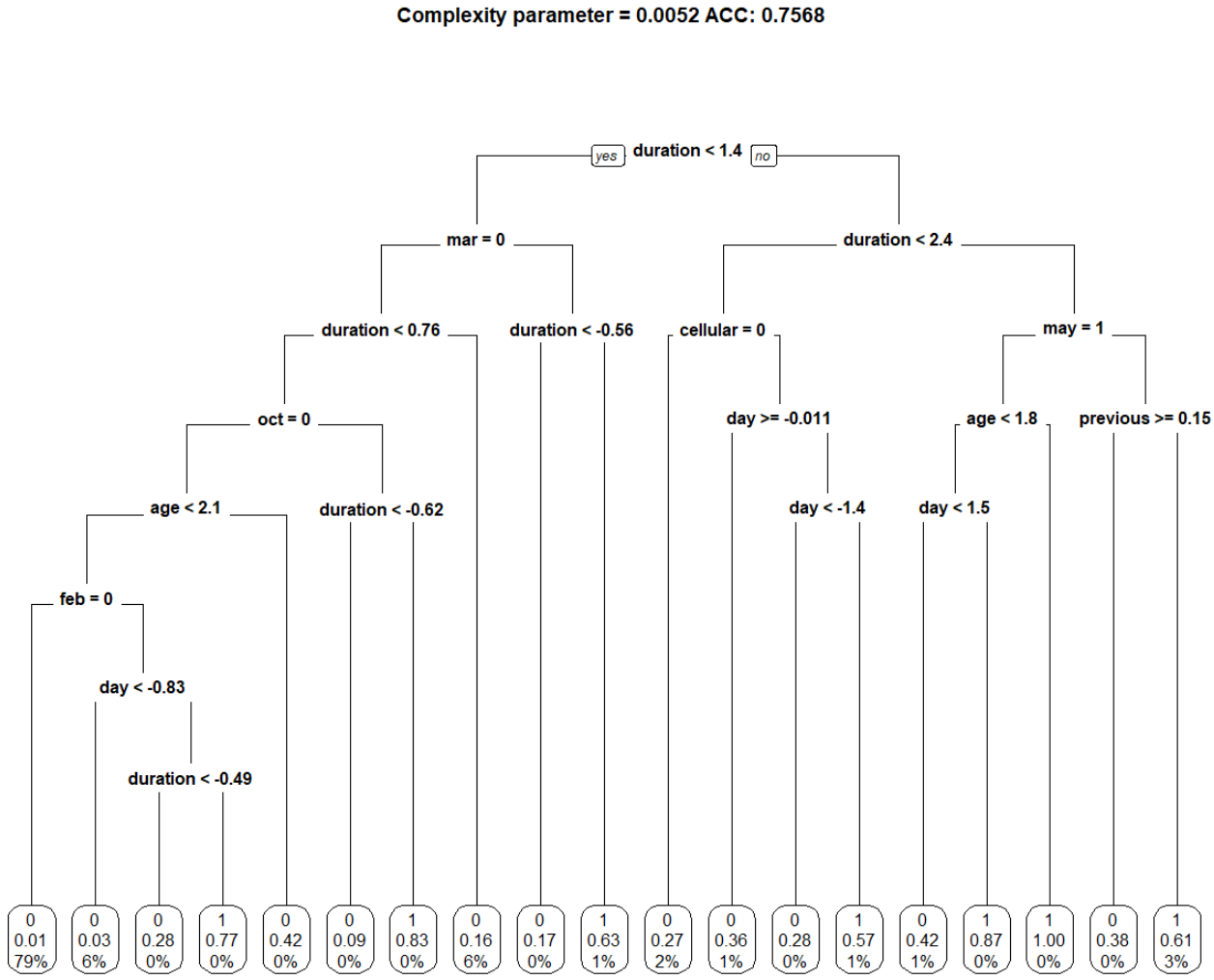


**Figure 8.** Large decision tree classification

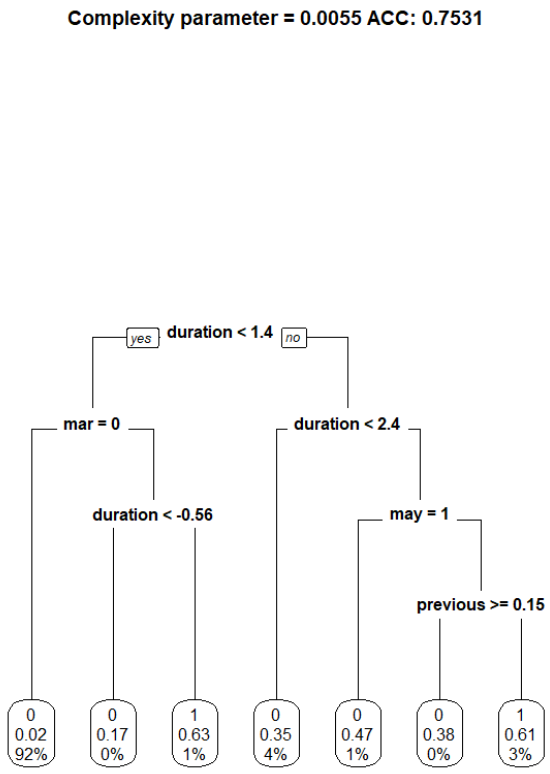


**Figure 9.** Medium size decision tree

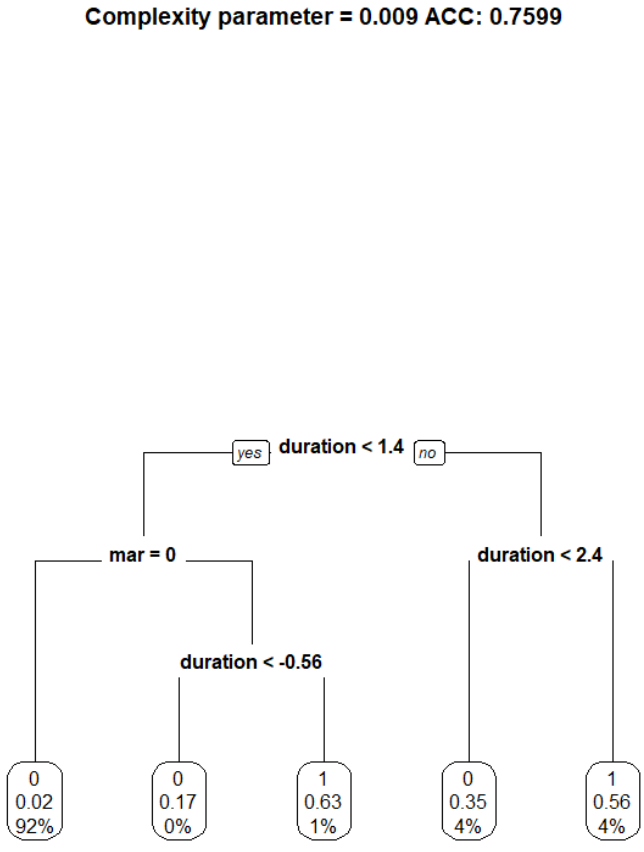


**Figure 10.** Small decision tree

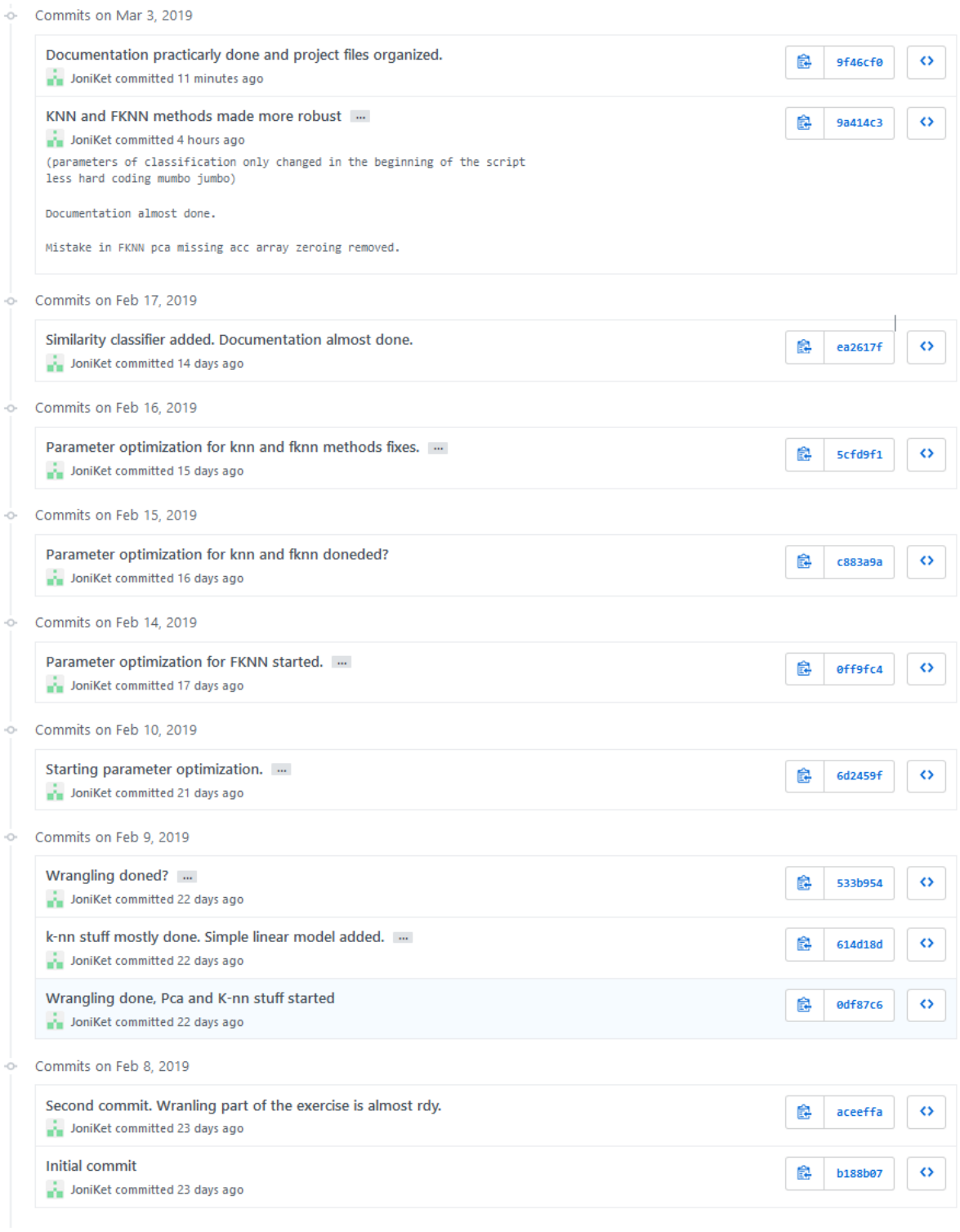


**Figure 11.** Project progress on Github. Private repository.