Learning systems

Project report

Fredrik Mårtensson

March 9, 2020

Abstract

This report is an assignment given by Halmstad University in the course Learning Systems. Machine learning, AI and data analysis provide opportunities for automation and streamlining of today's industrial society. The report analyzes six different datasets with different characteristics to predict a regression and classification model. Each model is trained with different hyper-parameters to determine which model performs best and is compared using cross-validation. The report summarizes and presents analytical data and how well each model performed on the testset.

Contents

1 Introduction

This report is an assignment given by Halmstad University in the course Learning Systems. The project is based on information from google crash course [1], Introduction to Machine Learning, Second Edition [1] and The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Second edition [2]. The report introduces results and procedures for calculating classification and regression problems. The report consists of a background describing state-of-the-art and relevant research into the work. The method consists of describing which models for classification and regression and its hyper-parameters where used during the project. Methods and algorithms for how the best model was presented and how the methods were implemented in the project. The method concludes with a description of the data and interesting discoveries during the analysis.

[1] `https://developers.google.com/machine-learning/crash-course`

2 Background

2.1 State-of-the-art

Machine learning, AI and data analysis provide opportunities for automation and streamlining of today's industrial society. By using methods such as neural networks (NN), evolution learning, deep learning or other techniques, it is possible to teach computers to interpret the environment and analyze data in a novel way. This involves scientific studies such as openAI [3, 4], Semantic Segmentation [5], Object detection[6] and image generation[7]. These are a few areas where technology can be utilized to facilitate everyday life and streamline systems. Machine learning can also be used to optimize systems such as Google's energy project [2] [3]. These are just a few of many possibilities of AI. In view of the efficiency that lies behind AI, this has made it easier for research in healthcare. Machine learning applications in cancer prognosis and prediction describe methods for predicting cancer using machine learning and propose different models to classify what is considered cancer or not. Other and more current are Deepmind which recently published models for the newly discovered virus COVID-19. Improved protein structure prediction using potentials from deep learning [8] describe models for developing protein structures. The resulting system is named AlphaFold and is also used to pick up and facilitate the research for COVID-19 which was published on deepmind's website [4]. International evaluation of an AI system for breast cancer screening [9] develops how to detect breast cancer using deep learning and presents relevant models and how it can be recreated using tensor flow.

Machine learning is something that has been around for a long time, but when google released tensorflow to the public [10], this gave the opportunity for everyday programmers to start exploring the technology. Due to the increased amount data collection, the requirements of data processing have increased. This has resulted in methods such as feature extraction, feature selection and normalisation becoming increasingly popular in order to reduce the amount of features in high-dimensional data and the construction of a more generalized model. In Feature Selection: A Data Perspective, 2017, it shows how feature selection is utilized to reduce irrelevant data. State-of-the-art methods for managing robustness within AI are becoming increasingly important in order to provide a more accurate picture of results and avoid cases for False Negative (FN) and False Positive (FP), which are two risk groups in cases for detecting cancer or within autonomous vehicles. Towards Evaluating the Robustness

[2] https://sustainability.google/projects/machine-learning/

[3] https://deepmind.com/blog/article/deepmind-ai-reduces-google-data-centre-cooling-bill-40

[4] https://deepmind.com/research/open-source/computational-predictions-of-protein-structures-a

of Neural Networks [11] describes methods to increase robustness as well as adversarial for strengthening NN models from attacks.

3 Method

3.1 Models

The tasks require models for both regression and classification. Each task consists of four different models that are tested with different hyper-parameters. These hyper-parameters should be evaluated in the training phase to determine which model performs best on the training set. The choice of models is based on presented models and solutions from the course "Learning systems" and from Scikit's documentation [5].

Regression	**Classification**
MLPRegression	MLPClassification
RandomForestRegressor	RandomForestClassifier
SVR	SVC
Lasso	LinearSVC
ElasticNet	KNeighborsClassifier

Table 1: Table of the models used for the training.

The selected models were built on sklearn model [12] to determine the relevant algorithm for creating regression and classification models. In addition to these models, a neural network model (NN) was also developed to determine how it performed against the remaining classification and regression models. Multilayer Perceptron (MLP) classifier and regression are the two NNs used to see how NN performs against the usual models.

3.2 Hyper-parameters

For each model specified in the table 1, the description and hyper-parameter used in the table 2 and 3. Each table contains the model, hyper-parameters and the values used. Each parameter is separated by commas. Due to performance limitations, only a limited number of parameters were implemented and the models that performed best (presented in results) were further tested to improve the model.

[5] `https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html`

Regression	***Values***
SVR	
C	0.1,10,100
Gamma	'scale', 'auto',2
Lasso	
alpha	0.1,1.0,2.0,5.0
max_iter	100,1000,5000
ElasticNet	
alpha	0.1,1.0,2.0,5.0
max_iter	100,1000,5000
RandomForestRegressor	
n_estimators	100,150,200
MLPRegressor	
hidden_layer_sizes	(100,),(100,50)
activation	'relu'
solver	'adam', 'lbfgs'
alpha	0.0001,0.001,0.1
max_iter	200,400
early_stopping	True, False

Table 2: Regression hyper-parameters

Classification	***Values***
LinearSVC	
penalty	'l1','l2'
C	0.1,0.5,1,1.5,2.0
dual	True,False
SVC	
C	1,0.5,1,5.2
kernel	'linear', 'poly', 'rbf', 'sigmoid'
degree	1,2,3
gamma	1,2,3
max_iter	1000,5000,10000
KNeighborsClassifier	
n_neighbors	5,3,6,8
algorithm	'auto', 'ball_tree', 'kd_tree', 'brute'
leaf_size	30,15,20,40
RandomForestClassifier	
n_estimators	100,150,200
MLPClassifier	
hidden_layer_sizes	(100,),(100,50)
activation	'relu'
solver	'adam', 'lbfgs'
alpha	0.0001,0.001,0.1
max_iter	200,400
early_stopping	True, False

Table 3: Classification hyper-parameters

3.3 Methods

The project used various methods including PCA[13], SelectKBest, GridSearchCV. These methods facilitated the investigation and management of the data. By implementing PCA, the number of interesting features could be extracted from the models. The features were selected after 95% of the interesting ones so PCA extracted enough features to meet the requirement of capturing a majority of the data. After the analysis of PCA has been performed, the number of features that met the 95% requirement is reused to pick up a feature selection through SelectKBest. The algorithm for regression was f_regression (Univariate linear regression) and f_classifi (ANOVA F-value) for classification according to Scikit learn. By plotting the distribution a good idea of the data is given and whether the information is a Gaussian, chi-squared, lognormal or uniform distribution. There are several other distributions, but these are a

few examples. When the selection of features is extracted, GridSearchCV is used to select hyper-parameters. The method facilitated the search of several parameters by contracting a pipeline.

3.4 Implementation

The implementation of the project follows the following model observed in the table 4. Stage 1, 2, 3 include the analytical section that analyzes how the dataset looks. This includes its size and warns if the dataset contains NaN or values that go between -inf and + inf. Subsequently, Principal Component Analysis (PCA), step 4, is performed to reduce the number of dimensions for further analysis. The default value is 95% to extract the majority of all features from the model. Since PCA acts as a reduction technique to model the dataset, the size of the reduced model is used to determine the number of features to use for feature selection. In feature selection, two methods f_regression and f_classifi are used by SelectKBest to develop and present which features have a major impact on the dataset.

Furthermore, in step 6, the hyper-parameters for each model are modified to determine how the model performs with different parameters in training using accuracy (classification) or mean squared error, MSE (regression). Each model is verified with k-folding of grade 10. Step 8, summarizes the best model and plots it in the form of training score and CV score, scalability of the model and performance of the model.

1	Data load
2	Data analysis
3	Preprocessing
4	Feature reduction
5	Feature selection
6	Model training & Hyper-parameter tuning
8	Summerise best model

Table 4: Flow of system

4 Data

The different datasets needs to be transposed and reshaped to get the right dimensions. If the data contains NaN, +inf or -inf or output consisted of not singular value further preprocessing needed to be performed.

4.1 Task 1

The first dataset consist of "Estimating the cetane number for diesel fuel" through a regression model. The data given is based upon infrared spectrum (IR) for each sample. The size of the set is 133 observations for training and 112 are retained for testing. The set contains a total of 401 features. The data distribution can be observed in figure 1.

The input for training and testing was transformed and the target was reshaped to fit the length of the input. Through an overview of the analysis in the data distribution it is clear to see that there are a lot of outliers but that the mean is fairly centred. Based on the data it could be assumed that the data is a gaussian distribution.

4.2 Task 2

Task 2 consist of "Modeling the need for cooling in a H2O2 process" through a regression model. The set consists of 4466 observations for training and 2971 retained for testing. The set contains a total of 65 features. The data includes timestamp that, according to the project description, were not considered a feature and were thus removed. The data distribution can be observed in figure 4.

The input for training and testing did not need to be transformed but that the target set needed to be reshaped according to the length of the input. The box plot in the data distribution appears to contain more centred data with a few outliers in the distribution. The box plot is somewhat reminiscent of a log-normal distribution.

4.2.1 Task 3

Task 3 consist of "Predicting power load" through a regression model. The set consists of 844 observations for training and 115 retained for testing. The set contains a total of 15 features. The data distribution can be observed in figure 7.

The input for training and testing was transformed and the target was reshaped to fit the length of the input. The data in the data distribution it is more like a normal distribution. The data seems to be more centered but with some outward projecting data points. According to PCA, there is considered to be only one feature that has great impact and then reduces the variance for each feature added.

4.2.2 Task 4

Task 4 consist of "Thyroid disease" through a classification model. The set consists of 5000 observations for training and 2200 retained for testing. The set contains a total of 21 features and the output is labelled as one-hot-encoding which means that the output should be converted into a numerical value between 0-2. The data distribution can be observed in figure 10.

Neither the input for training, testing nor target needed to be modified for the dataset. Task 4 contained very few features but some differed significantly from the rest. which can be observed in data distribution. Despite this, PCA decided that in order to cover 95% of the variance almost all features were needed. The variance of features appears to be quite stable between approximately 3-12 features. This could be according to the box plot where it is possible to see the compactness in the distribution between the features 2-15.

4.2.3 Task 5

Task 5 consist of "Breast cancer" through a classification model. The set consists of 400 observations for training and 169 retained for testing. The set contains a total of 30 features and the output is labelled as 0 = benign and 1 = malign. The data distribution can be observed in figure 13.

The input for training and testing was transformed and the target was reshaped to fit the length of the input. Task 5, according to PCA, had lower variance

and very closely mimicked a lognormal or chi2 distribution. Some features stand out more than others and may have taken a majority of the data points, which may have resulted in PCA resulting in a lower number of features. The distribution in the box plot also shows that several points lie as outliers and that this is frequent for each feature. It should therefore also be noted that the distribution is the lower (<25%) and higher (>25%) quantiles are quite wide and that the data contain a large spread of data points.

4.2.4 Task 6

Task 6 consist of "Electrocardiograms" through a classification model. The set consists of 200 observations for training and 100 retained for testing. The set contains a total of 312 features and the output is labelled as TI pattern (1) and a non-TI pattern (0). The data is divided into channels of 26 features and 12 channels. The hypothesis of the exercise is that in between 19-26 of each channel the most important features should be discovered. The data distribution can be observed in figure 13.

The input for training and testing did not need to be transformed but that the target set needed to be reshaped according to the length of the input. Task 6 was more difficult to analyze compared to previous data. According to the box plot, there were many points that were outliers of the data and some were very focused and included a very high density. PCA considered that just over 60 features were relevant and according to the project description, features between 19-26 in each channel would be interesting. The model for SelectKBest and f_classifi presented the following features (remaining presented in results): [5 17 38 75 101 121 158 165 168].

5 Result

Each dataset presented different results where different models presented different good depending on model. In some cases, this resulted in a lighter implementation such as random forest performing better than utilizing neural networks. The result for each task for regression is presented in a line combined with a scatter plot, which illustrates the actual value and guessed value from the model. In classification, the data is presented in a confusion matrix that shows true positive (TP), true negative (TN), false negative (FN), false positive (FP), of which FN and FP are two cases that want to be avoided in order not to signal wrong results. The respective results also present the training and validation score, scalability of the model, performance of model which describes how the model performs. The parameters print means only those parameters that were changed during training, in addition to the modified ones, only the standard parameters were used during hyper-parameter tuning. The presentation of predicted vs expected value in each plot is reduced to a level to enable the readability of the data. In addition to the best model, the mean and variance of the next top 3 models are also displayed to give a reference on how the best model performed.

5.1 Regression

Each model was trained and performed reasonably well, there were several outliers in the regression model, but if the results are analyzed it can be noted that the model performed well in several of the measurements. For the regression data it is possible to observe in prediction vs expected how the model performed against the actual value while in the performance figures it is possible to observe how well the model was trained.

Task 1	**Estimating cetane number for diesel fuel**
Features	165 166 167 168 169 241 242
Estimator	ElasticNet
Parameters	alpha: 0.1, max_iter:1000
MSE	-4.908993 ± 1.408985
Predicted vs Expected	Figure 2
Performance	Figure 3
MSE - Comparison	
ElasticNet	-4.908993 ± 1.408985
ElasticNet	-4.909149 ± 1.408483
ElasticNet	-4.979614 ± 1.466922

Table 5: Best model, including the selected features, estimator, parameters and the error. The parameters mentioned is the modified, everything else is defaulted to the sklearn model.

Task 2	**Modeling the need for cooling in a H2O2 process**
Features	1 2 6 7 8 11 12 14 15 16 18 19 24 25 27 28 30 32 34 35 36 43 44 47 54 56 63
Estimator	SVR
Parameters	C: 10
MSE	-79.1009267 ± 70.341284
Predicted vs Expected	Figure 5
Performance	Figure 6
MSE - Comparison	
SVR	-91.176367 ± 91.906112
Lasso	-91.176367 ± 91.906112
Lasso	-91.176367 ± 91.906112

Table 6: Best model, including the selected features, estimator, parameters and the error. The parameters mentioned is the modified, everything else is defaulted to the sklearn model.

Task 3	**Predicting power load**
Features	0 1 2 3 4 11
Estimator	MLPRegressor
Parameters	activation: 'relu', alpha: 0.001, early_stopping: True, hidden_layer_sizes: (100, 20), max_iter: 200, solver: 'lbfgs'
MSE	-11098.585795 ± 3237.846183
Predicted vs Expected	Figure 8
Performance	Figure 9
MSE - Comparison	
MLPRegressor	-11287.191160 ± 3550.459524
MLPRegressor	-11293.641862 ± 3326.795473
MLPRegressor	-11300.521074 ± 3407.391361

Table 7: Best model, including the selected features, estimator, parameters and the error. The parameters mentioned is the modified, everything else is defaulted to the sklearn model.

5.2 Classification

All models performed with an accuracy of 80% or higher. In predicted vs expected, a confusion matrix is observed which shows how each model guessed. There was generally a higher chance for the models to guess at the FN which in the case of cancer, the result could be that the wrong diagnosis is dropped, which makes the model in Task 5 perform worse than the actual accuracy. The features described in Task 6 were not the same as f_classifi decided, which may have been the reason why the model performed worse than expected.

Task 4	**Thyroid disease**
Features	0 1 2 3 4 5 6 7 8 9 11 12 13 15 16 17 18 19 20
Estimator	RandomForestClassifier
Parameters	n_estimators: 100
Accuracy	0.9952 ± 0.002400
Predicted vs Expected	Figure 11
Performance	Figure 12
Accuracy - Comparison	
RandomForestClassifier	0.9950 ± 0.002864
RandomForestClassifier	0.9948 ± 0.002857
RandomForestClassifier	0.9868 ± 0.002857

Table 8: Best model, including the selected features, estimator, parameters and the error. The parameters mentioned is the modified, everything else is defaulted to the sklearn model.

Task 5	**Breast cancer**
Features	0 2 3 6 7 20 22 23 26 27
Estimator	SVC
Parameters	C: 5.2, degree: 1, gamma: 'auto', kernel: 'rbf', max_iter: 1000
Accuracy	0.9625 ± 0.032113
Predicted vs Expected	Figure 14
Performance	Figure 15
Accuracy -	
SVC	0.9625 ± 0.032113
MLPClassifier	0.9625 ± 0.032113
SVC	0.9625 ± 0.032113

Table 9: Best model, including the selected features, estimator, parameters and the error. The parameters mentioned is the modified, everything else is defaulted to the sklearn model.

Task 6	**Electrocardiograms**
Features	5 17 38 75 101 121 158 165 168 194 195 196 197 200 201 202 203 208 212 213 214 215 219 224 225 226 227 243 244 248 249 250 251 255 256 257 260 261 262 263 267 268 269 272 273 274 275 279 280 281 284 285 286 287 291 292 293 296 297 298 299 303 304 305 308 309 310 311
Estimator	KNeighborsClassifier
Parameters	algorithm: 'auto', leaf_size: 30, n_neighbors: 3
MSE	0.8 ± 0.074162
Predicted vs Expected	Figure 17
Performance	Figure 18
Accuracy - Comparison	
KNeighborsClassifier	0.8 ± 0.074162
KNeighborsClassifier	0.8 ± 0.074162
KNeighborsClassifier	0.8 ± 0.074162

Table 10: Best model, including the selected features, estimator, parameters and the error. The parameters mentioned is the modified, everything else is defaulted to the sklearn model.

6 Discussion

The project shows that the models produce an arbitrary result. To improve the models, additional models can be tested but go because of limited hardware this was not possible at the present time. Several models for NN were tested but had to be scrapped even though some tests performed better than current results but the weighting between time and performance became too great. Alternative solutions include testing more models for feature selection to get closer to the results described for task 6. The models can be considered to perform well on paper but in the absence of data a better model will not be possible. For example, in Task 5 on cancer, a model with an accuracy of 96% can be considered good, but this also means that 4 out of 100 people will be diagnosed incorrectly, which also shows in the confusion matrix for Task 5 how the majority of the errors lie in the FN and thus that the program will miss that the patient has cancer. The report could not find comparisons for all tasks but that a majority could be compared and performed good in several cases.

In the Artificial Neural Network for Predicting Cetane Number of Biofuel Candidates Based on Molecular Structure [14], the authors managed to reach an RMSE below 10 which, in comparison to the project's MSE of $\mu \approx$ -4.908993 and $\sigma \approx 1.408985$ which means that the model performed better if a conversion to MSE is done for RMSE.

In a comparative study on thyroid disease diagnosis using neural networks [15] from 2009, an analysis was performed using NN on thyroid disease diagnosis. The results presented reached an accuracy between $\approx$ 89-95%. The result performed worse than described in this report with $\mu \approx 0.9950$ and $\sigma \approx 0.002864$. The models described in the project may work but in comparison with other works such as AlphaFold [8] or International evaluation of an AI system for breast cancer screening [9] which presented a result with a reduction of 5.7% and 1.2% (USA and UK) in false positives and 9.4% and 2.7% in false negatives. The comparison in the result where task 5 presented a false negative of 7% which were outperformed by the published model.

Power load forecasting using support vector machine and ant colony optimization [16] is an alternative report that tests models such as ANN, SVM and ACO-SVM to predict the power load in a system. The results show errors up to just over 4% which in comparison of the figures for task 3 can perform better than the current model described in this project.

References

[1] Alpaydin E. Introduction to machine learning. MIT press; 2020.

[2] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media; 2009.

[3] Lowe R, Wu Y, Tamar A, Harb J, Abbeel OP, Mordatch I. Multi-agent actor-critic for mixed cooperative-competitive environments. In: Advances in neural information processing systems; 2017. p. 6379–6390.

[4] Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J, et al. Openai gym. arXiv preprint arXiv:160601540. 2016;.

[5] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 3431–3440.

[6] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 779–788.

[7] Gregor K, Danihelka I, Graves A, Rezende DJ, Wierstra D. Draw: A recurrent neural network for image generation. arXiv preprint arXiv:150204623. 2015;.

[8] Senior AW, Evans R, Jumper J, Kirkpatrick J, Sifre L, Green T, et al. Improved protein structure prediction using potentials from deep learning. Nature. 2020;p. 1–5.

[9] McKinney SM, Sieniek M, Godbole V, Godwin J, Antropova N, Ashrafian H, et al. International evaluation of an AI system for breast cancer screening. Nature. 2020;577(7788):89–94.

[10] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. Tensorflow: A system for large-scale machine learning. In: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16); 2016. p. 265–283.

[11] Carlini N, Wagner D. Towards evaluating the robustness of neural networks. In: 2017 ieee symposium on security and privacy (sp). IEEE; 2017. p. 39–57.

[12] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2011;12:2825–2830.

[13] Tipping ME, Bishop CM. Probabilistic principal component analysis. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 1999;61(3):611–622.

[14] Sennott T, Gotianun C, Serres R, Ziabasharhagh M, Mack J, Dibble R. Artificial neural network for predicting cetane number of biofuel candidates based on molecular structure. In: ASME 2013 Internal Combustion Engine Division Fall Technical Conference. American Society of Mechanical Engineers Digital Collection; 2013. .

[15] Temurtas F. A comparative study on thyroid disease diagnosis using neural networks. Expert Systems with Applications. 2009;36(1):944–949.

[16] Niu D, Wang Y, Wu DD. Power load forecasting using support vector machine and ant colony optimization. Expert Systems with Applications. 2010;37(3):2531–2539.

7 Appendix

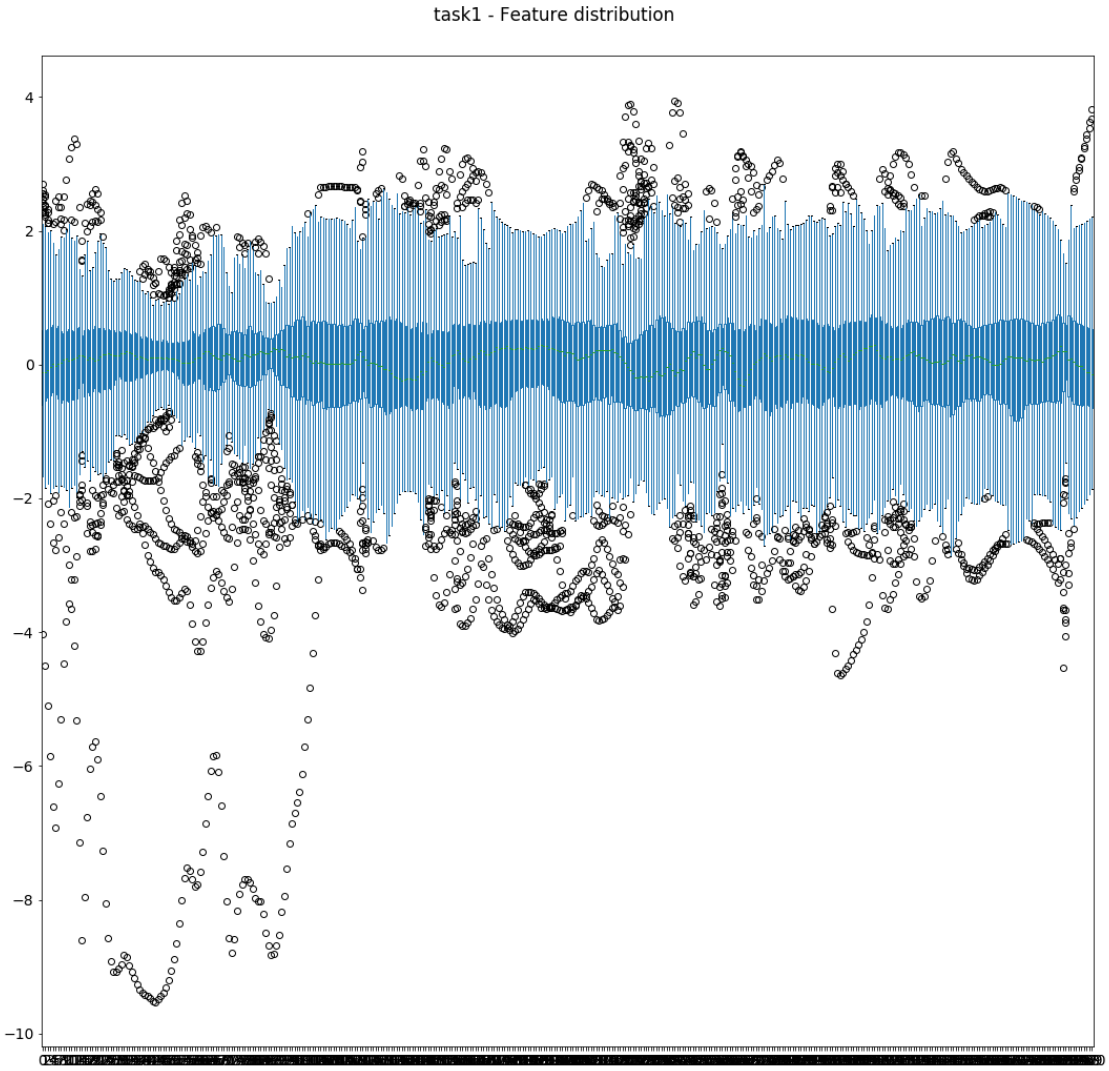

Figure 1: Task 1: Feature distribution after preprocessing with the help of a boxplot describing each feature and the distributions. The green line is the median, the darker blue is the interquartile range, and the dots is concidered outliers of the data points.

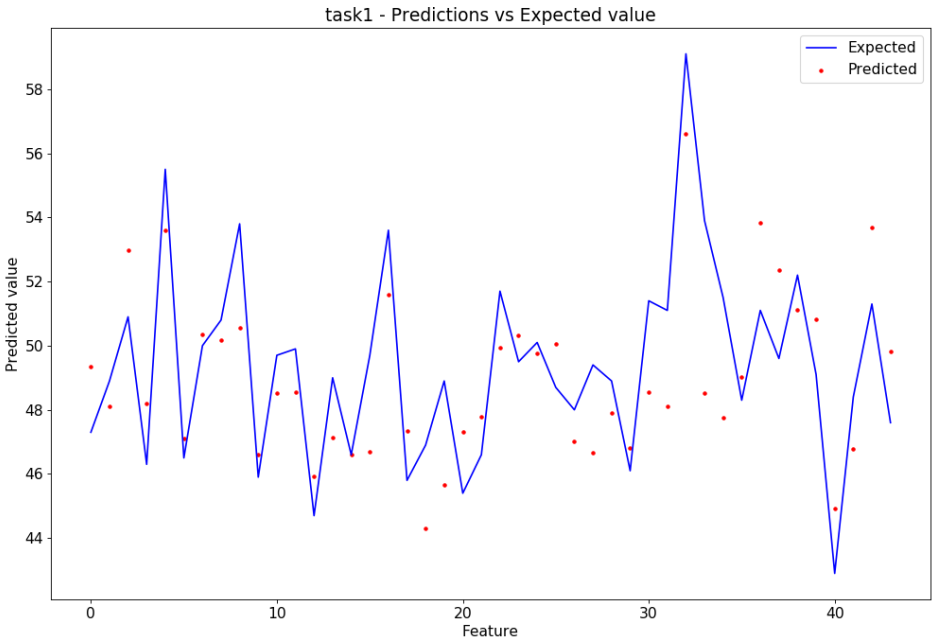

Figure 2: Task 1: Prediction vs expected where line is expected and the red dots are the predicted values

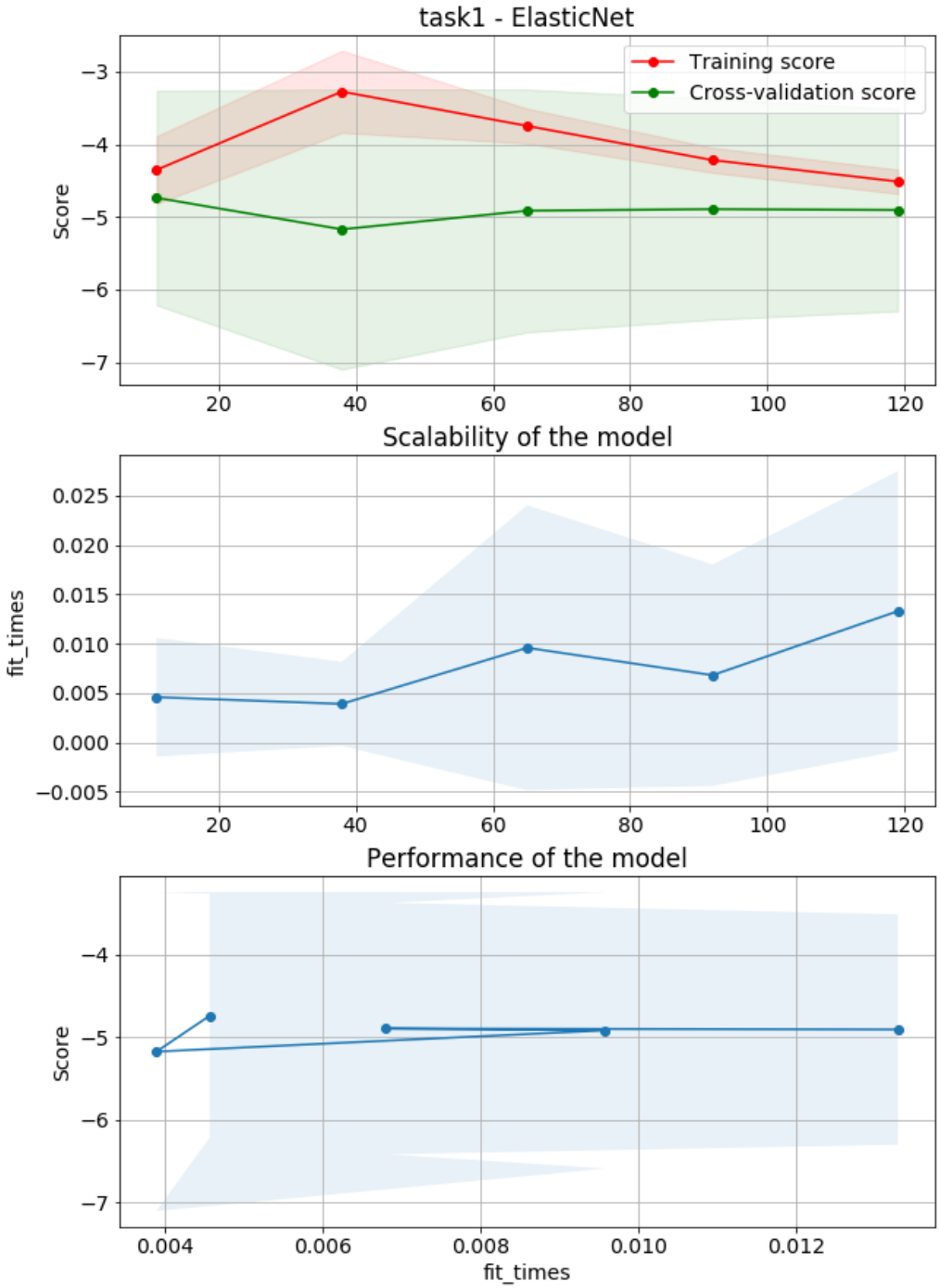

Figure 3: Task 1: Validation curve of training and cross-validation, scalability and performance of the model.

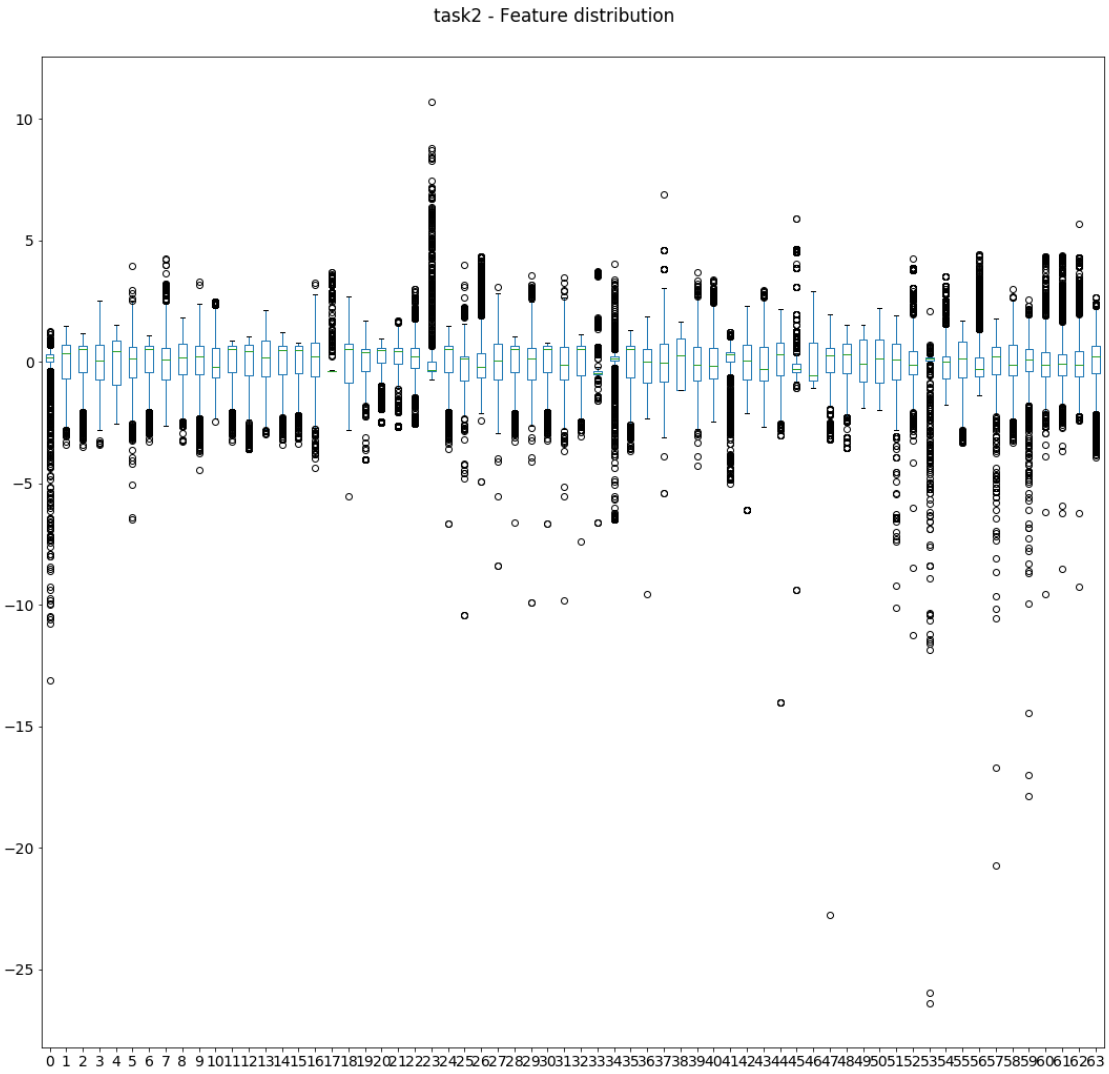

Figure 4: Task 2: Feature distribution after preprocessing with the help of a boxplot describing each feature and the distributions. The green line is the median, the darker blue is the interquartile range, and the dots is concidered outliers of the data points.

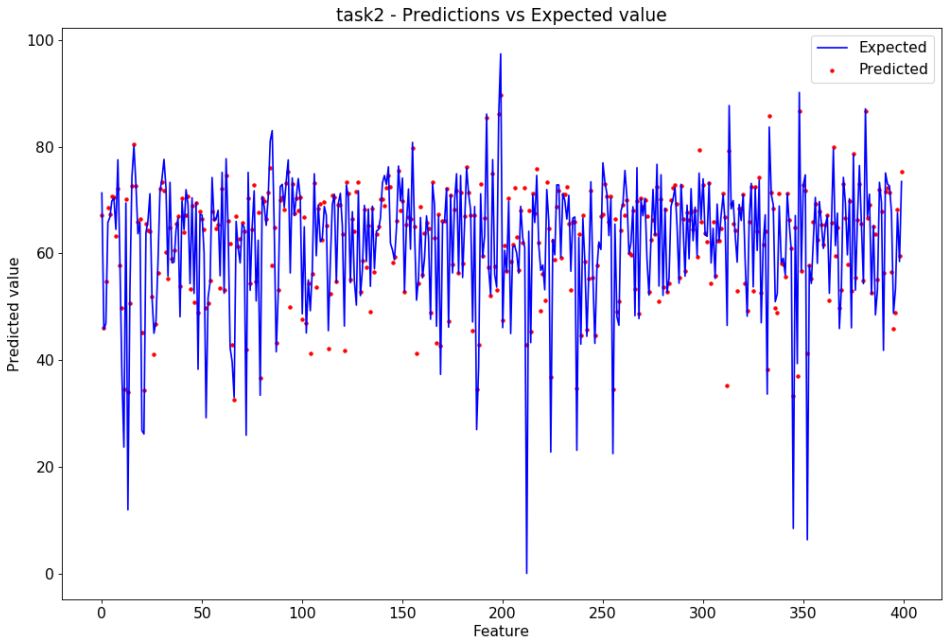

Figure 5: Task 2: Prediction vs expected where line is expected and the red dots are the predicted values

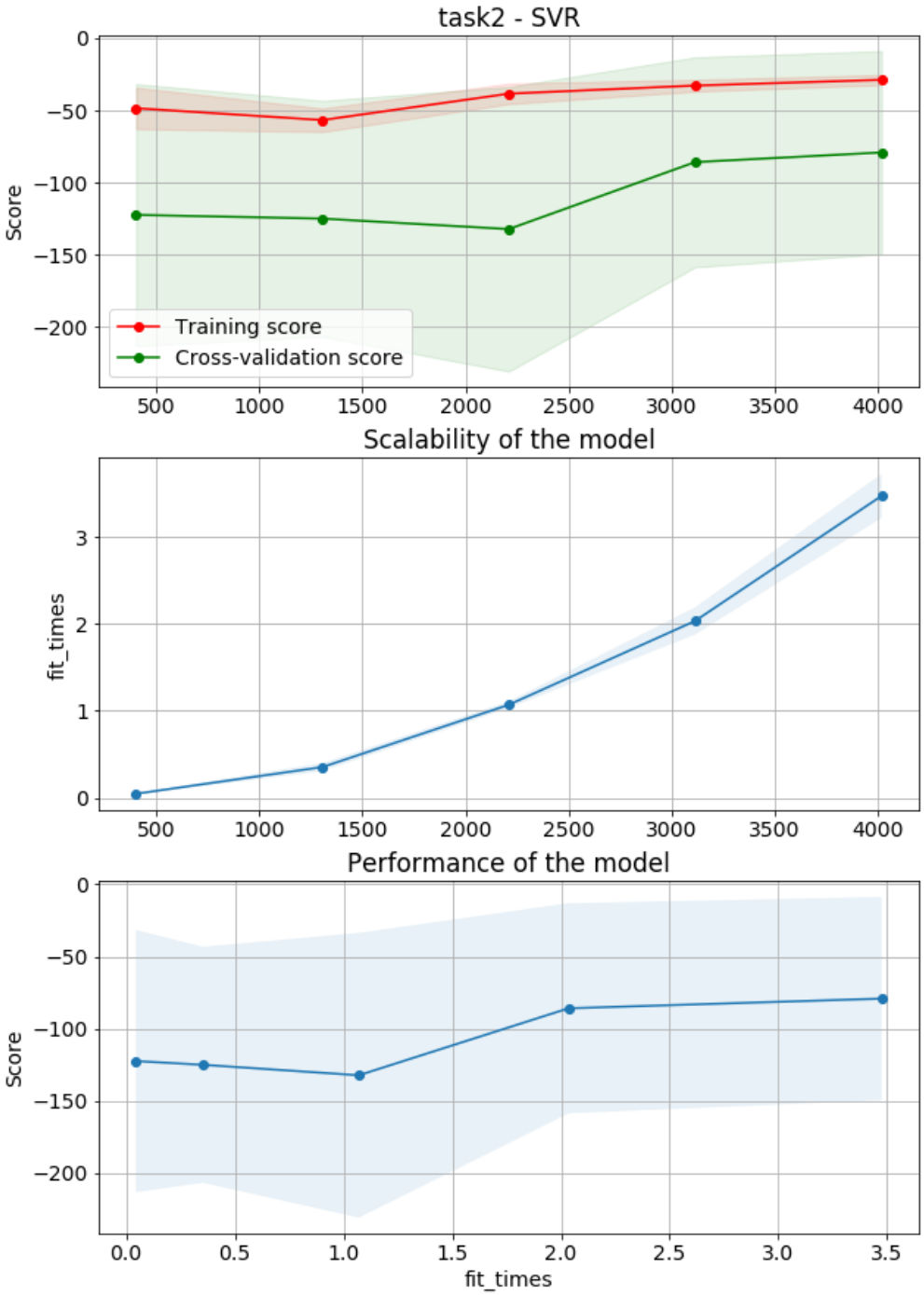

Figure 6: Task 2: Validation curve of training and cross-validation, scalability and performance of the model.

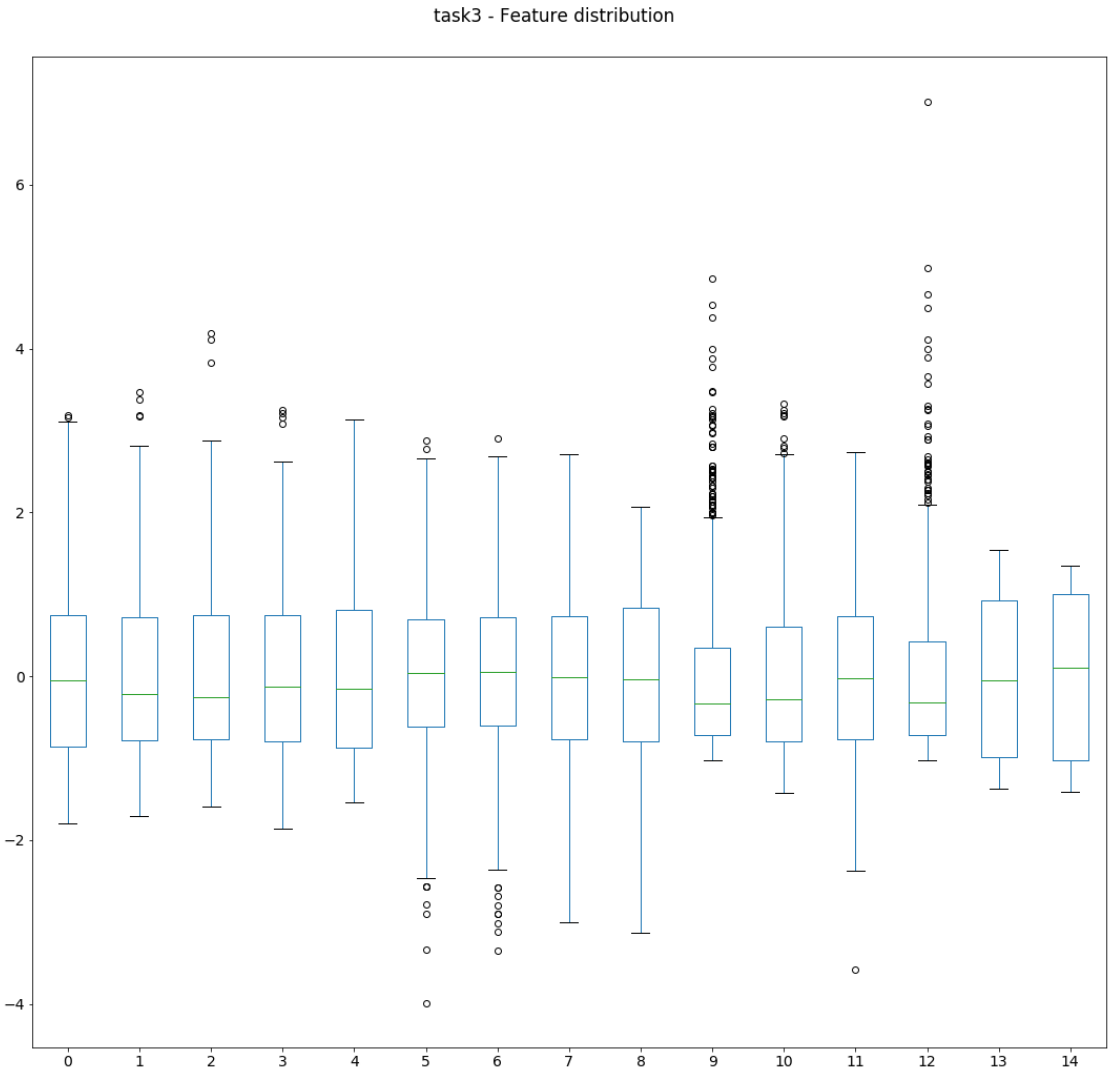

Figure 7: Task 3: Feature distribution after preprocessing with the help of a boxplot describing each feature and the distributions. The green line is the median, the darker blue is the interquartile range, and the dots is concidered outliers of the data points.

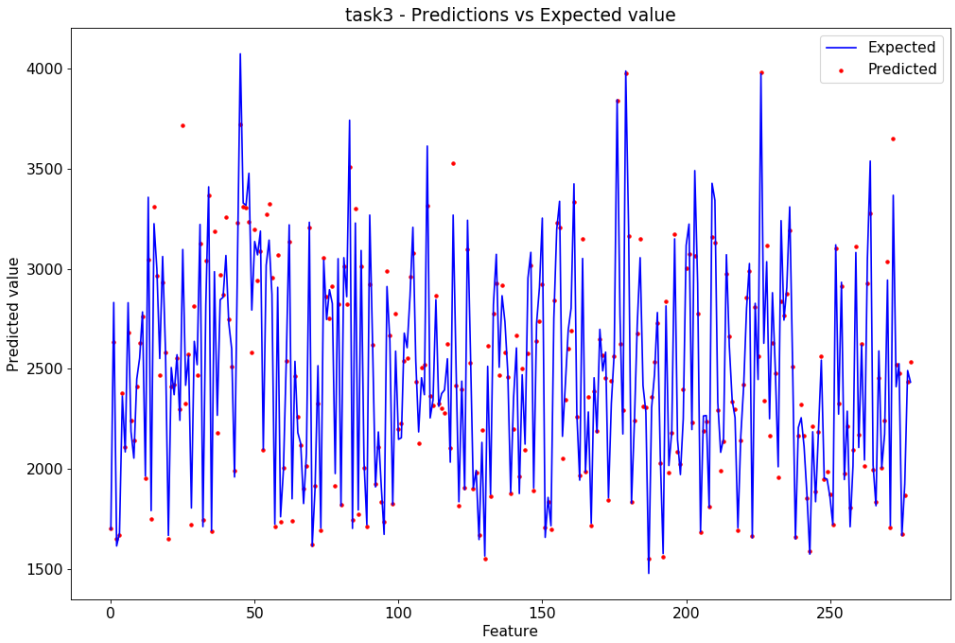

Figure 8: Task 3: Prediction vs expected where line is expected and the red dots are the predicted values

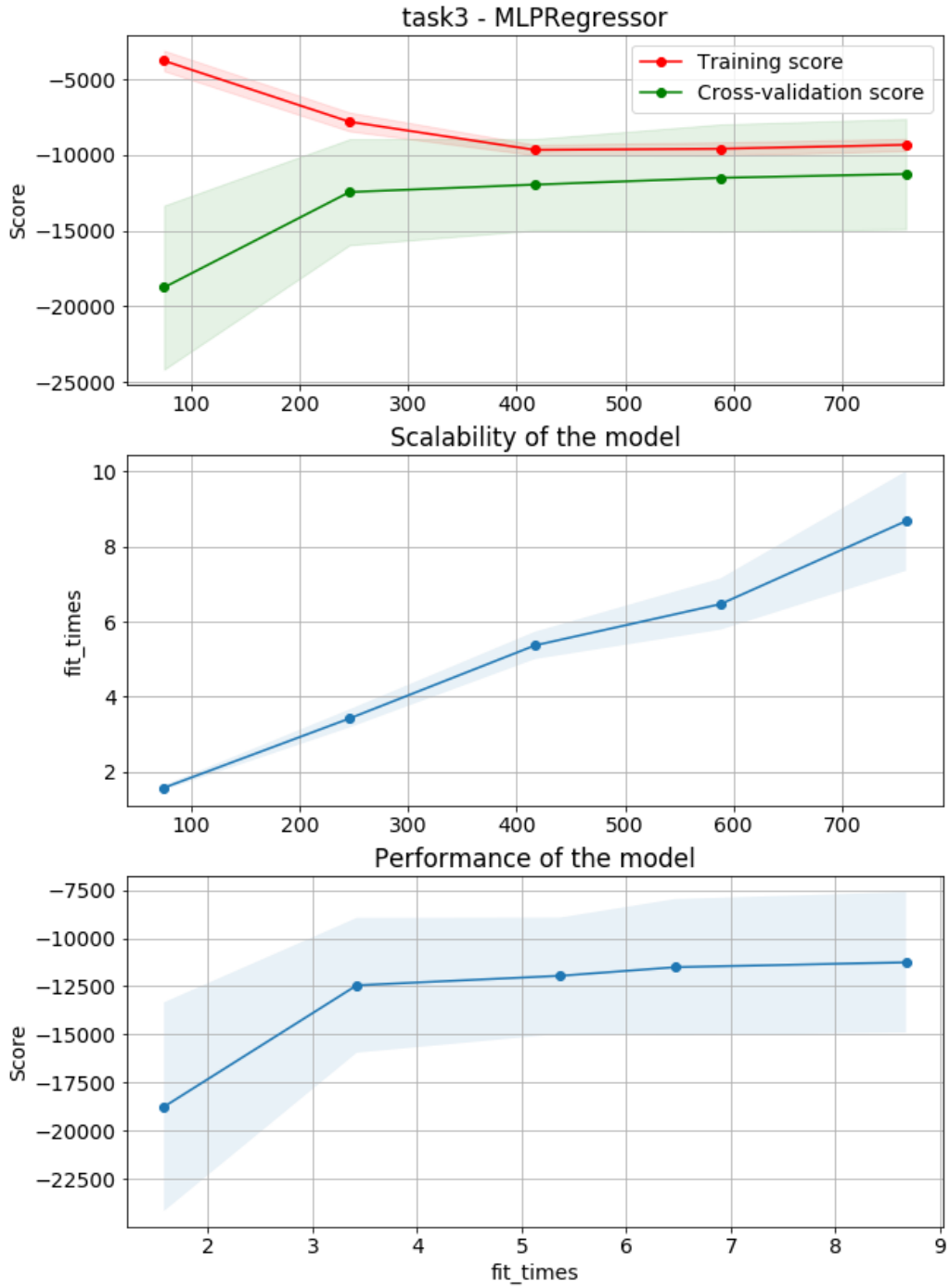

Figure 9: Task 3: Validation curve of training and cross-validation, scalability and performance of the model.

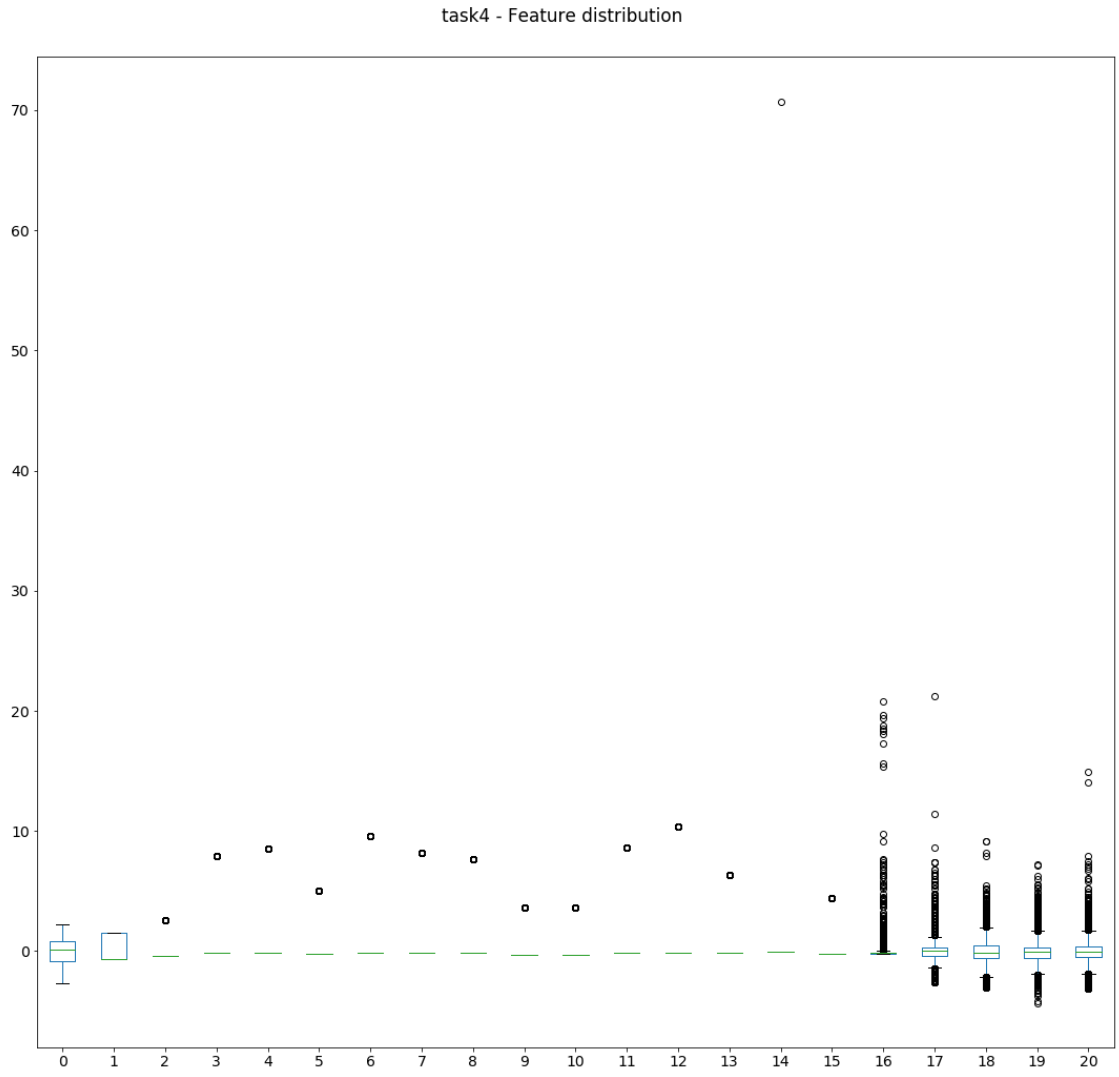

Figure 10: Task 4: Feature distribution after preprocessing with the help of a boxplot describing each feature and the distributions. The green line is the median, the darker blue is the interquartile range, and the dots is concidered outliers of the data points.

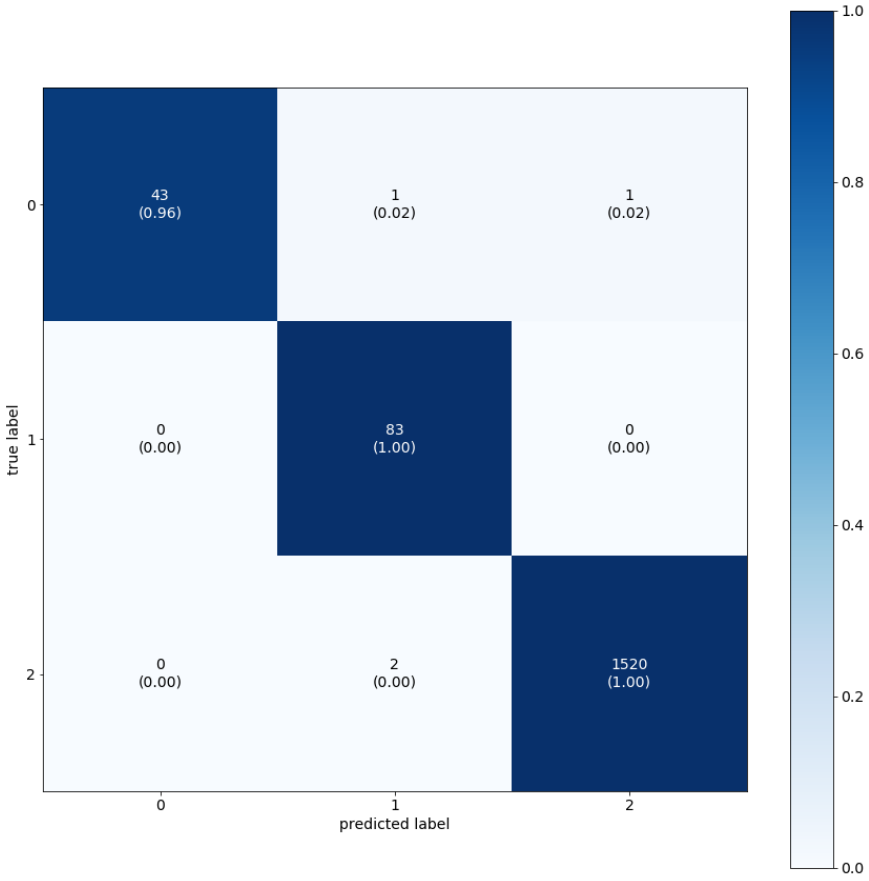

Figure 11: Task 4: Prediction vs expected where line is expected and the red dots are the predicted values

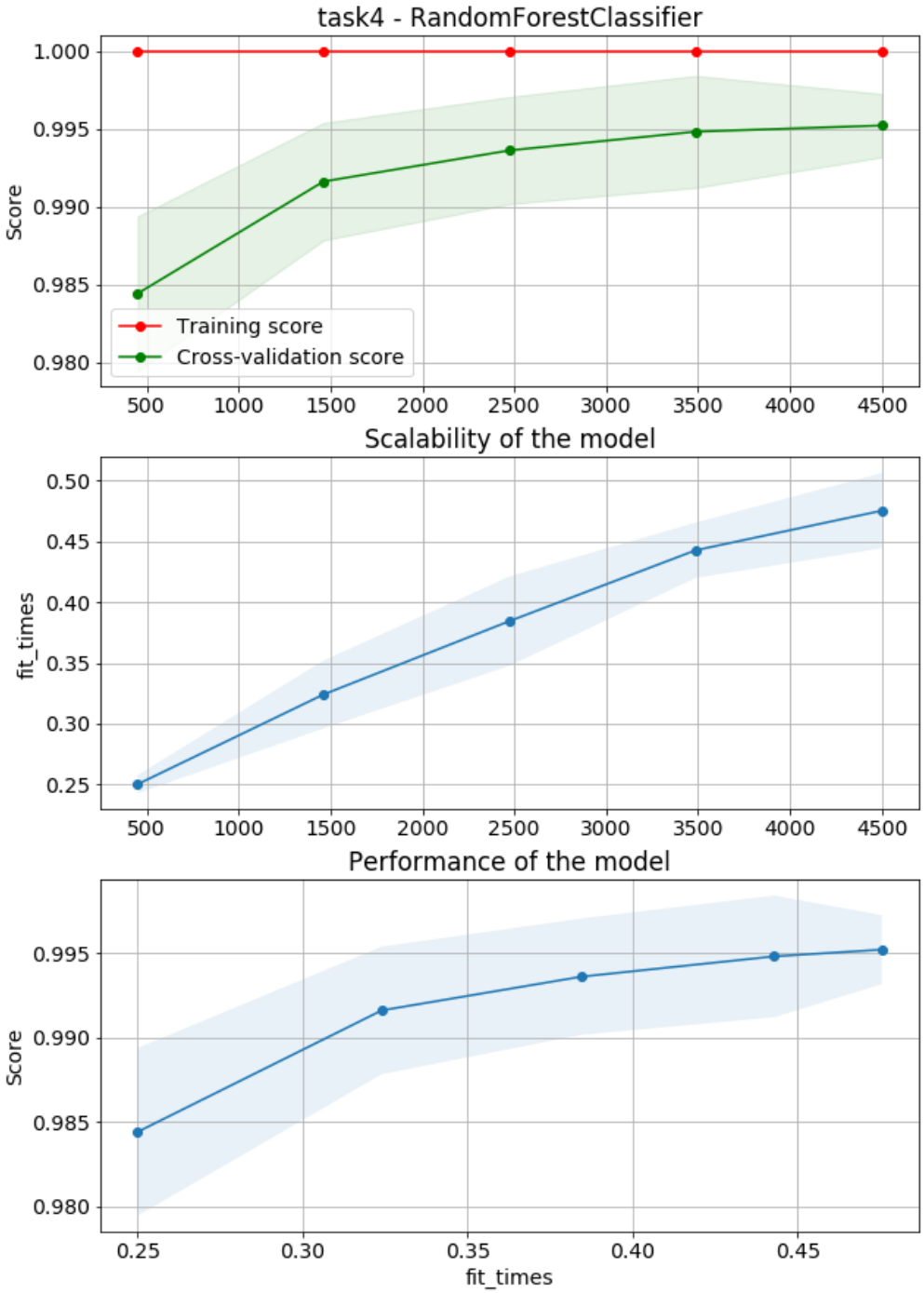

Figure 12: Task 4: Validation curve of training and cross-validation, scalability and performance of the model.

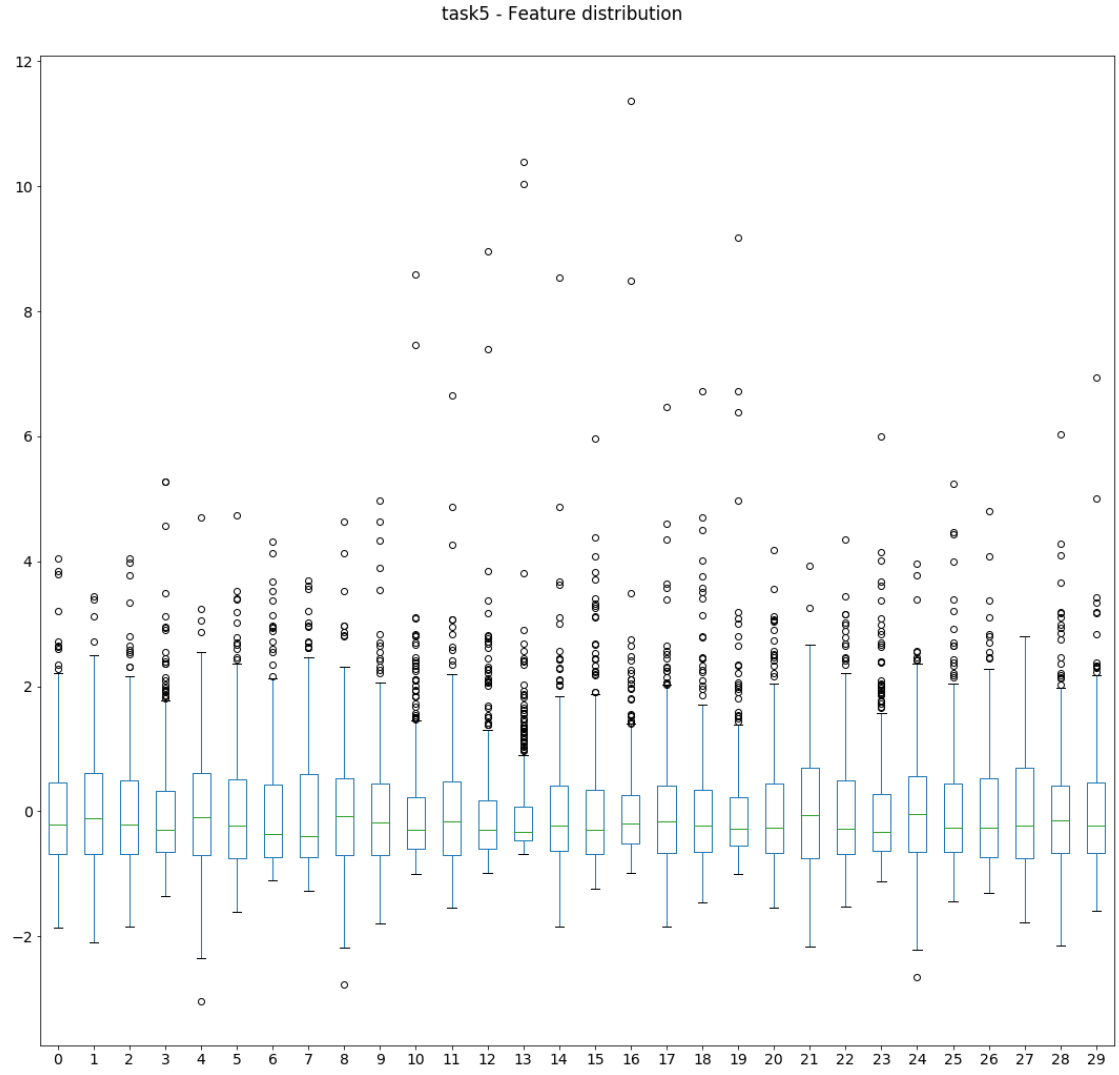

Figure 13: Task 5: Feature distribution after preprocessing with the help of a boxplot describing each feature and the distributions. The green line is the median, the darker blue is the interquartile range, and the dots is concidered outliers of the data points.

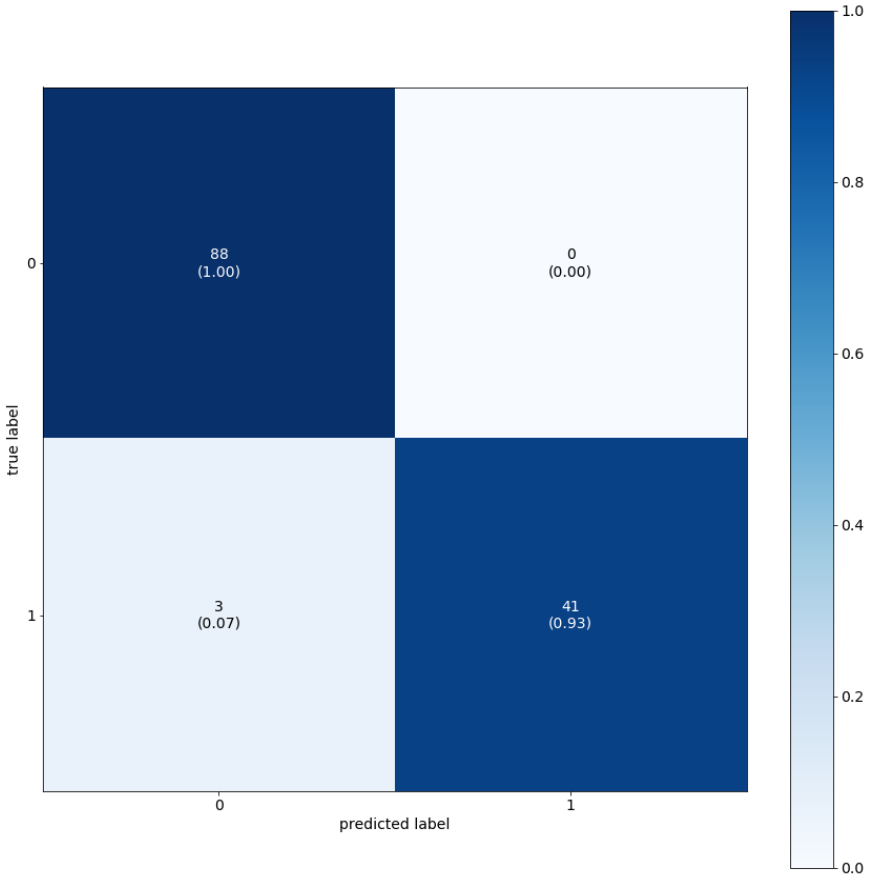

Figure 14: Task 5: Prediction vs expected where line is expected and the red dots are the predicted values

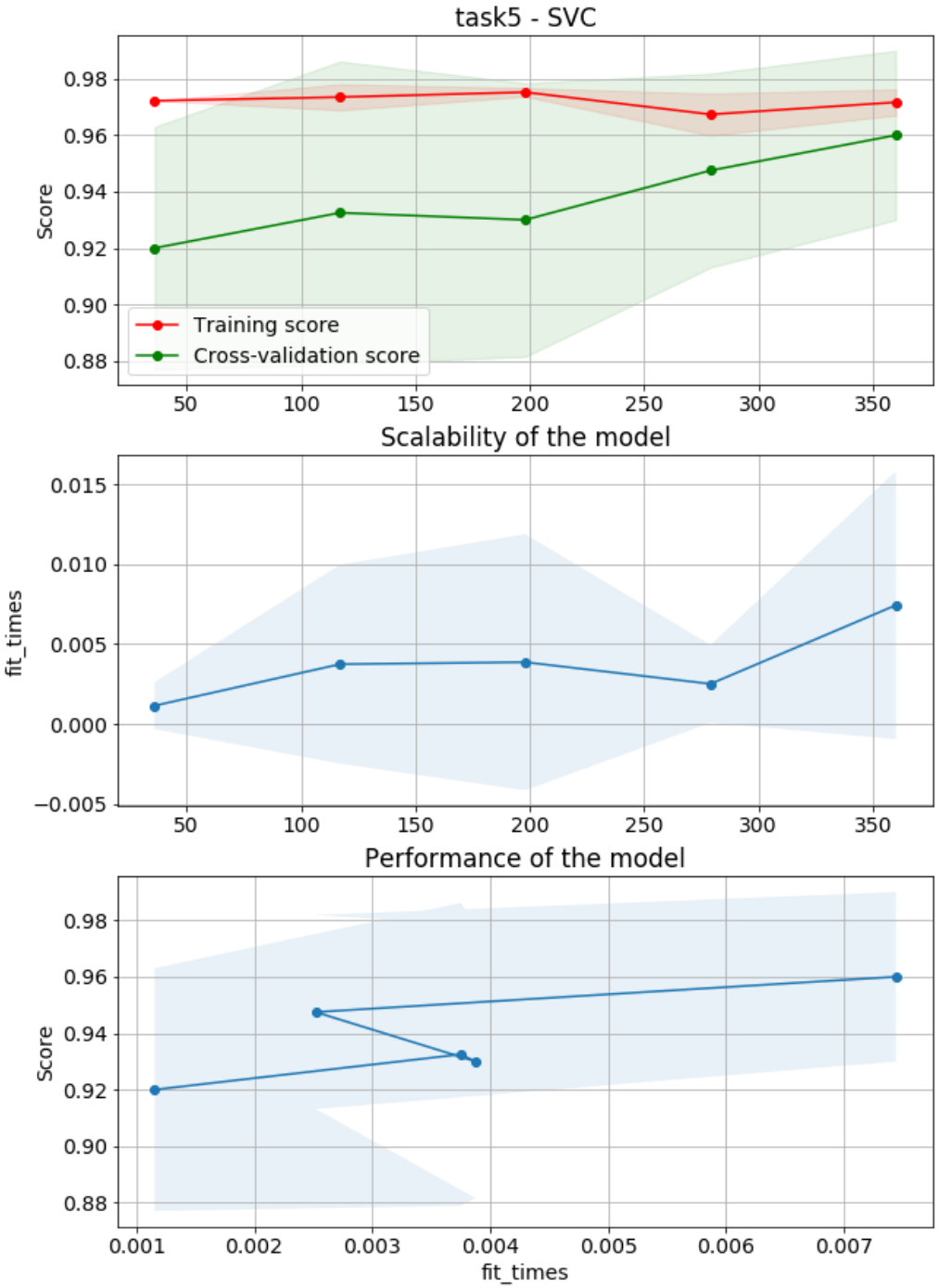

Figure 15: Task 5 Validation curve of training and cross-validation, scalability and performance of the model.

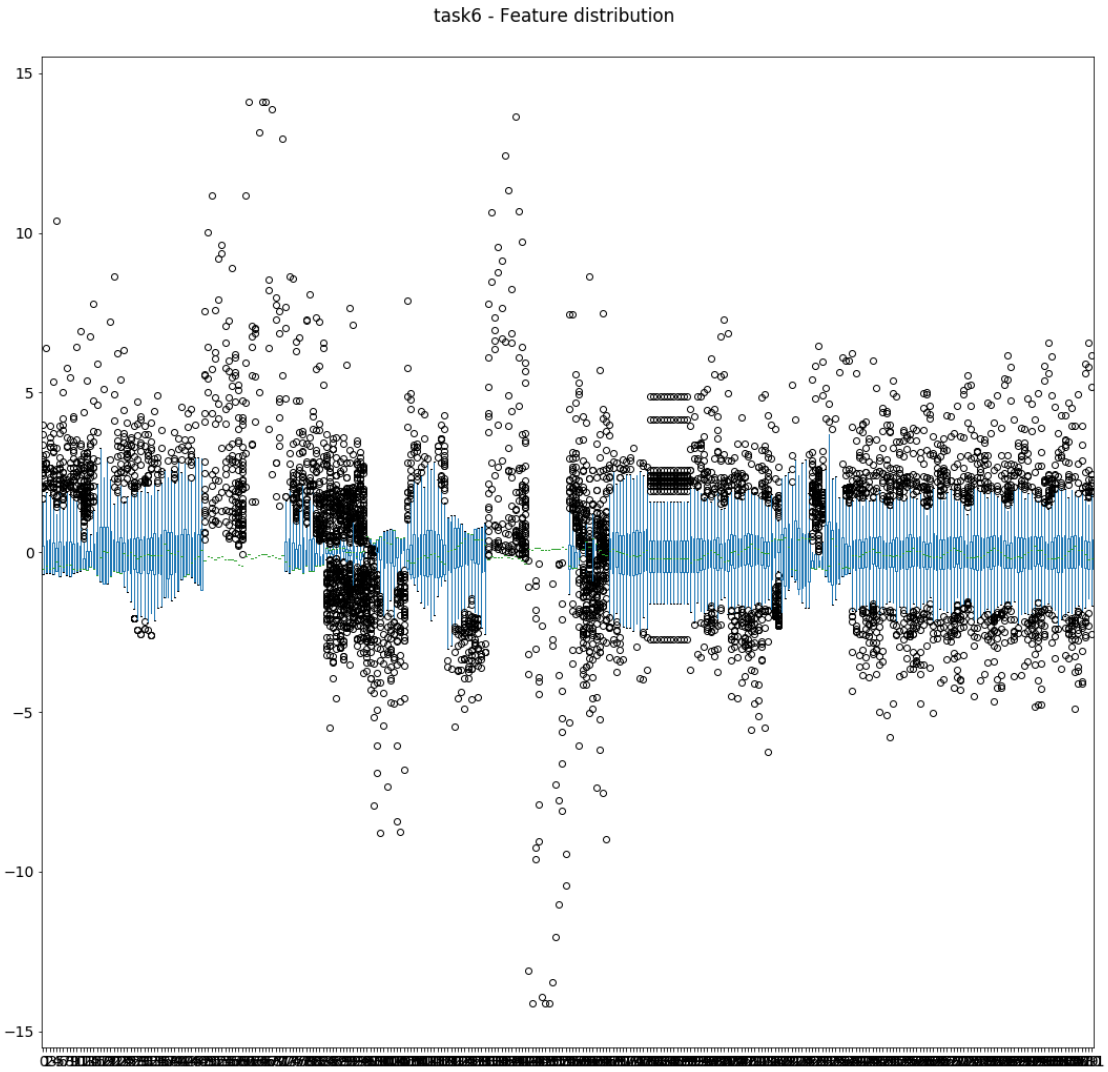

Figure 16: Task 6: Feature distribution after preprocessing with the help of a boxplot describing each feature and the distributions. The green line is the median, the darker blue is the interquartile range, and the dots is concidered outliers of the data points.

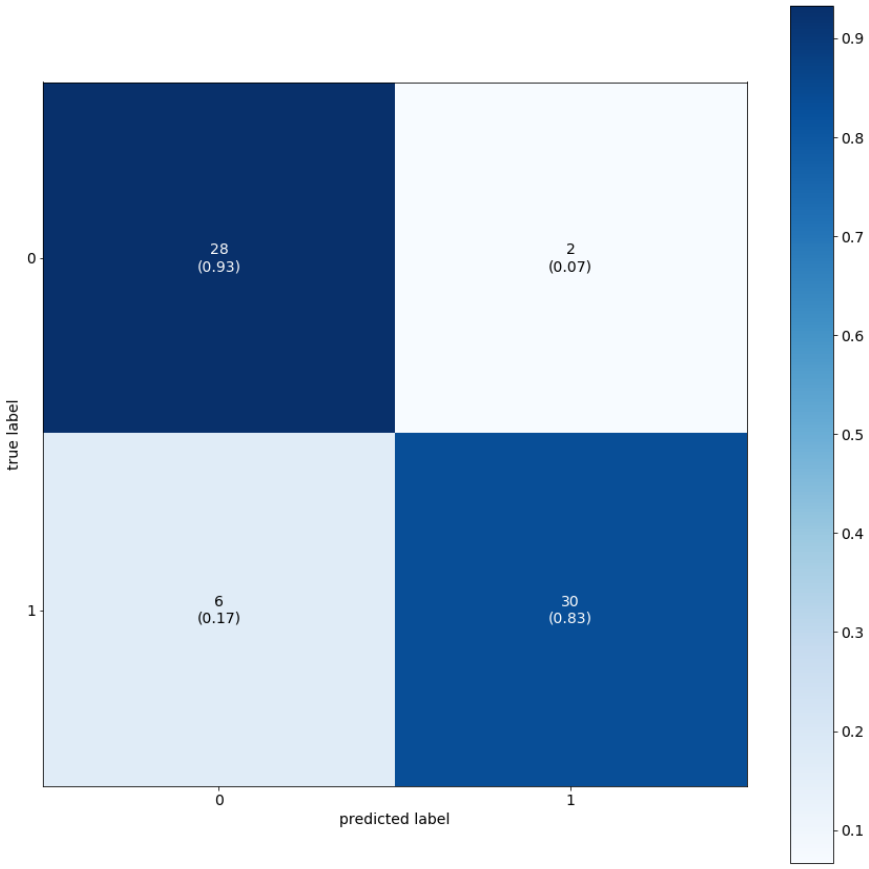

Figure 17: Task 6: Prediction vs expected where line is expected and the red dots are the predicted values

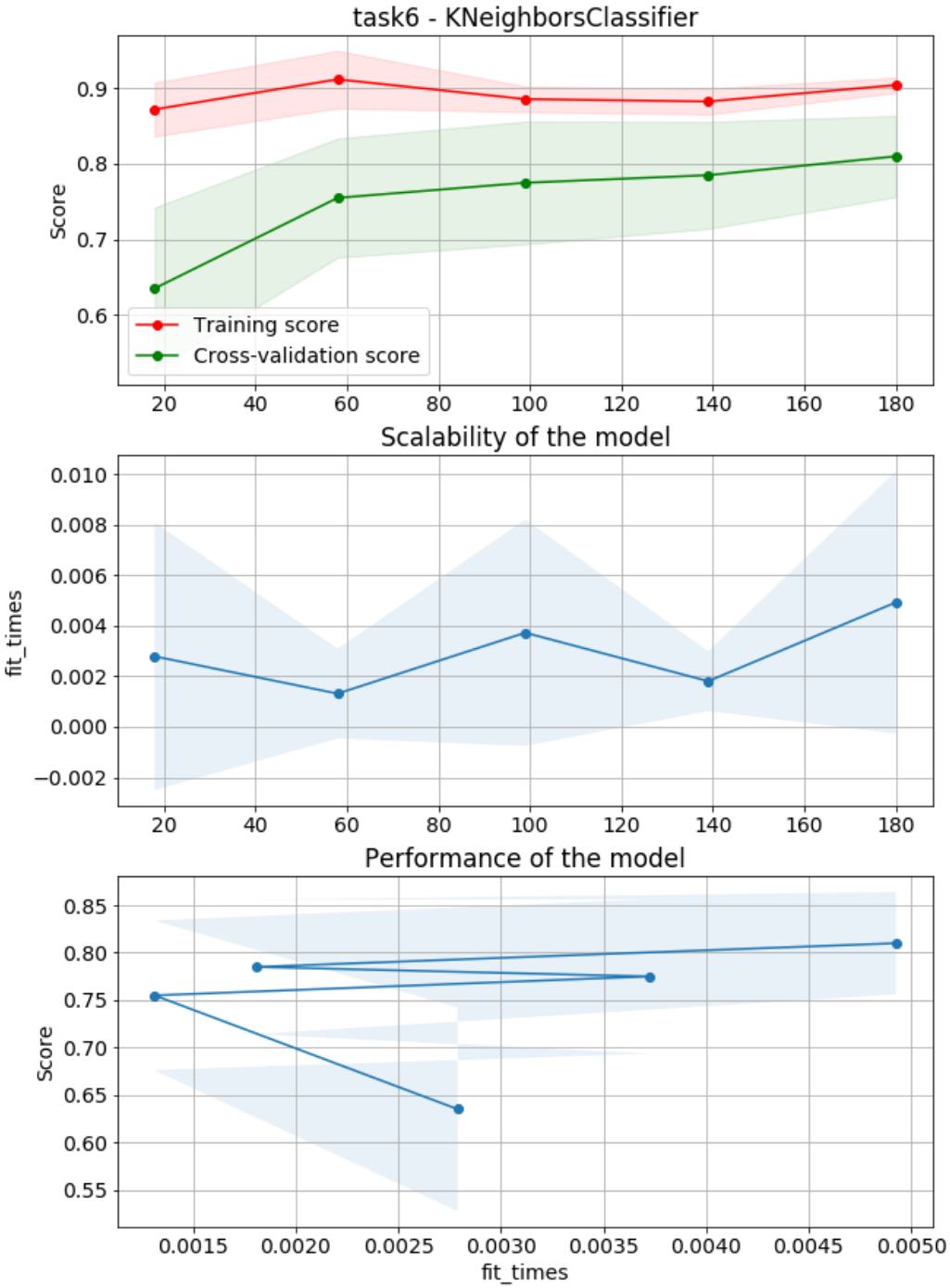

Figure 18: Task 6: Validation curve of training and cross-validation, scalability and performance of the model.