Machine Learning in Medical Diagnostics

1. Introduction

Getting the right diagnosis is probably the most important aspect of health care: It explains a patient's health problem and provides the basis for subsequent health care decisions. But medical diagnoses are known to be extremely complex: Going through a patient's medical history, correctly detecting diseases and recommending the optimal treatment are not only time-consuming but also highly subjective, as those tasks greatly depend on the physician's expertise level. Because of this, diagnostic errors contribute to approximately ten percent of patient deaths and continue to represent a blind spot in the delivery of quality health care. To address these challenges modern research is turning to artificial intelligence. With rapid progress made in the field of machine learning, it is possible to train algorithms to automatically detect patterns of certain diseases within a patient's medical record and inform clinicians of any anomalies. The use of machine learning can therefore improve medical diagnostics significantly.

In this project, we want to give two examples on how machine learning can be applied to medical data for the purpose of detecting diseases and supporting clinicians in their decision-making process. We have worked on two very different sets of data and subsequently implemented different machine learning methods to diagnose the data. The first half of the project deals with the prediction of diabetes based on demographic, questionnaire data and laboratory examinations. Diabetes is a chronic disease that directly causes an estimated 1.6 million deaths per year. The number of people affected by the disease has risen rapidly over the past years, amounting around 422 million today [1]. With early detection of diabetes and prediabetes, the burden of the disease can be significantly reduced or even effectively prevented. On our dataset, we implemented random forest, support vector machine and multilayer perceptron algorithms to predict whether a person is "not diabetic", "at risk for diabetes" or "diabetic". Since there already are rather simple and inexpensive tests that can easily detect diabetes, our main motivation for this part of the project lies in finding out whether or not and which of our lifestyle choices are a factor for developing diabetes.

The second dataset we worked on contains images of blood smears, half of which represent healthy samples while the other half shows malaria infected samples. Malaria is a

life-threatening disease caused by parasites that are transmitted to people through mosquitoes. Though it is preventable and easily curable, it still causes more than 400.000 deaths per year [2]. The most widely used method for malaria detection is examining thin blood smears under a microscope, and visually searching for infected cells. Especially in countries that carry a high share of the global malaria burden and where most people do not have regular access to health care, this approach is too time-consuming. Opposing the traditional method, we implemented a convolutional neural network as well as a support vector machine algorithm to classify healthy and malaria infected blood smears.

2. Prediction of Diabetes

The idea of this part of the project was to find a dataset with lifestyle data (nutrition, physical activity, stress levels) and health examination data and try to predict health condition based on the lifestyle of a person. Our motivation is the following: to show indirectly that health and wellbeing is a straight consequence of our everyday choices. We have chosen diabetes for two reasons: First, because it is a widely distributed chronic disease and, second, because it (at least the type 2 diabetes mellitus) is usually developed in adulthood due to unhealthy diet, obesity and little physical activity [3]. The primary question was: Is it possible to predict diabetes using a set of demographic, body measurements and lifestyle data? Secondary questions were: How good can we distinguish between three classes (“no diabetes”, “pre-diabetes”, “diabetes”)? How good can we distinguish between two classes (“no diabetes”, ‘’pre-diabetes or diabetes” merged together)? Which features have the most predictive power? How do we interpret the accuracy of predictions? How do different models perform on the same task?

The code was written in Python 3 using its libraries including pandas, numpy, pyplot, seaborn, scikit-learn and other. The Jupyter notebook including some descriptions, step by step code and graphs can be found on GitHub under [M1].

2.1 Dataset and preprocessing

The dataset on the lifestyle and health from 2013-2014 was found on Kaggle [M2]. It originates from the program of studies designed to assess the health and nutritional status of adults and children in the United States and is called National Health and Nutrition Examination Survey (NHANES). The dataset consists of 6 tables, however for our analysis we used information from 4 of them (demographic.csv, questionnaire.csv, labs.csv and examination.csv). After merging these 4 tables together we got one with a total of 9813

entries and 1645 attributes. In the case of this dataset, feature selection and preprocessing was the most time-consuming part. Also it was something new, since we didn't do it much during our practical course and always had ready-for-use data sets instead.

First, we studied the dataset and the description of the columns trying to figure out what could be relevant for the diabetes prediction. Initially the following set of attributes was selected:

- From the demographics data: id, age, gender, educational level, annual family income;
- From the questionnaire data: minutes sedentary activity, vigorous recreational activities, minutes vigorous recreational activities, moderate recreational activities, minutes moderate recreational activities, frequency of meals from fast-food or pizza places, number of frozen meals/pizza in the past 30 days, trouble sleeping or sleeping too much, feeling tired or having little energy, how do you consider your weight;
- From the examination data: BMI (Body Mass Index) and waist circumference;
- From the labs.csv table: blood manganese;

Demographics data was taken to look at the basic statistics and distribution of the disease among people from different groups. The questionnaire data was used to assess physical activity and nutritional habits of the respondents. BMI and waist circumference were reported to be major obesity markers associated with diabetes [4] and we wanted to see how it will help in our prediction task. Some studies also reported that diabetic patients have lower blood levels of manganese so it was also included in this analysis [5]. For the units of measurement and value code description please refer to the notebook [M1].

Second, we decided to split the data into three categories based on the glycohemoglobin measurements since this was the only test performed on almost all the study participants. This test does not distinguish between type 2 and type 1 diabetes mellitus, hence further on we do not talk about the types. The split was done based on the laboratory procedure manual found on the official NHANES website where it is written that values between 4 and 6% are normal and results higher than 6,5% are indicative of diabetes [6]. The values in between were marked by us as "prediabetic".

For the analysis, we only took people older than 25. This was chosen arbitrarily as we wanted to look at adult people who had time to accumulate consequences of their lifestyle. After transforming some features and cleansing the data from NA-values and responses like

“refused” and “don’t know” we ended up with 1579 rows in our table. In the Figure 2.1 we see the total counts for categories by gender. We had approximately 200 people in the risk zone for diabetes, 200 diabetic and 1200 people without diabetes.

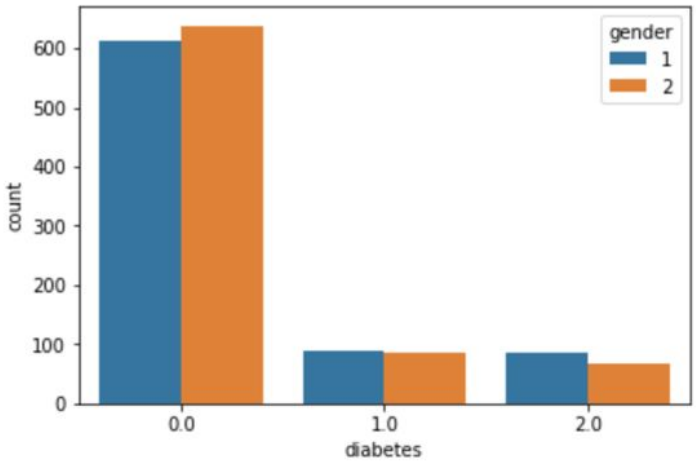

Figure 2.1.1 Total counts in three categories (0 - no diabetes, 1 - prediabetes, 2 - diabetes) by gender (1 – male, 2 - female)

As next we looked at the distribution of different features over three categories (see Figure 2.1.2 and Figure 2.1.3 for an example).

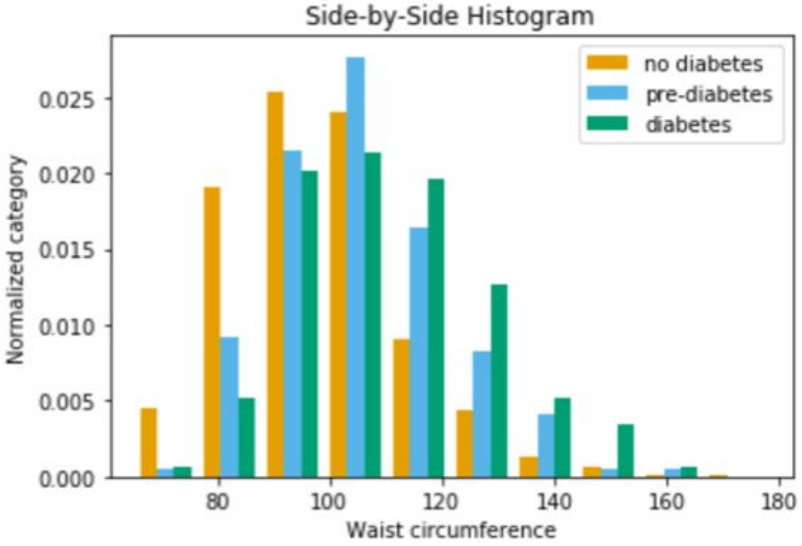

Figure 2.1.2 Distribution of waist circumference for three classes

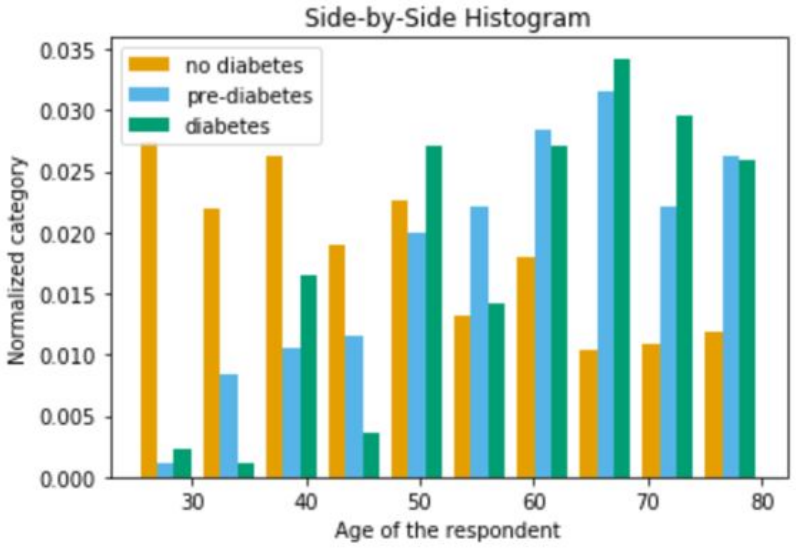

Figure 2.1.3 Distribution of the age for three classes

In both figures we see that prediabetic and diabetic people tend to have a distribution skewed to the right compared to people with negative diagnosis. The conclusion: both attributes can be helpful in training a predictive model. At the same time it is clear that it will be hard to distinguish between prediabetic and diabetic class, more on that in the results section.

2.2 Models

After the dataset was ready for use for our predictive purposes, we looked at the performance of several models. Here we would like to present three of them: random forest, support vector machine and MLP.

Random forest consists of a large number of decision trees and represents a typical ensemble algorithm. Each tree votes for a certain class for a particular data entry and the class with the most votes becomes our model's prediction. In our study we used 100 estimators with max depth of 3.

SVM algorithm is described later in this report. For the diabetes part we used the following hyperparameters: C = 1, rbf kernel with gamma = 50.

MLP classifier is a simple feedforward artificial neural networks with at least three layers. In our classification we used one hidden layer with 49 nodes and a maximum of 50 iterations, other parameters were default.

2.3 Results

All three models have reached approximately the same accuracy on the test set – around 79% (see Table 2.3.1). It looked good at first sight, but the confusion matrix showed that the algorithms learned the simplest strategy to classify almost all the entries as ‘not diabetic’ and with that reaching the accuracy rates below. Note that the dataset was imbalanced having 80% of non-diabetic entries.

We also looked at the feature importance graph for random forest (see Figure 2.3.1). The first thing we saw was the fact that from all the attributes only age, BMI and waist circumference had some predictive power.

These unsatisfactory results made us rethink the design of the classification. First, looking at the histograms in Figures 2.1.2 and 2.1.3, we see that values for two classes “diabetes” and “prediabetes” go hand in hand. So, using the set of chosen features it was impossible to separate them from each other. As the number of people without diagnosis made almost 80% of the dataset, the most efficient way to assign entries to the classes was the naive strategy just mentioned. One possible solution to that problem was to merge “diabetes and prediabetes” into one single class and perform a binary classification.

Table 2.3.1 Accuracy of diabetes prediction in the 3 classes case

Model	Accuracy
RF	0.791
SVM	0.79
MLP	0.794

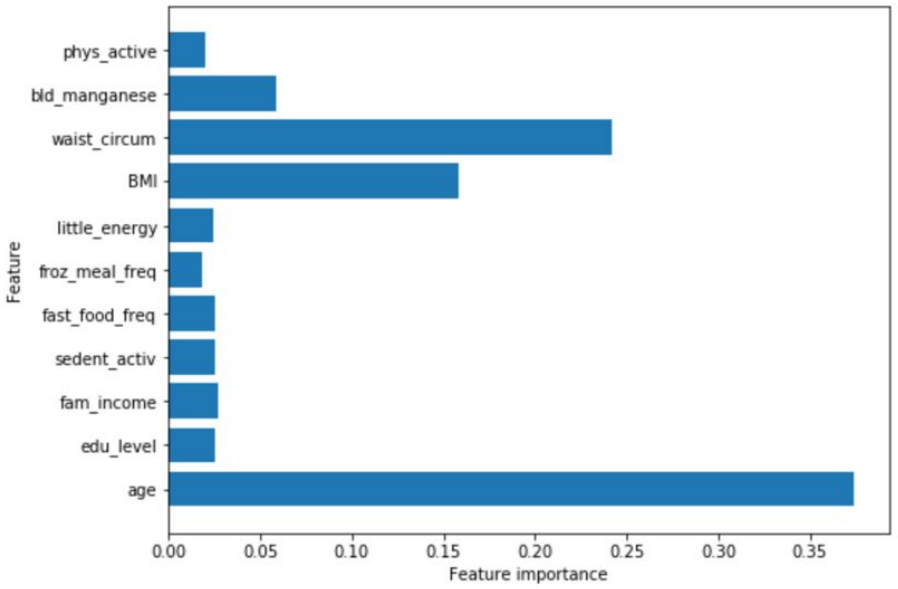

Figure 2.3.1 Feature importance in RF

In the final version of the binary classification we included only 4 features from the initial set, because it gave us the best results (blood manganese was excluded since this column contained many NA values resulting in a much lower dataset size). We had a total of 4644 entries, 1092 of them belonging to the “diabetic or at risk” class. Prediction accuracy did not get higher than it was in the three classes case, but the algorithm started to distinguish between the classes and changed the naive strategy to a different one. All three algorithms performed in a similar way, the accuracy reaching 0.784 for random forest, 0.77 for SVM and 0.782 for MLP. In the Figure 2.3.2 you see an example decision tree from RF algorithm, that classified the samples as “diabetic or at risk” when the person was older than 46 years and had a waist circumference greater than 112.65 cm.

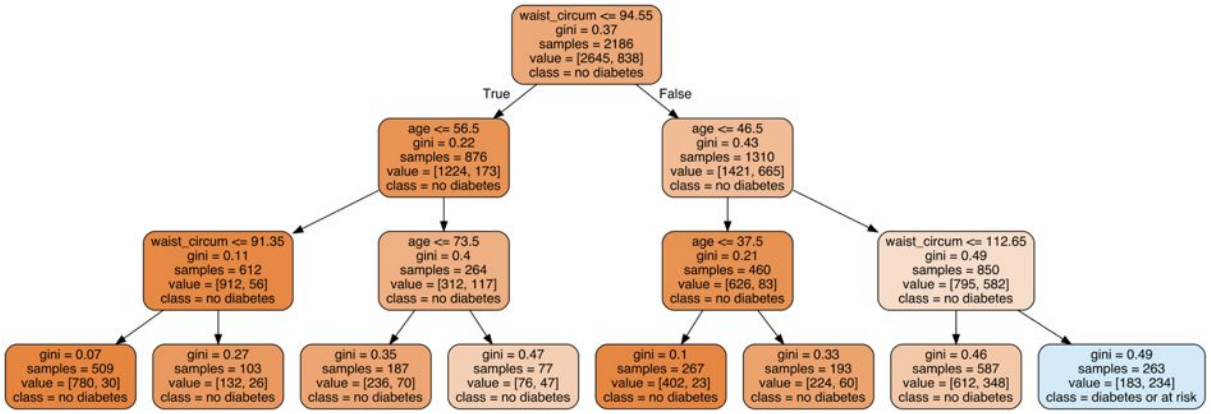

Figure 2.3.2 One of the decision trees from the RF algorithm

Below you see the confusion matrix for the RF algorithm and the ROC curve giving a better idea of the performance.

Table 2.3.2 Confusion matrix RF

True \ Predicted	0	1
0	869	19
1	232	41

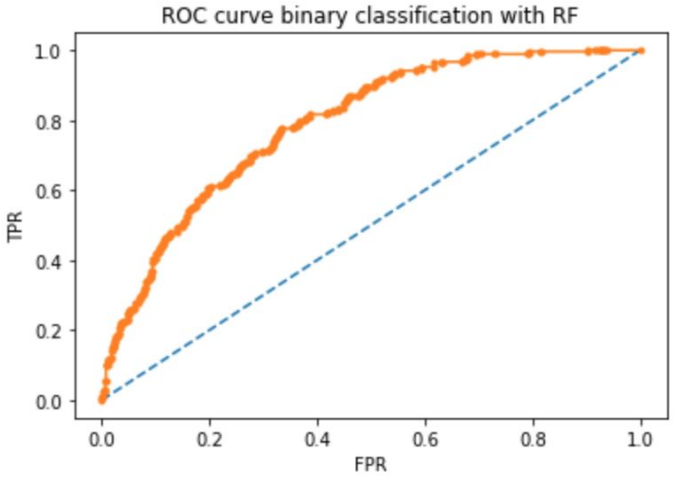

Figure 2.3.3 ROC Curve RF

We see that the algorithm assigned 869 input-values to the non-diabetic class and 41 to the diabetic or at risk, the area under the ROC curve reaches 0.791. The model cannot distinguish between non-diabetic and diabetic people perfectly, but the accuracy of almost 80% is still quite good.

2.4 Discussion

Our initial try to separate non-diabetic, pre-diabetic and diabetic people was rather a failure. Two latter groups had very similar profile when considering selected features. Another problem with this approach was the unbalanced dataset: we had much more non-diabetic entries as diabetic or pre-diabetic. Change of approach (binary classification plus reduced features) resulted in an overall better performance.

An interesting question is, why actually features like ‘physical activity’ or ‘fast food frequency’ did not help in the classification? One possible reason is that the behavior of people changes once they get a positive diagnosis for diabetes. They probably start reducing unhealthy products in their diet and start to do more exercise. Mostly it doesn’t help to cure the disease, but at least it slows down its progress. If so, then we find healthy people eating little fast food and sick people eating the same little amount and fail to separate them in the classification. If we would have values over a long enough period with nutritional behavior and physical activity data and could follow the onset of diabetes depending on that, then it might be possible to predict the risks of getting sick given the lifestyle. Considering the features that did help to separate the classes, we can see that indirectly they also mirror people’s lifestyle. High Body Mass Index or large waist circumference means obesity. Obesity is in turn very often a result of a certain lifestyle and unhealthy everyday choices. With that in mind we should become conscious about putting ourselves under high risk of diabetes if we don’t pay attention to the wellbeing of our bodies. The older we get the higher the risk.

3. Detecting cells infected with malaria

Given another dataset that contains images of blood smears our goal is to use machine learning algorithms to classify malaria infected cells and healthy cells. To reach a high accuracy we decided to use a Convolutional Neural Network and a Support Vector Machine algorithm, both methods have proven to be a good choice for binary classifications. The main questions are: Which method acquires a higher accuracy? How can we tune the hyperparameters? Are those methods good alternatives to manually classifying cells by human resources?

The implemented algorithms can be found under [M3]. Mainly we used Python 3 libraries like Pandas, Matplotlib, Numpy, Scikit-learn and Tensorflow.

3.1 Dataset

The malaria dataset is provided by the Lister Hill National Center for Biomedical Communications, part of the National Library of Medicine. Therefore, the researchers developed a mobile application running on Android smartphones that is attached to a light microscope. Giemsa-stained thin blood smear slides from 150 malaria infected patients and 50 healthy patients were collected at Chittagong Medical College Hospital in Bangladesh.

With the help of the mobile application the researchers acquired images of slides for each microscopic field of view. A level-set based algorithm was applied to detect and segment the red blood cells.

The dataset consists of 27.558 cell images, half of which are infected cells while the other half are healthy cells. The images of each cell are saved as png files. The file name provides information about the matching patient-ID [M3].

	filename	label
0	./cell_images/Parasitized/C180P141NThinF_IMG_2...	Infected
1	./cell_images/Parasitized/C84P45ThinF_IMG_2015...	Infected
2	./cell_images/Uninfected/C204ThinF_IMG_2015102...	Healthy
3	./cell_images/Uninfected/C76P37ThinF_IMG_20150...	Healthy
4	./cell_images/Uninfected/C141P102ThinF_IMG_201...	Healthy

Figure 3.1.1 Dataframe of the cell images created with the Python module Pandas

---

The images are provided in different sizes since the blood smears and cell images vary based on the individual patient. After exploration we got following data concerning the image data:

Table 3.1.1 Information about the shape of the data arrays of images

	#pixels width	#pixels height	#color channels
Minimum	40	46	3
Maximum	385	394	3
Average	132.98	132.49	3
Median	130	130	3

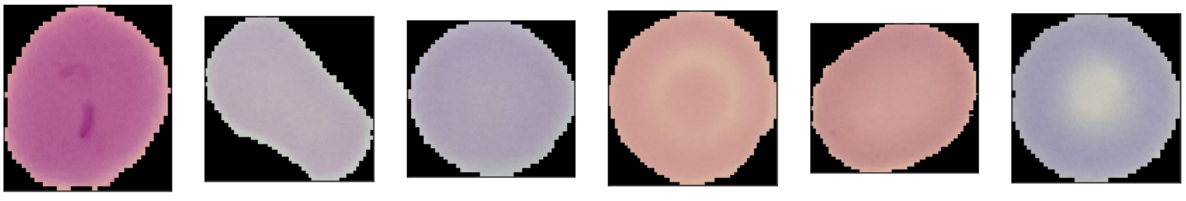

Figure 3.1.2 Examples of healthy cells

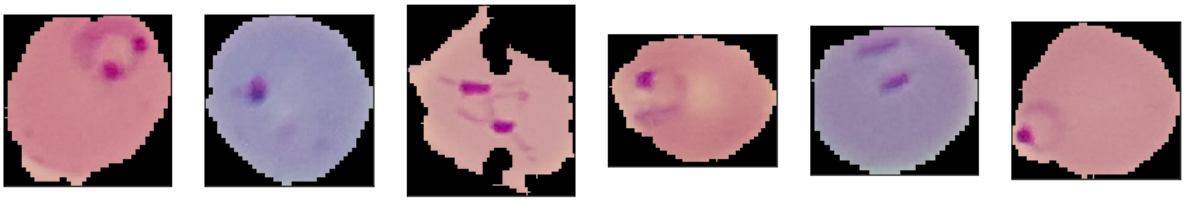

Figure 3.1.3 Examples of infected cells

In Figure 3.1.2 and Figure 3.1.3 we can detect firsthand differences between infected and healthy cells. While stained spaces can be seen in the images of infected cells, there are almost no magnificent stained spaces in the healthy cells. Therefore, we want our models to detect those characteristics and classify the cells.

3.2 Methods

We used two different methods in order to classify the infected and healthy cell images. First, we decided to use an artificial neural network. More specific we implemented a Convolutional Neural Network (CNN) which is a deep learning architecture specifically developed for image and pattern recognition. A typical CNN consists of three types of layers: The convolutional layer for pattern detection, the pooling layer for reducing the input size, and the fully connected layer for computing the output class.

The convolutional layer consists of filters, also called kernels. Those kernels can vary in size, number and stride. Every kernel is specialized on detecting a specific pattern, for example some kernels detect lines, others detect edges, or spikes. The deeper we get into the network, the more sophisticated the filters become, being able to detect whole objects, faces, and so on. The pooling layer, also known as the down-sampling layer, reduces the size of its input. Variable parameters are the pooling size and the pooling stride. There are different types of pooling operations, the most widely used being max pooling which outputs the maximum of the input values. At the end, the pooling outputs are flattened and concatenated before serving as an input for the fully connected layer which determines the output class.

However looking at Figure 3.1.2 and Figure 3.1.3, we assume that the specific pattern detections made by CNN's kernels are probably not required to detect anomalies in the given blood smears and that it is possible to build a much simpler, binary classifier to diagnose malaria on the given dataset. For instance, the usage of a Support Vector Machine

(SVM) algorithm could also solve our problem. SVM is mainly used for binary classification. If the training data is plotted n-dimensionally the algorithm tries to separate two groups of data points with a hyperplane aiming for maximum distance between the line and the closest data points of each group [7].

3.3 Preprocessing

In case of the CNN model we decided to resize the images to 50x50 pixels by interpolation. In order to solve this fast we used parallel processing. Furthermore, we split the data into training, validation and test set by the ratio 6:1:3. In the following you can see 25 samples of the new dataset:

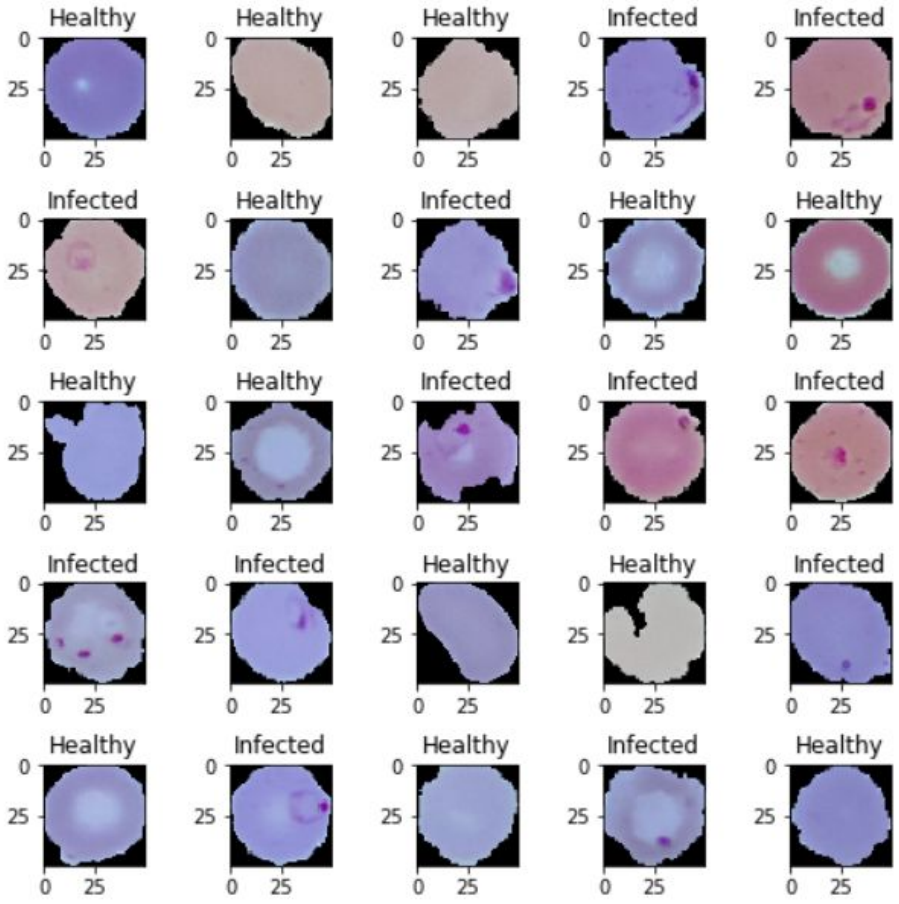

Figure 3.3.1 Examples of resized images (50x50 pixels)

Since the kernels of the CNN automatically extract and detect features on which the classification will function, there is no need for further data preprocessing.

For our model with SVM we preprocessed the dataset similar to the CNN model. In order to reduce the dimensionality of our problem even further we transformed all cell images into

grayscale. We integrated this task in the parallel process of resizing the images. In the following it is obvious that infected cells can still be distinguished from healthy cells:

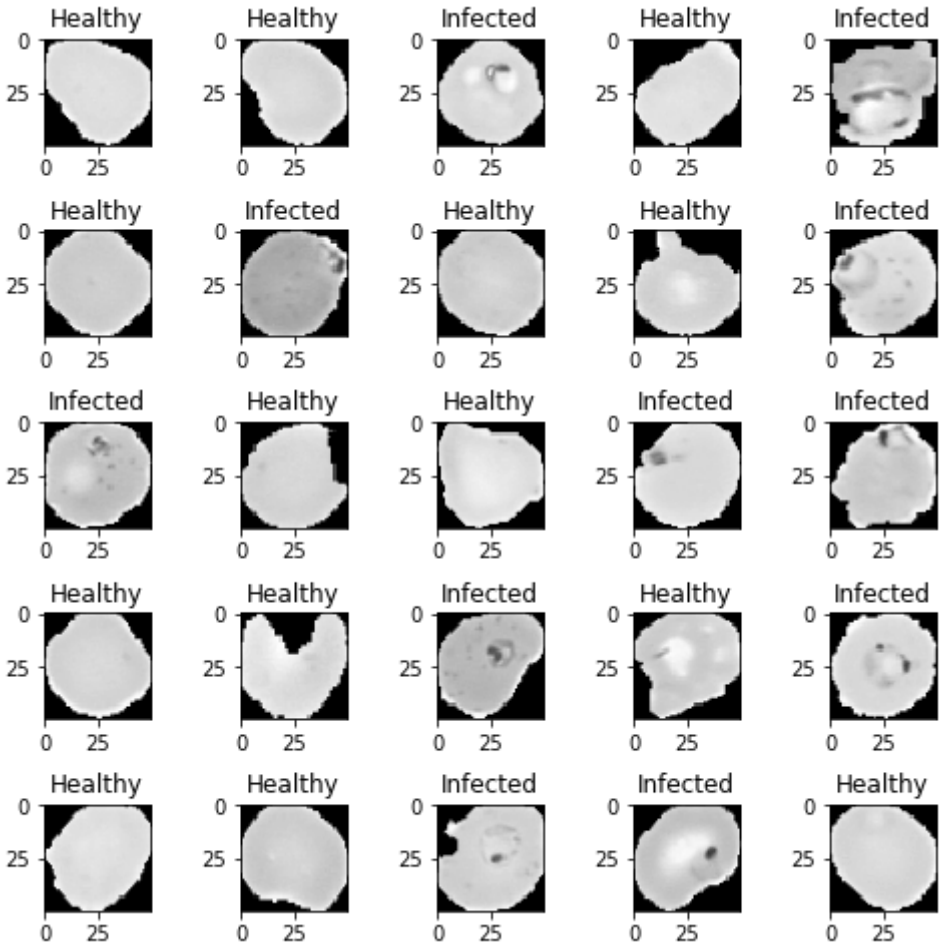

Figure 3.3.2 Examples of resized images in grayscale (50x50 pixels)

Furthermore, we used the function histnorm_cell_images in order to compute a normalized histogram of colors for each image. Finally we got a numpy array with features for each image, a so-called feature matrix. The visualisation of the normalized color histogram data can be seen here:

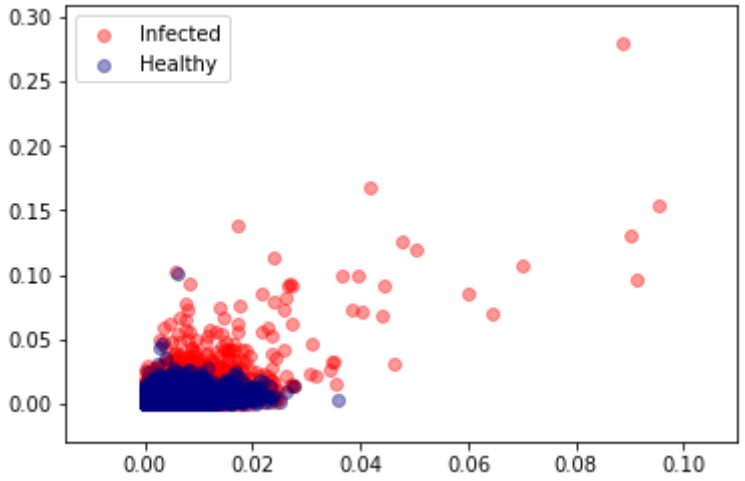

Figure 3.3.3 Normalized color histogram data plotted two-dimensional

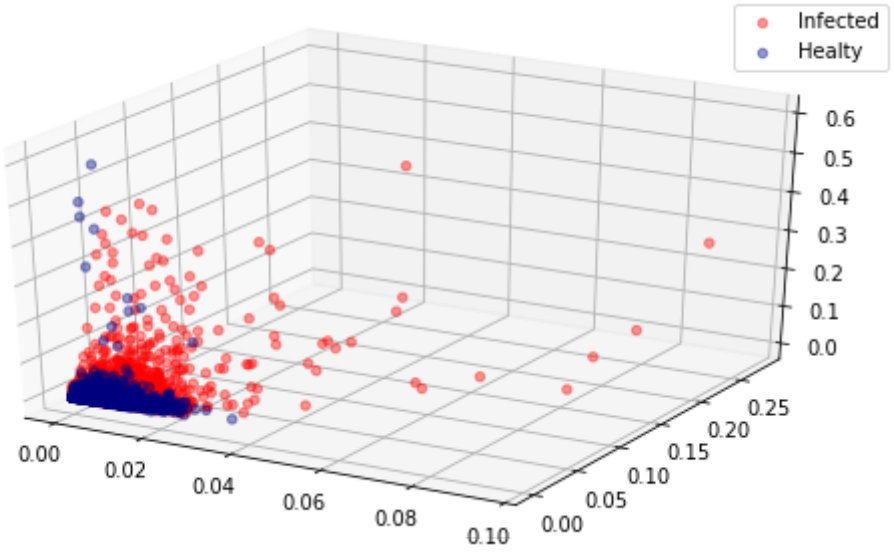

Figure 3.3.4 Normalized color histogram data plotted three-dimensional

---

The dataset will be split into train and test dataset by a ratio of 7:3.

3.4 Models

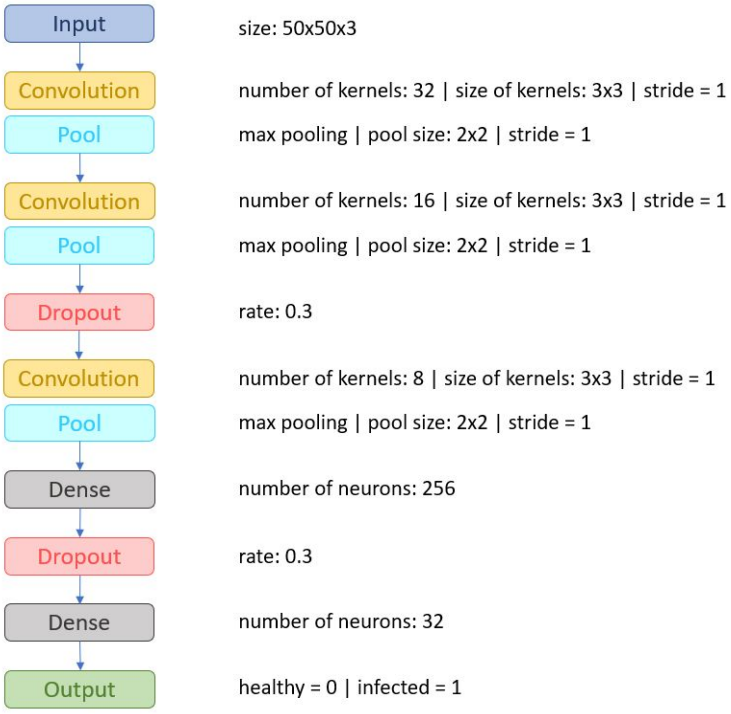

Figure 3.4.1 Model of the CNN

The CNN consists of three layers of convolution and max pooling each followed by two dense layers. The first convolution layer has 32 kernels, the second convolution layer has 16 kernels and the last one has eight kernels. Each kernel is 3x3 in size. The pooling size in each of the pooling layers is 2x2. The first dense layer consists of 256 neurons while the second only has 32 neurons. Overall our CNN model has 603,890 trainable parameters. All layers use ReLU for activation except for the output layer, which uses Softmax. To prevent the model from overfitting we added two dropouts, each with a dropout rate of 0.3. One dropout is added after the second pooling layer, another one between the dense layers. The model was trained for 20 epochs with a batch size of 64.

In order to choose an appropriate type of kernel for our SVM model we need to consider Figure 3.3.3 and Figure 3.3.4 more in detail. Seeing both visualisations two- and three-dimensional it is clear that we cannot draw a hyperplane to separate differently labeled data. We decided for SVM with Radial basis function kernel (RBF) kernel that is a non-linear

classifier. It is also known as Gaussian kernel since the kernel is in the form of a Gaussian function [8].

To use SVM the number of features needs to be reduced. PCA transforms the data in a way so that most of the information in the data is contained within a smaller number of features, the so-called components. To figure out what the best number of components is to reach a high accuracy we determine how many components we need to explain 0.95 of variance.

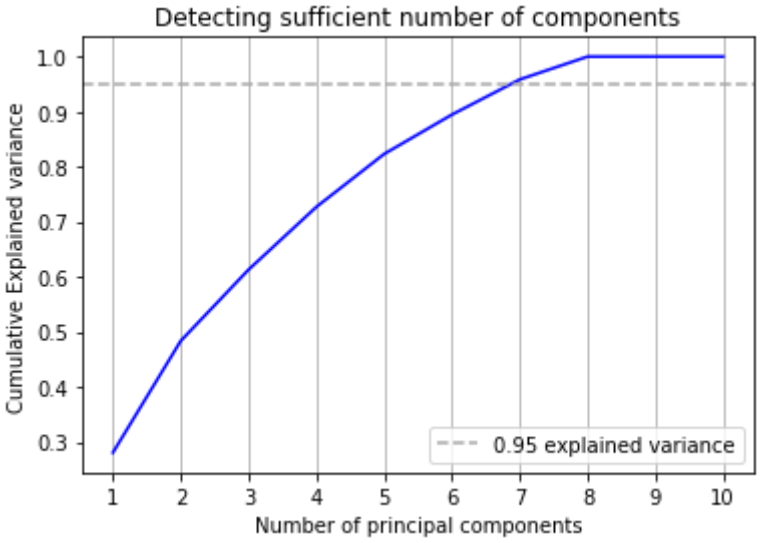

Figure 3.4.2 Plot of cumulative explained variance depending on number of principal components

As can be seen seven components are enough to explain 0.95 of the variance.

Also, it is relevant to have a look on the hyperparameters gamma and C for our SVM model with RBF kernel. Gamma defines intuitively how far the influence of a single training example reaches where low values mean ‘far’ and high values mean ‘close’. The hyperparameter C trades off correct classification of training examples against maximization of the decision function’s margin. In other words C can be seen as a regularization parameter. A high value of C means that a small margin will be accepted if the decision function is better at classifying all training points correctly. A low value of C means that a large margin will be accepted with a simpler decision function [9].
To choose the best values for the hyperparameters gamma and C we use cross validation and do grid search to fit multiple SVM models on the cell data. Beside using seven components we create a pipeline so we can standardize the data and transform it with PCA

on each fold. According to the grid search the best accuracy of 82,3% can be reached with gamma=1.0 and C=10.0.

The support vector machine can now be defined with the built-in function SVC and trained by svm.fit from the module sklearn.

3.5 Results

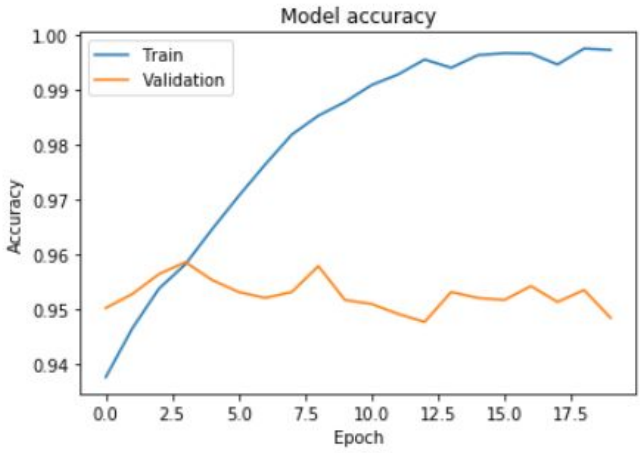

Figure 3.5.1 Training and validation accuracy

The CNN model reached an accuracy of 0.99 on the training set and an accuracy of 0.95 on the validation set. On the test set, the CNN performed with an accuracy of 0.94.
The area under the curve (AUC) is 0.98.

Table 3.5.1 Confusion matrix CNN

Predicted / True	0	1
0	4056	146
1	312	3754

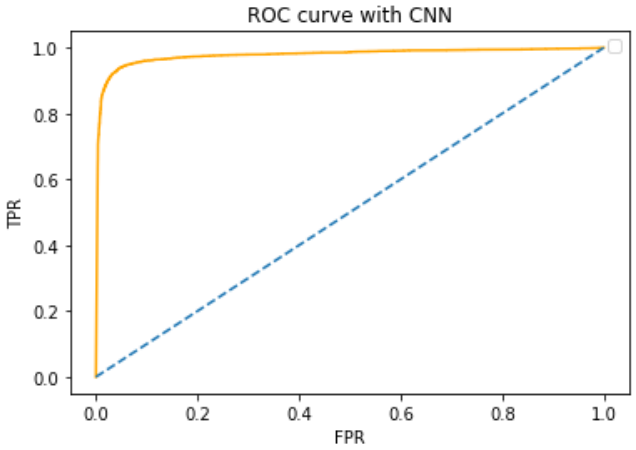

Figure 3.5.2 ROC curve CNN

The accuracy of the SVM model was measured by using the test set. The model reached an accuracy of 0.819. The area under the curve (AUC) is 0.896.

Table 3.5.2 Confusion matrix SVM

True \ Predicted	0	1
0	3479	596
1	900	3293

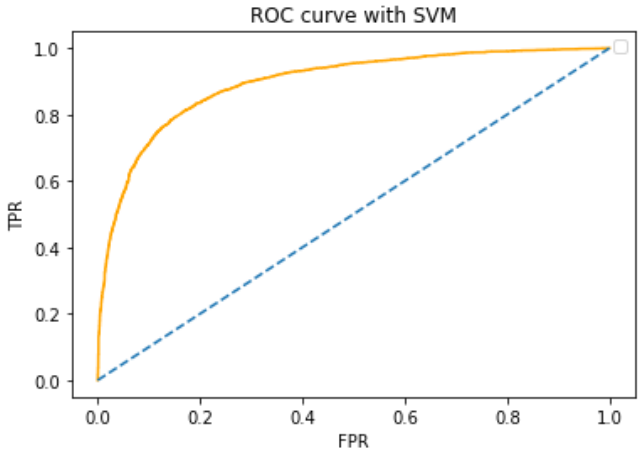

Figure 3.5.3 ROC Curve SVM

3.6 Discussion

With an accuracy of 0.94 on the test set, the CNN did a rather good job on classifying healthy and infected blood cells. We assume that it is possible to reach an even higher accuracy on this task with the CNN model. One reason why our model did not do an even better classification could be the fact that we decided to resize the input images to 50x50 pixels. This could have caused an information loss. Still, we opted for downsizing the input data for better runtime. Also note that our model slightly overfits: The training accuracy is higher than the validation and test accuracy. Since the traditional method for detecting malaria is a rather simple though time-consuming visual classification, it is obvious that the CNN model, that combines the processing power of modern computers with the ability to recognize patterns in images, is an appropriate choice for the given problem.

Even though the SVM is - unlike the CNN - not designated for image classification problems, our model still achieved an accuracy of almost 82 percent. For this method, we had to reduce the complexity of the input data quite a lot by not only resizing the images to 50x50 pixels but also by transforming them to grayscale images. It can be assumed that this feature extraction is not optimal and that the SVM could have performed better on images that are preprocessed in another way.

The value of AUC of our CNN model is 0.978 and of our SVM model is 0.896 meaning that the models classifies infected cells as infected cells especially well. Since it is important to prioritise true positive cases when it comes to diseases these classifiers succeed at this real life problem.

4. Conclusion

In the diabetes part of our project we went the way from an abstract idea through the data search, dataset preparation and experiments with different approaches to classification. Selection of the features was probably not the most optimal and surely there is a lot of space for further experiments with the dataset. However, we reached an accuracy of almost 80 percent and gained some valuable insights into the diabetes problem.

Both the CNN and the SVM did an acceptable job on the malaria part of the project. The quite straightforward SVM was able to reach an accuracy of over 80 percent on a rather complex dataset, demonstrating that even very simple machine learning architectures can be powerful. And even though our CNN results are not perfect, the performance of the model

proves that it is possible to use machine learning algorithms for supporting clinicians with detecting anomalies in medical data.

References

[1] **World Health Organization.** *Diabetes.* [Online;accessed 2019-09-20]. URL: https://www.who.int/health-topics/diabetes.

[2] **World Health Organization.** *Malaria.* [Online;accessed 2019-09-20]. URL: https://www.who.int/malaria/en/.

[3] **David E. Laaksonen, Hanna-Maaria Lakka, Leo K. Niskanen, George A. Kaplan, Jukka T. Salonen and Timo A. Lakka.** *Metabolic Syndrome and Development of Diabetes Mellitus: Application and Validation of Recently Suggested Definitions of the Metabolic Syndrome in a Prospective Cohort Study.* American Journal of Epidemiology, 156(11):1070–1077, 2002.

[4] **Gabriela Vazquez, Sue Duval, David R. Jacobs, Jr. and Karri Silventoinen.** *Comparison of Body Mass Index, Waist Circumference, and Waist/Hip Ratio in Predicting Incident Diabetes: A Meta-Analysis.* Epidemiologic Reviews, 29(1):115–128, 2007.

[5] **Tasneem Gul Kazi, Hassan Imran Afridi, Naveed Kazi, Mohammad Khan Jamali, Mohammad Bilal Arain, Nussarat Jalbani and Ghulam Abbas Kandhro.** *Copper, Chromium, Manganese, Iron, Nickel, and Zinc Levels in Biological Samples of Diabetes Mellitus Patients.* Biological Trace Element Research, 122(1):1–18, 2008.

[6] **Randie Little.** *Laboratory Procedure Manual.* University of Missouri at Columbia, 2013.

[7] **Ancud IT.** *Einführung in Machine Learning mit Python – Support Vector Machines.* [Online;accessed 2019-09-20]. URL: https://www.ancud.de/support-vector-machine-svn/.

[8] **Matthew N. Bernstein.** *The Radial Basis Function Kernel.* University of Wisconsin, 2017.

[9] **Scikit-learn developers.** *RBF SVM parameters.* [Online;accessed 2019-09-20]. URL: https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html.

Materials

[M1] *Implemented Software.* URL: https://github.com/ElviraMingazova/PRML_Project.

[M2] **Centers for Disease Control and Prevention.** *National Health and Nutrition Examination Survey.* [Online;accessed 2019-09-10]. URL: https://www.kaggle.com/cdc/national-health-and-nutrition-examination-survey.

[M3] **National Library of Medicine.** *Malaria Datasets.* [Online;accessed 2019-09-10]. URL: https://lhncbc.nlm.nih.gov/publication/pub9932.