UNIVERSITY OF AMSTERDAM

LEREN & BESLISSEN

Fall risk classifier

Authors
Elise STERK
Nienke DUETZ
Suzan ZUURMOND
Vivien VAN VELDHUIZEN

Supervisors
Shihan WANG
Tim DOOLAN
Kylian VAN GEIJTENBEEK

5 February, 2019

Contents

Chapter 1

Summary

Falling among elderly is a major problem in the Netherlands, but current screening tools that detect balance are expensive. Therefore the Digital Life Centre research lab has created a test to examine if more accessible screening tools can detect fall risk among people. For that reason, data was collected by the Bravo project, which included the location of a participant's mid spine point while walking without and with depth distorting glasses that worsen a participant's balance control.

The aim of this project is to create and compare multiple machine learning classifiers that could detect whether or not a participant is wearing glasses, thus has a worsened balance control. In addition, this project also aims to predict a participant's alcohol level, and it aims to predict the quality of a participant's standing balance control. The four implemented classifiers are logistic regression, naive Bayes, decision tree and K-NN. The features that were important to include for these classifiers in order to detect balance control, are the average step length, the gait time and the deviation of the mid spine point from the mean gait path.

For the glasses classification all the classifiers were capable of detecting a participant's condition with an accuracy of at least 84% and the best performing classifier, K-NN reaching an accuracy of 89%. This implies that the more accessible screening tools used in Bravo's test are sufficient to detect fall risk.

The alcohol and standing balance classification was more challenging, since the data was very imbalanced for these two classifications. Again, the K-NN classifiers was the best performing and the only classifier capable of predicting a participant's standing balance and alcohol level. This implies that there might be a correlation between a participant's standing balance and walking balance and between a participant's balance and alcohol level, with an increase in alcohol level leading to a decrease in balance control. However, more data is necessary to detect these correlations, so for future research it would be interesting to collect and process more data and more features.

Chapter 2

Introduction to the problem

Nowadays falling is a major problem for elderly people in the Netherlands. An increasing amount of people over 65 fall every day and the people in need of first aid as a result of falling has increased with 40% in the last ten years. Even more serious is the prediction that by 2030 this number will increase with 41% compared to 2017 (Stam, 2018).

To reduce this increase, it is essential to identify the elderly that have an increased risk of falling. However, current screening tools that could identify this risk are too expensive and require professional assistance. Therefore, the Digital Life Centre research lab at the Amsterdam University of Applied Sciences has set up the BRAVO project. This project's goal is to develop a screening test that can identify fall risk, while also being accessible and easy to use.

For this purpose, BRAVO has decided to test the use of Xbox Kinect as a screening tool, since it is inexpensive and easy to handle. The Kinect was tested on Dutch music festival Lowlands, where participants had to perform a series of actions while being monitored by the Kinect. They had to do this two times; one time without disabilities and one time while wearing vision distorting glasses, which affect depth perception.

Given this movement data, the goal of this project is to create multiple machine learning classifiers that are capable of identifying if a participant is wearing glasses or not. The classifiers should be able to identify this by processing the movement data acquired by the Kinect. By comparing the classifiers to each other, it should be determined which algorithm can classify the data with the highest accuracy.

Chapter 3

Theoretical framework

1 Technical design: About the data

As stated in the introduction, the data for this project was collected during an experiment held at Dutch music festival Lowlands. Participants had to perform two tests concerning their balance: a dynamic walking balance test and a static balance test, both with and without wearing vision distorting glasses. A total of 205 people participated in the experiment. Their test results, as well as their biometrics and alcohol consumption were included in the data set used for this project. This section will give an overview of all the data that was obtained and explain why and how this was done. Further details can be found in appendix A.1, which lists all specific variables and their descriptions.

1.1 Walking Test

The main problem of this project is to create a classifier that can tell if a participant is wearing depth distorting glasses or not. These glasses were worn during an experiment that tested a participant's dynamic balance: the walking test.

For the walking test, participants had to stand up from a chair, walk three metres back and forth four times at comfortable speed, then sit back in the chair. This test was repeated twice: once while wearing the depth perception distorting glasses (the "glasses" condition) and once without glasses ("control" condition). Data about the gait movement was collected via Xbox Kinect sensors. These sensors measured multiple body parts, but only data for the mid spine was included in the data set. Also excluded was the data for sitting and standing up, leaving just the walking part of the data. Throughout this report, the walking data will be referred to as "segment data", and will be the main focus for the glasses classification problem.

1.2 Biometrics

Besides the walking test, the participant's biometrics: age, height, weight and body mass index were included in the data set. This was done because research has shown that they can impact someone's balance control.

Age, for example, has been shown to have a negative effect on both dynamic and static balance. Older adults generally display more body sway in their gait than younger adults, as well as longer movement times when walking, which can lead to worsened balance control (Hageman, Leibowitz, & Blanke, 1995). Increased sway, both standing and walking, is also observed in individuals who are taller (Greve, Cuğ, Dülgeroğlu, Brech, & Alonso, 2013). This might explain why studies generally find a relationship between body height and balance control (Alonso et al., 2012). Furthermore, the influence of BMI or weight is less straightforward: variations in normal BMI do not appear to influence one's balance control, but it does so in extreme situations (i.e. obesity) (Alonso et al., 2012).

1.3 Alcohol

Another variable that was measured was the alcohol breath concentration in permillage. As stated by the Trimbos institute, alcohol could have a negative effect on someone's gait and balance, so considering every participant's alcohol consumption level will be useful in classifying the results of the walking test (Trimbos Institute, 2017).

Furthermore, this data could also form the basis of an additional problem: classifying alcohol level. If someone's alcohol consumption influences their balance and gait, their alcohol level might be observed from the segment data. Whether or not a classifier is able to predict a alcohol class based on the segment and balance data, will be an additional problem for this project to explore.

1.4 Balance Test

Separately from the walking test, a balance test was conducted for every participant. For this test, participants had to stand in place, one leg in front of them and one leg behind them, then hold this position for thirty seconds. During these thirty seconds, the Kinect sensors monitored how much the participants swayed from their central point. This test was again conducted twice, once with glasses and once without. Multiple variables were measured and were combined into the "Balancemetric" data. Also, it was already identified by BRAVO that only the tests without glasses needed to be included for this problem, since the glasses had no influence on the balance test.

The reason that this balance data was collected, was because it might be related to someone's performance in the walking test, which might be useful in classifying the glasses or no glasses condition. Another reason was to use it for building an additional classifier, one that could predict someone's balance score based on the other data, such as the segment data.

Some studies suggest that static balance, like this balance test, and dynamic balance, like the walking test, are not heavily correlated (Kiss, Schedler, & Muehlbauer, 2018) (Shimada et al., 2003) (Karimi & Solomonidis, 2011). This seems against trying to classify the static balance with dynamic balance. However, age, height, weight and alcohol level can influence both balances (Ikai, Tatsuno, & Miyano, 2006) (Trimbos Institute, 2017), which suggests a coherence between the two. Whether it is possible to classify standing balance score and testing which variables are the most useful for this, will be another additional problem this project is concerned with.

2 Technical design: Data Preprocessing

This section will detail the steps taken for preprocessing the data. At first, the discarding of data is underpinned. Secondly, it will be explained how the original data set is restructured to the data set used in this project. Finally, the additional preprocessing steps for both the alcohol and balance problems will be described.

Beforehand, it should be noted that the data was already partially preprocessed by the stakeholder. For example, erroneous data was removed, everything except for the data collected about a participant's mid spine was excluded and the segmentation of the gait sections was already done.

2.1 Discarding Incomplete Data

There were three participants who had missing data entries: subject 75 and 111 did not have alcohol scores and participant 146 did not have a value for weight. Thus, these three participants were excluded from the data. Also, not all of the participants walked four segments. Since the data is compared per participant, as described in the next part, the data of participants that walked more or less than four segments was discarded. By doing this, the data per participant would be based on an equal amount of data. Furthermore, a small group of participants walked multiple trials. This could have different causes: a fault in the filming equipment or the participant making a mistake. This might have caused a participant to get more accustomed to the depth distortion glasses, so this group is also discarded. Both these groups were only a small part of the data (see Figure 3.1).

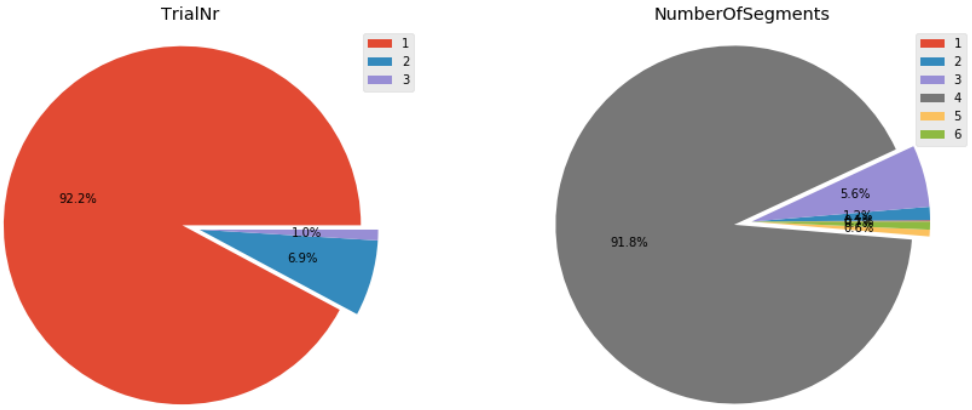


Figure 3.1: Percentage of data that was discarded from the original data set

2.2 Adjusting the Data set

The original data set is structured around single segments. This means that for every participant multiple segments are given. Since the classification problems ask for a classification per participant and not per segment, the data is restructured to a data set per participant. By doing this, multiple columns are dropped and other columns are combined into one.

Table 3.1 gives an overview of which functions are used for the new personal data set. The 'first' function selects the first value of the multiple segment rows. This function is used for all the variables that are equal for every segment a participant walks, for instance their height, alcohol level and ID. For other variables that differ per segment, the mean is taken. For instance, this is done for the step length, median walking speed and total walking time. All the variables in table 3.1 are explained in A.2. The variables *SegmentNr*, *WalkingDirection* and *SegmentData* are excluded from the adjusted data set, since they are not relevant anymore.

Variable	**Functions**	**Variable**	**Functions**
Conditie	first	*SubjectID*	first
Height	first	*Weight*	first
Age	first	*BMI*	first
Alcohol	first	*Alcohol*_Class	first
Balance_MLrange	first	*Balance_MLstdev*	first
Balance_MLmeanVelocity	first	*Balance_APrange*	first
Balance_APstdev	first	*Balance_APmeanVelocity*	first
Balance_MeanVelocity	first	*GaitVelocity*	mean
GaitTime	mean	*MovementVelocity*	mean
std_SegmentData	mean	*mad_SegmentData*	mean
mean_StepLength	mean		

Table 3.1: The functions used to restructure the data set to a personal data set

2.3 Preprocessing - Alcohol Problem

For the alcohol problem, the task is to identify if a classifier is capable of predicting a participant's alcohol level. Including segments with the glasses condition will make this more difficult, since participants without alcohol will also have less balance control. Therefore, all the data that was collected with glasses is excluded for this particular problem.

The alcohol score was measured in continuous permillage, as opposed to the predefined classes of *Condition*. In order to classify the alcohol level, three specific labels were identified from the continuous data: no, low and high alcohol concentration. The labels are based on the Dutch mental health and addiction institute Trimbos, which states that a person's balance already slightly declines when they are tipsy (0.5 - 1.5 permillage) and declines even more when they are drunk (1.5 - 3.0 permillage) (Trimbos Institute, 2017). With these classes, a participant's alcohol level could be predicted based on data like the balance score, segment data or other variables. The continuous alcohol data was also kept in the data set, as alcohol breath concentration might play a roll in the other classification problems.

Over fifty percent of the participants was sober when they did the experiment. This means that the number of participants between no, low and high is not equally divided (see 3.2). Applying a machine learning classifier on such an unequally divided data set will lead to unreliable and skewed results. Therefore, a different `Scikit library` was necessary to divide the data more equally, without having to delete data. This will be explained in section 4: Machine Learning.

Another method that was used to more accurately predict the classes for the alcohol and balance problems, was oversampling of the data. Oversampling involves generating new data for the classes that are less represented in the data. For the alcohol classification this consists of creating extra data for the low and high classes. To randomly create this extra data, *Python's imbalanced learn library* was used, specifically it's *RandomOverSampler()* (Imbalanced Learn, n.d.).

Alcohol_class	**Participants**
No	90
Low	60
High	4

Table 3.2: Alcohol class unequal division

2.4 Preprocessing - Balance Problem

For the balance problem, the task is to identify if a classifier is capable of predicting a participant's standing balance. Since it was already identified by BRAVO that the glasses had no influence on the balance test only the tests without glasses are included for this problem.

The standing balance test, described in section 1.4, resulted in multiple variables that made the balance metric. The variables *Balance_MLrange*, *Balance_MLstdev* and *Balance_MLmeanVelocity* describe the medial/lateral instability, while *Balance_APrange*, *Balance_APstdev* and *Balance_APmeanVelocity* describe the anterior/posterior instability. The variable *Balance_MeanVelocity* can be seen as a describing value that takes both directions in account. Therefore, it was decided to use only this last variable. The other six variables are excluded.

Similar to the alcohol problem, the outcome of the balance test is on a continuous scale. In order to classify the alcohol level, two labels were identified from the continuous data: balance and no balance. By analyzing the distribution of the *Balance_MeanVelocity* variable, the boundary is set at 6.0. This corresponds to the boundary that is set by the stakeholder. A score of less than 6.0 corresponds to a good balance, while a score of 6.0 and above corresponds to no balance. Since this split results in two unequally sized groups (see figure 3.2), a different `Scikit library` was necessary to divide the data more equally. This is explained in section 4: Machine Learning.

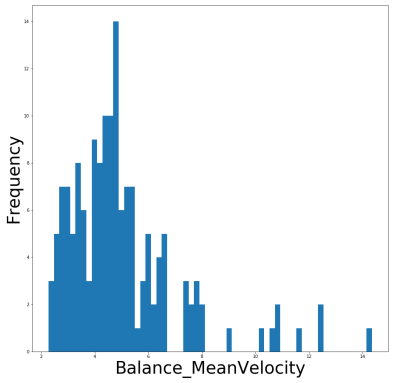


Figure 3.2: The distribution of the outcome of the standing balance test

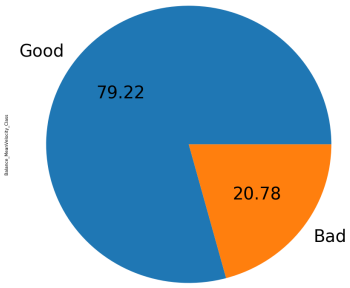


Figure 3.3: The division of the two balance groups

3 Technical Design: Segment Data Preprocessing and Visualization

To make a classifier for the three different problems (condition, alcohol and balance) the segment data was analyzed. The segment data consists of multiple x,y,z-coordinate triples, which represent the middle of the spine of the participant while walking. The x-coordinate represents the participant's movement from left to right, the y-coordinate from up to down and the z-coordinate the distance that the participant has walked. With these points it is possible to make a 3D-representation (3.4) and compare different walks. From comparing these 3D-representations, multiple describing variables have been found and calculated. The descriptors that were calculated were the standard deviation, the mean absolute deviation, the average step length and the walking time.

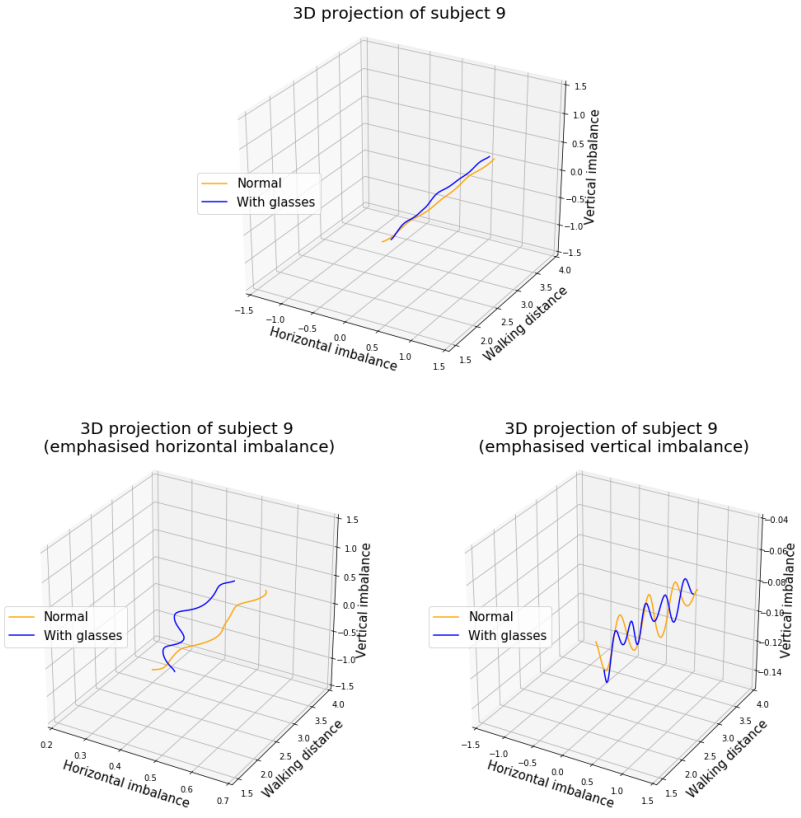


Figure 3.4: 3D projection of a gait of the same subject with and without wearing depth distorting glasses

3.1 Standard and Mean Absolute Deviation

Research suggests that people with distorted balance tend to walk less straight and with wider steps (Kawamura, Tokuhiro, & Takechi, 1991). This was visible in the segment data. If the x-coordinates are plotted against the z-coordinates, a difference is seen in deviation from the straight line that should have been walked (figure 3.2). To compare different walks and conditions with each other, both the standard deviation and the mean absolute deviation of the x-coordinates were calculated.

At first, the standard deviation is calculated, since this gives a standardized value to the amount of deviation in a set of data points.

To improve accuracy, a switch was made from using standard deviation to the mean absolute deviation. This measure takes all absolute deviations from the mean and computes their average. The result is a number that represents the total amount of variability for that data. Mean absolute deviations are more robust and less susceptible to outliers, making them a good fit for out data (Pham-Gia & Hung, 2001).

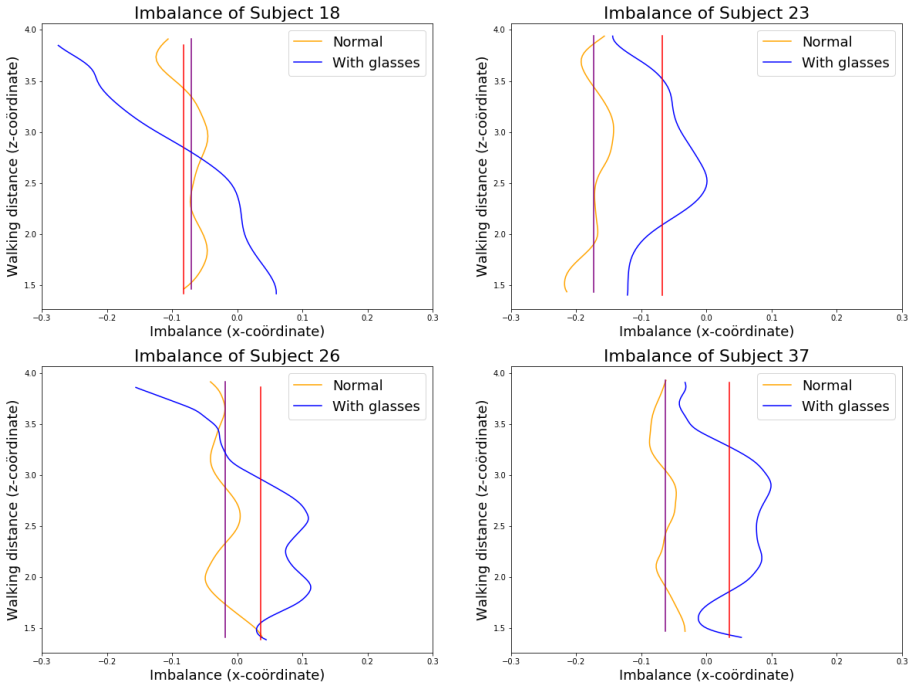


Figure 3.5: Visualisation of the horizontal deviation for various subjects

3.2 Average Step Length

Another feature that is used as an indicator for distorted balance is the length of a person's stride. Multiple studies suggest that an impaired sense of balance causes people to adjust their stride: their stride width is increased, as well as their step frequency, and most significantly, their step length is shortened (Hak, Beek, & van Dieën, 2013) (Hak et al., 2012). By analyzing the y and z-coordinates, the average stride length of one gait could be calculated. This was done by calculating the distance between the y-coordinates of every local maxima and taking the average of these distances per gait (see figure 3.6).

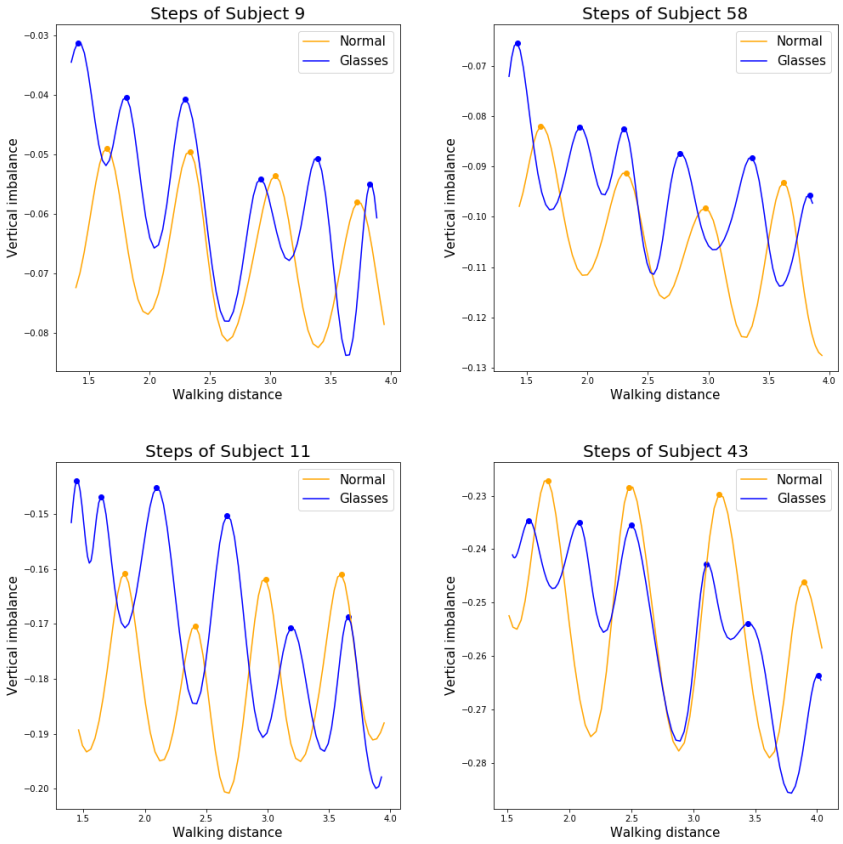


Figure 3.6: Visualisation of the step length and local maxima for various subjects

3.3 Gait Time

The last feature that is used to indicate someone's imbalance is someone's walking speed, where an imbalanced person prioritizes stability over speed (Hak et al., 2013). Since the Kinect makes thirty dataframes (coordinate triples) per second, it is possible to derive the time a participant takes for one gait from the length of the segment data. This gait time, together with the *GaitVelocity* (the median gait velocity of head during gait segment in Z-direction), give an indication of someones walking speed.

4 Machine Learning

Classifiers

The aim of this research is to compare different classifiers and to identify the classifier with the best performance. For this purpose, logistic regression, naive Bayes, K-nearest neighbors (K-NN) and decision trees were chosen as the main classifiers. All four are specifically used for supervised learning problems, making them well suited for the data set. Another feature these classifiers all have in common, is that they do not require any complicated optimization. This makes them more convenient, since the main task of this project is to compare multiple classifiers and having to do a lot of optimization might take up unnecessary time. Specific details about the four classifiers and why they were chosen will be specified later this section.

Near the end of the project it was decided to implement three more classifiers to see if an even higher accuracy could be achieved. However, these classifiers were only implemented superficially; no optimization was conducted and they were only implemented for the glasses problem. Overall, this project's main focus is on the first four classifiers, therefore these classifiers will not be discussed in too much detail.

Given the task to compare multiple classifiers and given the time frame for this project, it was not feasible to build all classifiers from scratch. Therefore, the classifiers were all implemented with Python's
`Scikit-learn package`. Furthermore, the code was written in Python 3.6 in a Jupyter Notebook environment.(Pedregosa et al., 2011), (Kluyver et al., 2016). The decision to work with Jupyter Notebook was made because it is very convenient for visualizing data and plotting graphs, resulting in a clear overview of the available data.

Feature Scaling and PCA

Before implementing the classifiers, feature scaling and dimensionality reduction by principal component analysis (PCA) was applied to the data. Feature scaling and dimensionality reduction are important steps to conduct before running the classifier on the data (Scikit Learn, n.d.). Standardization of the data means rescaling the values of the independent variables so that they are normally distributed. Since the range of values for specific variables varies widely in this project's data and the properties that are measured differ for every variable, standardizing the independent variables might lead to a higher accuracy.

Dimensionality reduction could lead to better results when more than two variables are being included in a classifier, since PCA determines which variables are more important for the classification. PCA is also important for plotting more than two variables, otherwise the plots will not visualize the data correctly. It was decided to run the classifiers both on the scaled and reduced data and on the normal data, to then choose the data that leads to the highest

accuracy. Again `Scikit-learn packages` were used for implementing feature scaling and dimensionality reduction.

Finding Correlation

Since there are many variables in the data, it can be a challenge to find the variables with the highest or lowest correlation. The variables that would lead to the highest accuracy were included together in the classifier. To achieve the highest accuracy a feature selection function from `seaborn` (a Python data visualization library) (Seaborn, n.d.) was used to compare all the variables in the data. This function returns a correlation matrix with colors for how correlated two variables are. Using this matrix to select the best variables will lead to the highest possible accuracy.

Training en Validation Sets

Moreover, an important part of machine learning is dividing the data set into a training set and a validation set. The training set is used to train the classifier and the validation set is used to test the accuracy of the trained classifier and test if the classifier does not overfit. For this project the ratio for the training and validation set was 70/30. Since every participant appeared twice in the data, with and without glasses, it was important for the training/validation split function that both the glasses and no glasses data from the same participant was divided into the same set. Therefore, it was decided to not use a `Scikit-learn package`, but to build a customized split function from scratch.

The situation was different for the alcohol and balance classification, since all the glasses data was deleted for these two tasks. Therefore, every participant only appears once in the data and no customized split function was necessary. `Scikit's train_test_split()` function was used to divide the participants between a training and validation set with as ratio 70/30. An important parameter to set for this split function is the `stratify = y` parameter. This is important since both the data for the alcohol and balance classification is skewed, since the number of participants are not equally divided among the different classes. The `stratify = y` parameter confirms that the training data equals the classes division from the data set.

Moreover, the data gets shuffled every time the notebooks are being loaded. This results in a slightly different accuracy for every shuffle. Therefore, a code was written to loop through the data 100 times and calculate the accuracy every time. The mean of the 100 accuracy scores was taken and returned by the algorithm. This ensured that the accuracy was reliable and accurate.

Performance Metrics

Lastly, for the glasses classification only the accuracy score is reported. This score is very reliable for the glasses classification, because the data is perfectly balanced, meaning there are equal numbers of no glasses and glasses classes in the data. As explained, this is not the case for the alcohol and balance classification. Therefore, the F1-score instead of the accuracy score was calculated for the stratified data for the balance and alcohol classification. For the classification including the oversampled data, only the accuracy score was calculated and reported, since the oversampled data is balanced.

4.1 Logistic Regression

Logistic regression is a supervised learning classifier that has a dependent variable with two possible values namely "0" and "1". For this project the "0" and "1" are the no glasses and the glasses labels. To predict the labels of the data points at least two independent variables need to be included in the logistic regression classifier. The classifier can then use the data from these independent variables to classify a participant's label. How many and which variables need to be included to achieve the highest accuracy score is unknown. Therefore, feature selection was implemented and different combinations of variables were included, tested and plotted.

The logistic regression classifier was not built from scratch. Rather, variables were loaded into `Scikit's` logistic regression classifier. However, some parameters had to be altered for the data set. To include the best parameters in the classifier, multiple different parameter settings were tested and the parameters with the highest performance were set. This differs for the three sub problems. Namely, the alcohol and balance test are multiclass problems, therefore the parameter of the classifier had to include `multi_class='multinomial'` and `solver='lbfgs'`. Defining `multi_class='multinomial` was not necessary for the glasses problem. The solver is the same for the glasses problem and the `max_iter` was set to 1000 for both the multiclass and glasses problems.

4.2 Naive Bayes

Naive Bayes is similar to logistic regression, with the difference being that `Scikit's GaussianNB()` classifier was used. Naive Bayes is also a '0'/'1' classification problem, but can be used to classify more classes, such as for the alcohol and balance problem. Similar to logistic regression, to find the highest accuracy, many different combinations of independent variables were included, tested and plotted by the classifier.

No parameters were included in the classifier, since there are not many parameters to choose from, and none were necessary to achieve a higher accuracy.

4.3 K-NN

K-NN is similar to the above two classifiers and
`Scikit's KNeighborsClassifier()` was used. Just as the previous two classifiers, K-NN is a supervised classification algorithm, the difference being that all the data points in K-NN are being divided based on the labels of the k nearest data points, whereas for logistic regression and naive bayes decision boundaries between data points are being calculated.

The only parameter specified in the K-NN classifier is the `n_neighbors=k` parameter. Essential for specifying this parameter is an algorithm that determines the amount of neighbors (k) with the lowest mean error. Using this specific amount of neighbors leads to the highest accuracy. This algorithm was included in the code that loops through the data 100 times and calculates the mean of the accuracy scores.

4.4 Decision Trees

Another classifier that was used is the decision tree. This classifier splits data into an increasing amount of smaller subsets, grouping similar data points together, until ultimately a classification can be made. The groups are split in a way that the first few splits are decided by the most important features, which allows for a neat overview of the made decisions and the features that had the most impact in classifying the data. The main convenience of decision trees over the other used algorithms is in this visualizability. Plotting a decision tree results in easy to read tree structure, which displays exactly what rules were used to split at what points. It is also easy to get a sense of the data; it shows the relations between data points, while also telling which features are important.

When it comes to decision trees, setting the right parameters is important, since a standard tree will often be too big and complicated to read. Rather, it needs to be "pruned", that is a limit needs to be set to the depth and wideness. For this problem, the parameters `max_depth`, `min_samples_split` and `min_samples_leaf` were adjusted. Optimal values for these parameters were found by plotting the accuracy scores of different classifier parameters (Fraj, 2017). These plots and more in depth explanations can be found in the results section.

4.5 Other classifiers

Neural network

For the neural network implementation `Scikit's MLPClassifier()` was tested. This is a supervised multi-layer perceptron classifier and requires many parameter optimizations. Due to time limits only a few parameter settings were tried and set, meaning that the classifier is not optimal.

Ensemble

Another classifier that was tested is an ensemble learning classifier. This classifier combines multiple models into one classifier. `Scikit's VotingClassifier()` was used to combine logistic regression, naive bayes, K-NN and decision tree into one classifier to predict class labels.

Random Forests

In order to improve the reliability of a decision tree, a multitude of decision trees can be computed and combined. This is the principle behind random forests, an ensemble method that computes multiple decisions trees and returns the class that was the most frequent output among all decision trees.

Random forests essentially work the same way as regular decision trees do, except an extra parameter needs to be set; namely the desired amount of decision trees in one random forest. In order to approximate this value the best way possible, the accuracy scores for these *n_estimators* were also plotted.

Chapter 4

Results

1 Feature scaling

In section four of the theoretical framework it was mentioned that the classifiers were applied to both the normal data and the standardized and reduced data. Applying the classifiers to the standardized data leads to a slightly higher result (1-2% accuracy increase) for all classifiers. Therefore, the results section includes the results from the standardized data set and dimensionality reduction to two dimensions has been applied on the data from this section.

Before defining the classifiers, feature selection was conducted. The correlation matrix is displayed by figure 4.1. In the matrix it is clearly visible that features either have a high correlation (red squares), no correlation (light squares) or have a low correlation (blue squares). The features with high correlation are very similar to each other and therefore not optimal to include together in a classifier. On the other hand, the features with a low correlation differ significantly from one another, thus they might lead to good results when included together. From the matrix it is visible that the variables calculated from the SegmentData *GaitTime*, *mean_Steplength*, *mad_SegmentData* and *std_SegmentData* and the variables *GaitVelocity*, *Balance_MeanVelocity* and *MovementVelocity* are all variables that should be tested in different formations for the classifiers. The matrix does not imply that the variables *Age* and *Height* should be included in the classifiers. However, previous research suggests that an increased height and age results in less balance control. Therefore, it was decided to include both variables for the classification.

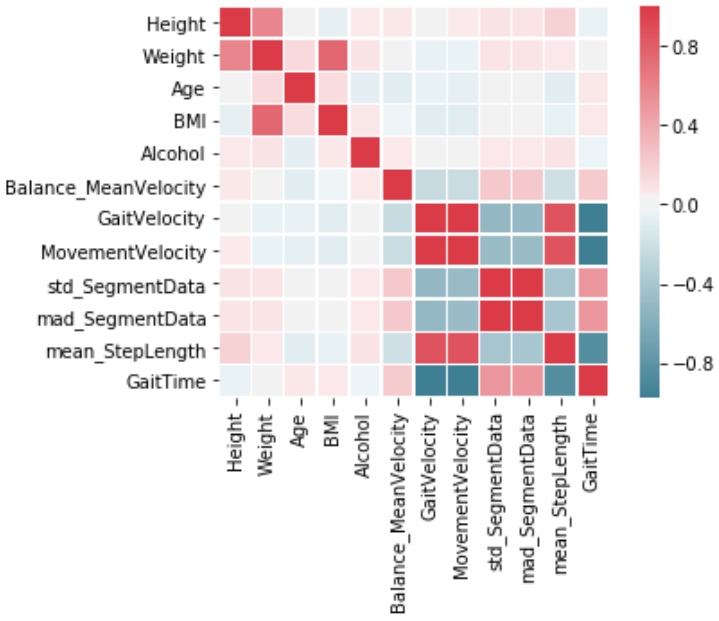


Figure 4.1: Correlation matrix feature selection

2 Glasses vs. No Glasses

2.1 Logistic Regression

The variables identified by the correlation matrix in figure 4.1 were included in different combinations for the logistic regression classifier. The accuracy of all these combinations is displayed in table 4.1. This accuracy is the mean of 100 loops as explained in the machine learning section. Not all possible combinations are included in the table in this report. Only the combinations that lead to the highest accuracy and the combinations that show a significant decrease in accuracy are displayed in the tables. This is true for all the tables in the results section.

From table 4.1, it is clear that most variable combinations lead to an accuracy between 84% and 87%. Including *Height* and *Age* does not necessary decrease accuracy. On the contrary, the variable combination *GaitTime*, *mean_StepLength* and *Height* leads to the second highest accuracy of 87.13 %. However, when the variables *Age* and *Height* are included in a combination without *GaitTime* or *mean_Steplength* the accuracy drops significantly. The combination *std_SegmentData*, *GaitTime* and *mean_StepLength* results in the highest accuracy score of 87.76%.

Included variables	**Accuracy**
mad_SegmentData and mean_StepLength	85.79%
mean_StepLength and std_SegmentData	86.2%
GaitTime, mean_StepLength and mad_SegmentData	86.06 %
std_SegmentData, GaitTime and mean_StepLength	87.76%
GaitTime, mean_StepLength and Height	87.13%
GaitTime, mean_StepLength and Age	84.66 %
mean_StepLength and Height	86.26 %
mean_StepLength and Age	84.76 %
GaitTime and Height	84.85 %
std_SegmentData and Height	72.28 %
mad_SegmentData and Age	71.78 %
std_SegmentData, GaitVelocity, MovementVelocity, mean_StepLength	85.38 %
GaitTime, GaitVelocity, MovementVelocity, mean_StepLength, std_SegmentData	84.31 %

Table 4.1: Logistic regression accuracy scores

The different variables were plotted for visualization and understanding of the classifier. Figure 4.2 is a plot with as independent variables *mean_StepLength*, *GaitTime* and *mad_SegmentData*. These three variables have been normalized and PCA has been applied. There are no x and y labels, since dimensionality reduction alters the variables into two principal components. This results in no specific variable names for the x and y axis. Also, the values plotted on the x and y axis are normalized, which means that they are different from the values of the variables included in the classifier. There is a clear decision boundary visible in the plot. This decision boundary was plotted based on the training set. The data points that are scattered in the plot are from the validation set. As a result, the plot indicates the data points that were misclassified.

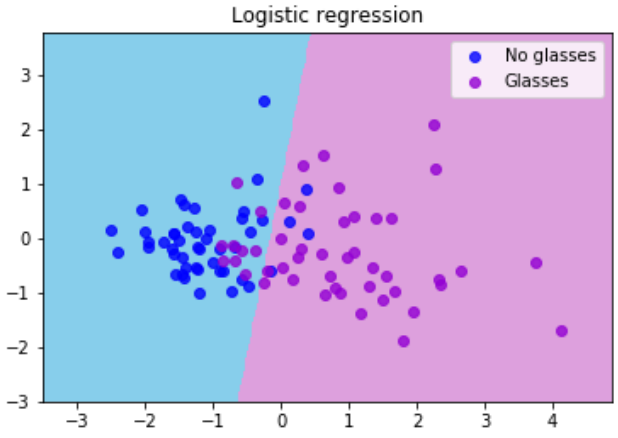


Figure 4.2: Logistic regression plot

2.2 Naive Bayes

Table 4.2 shows the accuracy scores from the naive Bayes classifier. The accuracy for the different combinations is relatively stable and ranges from 84% to 86%. This is a slightly lower performance compared to the logistic regression classifier. Again, including *Height* and *Age* without *GaitTime* or *mean_StepLength* leads to a performance drop of approximately 12%. But including *Height* with *GaitTime* and *mean_stepLength* results in the best performance from the bayes classifier.

Included variables	Accuracy
mad_SegmentData and mean_StepLength	84.79%
mean_StepLength and std_SegmentData	86.2%
GaitTime, mean_StepLength and mad_SegmentData	85.06 %
std_SegmentData, GaitTime and mean_StepLength	84.99%
GaitTime, mean_StepLength and Height	86.33%
GaitTime, mean_StepLength and Age	85.66 %
mean_StepLength and Height	86.26 %
mean_StepLength and Age	84.76 %
GaitTime and Height	78.85 %
std_SegmentData and Height	71.28 %
mad_SegmentData and Age	70.78 %
std_SegmentData, GaitVelocity, MovementVelocity, mean_StepLength	84.72 %
GaitTime, GaitVelocity, MovementVelocity, mean_StepLength, std_SegmentData	84.21 %

Table 4.2: Naive Bayes accuracy scores

Figure 4.3 is a plot similar to figure 4.2 containing the variables *GaitTime*, *mad_SegmentData* and *mean_StepLength*. In the figure a clear decision boundary is drawn and the misclassified data points are displayed.

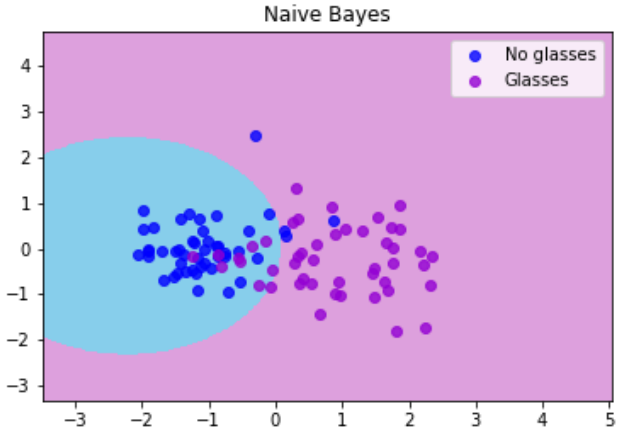


Figure 4.3: Naive Bayes plot

2.3 K-NN

To determine the optimal number of neighbors, an algorithm was used that calculates the mean error of every k-value from the range of one to forty and plots the errors in a graph (figure 4.4) (Robinson, 2018). In this figure it is visible that the optimal number of k for a specific data shuffle is 3.

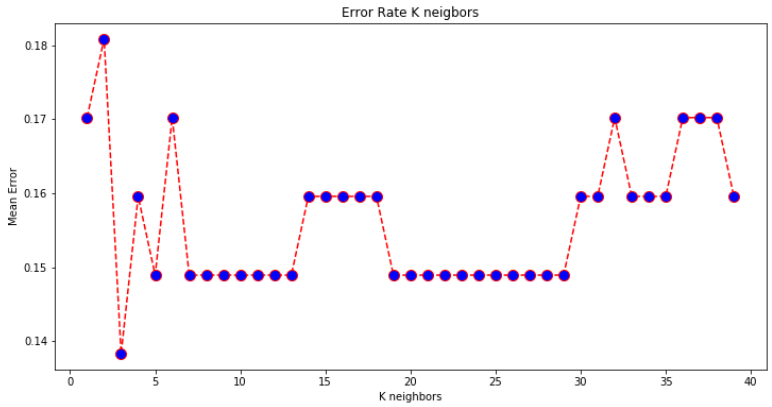


Figure 4.4: Error rate k neighbors

Table 4.3 shows the accuracy scores for the K-NN classifier. The scores are slightly higher compared to the previous two classifiers and range from 86% to 89%. Again, there is a significant performance drop for the variable combination *std_SegmentData* and *Height* and for the combination *mad_SegmentData* and *Age*. The best performance is achieved when *std_SegmentData*, *GaitTime* and *mean_StepLength* are included together.

Included variables	Accuracy
mad_SegmentData and mean_StepLength	88.25%
mean_StepLength and std_SegmentData	87.23%
GaitTime, mean_StepLength and mad_SegmentData	88.44 %
std_SegmentData, GaitTime and mean_StepLength	88.61%
GaitTime, mean_StepLength and Height	88.46%
GaitTime, mean_StepLength and Age	87.34 %
mean_StepLength and Height	87.84 %
mean_StepLength and Age	87.45 %
GaitTime and Height	86.19 %
std_SegmentData and Height	73.61 %
mad_SegmentData and Age	74.02 %
std_SegmentData, GaitVelocity, MovementVelocity, mean_StepLength	87.49 %
GaitTime, GaitVelocity, MovementVelocity, mean_StepLength, std_SegmentData	87.11 %

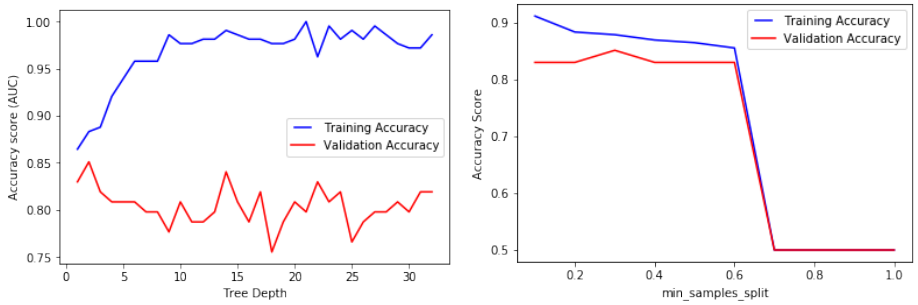
Table 4.3: K-NN accuracy scores

2.4 Decision tree

As was mentioned before, the optimal parameters for the decision trees were found by analyzing a range of plotted parameters (see figure **??**). For example, figure 4.5a shows that a greater maximum tree depth value, causes the model to overfit quite easily, meaning a small tree depth should be used. Furthermore, figure 4.5b shows that the minimum number of samples needed to split a node should also not be too high, as a values greater than 0.7 causes the model to be unable to learn anything. The same is true for figure 4.5c, which tells us that the minimum amount of samples contained in a leaf node should not be greater than 0.3, as more will cause the model to underfit.

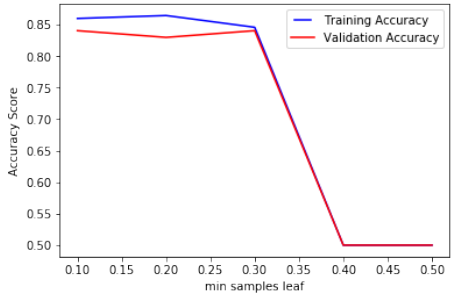
Table 4.4 shows the achieved accuracy scores for the decision tree classifier. The accuracy scores generally range from 81% to 83%, which is relatively lower than the other classifiers. *GaitTime, mean_StepLength* and *Height* scores the highest with 83.86% accuracy. This is in line with the other classifiers, where *GaitTime, mean_StepLength* and *Height* often scores high as well. One variable combination that scored relatively low compared to the other classifiers is *GaitTime, GaitVelocity, MovementVelocity, mean_StepLength* and *std_SegmentData*. It scored a 79,88% accuracy, while this feature combination scored generally high for the other three classifiers.

A visualization of the decision tree can be seen in figure 4.6. An orange color represents the "control" class while blue represents "glasses". The intensity of the colors indicates how certain the algorithm is about it's predictions. This plot shows that classifying is fairly straightforward. Orange and blue are mostly grouped together and confidence is overall high. This decision tree can also be found in appendix B.



(a) max_depth

(b) min_samples_split



(c) min_samples_leaf

Figure 4.5: Plots of accuracy for range of different parameter values

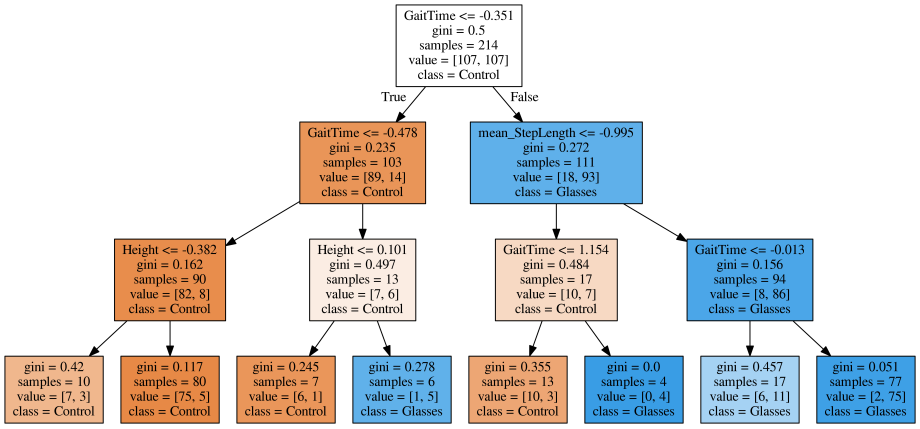


Figure 4.6: Decision tree for classification of glasses, using *GaitTime*, *Mean_StepLength* and *Height*

Included Variables	Accuracy
mad_SegmentData, mean_StepLength	82.43 %
mean_StepLength, std_SegmentData	82.65 %
GaitTime, mean_StepLength, mad_SegmentData	83.36 %
GaitTime, mean_StepLength, Height	83.86 %
GaitTime, mean_StepLength, Age	81.62 %
mean_StepLength, Height	81.37 %
mean_StepLength, Age	83.32 %
GaitTime, Height	78.65 %
std_SegmentData, Height	65.89 %
mad_SegmentData, Age	67.41 %
std_SegmentData, GaitVelocity, MovementVelocity, mean_StepLength	82.03 %
GaitTime, GaitVelocity, MovementVelocity, mean_StepLength, std_SegmentData	79.88 %

Table 4.4: Decision tree accuracy scores

2.5 Other classifiers

Random Forest

The results of the random forest classifier are similar to those of the decision tree, but generally lie about 3%-4% higher. *GaitTime, mean_StepLength* and *Height* is the best scoring feature combination, while *std_SegmentData* and *Height* performs the worst with 67.07%.

Included Variables	Accuracy
mad_SegmentData, mean_StepLength	84.12 %
mean_StepLength, std_SegmentData	84.05 %
GaitTime, mean_StepLength, mad_SegmentData	84.36 %
GaitTime, mean_StepLength, Height	86.1 %
GaitTime, mean_StepLength, Age	84.28 %
mean_StepLength, Height	83.47 %
mean_StepLength, Age	84.37 %
GaitTime, Height	80.44 %
std_SegmentData, Height	67.07 %
mad_SegmentData, Age	68.56 %
std_SegmentData, GaitVelocity, MovementVelocity, mean_StepLength	84.11 %
GaitTime, GaitVelocity, MovementVelocity, mean_StepLength, std_SegmentData	83.35 %

Table 4.5: Accuracy scores for random forest classifier

Ensemble classifier

The accuracy scores from the ensemble classifier are displayed in table 4.6. The scores from most combinations are between 84% and 86%. The combination *GaitTime*, *mean_StepLength* and *Height* is the only combination that scores 86% accuracy.

[h]

Included variables	**Accuracy**
mad_SegmentData and mean_StepLength	85.81%
mean_StepLength and std_SegmentData	85.81%
GaitTime, mean_StepLength and mad_SegmentData	85.76 %
std_SegmentData, GaitTime and mean_StepLength	86.11%
GaitTime, mean_StepLength and Height	86.66%
GaitTime, mean_StepLength and Age	85.72 %
mean_StepLength and Height	85.84 %
mean_StepLength and Age	85.1 %
GaitTime and Height	83.0 %
std_SegmentData and Height	71.09 %
mad_SegmentData and Age	72.02 %

Table 4.6: Ensemble classifier accuracy scores

Neural Network

Table 4.7 shows the accuracy scores of the neural network classifier. The average result of this classifier was around 85%, which is not higher than the main four classifiers. In general the results are very similar to the ensemble classifier.

Included variables	**Accuracy**
mad_SegmentData and mean_StepLength	85.38%
mean_StepLength and std_SegmentData	84.88%
GaitTime, mean_StepLength and mad_SegmentData	85.18 %
std_SegmentData, GaitTime and mean_StepLength	85.35%
GaitTime, mean_StepLength and Height	86.6%
GaitTime, mean_StepLength and Age	84.43 %
mean_StepLength and Height	85.16 %
mean_StepLength and Age	84.05 %
GaitTime and Height	83.87 %
std_SegmentData and Height	70.98 %
mad_SegmentData and Age	70.63 %

Table 4.7: Neural network accuracy scores

3 Alcohol classification

As mentioned, the continuous alcohol score was divided into three classes. However, to improve accuracy there was also experimented with two and four classes. Changing the number of classes did not result in a better performance, therefore the result section includes only the results from the three classes classification. Moreover, while experimenting to reach a better accuracy, the classifiers were also tested on the data set including the glasses condition. Using the data set with the glasses condition resulted in an accuracy decrease of 10% compared to using the data without the glasses condition. Therefore, only the data without the glasses condition is included in the results section.

3.1 Logistic regression

Extra dummy data has been added to the data set, to test if the accuracy could be increased. Figure 4.7 shows the confusion matrix of logistic regression composed of the variables *GaitTime*, *mean_StepLength* and *Height*. This matrix is plotted from the validation set and it shows that data has been added, since there are more than four high class alcohol data points. As discussed in the data preproccesing section, there are only four high class alcohol data points in the original data set. Figure 4.7 shows that many data points are misclassified, for example, 14 high class data points were predicted as being low class. This confusion matrix indicates that the accuracy scores for the oversampled data will not be very high. Which is confirmed in table 4.8, which displays the accuracy scores for the logistic regression classifier.

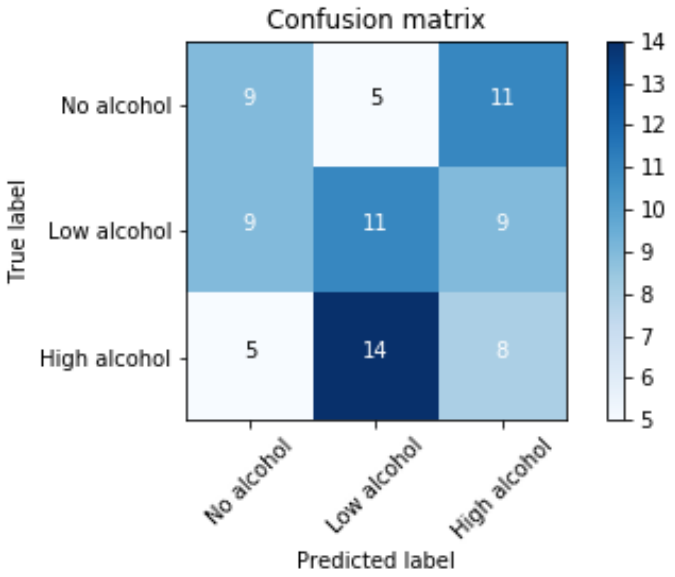


Figure 4.7: Logistic regression confusion matrix alcohol classification

Included variables	**Accuracy**
mad_SegmentData and mean_StepLength	50.62%
mean_StepLength and std_SegmentData	49.38%
GaitTime, mean_StepLength and mad_SegmentData	45.68 %
GaitTime, mean_StepLength and Height	34.57%
GaitTime, mean_StepLength and Age	45.68 %
mean_StepLength and Height	44.44 %
mean_StepLength and Age	38.27 %
GaitTime and Height	51.85 %
std_SegmentData and Height	57.85 %
mad_SegmentData and Age	54.32 %
std_SegmentData, GaitTime, mean_StepLength	45.68 %
std_SegmentData, GaitVelocity, MovementVelocity, mean_StepLength	50.62 %
GaitTime, GaitVelocity, MovementVelocity, mean_StepLength, std_SegmentData	50.62 %

Table 4.8: Logistic regression oversampled data accuracy scores for the alcohol classification

Table 4.8 shows that the accuracy scores are very low compared to the glasses problem, with the best performance being a 58% score. Interesting is that in comparison to the glasses problem, including *Age* and *Height* without *GaitTime* and *mean_StepLength* does not result in a significant performance drop. Furthermore, the scores differ significantly for the different variable combinations, which is interesting since the scores for the glasses classification were relatively stable across different combinations.

Not including the extra data, but stratifying the data leads to the F1-scores in table 4.9. Notable is that many variable combinations lead to a similar F1-score of 0.25, which is very low.

Included variables	**F1-score**
mad_SegmentData and mean_StepLength	0.25
mean_StepLength and std_SegmentData	0.25
GaitTime, mean_StepLength and mad_SegmentData	0.25
GaitTime, mean_StepLength and Height	0.32
GaitTime, mean_StepLength and Age	0.25
mean_StepLength and Height	0.25
mean_StepLength and Age	0.25
GaitTime and Height	0.32
std_SegmentData and Height	0.29
mad_SegmentData and Age	0.25
std_SegmentData, GaitTime, mean_StepLength	0.25
std_SegmentData, GaitVelocity, MovementVelocity, mean_StepLength	0.25
GaitTime, GaitVelocity, MovementVelocity, mean_StepLength, std_SegmentData	0.25

Table 4.9: Logistic regression F1-scores for the alcohol classification

The low performance for the alcohol classification becomes understandable when the data points from the stratified data are plotted (figure 4.8). The variables included in this plot are *GaitTime*, *Height* and *mean_StepLength*. To indicate how difficult it is for the classifier to classify alcohol levels correctly, the points that are scattered are the training points instead of the validation points. The decision boundary in the plot is not very accurate and many points get misclassified. All data points are scattered through the plot and the three classes overlap. Also, there is no clear boundary between the classes visible that a logistic regression classifier could predict.

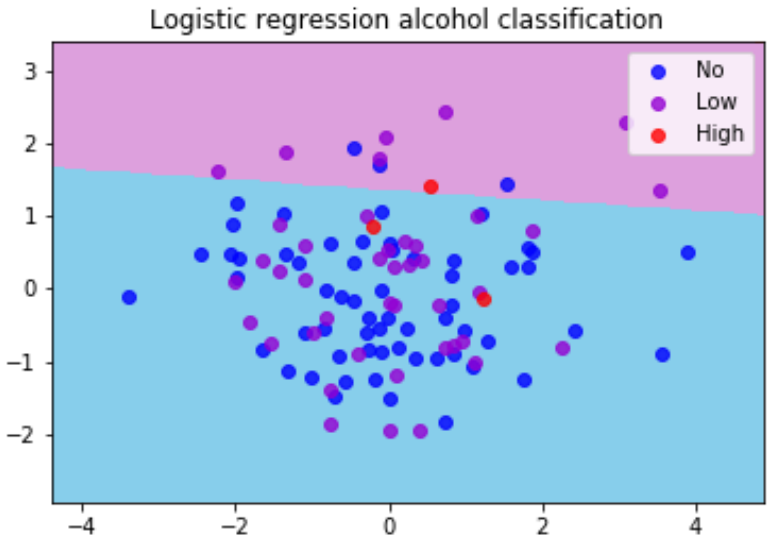


Figure 4.8: Logistic regression alcohol classification plot

3.2 Naive Bayes

The accuracy scores for the naive Bayes oversampled data are shown in table 4.10. These scores are similar to the previous classifier and range between 34% and 64%. The accuracy drops significantly for the combination *GaitTime*, *mean_StepLength* and *Height*, whereas it reaches 64.2% for the combination *GaitTime*, *mean_StepLength* and *std_SegmentData*.

Included variables	Accuracy
mad_SegmentData and mean_StepLength	51.85 %
mean_StepLength and std_SegmentData	50.62%
GaitTime, mean_StepLength and mad_SegmentData	53.09 %
GaitTime, mean_StepLength and Height	33.33%
GaitTime, mean_StepLength and Age	40.74 %
mean_StepLength and Height	48.15 %
mean_StepLength and Age	44.44 %
GaitTime and Height	48.15 %
std_SegmentData and Height	56.13 %
mad_SegmentData and Age	62.96%
std_SegmentData, GaitTime, mean_StepLength	64.2%
std_SegmentData, GaitVelocity, MovementVelocity, mean_StepLength	58.02 %
GaitTime, GaitVelocity, MovementVelocity, mean_StepLength, std_SegmentData	54.32 %

Table 4.10: Naive Bayes oversampled data accuracy scores for the alcohol classification

Table 4.11 shows the F1-scores for the stratified data. The scores differ more across the various variable combinations than for the logistic regression classifier, but it's performance is similar.

Included variables	Accuracy
mad_SegmentData and mean_StepLength	0.33
mean_StepLength and std_SegmentData	0.33
GaitTime, mean_StepLength and mad_SegmentData	0.25
GaitTime, mean_StepLength and Height	0.35
GaitTime, mean_StepLength and Age	0.24
mean_StepLength and Height	0.34
mean_StepLength and Age	0.25
GaitTime and Height	0.34
std_SegmentData and Height	0.38
mad_SegmentData and Age	0.30
std_SegmentData, GaitTime, mean_StepLength	0.25
std_SegmentData, GaitVelocity, MovementVelocity, mean_StepLength	0.30
GaitTime, GaitVelocity, MovementVelocity, mean_StepLength, std_SegmentData	0.29

Table 4.11: Naive Bayes F1-scores for the alcohol classification

Figure 4.9 plots the variables *mean_StepLength*, *GaitTime* and *mad_SegmentData* from the stratified data. Again, it is understandable why both the accuracy and the F1-scores are so low. All the data points overlap and no clear decision boundary is visible.

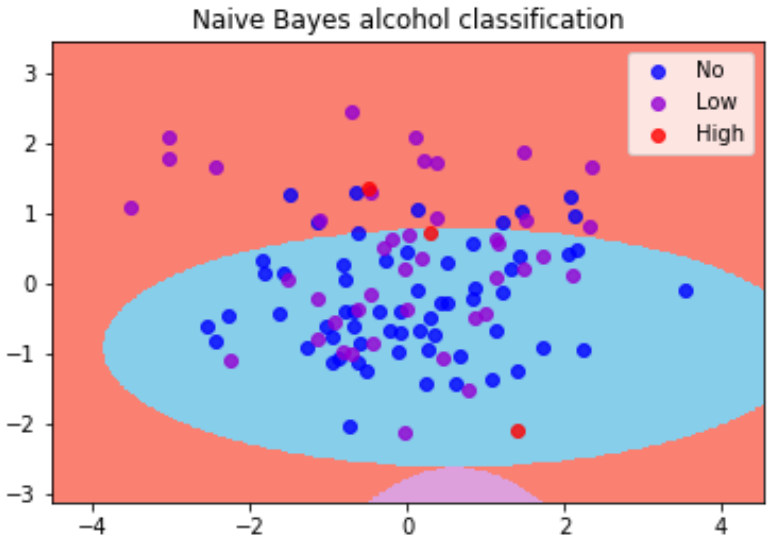


Figure 4.9: Naive Bayes alcohol classification plot

3.3 K-NN

The accuracy scores for the K-NN classifier including the extra data are displayed in table 4.12. In contrast with the previous two classifiers, the K-NN accuracy scores are stable and range between 72% and 83%. The scores are also higher and no significant drops occur for specific variable combinations.

Included variables	**Accuracy**
mad_SegmentData and mean_StepLength	75.31%
mean_StepLength and std_SegmentData	71.6%
GaitTime, mean_StepLength and mad_SegmentData	74.07 %
GaitTime, mean_StepLength and Height	72.84%
GaitTime, mean_StepLength and Age	76.54%
mean_StepLength and Height	76.54%
mean_StepLength and Age	74.07%
GaitTime and Height	74.07 %
std_SegmentData and Height	71.6%
mad_SegmentData and Age	72.06%
std_SegmentData, GaitTime, mean_StepLength	74.07%
std_SegmentData, GaitVelocity, MovementVelocity, mean_StepLength	82.72%
GaitTime, GaitVelocity, MovementVelocity, mean_StepLength, std_SegmentData	77.78 %

Table 4.12: K-NN oversampled data accuracy scores for the alcohol classification

Some of the F1-scores of the K-NN classifier are also higher compared to the previous two (table 4.13), with the best performance being 0.44 and an overall performance of around 0.37.

Included variables	**Accuracy**
mad_SegmentData and mean_StepLength	0.39
mean_StepLength and std_SegmentData	0.44
GaitTime, mean_StepLength and mad_SegmentData	0.39
GaitTime, mean_StepLength and Height	0.41
GaitTime, mean_StepLength and Age	0.28
mean_StepLength and Height	0.36
mean_StepLength and Age	0.29
GaitTime and Height	0.35
std_SegmentData and Height	0.41
mad_SegmentData and Age	0.42
std_SegmentData, GaitTime, mean_StepLength	0.41
std_SegmentData, GaitVelocity, MovementVelocity, mean_StepLength	0.36
GaitTime, GaitVelocity, MovementVelocity, mean_StepLength, std_SegmentData	0.35

Table 4.13: K-NN F1-scores for the alcohol classification

For comparison, figures 4.10 and 4.11 display the confusion matrix of the variable combination *GaitTime*, *mean_StepLength* and *Height* for the stratified data and the oversampled data. Clear from the matrices is that the classifier is better at classifying data points when more data is included. When there is more than one "High" alcohol class data point, K-NN is capable of classifying all "High" alcohol class data points correctly. The "No" and "Low" classes are slightly more difficult, however it is a great improvement compared to the stratified confusion matrix, where almost all data points are classified as no alcohol.

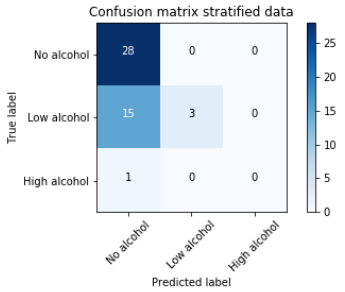


Figure 4.10: K-NN confusion matrix of the stratified data

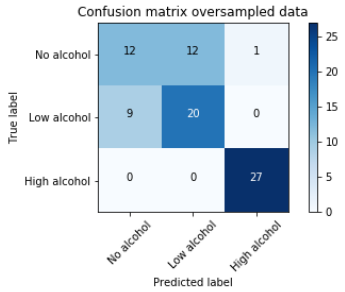


Figure 4.11: K-NN confusion matrix of the oversampled data

Lastly, since the K-NN oversampled data accuracy scores are significantly higher than the scores from the other three classifiers, it was decided to also test the precision and recall score for the oversampled data. These two scores were calculated using Scikit's library. Both the precision score and the recall score are around 0.75. These scores match with the calculated accuracy scores.

3.4 Decision tree

The dummy data accuracy scores for the decision tree, as seen in table 4.14, are quite stable and all similar to each other. They range around 68%-70%, with the only outlier being *GaitTime, mean_StepLength* and *Height*, which scores 64,63%.

Included Variables	**Accuracy**
mad_SegmentData, mean_StepLength	69.14 %
mean_StepLength, std_SegmentData	71.6 %
GaitTime, mean_StepLength, mad_SegmentData	69.8 %
GaitTime, mean_StepLength, Height	64.63 %
GaitTime, mean_StepLength, Age	70.37 %
mean_StepLength, Height	66.02 %
mean_StepLength, Age	66.67 %
GaitTime, Height	71.6 %
std_SegmentData, Height	67.9 %
mad_SegmentData, Age	68.41 %
std_SegmentData, GaitTime, mean_StepLength	69.65 %
std_SegmentData, GaitVelocity, MovementVelocity, mean_StepLength	70.37 %
GaitTime, GaitVelocity, MovementVelocity, mean_StepLength, std_SegmentData	71.6 %

Table 4.14: Accuracy scores of alcohol classification with decision tree, including extra data

This is also the case for the F1-scores, for the stratified data. These scores all range around 0.3, while the last variable combination scores the highest with 0.36.

In figure 4.12, an example of a visualized decision tree for the stratified data can be seen. Notable is that height is used as an important feature for classifying and a greater height is more likely to be classified as having light alcohol levels. Furthermore, the high alcohol level has not been classified at all and is missing from the tree.

Included Variables	F1-score
mad_SegmentData, mean_StepLength	0.34
mean_StepLength, std_SegmentData	0.33
GaitTime, mean_StepLength, mad_SegmentData	0.35
GaitTime, mean_StepLength, Height	0.33
GaitTime, mean_StepLength, Age	0.32
mean_StepLength, Height	0.32
mean_StepLength, Age	0.31
GaitTime, Height	0.33
std_SegmentData, Height	0.32
mad_SegmentData, Age	0.29
std_SegmentData, GaitTime, mean_StepLength	0.33
std_SegmentData, GaitVelocity, MovementVelocity, mean_StepLength	0.33
GaitTime, GaitVelocity, MovementVelocity, mean_StepLength, std_SegmentData	0.36

Table 4.15: Decision tree alcohol classification F1-scores

In figure 4.12, it can be seen that the three alcohol classes are not predicted very accurately. The "No" and "Low Alcohol" classes are not clearly divided and the "High" alcohol class is not predicted at all.

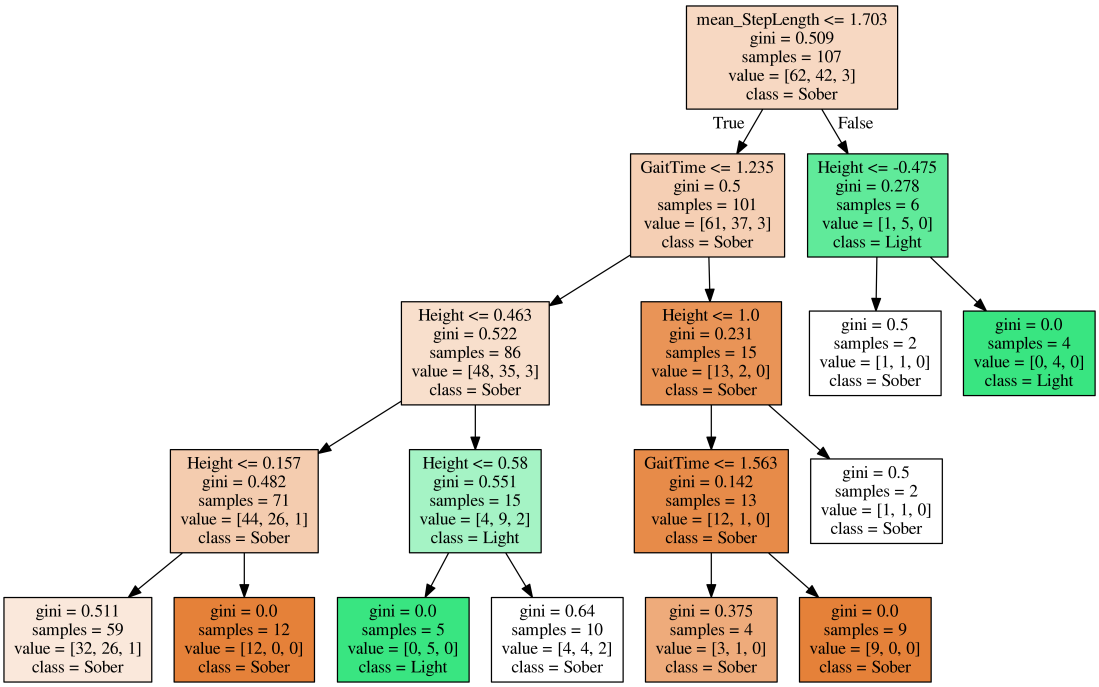


Figure 4.12: Decision Tree for Alcohol Classification

4 Balance classification

4.1 Logistic regression

Table 4.16 displays the accuracy scores for the oversampled data. These scores are significantly lower compared to the glasses classification, but similar to the alcohol classification and range from 49% to 55%.

Included variables	**Accuracy**
mad_SegmentData and mean_StepLength	56.76%
mean_StepLength and std_SegmentData	55.41%
GaitTime, mean_StepLength and mad_SegmentData	56.76 %
GaitTime, mean_StepLength and Height	54.05%
GaitTime, mean_StepLength and Age	50.0%
mean_StepLength and Height	54.05 %
mean_StepLength and Age	48.65 %
GaitTime and Height	50.0 %
std_SegmentData and Height	50.0 %
mad_SegmentData and Age	48.65 %
std_SegmentData, GaitTime, mean_StepLength	55.41 %
std_SegmentData, GaitVelocity, MovementVelocity, mean_StepLength	51.35 %
GaitTime, GaitVelocity, MovementVelocity, mean_StepLength, std_SegmentData	52.7%

Table 4.16: Logistic regression oversampled data accuracy scores for the balance classification

The F1-scores from the stratified data in table 4.17 are interesting, since the F1-score for all the different variable combinations is 0.44, which is higher than the classifier's F1-scores for the alcohol classification.

Included variables	Accuracy
mad_SegmentData and mean_StepLength	0.44
mean_StepLength and std_SegmentData	0.44
GaitTime, mean_StepLength and mad_SegmentData	0.44
GaitTime, mean_StepLength and Height	0.44
GaitTime, mean_StepLength and Age	0.44
mean_StepLength and Height	0.44
mean_StepLength and Age	0.44
GaitTime and Height	0.44
std_SegmentData and Height	0.44
mad_SegmentData and Age	0.44
std_SegmentData, GaitTime, mean_StepLength	0.44
std_SegmentData, GaitVelocity, MovementVelocity, mean_StepLength	0.44
GaitTime, GaitVelocity, MovementVelocity, mean_StepLength, std_SegmentData	0.44

Table 4.17: Logistic regression F1-scores for the balance classification

The low performance for the balance classification becomes understandable when the data points from the stratified data are plotted (figure 4.13). The variables included in this plot are *GaitTime*, *Height* and *mean_StepLength*. Again, the points that are scattered are the training points instead of the validation points. All the data points are scattered through the plot and the two classes overlap. Also, there is no clear boundary between the classes visible that a logistic regression classifier could predict.

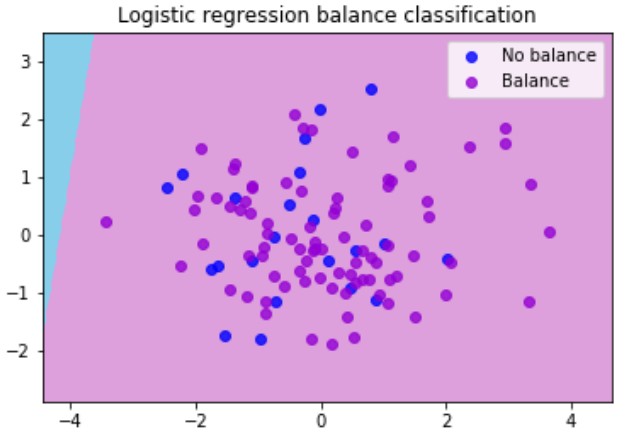


Figure 4.13: Logistic regression balance classification plot

4.2 Naive Bayes

Table 4.18 displays the naive Bayes accuracy scores for the oversampled data. Bayes's performance is slightly better than the previous classifier, however it is still very low and does not reach an accuracy higher than 60.81%.

Included variables	**Accuracy**
mad_SegmentData and mean_StepLength	56.76%
mean_StepLength and std_SegmentData	54.05%
GaitTime, mean_StepLength and mad_SegmentData	60.81 %
GaitTime, mean_StepLength and Height	51.35%
GaitTime, mean_StepLength and Age	58.11%
mean_StepLength and Height	52.7 %
mean_StepLength and Age	48.65 %
GaitTime and Height	60.81 %
std_SegmentData and Height	41.89 %
mad_SegmentData and Age	62.16%
std_SegmentData, GaitTime, mean_StepLength	54.05%
std_SegmentData, GaitVelocity, MovementVelocity, mean_StepLength	51.35 %
GaitTime, GaitVelocity, MovementVelocity, mean_StepLength, std_SegmentData	56.76%

Table 4.18: Naive Bayes oversampled data accuracy scores for the balance classification

The F1-scores in table 4.19 are similar to the logistic regression classifier, meaning that the scores are 0.44 for almost every variable combination.

Included variables	**Accuracy**
mad_SegmentData and mean_StepLength	0.44
mean_StepLength and std_SegmentData	0.44
GaitTime, mean_StepLength and mad_SegmentData	0.44
GaitTime, mean_StepLength and Height	0.44
GaitTime, mean_StepLength and Age	0.44
mean_StepLength and Height	0.54
mean_StepLength and Age	0.44
GaitTime and Height	0.44
std_SegmentData and Height	0.44
mad_SegmentData and Age	0.44
std_SegmentData, GaitTime, mean_StepLength	0.44
std_SegmentData, GaitVelocity, MovementVelocity, mean_StepLength	0.44
GaitTime, GaitVelocity, MovementVelocity, mean_StepLength, std_SegmentData	0.44

Table 4.19: Naive Bayes F1-scores for the balance classification

4.3 K-NN

The scores in table 4.20 are very high compared to the previous two classifiers. The K-NN classifier performs very well with the oversampled data and reaches accuracy scores ranging from 79% till 88%.

Included variables	**Accuracy**
mad_SegmentData and mean_StepLength	87.84%
mean_StepLength and std_SegmentData	86.49%
GaitTime, mean_StepLength and mad_SegmentData	80.17 %
GaitTime, mean_StepLength and Height	81.08%
GaitTime, mean_StepLength and Age	89.19%
mean_StepLength and Height	85.14 %
mean_StepLength and Age	82.43 %
GaitTime and Height	79.73 %
std_SegmentData and Height	83.78 %
mad_SegmentData and Age	87.84%
std_SegmentData, GaitTime, mean_StepLength	87.84%
std_SegmentData, GaitVelocity, MovementVelocity, mean_StepLength	82.43 %
GaitTime, GaitVelocity, MovementVelocity, mean_StepLength, std_SegmentData	83.78%

Table 4.20: K-NN oversampled data accuracy scores for the balance classification

From table 4.21 it becomes clear that overall the K-NN classifier performs better on the balance problem than the naive Bayes and logistic regression classifiers. The F1-scores are significantly higher and even reach a score of 0.88 for the variable combination *mean_StepLength* and *std_SegmentData*.

Included variables	Accuracy
mad_SegmentData and mean_StepLength	0.58
mean_StepLength and std_SegmentData	0.88
GaitTime, mean_StepLength and mad_SegmentData	0.54
GaitTime, mean_StepLength and Height	0.72
GaitTime, mean_StepLength and Age	0.62
mean_StepLength and Height	0.7
mean_StepLength and Age	0.54
GaitTime and Height	0.54
std_SegmentData and Height	0.6
mad_SegmentData and Age	0.52
std_SegmentData, GaitTime, mean_StepLength	0.88
std_SegmentData, GaitVelocity, MovementVelocity, mean_StepLength	0.54
GaitTime, GaitVelocity, MovementVelocity, mean_StepLength, std_SegmentData	0.52

Table 4.21: K-NN F1-scores for the balance classification

For comparison, figures 4.14 and 4.15 display the confusion matrix of the variable combination *GaitTime*, *mean_StepLength* and *Height* for the stratified data and the oversampled data. Clear from the matrices is that the K-NN is better at classifying data points when more data is included. For example, K-NN performs better on the oversampled data when predicting "No balance" classes, whereas almost all data points are classified as "Balance" with the stratified data.

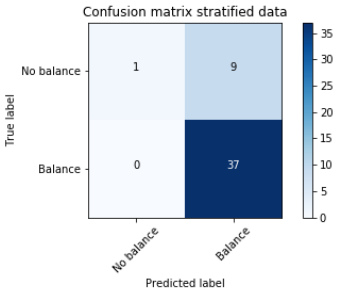


Figure 4.14: K-NN confusion matrix stratified data for the balance classification

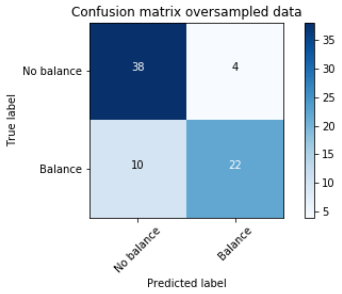


Figure 4.15: K-NN confusion matrix oversampled data for the balance classification

Again, since the K-NN oversampled data accuracy scores are significantly higher than the scores from the other three classifiers, it was decided to also test the precision, recall, roc_auc and average precision score for the K-NN extra data classifier. These four scores were calculated using Scikit's library. The precision score is around 0.87, both the recall and the average precision score are around 0.77 and the roc_auc score is around 0.86. These scores match with the calculated accuracy scores.

4.4 Decision tree

The accuracy scores for the decision tree with extra data are a little more spread out than other results, with the lowest scores being around 60% and the highest around 77%. Overall these scores are not as good as the K-NN accuracy, but higher than logistic regression and naive Bayes.

Included Variables	**Accuracy**
mad_SegmentData, mean_StepLength	72.97 %
mean_StepLength, std_SegmentData	67.57 %
Velocity, mean_StepLength, mad_SegmentData	74.32 %
std_SegmentData, GaitVelocity, MovementVelocity, mean_StepLength	64.86 %
Velocity, GaitVelocity, MovementVelocity, mean_StepLength, std_SegmentData	67.57 %
Velocity, mean_StepLength, Height	61.49 %
Velocity, mean_StepLength, Age	66.22 %
mean_StepLength, Height	70.27 %
mean_StepLength, Age	68.92 %
Velocity, Height	77.03 %
std_SegmentData, Height	71.62 %
mad_SegmentData, Age	60.81 %
std_SegmentData, Velocity, mean_StepLength	74.22 %

Table 4.22: Decision tree with dummy data balance accuracy scores

This is also true for the F1-scores. While they are not as high as the scores for K-NN, the decision tree with stratified data performs a little better than the other two classifiers. The best performance is delivered by *mad_SegmentData* and *mean_StepLength*, which reaches an F1-score of 0.55, while the worst performing feature combination is *std_SegmentData* and *Height*. This is different from the accuracy scores of table 4.22, where both of these features combinations perform quite well.

Included Variables	F1-score
mad_SegmentData, mean_StepLength	0.55
mean_StepLength, std_SegmentData	0.52
Velocity, mean_StepLength, mad_SegmentData	0.52
std_SegmentData, GaitVelocity, MovementVelocity, mean_StepLength	0.45
Velocity, GaitVelocity, MovementVelocity, mean_StepLength, std_SegmentData	0.44
Velocity, mean_StepLength, Height	0.48
Velocity, mean_StepLength, Age	0.49
mean_StepLength, Height	0.41
mean_StepLength, Age	0.51
Velocity, Height	0.42
std_SegmentData, Height	0.39
mad_SegmentData, Age	0.44
std_SegmentData, Velocity, mean_StepLength	0.5

Table 4.23: Decision tree balance classification F1-scores

Figure 4.16 shows the visualized balance classification tree, for the stratified data. It can be seen that very big step length causes the algorithm to immediately classify the balance as bad, with a pretty high confidence as well. Bad balance isn't classified often though and when it is, the prediction is not as certain.

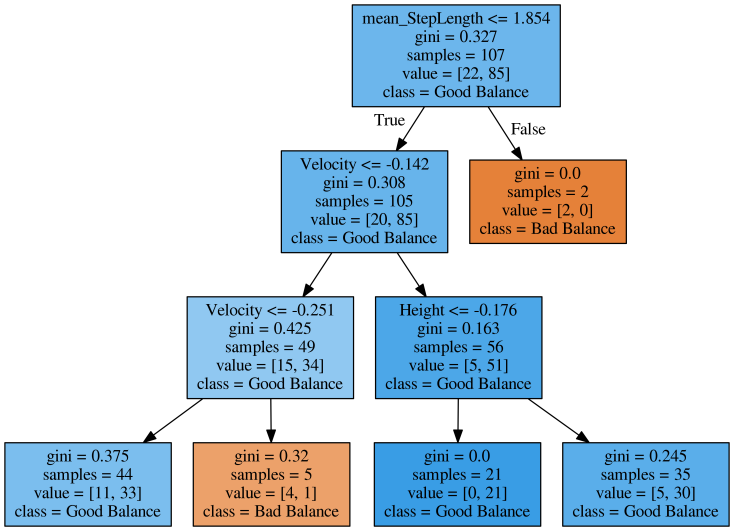


Figure 4.16: Decision Tree for Balance Classification

Chapter 5

Conclusion and Discussion

1 Glasses Problem

Multiple classifiers have been implemented and compared for this project. The results from all classifiers are similar and are all relatively high, ranging from 84% to 89% accuracy. However, the K-NN classifier performs slightly better than the other classifiers, reaching an accuracy of nearly 89%, thus making it the best performing classifier of this project. A possible reason for this high performance score is that K-NN looks at the classes of neighboring data points. Since the data points with different classes were located relatively close to one another in this data set, K-NN's method was a good fit for this project.

One interesting result is that the variables *Height* and *Age* were not indicated by seaborn's feature selection to result in a high accuracy. However, it turned out that including these two variables in different variable combinations could lead to a good performance. Including *Height* even led to the highest accuracy score for the logistic regression classifier. These results correspond with previous research, which indicates that an increase in height and age results in less balance control (Alonso et al., 2012) (Greve et al., 2013).

Moreover, the variables *mean_StepLength*, *mad_SegmentData*, *std_SegmentData* and *GaitTime* were created based on research suggesting that people with less balance take more steps, sway more and walk slower than people with normal balance control.(Hak et al., 2013), (Kawamura et al., 1991), (Hak et al., 2012). The results from this project are consistent with this research, since including these four variables lead to good performing classifiers that are capable of accurately identifying whether a participant was wearing glasses or not, thus whether they had less balance control. This also becomes clear when all four variables are completely disregarded in classifying, which causes the accuracy to drop significantly. Overall, the results indicate that people's walking speed, step length and sway are important factors in the classification of fall risk.

Conclusively, all the classifiers implemented for this research were capable of identifying whether or not a participant was wearing glasses. This suggests that the Xbox Kinect has the potential to be a sufficient screening tool for detecting fall risk in elderly people.

1.1 Alcohol Problem

For the classification of alcohol level, logistic regression, Naive Bayes, K-NN and a decision tree were implemented. These classifiers had a lower performance than the glasses classification, reaching a F1-score of around 0.6 with a few outliers towards 0.7. Research suggests that a breath alcohol concentration of more than 0.5 permillage already leads to less balance control (Trimbos Institute, 2017). Therefore it would be presumable that participants with an alcohol level higher than 0.5 permillage had less balance control and different segment data than sober participants. However, the results from the alcohol classification problem show no clear correlation between alcohol level and balance control.

One possible explanation might be that the Kinect sensors weren't sensitive enough to pick up on the gait change of participants with alcohol consumption, or maybe the mid spine point data alone wasn't sufficient. Adding more Kinect data points might improve performance. Another possible reason for these results is the skewedness of the alcohol classes, with a great majority of the participants being sober or having only very low alcohol concentration levels. Having insufficient data on high alcohol levels, might have negatively impacted the performance of the classifiers. This is a well substantiated reason, since adding extra data leads to relatively high accuracy scores for the K-NN classifier, with the best score being 82.72%. Adding extra data does not improve the logistic regression and naive Bayes performances and only slightly improves the decision tree performance.

Overall, the K-NN classifier with the oversampled data easily results in the best performance for the alcohol classification problem. The results from the three other classifiers never imply that a correlation between alcohol level and balance control exists, however, K-NN's results could prove that that correlation does exist, but that more data is necessary to discover such a correlation with machine learning.

1.2 Balance Problem

Concerning the balance problem, the same four classifiers (as for the alcohol problem) were implemented and tested. Their task was to predict the balance test score of a participant, given their walking segments. In this way, it could be seen whether or not there was a correlation between a participant's walking balance and their standing balance.

The results show that the classifiers were only partially capable of classifying the right balance class for every participant. The logistic regression and naive Bayes classifiers were not able to classify the right balance class for every participant. The decision tree classifier performed reasonably well with the oversampled data, reaching a accuracy score of 77.03%. However, K-NN is again the best performing classifier and reaches an accuracy of 89.19% on the oversampled data. This is a very good performance and suggests that, with enough data, K-NN should be able to find a correlation between a person's walking balance and standing balance. Even though research suggests that a clear correlation between static an dynamic balance can't be found (Sell, 2012) (Karimi & Solomonidis, 2011), the results from this project seem to indicate that a correlation indeed exists, provided that enough data is collected.

2 Limitations

One of the greatest limitations that was encountered was the unevenly distributed data. Skewed variables were seen in multiple features. For example, the age of participants was generally low. Most people were in their twenties or early thirties, with only a few cases of older adults. Their alcohol consumption was also mostly low. The majority of the participants were sober, a few had a light alcohol level and only a few people were drunk. This posed a problem mainly in classifying the alcohol level, since there was not enough data to work with. This was also true for the balance classification. Almost eighty percent of the participants had normal balance, so classifying them well proved to be difficult. This limitation was tried to overcome by stratifying and oversampling the data, but a more evenly distributed dataset would have been more efficient and reliable.

3 Future Research

Future studies might benefit from having a bigger, more balanced data set. Representing the different features equally, will lead to more grounded, reliable predictions and also eliminates the need of having to normalize or drop a majority of the data.

Furthermore, including more points from the Kinect sensor might lead to interesting results. For this project, only the mid spine was used for classification, but other body parts could be included as well. Data about feet position, for example, might be a good indicator of bad balance or fall risk, but arm movement or head movement could provide new information as well (Stam, 2018), (Kiss et al., 2018).

Another feature that might be interesting to include, is gender. In the original data set, gender was not included, but there are studies that suggest that gender is of influence on balance control. Some factors, like height, weight or sway, can more heavily influence balance control depending on someone's gender (Hageman et al., 1995). It would be interesting if these findings could be confirmed by using them for classifying.

Finally, performance might be increased even more by writing original code for the classifiers. Since the goal of this project was to compare many classifiers, and time was limited, mostly builtin functions were used for coding the classifiers. If the classifiers were to be used in a future project, it would be advisable to write custom code, so that they may be more tailored towards the project.

Appendix

A Data sets

A.1 Original Data set

Row	Description
TestID	Unique ID of Test; For example "169B03" is subject ID 169; condition B ; third walk
Conditie	With or without distorted depth perception
SubjectID	Unique ID of Participant
TrialNr	N-th time the participant performed the test
NumberOfSegments	Total number of gait segments in this test
SegmentNr	N-th gait segment of the test
WalkingDirection	Whether subject is walking towards or away from the sensor
SegmentData	Contains the actual xyz data of the spine-mid point
Alcohol	Breath alcohol concentration
Height	Subject height
Weight	Subject weight
Age	Subject age
BMI	Subject Body Mass Index
Balance_MLrange	
Balance_MLstdev	
Balance_MLmeanVelocity	
Balance_APrange	
Balance_APstdev	
Balance_APmeanVelocity	
Balance_MeanVelocity	Preferred balance metric
Gait_Velocity	Median gait velocity of head during gait segment in Z-direction
Movement_velocity	Median velocity of head during gait segment in the horizontal plane

A.2 Adjusted Data set

Row	Description
SubjectID	Unique ID of Participant
Height	Subject height
Weight	Subject weight
Age	Subject age
BMI	Subject Body Mass Index
Alcohol	Breath alcohol concentration
Alcohol_Class	Class corresponding to ABC: no/sober/0, low/tipsy/1 or high/drunk/2
Balance_MeanVelocity	Preferred balance metric
GaitVelocity	Mean of the median gait velocity of head during the four gait segments in z-direction
MovementVelocity	Mean of the median velocity of head during the four gait segments in the horizontal plane
GaitTime	Mean of the walking time for the four segments
std_SegmentData	Mean of the standard horizontal deviation of the four segments
mad_SegmentData	Mean of the mean average horizontal deviation of the four segments
mean_StepLength	Mean of the mean of the step length of the four segments

B Decision Tree

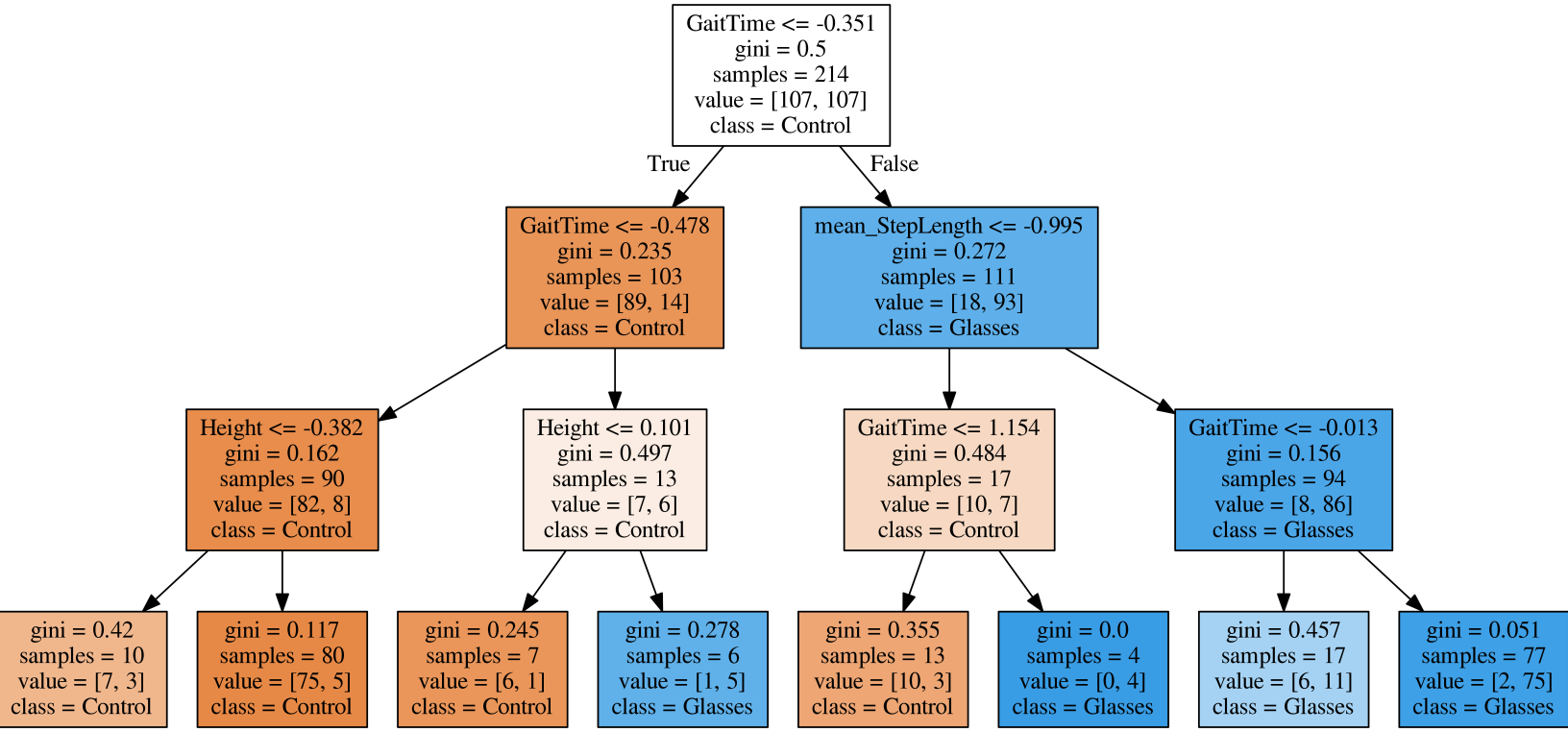


Figure 1: Decision Tree for Balance Classification

References

Alonso, A. C., Luna, N. M. S., Mochizuki, L., Barbieri, F., Santos, S., & Greve, J. M. D. (2012). The influence of anthropometric factors on postural balance: the relationship between body composition and posturographic measurements in young adults. *CLINICS*, *67*. doi: 10.6061/clinics/2012(12)14

Fraj, M. B. (2017). *In depth: Parameter tuning for random forest.* `https://medium.com/all-things-ai/in-depth-parameter-tuning-for-random-forest-d67bb7e920d`. (Accessed: 21-01-2019)

Greve, J. M. D., Cuğ, M., Dülgeroğlu, D., Brech, G. C., & Alonso, A. C. (2013). Relationship between anthropometric factors, gender, and balance under unstable conditions in young adults. *BioMed Research International*, *2013*. doi: https://doi.org/10.1155/2013/850424

Hageman, P. A., Leibowitz, J. M., & Blanke, D. (1995). Age and gender effects on postural control measures. *Arch Phys Med Rehabll*, *76*. Retrieved from `https://www.archives-pmr.org/article/S0003-9993(95)80075-1/pdf`

Hak, L., Beek, H. H. P., & van Dieën, J. H. (2013, 12). Steps to take to enhance gait stability: The effect of stride frequency, stride length, and walking speed on local dynamic stability and margins of stability. *PloS one*, *8*, e82842. doi: 10.1371/journal.pone.0082842

Hak, L., Houdijk, H., Steenbrink, F., Mert, A., van der Wurff, P., Beek, P. J., & van Dieën, J. H. (2012). Speeding up or slowing down?: Gait adaptations to preserve gait stability in response to balance perturbations. *Gait Posture*, *36*(2), 260 - 264. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0966636212000811` doi: https://doi.org/10.1016/j.gaitpost.2012.03.005

Ikai, T., Tatsuno, H., & Miyano, S. (2006). Relationship between walking ability and balance function. *The Japanese Journal of Rehabilitation Medicine*, *43*(12), 828-833. doi: 10.2490/jjrm1963.43.828

Imbalanced Learn. (n.d.). *Over-sampling.* `https://imbalanced-learn.readthedocs.io/en/stable/over_sampling.html`. (Accessed: 01-02-2019)

Karimi, M. T., & Solomonidis, S. (2011). The relationship between parameters of static and dynamic stability tests. *Journal of research in medical sciences: the official journal of Isfahan University of Medical Sciences*, *16*(4), 530.

Kawamura, K., Tokuhiro, A., & Takechi, H. (1991). Gait analysis of slope walking: a study on step length, stride width, time factors and deviation in the center of pressure. *Acta Medica Okayama*, *45*(3), 179–184.

Kiss, R., Schedler, S., & Muehlbauer, T. (2018). Associations between types of balance performance in healthy individuals across the lifespan: a systematic review and meta-analysis. *Frontiers in physiology*, *9*.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., ... Willing, C. (2016). *Jupyter notebooks – a publishing format for reproducible computational workflows* (F. Loizides & B. Schmidt, Eds.).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pham-Gia, T., & Hung, T. (2001). The mean and median absolute deviations. *Mathematical and Computer Modelling*, *34*(7), 921 - 936. Retrieved from http://www.sciencedirect.com/science/article/pii/S0895717701001091 doi: https://doi.org/10.1016/S0895-7177(01)00109-1

Robinson, S. (2018). *K-nearest neighbors algorithm in python and scikit-learn.* https://stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn/. (Accessed: 22-01-2018)

Scikit Learn. (n.d.). *Importance of feature scaling*. https://scikit-learn.org/stable/auto_examples/preprocessing/plot_scaling_importance.html. (Accessed: 21-01-2019)

Seaborn. (n.d.). *Plotting a correlation matrix.* https://seaborn.pydata.org/examples/many_pairwise_correlations.html. (Accessed: 22-01-2019) doi: 10.5281/zenodo.12710

Sell, T. C. (2012). An examination, correlation, and comparison of static and dynamic measures of postural stability in healthy, physically active adults. *Physical Therapy in Sport*, *13*(2), 80–86.

Shimada, H., Shuichi, O., Kamide, N., Shiba, Y., Okamoto, M., & Kakurai, S. (2003, 08). Relationship with dynamic balance function during standing and walking. *American journal of physical medicine rehabilitation / Association of Academic Physiatrists*, *82*, 511-6. doi: 10.1097/01.PHM.0000064726.59036.CB

Stam, C. (2018, september). *Privé-valongevallen bij ouderen* (Ongevalscijfers 2017). VeiligheidNL.

Trimbos Institute. (2017). *Alcohol info wat zijn de effecten per glas?* Retrieved from https://www.alcoholinfo.nl/publiek/veelgesteldevragen/resultaten/antwoord/?vraag=29203 (Accessed: 02-02-2019)