Assignment 3 - Group 32[*]

Bas van der Bijl (sbl206), Max de Bruijne (mbe345), and Tom de Valk (tvk550)

Vrije Universiteit Amsterdam, 1081 HV Amsterdam, The Netherlands

1 Introduction

Using the sensors in smart devices, everyday activities can be quantified. This can potentially be done using a combination of wrist-worn smart devices and pocket-carried smart devices. This sensory data becomes even more valuable when it is labeled with an activity label. This activity label describes the activity that the user was doing when the measurements were made. When using the smart devices for long periods of time, it becomes possible to analyse daily activities and routines, potentially identify bad habits in the lifestyle of the user. This identification may lead to more awareness for the user and, eventually, a healthier lifestyle.

The aim of this paper is to predict the activity that the user is engaged in for each measurement. Before this can be done, the data is pre-processed, outliers and missing values are dealt with and extra features are added to the data. After that the instances are clustered and modelled using seven different models. The practical relevance of the prediction of the activity label is that this labelling is currently done by hand. Automating this will yield a much faster and easier analysis of daily activities and routines.

In Section 2, a description of the data is given, after which the pre-processing and feature engineering steps will be discussed in Section 3 and Section 4 respectively. In Section 5, the proposal for the different models is given, after which, the method is discussed in Section 6. Section 7 provides an elaborate description and discussion of the results and in Section 8, the conclusion and suggestions future work will be discussed.

2 Data Description

The data that is used in this study was gathered by G. M. Weiss [13]. The data contains accelerometer (x, y, z) and gyroscope (x, y, z) data, originating from a smartphone (Google Nexus 5/5X or Samsung Galaxy S5) and a smartwatch (LG G Watch). This data is labeled with an activity label that describes one of eighteen activities: walking, jogging, stairs, sitting, standing, typing, brushing teeth, eating soup, eating chips, eating pasta, drinking from cup, eating sandwich, kicking soccer ball, playing catch with a tennis ball, dribbling with a basketball, writing, clapping and folding clothes. The data for these activities was collected

[*] Part of the 2021 Machine Learning for the Quantified Self course at VU Amsterdam, The Netherlands

by 51 different users for three minutes per activity, with 20 measurements per second. In total, this led to 15,630,426 measurements, which could be grouped to 1,360,787 instances.

Figure 1 shows the distribution of the labels in the dataset. It can be seen that sitting occurs around twice as often as the other labels. Considering the large number of labels, this should not lead to a problematic class imbalance. Figure 2 shows the number of measurements per user. It can be seen that the number of measurements per user varies drastically. User 1629, for instance, has approximately 237,000 measurements while user 1640 has only 3600 measurements.

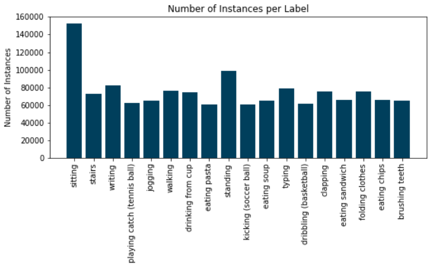


Fig. 1. Distribution of labels

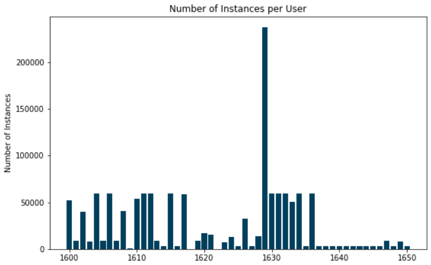


Fig. 2. Distribution of users

3 Data Pre-processing

To ensure that the data can be clustered and modelled, the measurements in the dataset need to be validated. This means that the data should not contain missing values or outliers. Any potential missing values or outliers should, therefore, be identified and dealt with accordingly.

Missing Values The original dataset contains slightly more accelerometer measurements than gyroscope measurements. This means that when the measurements are merged to the same dataset to create instances, there are multiple missing values for the gyroscope data. These missing values need to be filled to prepare the dataset for modelling. The missing values can be filled in a number of ways, using the mean, the median or linear interpolation, for instance. Since the accelerometer and gyroscope data are generally distributed around zero, filling the missing values with zero may also be a possibility.

Outliers For the detection of potential outliers, four different methods are considered: Chauvenet's criterion, mixture models, distance-based methods, the Kalman Filter and the Local Outlier Factor (LOF) method. According to Hoogendoorn and Funk, Chauvenet's criterion assumes a normal distribution of the data [3]. In Section 2, it was observed that this assumption is not always valid. Therefore, Chauvenet's criterion is not regarded to be a suitable outlier detection method for problem at hand. Since there was no clear segregation of the data points over different distributions, mixture models were also not regarded to be suitable. The Kalman Filter can only be used for a single feature per execution and is, therefore, solely useful for the detection of outlying values, rather than outlying instances. Lastly, the LOF algorithm is extremely computationally expensive and is therefore disregarded for outlier detection [4].

4 Feature Engineering

Next to the features that were described in Section 2, a number of other features can be derived. These features could be of use when modelling the data. The additional features can be defined as principal component analysis, frequency-based, time-based and clustering features.

Principal Component Analysis Principal component analysis (PCA) is an unsupervised algorithm with the goal to extract useful information from a dataset, in order to create new orthogonal variables (principal components) [1]. The number of principal components is determined using the amount of explained variance.

Time-Based Another way in which features can be generated, is by using the time domain. Using a time frame of length λ, a descriptive value, such as the mean, minimum, maximum or standard deviation, is calculated and used to describe each numerical feature [5].

Frequency-Based Next to time-based features, frequency-based features can also be used to summarize the data within a time frame of length λ. For this, a Fourier transformation[1] is used to observe the number of frequencies per time frame. This leads to features such as the highest amplitude frequency, the frequency weighted signal average and the power spectrum entropy [5].

Clustering Clustering is an unsupervised machine learning method which allows one to group instances with similar characteristics together. In the context of this study, the use of clustering may give more context to groups of features, which may lead to an increase in the predictive performance of models. One way in which this clustering can be done, is through k-means. K-means allows for the clustering of large amounts of data because of the iterative approach of the algorithm [7].

[1] A Fourier series is an expansion of a periodic function in terms of an infinite sum of sines and cosines. It is predominantly used to break up periodic functions to calculate them individually [14].

Finding the appropriate number of clusters is essential to maximize the value of the groups of observations. Determining this appropriate number of clusters can be done using several different metrics. One of these metrics is the silhouette coefficient, which is a measure of similarity between the instances within a cluster and between different clusters. The silhouette score ranges between -1 and $+1$, where a value of $+1$ indicates high intra-cluster similarity and low inter-cluster similarity [10].

5 Models

The aim of this paper is to label measurements with the corresponding activities. This can be translated to a classification problem, where one instance is classified to be one of the given eighteen activities. Since there are eighteen activity labels, this is a multi-class classification problem. According to M. Aly, the most suitable models for multi-class classification are random forest (RF), neural network (NN), k-nearest neighbors (kNN), decision tree (DT) and naive Bayes (NB) [2]. Hoogendoorn and Funk suggest that support vector machine (SVM) models may also perform well for multi-class classification. Moreover, they suggest that ensembling multiple models may also improve classification performance. Another model that may be useful for multi-class classification is XGBoost (XGB), which should, theoretically, outperform random forest and decision tree models [8].

To determine how good these models perform in relation to each other, several evaluation metrics can be used in combination with visual evaluation methods. One way in which the models can be evaluated visually is through the use of a confusion matrix. This is a matrix that shows the relation between the observed and predicted values for each class [11]. This confusion matrix can also be used to determine the different evaluation metrics. Evaluation metrics such as the precision, recall and F1-score are used to focus on false positives and negatives in particular. This is especially helpful for data with large class imbalances where incorrect predictions can be costly [6]. The accuracy shows the number of correctly-classified instances with respect to all of the instances in the data. Since the data contains no extreme class imbalances, the accuracy seems to be an adequate measure for the evaluation of model performance.

6 Methodology

Missing Values Considering that the activities are continuous and that time intervals between measurements are very small, linear imputation of the missing values seems to be the most suitable method for dealing with missing values. This was not possible for smartphone gyroscope data of the users 1625, 1627 and 1635 as these users did not collect any gyroscope data. The data that is associated with these users was then removed from the dataset. Linear imputation was also not an option when certain users had leading missing values. These missing values were backfilled.

Outlier Detection To reduce the required amount of computational resources, outlier detection has been carried out per user. This reduction turned out to be insufficient to carry out LOF outlier detection. As a result, only distance-based outlier detection was used to detect the outliers per user. The distance-based method did not identify any outliers for the entire dataset. The tuning of the parameters d_{min} and f_{min} to 0.001 and 0.6, respectively, had no effect on the identification of outliers for the data. A possible cause for this is that the sensors of the smart devices have maximum detection ranges. Figure 3, for instance, shows a three-dimensional plot of the smartwatch accelerometer data for dribbling a basketball. It can be seen that there are no values that exceed 18 or -18 for x, y and z.

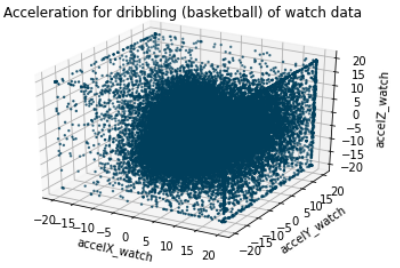


Fig. 3. 3D plot of the watch measurements for dribbling with a basketball.

Feature Engineering - PCA A plot of the explained variance per additional principal component of the PCA for user 1626, is shown in Figure 4. The explained variance for the other users was extremely similar to this. The explained variance decreases drastically until component four, after which the line decreases somewhat slower until the sixth component. Since, more than 95% of the variance is explained within six principal components, six principal components (six new features), are added to the dataset.

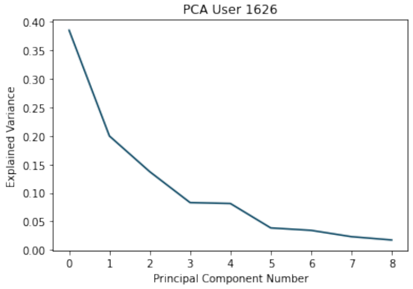


Fig. 4. The explained variance per principal component for User 1626

Feature Engineering - Time-Based For the acceleration and gyroscope features from the smartphone and smartwatch, three window sizes were used for time-based feature engineering: 30, 60 and 120 measurements. Considering the measurement frequency of 20 per second, this amounts to window sizes of 1.33, 2.66 and 5.32 seconds, respectively. Each of these windows was used to generate two new features per original attribute, describing the mean and standard deviation within the window.

Feature Engineering - Frequency-Based Next to the additional time-based features, frequency-based features were also added. These features were generated using a window size of 20 measurements, representing 1 second of data. In the selection of this window size, a trade-off was made between added value and computational cost.

Since the use of the window sizes of time-based and frequency-based features lead to a large number of largely similar instances, the overlap between the time frames is adjusted to 50%. 50% overlap of the windows was deemed appropriate for the study conducted by M. Shoaib et al. [9], so it seemed to be a good starting point for this study. Considering that multiple window sizes are used, the overlap is executed using a window size of 30 measurements.

Feature Engineering - Clustering K-means clustering is applied to the individual sensor features to add context about the measurements of these sensors. This results in separate clusterings for the accelerometer and gyroscope data for both the smartphone and the smartwatch. For each sensor, the most appropriate number of clusters is determined by maximizing the silhouette score. Figure 5 shows the silhouette score of clusters for the smartphone accelerometer data in the process of selecting the appropriate number of clusters. The plot shows that when applying k-means clustering with eight clusters, the highest silhouette score is achieved. The same procedure was executed for the remaining sensors. This resulted in eight, two and eight clusters for the smartwatch accelerometer, smartphone gyroscope and smartwatch gyroscope data, respectively.

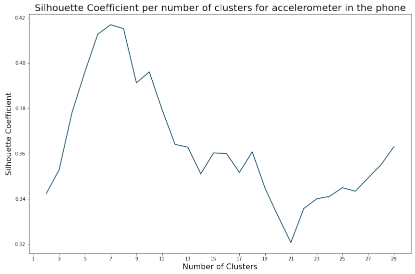


Fig. 5. The silhouette score per number of clusters for the accelerometer in the phone.

Modelling To reduce the required computational resources as well as the probability of overfitting, feature selection is implemented before the modelling of the data. For this, forward selection using decision trees is carried out. Forward selection is preferred over backward selection as the dataset contains a large number of features. In Figure 6, it can be seen that the model accuracy stabilizes after the addition of the first ten features. For that reason, only the ten most suitable features are used for modelling.

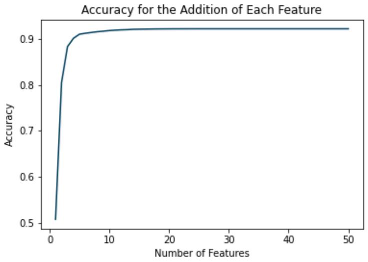


Fig. 6. Forward feature selection

Each of the models is trained for three different feature sets: (1) the original features, (2) the original features in addition to the engineered features and (3) the ten features that were selected with feature selection.

The models are trained using a training and test set, with 70% of the users being assigned to the training set. The following users are in the test set and are not used in the training of the model: 1600, 1602, 1603, 1607, 1611, 1621, 1630, 1631, 1632, 1636, 1643, 1646, 1650 and 1609. Using these training and test sets, the different models were all tuned using a grid-search approach.

To determine whether the different models are of value in the prediction of the activity, the model performance is compared to a random baseline, which has an accuracy of 5.56%.

7 Results

The performance of each model with the different feature sets are shown in Figure 7. It can be observed that the models all perform (far) better on the training set than on the test set. This suggests overfitting, which could not be reduced during the tuning of the models. Figure 7 also shows that all models perform best when the full set of features is used. This means that the engineered features add value to the models. It can also be seen that the feature set that originates from feature selection, generally leads to worse performance than the original dataset features.

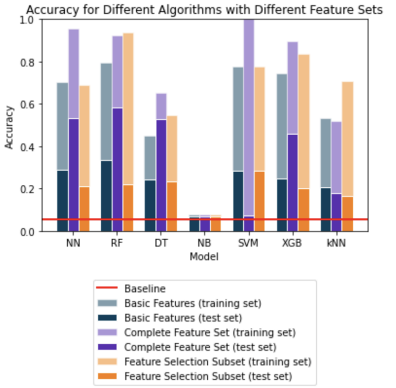
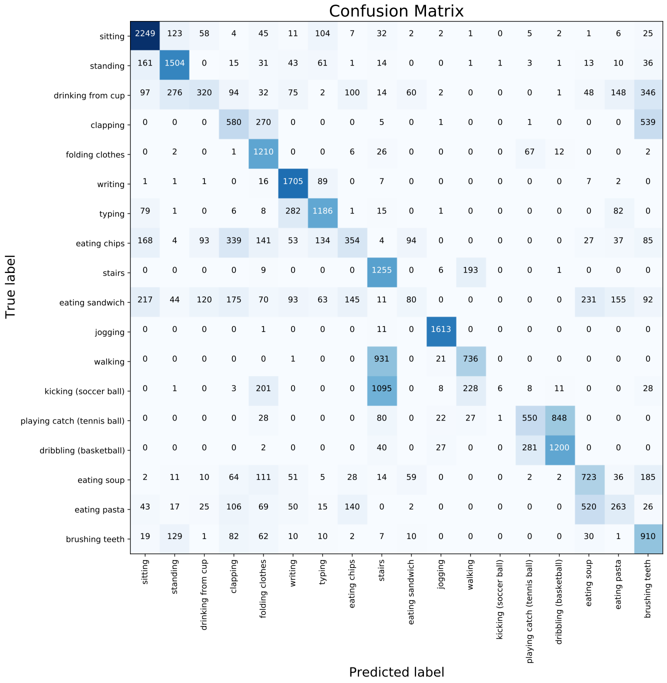


Fig. 7. Model performance with different feature sets in terms of accuracy

Figure 7 suggests that the random forest model performs best. This model uses all of the features and has an accuracy of 92.25% on the training set and 58.08% on the test set. While the model is still overfit on the training data, 58.08% accuracy on the test set is still a remarkable improvement with respect to the random baseline. The confusion matrix for this model is shown in Figure 8. It can be seen that the most incorrect predictions are made between similar labels such as *eating pasta*, *eating soup* or *eating sandwich*. As seen in the confusion matrix, the activity labels *walking* and *kicking (soccer ball)* are often mistaken for *stairs*.

8 Conclusion

Considering the results that were described in the previous section, it can be stated that a random forest model, among others, can be used to gain a notable edge over a random baseline in the prediction of user activity based on the accelerometer and gyrometer data. However, the results also show remarkable



	sitting	standing	drinking from cup	clapping	folding clothes	writing	typing	eating chips	stairs	eating sandwich	jogging	walking	kicking (soccer ball)	playing catch (tennis ball)	dribbling (basketball)	eating soup	eating pasta	brushing teeth
sitting	2249	123	58	4	45	11	104	7	32	2	2	1	0	5	2	1	6	25
standing	161	1504	0	15	31	43	61	1	14	0	0	1	1	3	1	13	10	36
drinking from cup	97	276	320	94	32	75	2	100	14	60	2	0	0	0	1	48	148	346
clapping	0	0	0	580	270	0	0	0	5	0	1	0	0	1	0	0	0	539
folding clothes	0	2	0	1	1210	0	0	6	26	0	0	0	0	67	12	0	0	2
writing	1	1	1	0	16	1705	89	0	7	0	0	0	0	0	0	7	2	0
typing	79	1	0	6	8	282	1186	1	15	0	1	0	0	0	0	0	82	0
eating chips	168	4	93	339	141	53	134	354	4	94	0	0	0	0	0	27	37	85
stairs	0	0	0	0	9	0	0	0	1255	0	6	193	0	0	1	0	0	0
eating sandwich	217	44	120	175	70	93	63	145	11	80	0	0	0	0	0	231	155	92
jogging	0	0	0	0	1	0	0	0	11	0	1613	0	0	0	0	0	0	0
walking	0	0	0	0	0	1	0	0	931	0	21	736	0	0	0	0	0	0
kicking (soccer ball)	0	1	0	3	201	0	0	0	1095	0	8	228	6	8	11	0	0	28
playing catch (tennis ball)	0	0	0	0	28	0	0	0	80	0	22	27	1	550	848	0	0	0
dribbling (basketball)	0	0	0	0	2	0	0	0	40	0	27	0	0	281	1200	0	0	0
eating soup	2	11	10	64	111	51	5	28	14	59	0	0	0	2	2	723	36	185
eating pasta	43	17	25	106	69	50	15	140	0	2	0	0	0	0	0	520	263	26
brushing teeth	19	129	1	82	62	10	10	2	7	10	0	0	0	0	0	30	1	910

Fig. 8. Confusion matrix of the best-performing random forest model

amounts of overfitting for nearly all of the different models. This is likely a result of the user-based nature of the data, which suggests that the models can be used to predict the activity for a certain user, but that these models do not generalize well to other users. It can also be stated that the predictive performance of the models is limited due to the data containing an array of similar activity labels. Activities such as *eating pasta*, *eating soup* and *eating sandwich* are logically described by similar movements of the user, making the classification of these activities difficult. Combining similar labels into one generic label, may improve the performance of the models.

The study that was carried out also leaves much room for improvement. For instance, it can be seen that there are multiple models that perform adequately in the prediction of the activity. Ensembling these models may yield improved predictive performance as multi-model ensembling has been effective for classification in the past [12]. It may also be valuable to add more features to the dataset. Features such as heart rate or magnetometer data may lead to more accurate predictions of activity labels. Lastly, it was not possible to identify different patterns between activities as the activities were carried out one by one. This does not generally resemble the daily routines of the user. Having users carry the smart devices throughout their regular daily routines may lead to other valuable insights regarding the patterns between different activities.

References

1. Abdi, H., Williams, L.J.: Principal component analysis. Wiley interdisciplinary reviews: computational statistics **2**(4), 433–459 (2010)
2. Aly, M.: Survey on multiclass classification methods. Neural Netw **19**, 1–9 (2005)
3. Barbato, G., Barini, E., Genta, G., Levi, R.: Features and performance of some outlier detection methods. Journal of Applied Statistics **38**(10), 2133–2149 (2011)
4. Gao, J., Hu, W., Zhang, Z.M., Zhang, X., Wu, O.: Rkof: Robust kernel-based local outlier detection. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) Advances in Knowledge Discovery and Data Mining. Springer Berlin Heidelberg (2011)
5. Hoogendoorn, M., Funk, B.: Machine learning for the quantified self. On the art of learning from sensory data (2018)
6. Juba, B., Le, H.S.: Precision-recall versus accuracy and the role of large data sets. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 4039–4048 (2019)
7. Likas, A., Vlassis, N., Verbeek, J.J.: The global k-means clustering algorithm. Pattern recognition **36**(2), 451–461 (2003)
8. Mustika, W.F., Murfi, H., Widyaningsih, Y.: Analysis accuracy of xgboost model for multiclass classification-a case study of applicant level risk prediction for life insurance. In: 2019 5th International Conference on Science in Information Technology (ICSITech). pp. 71–77. IEEE (2019)
9. Shoaib, M., Bosch, S., Scholten, H., Havinga, P.J., Incel, O.D.: Towards detection of bad habits by fusing smartphone and smartwatch sensors. In: 2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops). pp. 591–596. IEEE (2015)
10. Thinsungnoena, T., Kaoungkub, N., Durongdumronchaib, P., Kerdprasopb, K., Kerdprasopb, N.: The clustering validity with silhouette and sum of squared errors. learning **3**(7) (2015)
11. Visa, S., Ramsay, B., Ralescu, A.L., Van Der Knaap, E.: Confusion matrix-based feature selection. MAICS **710**, 120–127 (2011)
12. Weigel, A.P., Liniger, M., Appenzeller, C.: Can multi-model combination really enhance the prediction skill of probabilistic ensemble forecasts? Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography **134**(630), 241–260 (2008)
13. Weiss, G.M.: Wisdm smartphone and smartwatch activity and biometrics dataset. Kaggle https://www.kaggle.com/sohelranaccselab/smartphone-smartwatch-activity-biometrics
14. Weisstein, E.W.: Fourier series (2020), https://mathworld.wolfram.com/FourierSeries.html