Activities Recognition

1st Óscar Palacín Dominguez
MUAR
UPC
Barcelona, Spain
oscar.palacin@estudiantat.upc.edu

2nd Javier Pedrosa Alias
MUAR
UPC
Barcelona, Spain
javier.pedrosa@estudiantat.upc.edu

***Abstract*—This report describes the development of the final project of the Pattern Recognition and Machine Learning subject of the University Master in Automatic Control and Robotics of the UPC. It lies in choosing a data set with which apply some of the algorithms learned during the subject at stake, that includes data handling techniques; classifiers; machine learning methods; etc. In order to put them in practice performing a pattern identification and a class prediction processes.**

***Index Terms*—python, pattern recognition, machine learning, classifiers.**

I. Introduction

The project included in this report is contextualized in activities recognition, understanding activities as actions implying displacement, such as walking, running, or travelling by any vehicle. The data set including these activities was built by ourselves, by collecting the samples from a well-known online application, called Strava, for tracking human exercise but also incorporating social network features. Then, taking advantage of some friends past records, and adding some new ones during the development of the project, we have achieved more than 250 activity registers, enough amount to be able to identify the most basic types of activities.

II. State of the Art

The recognition of human activity and motion has been the objective of many prior studies. For example in [1], they collected data from a triaxial accelerometer, which was worn near the pelvic region while the subject performed the following activities:

- Standing
- Walking
- Running
- Climbing up stairs
- Climbing down stairs
- Sit-ups
- Vacumming
- Brushing teeth

Then, they extracted four features (mean, standard deviation, energy and correlation) from each of the three axes of the accelerometer, giving a total of twelve attributes to build a classifier to identify them. They achieved the best performance with Decision tree classifiers, recognizing activities with an overall accuracy of 84%.

Moreover, the extraction of some statistical features is a common method to get useful information from this type of data, such it is shown in papers [2] and [3]. Some of the features used in these last are standard deviation, mean, minimum, maximum, five different percentiles (10, 25, 50, 75, and 90), and a sum and square sum of observations above/below certain percentile (5, 10, 25, 75, 90, and 95); while the target activities correspond to:

- Walking
- Running
- Cycling
- Driving a car
- Idling

Obtaining performances of 86% by Decision Tree, and 95% by KNN and QDA, respectively.

In addition, in recent years, there has been a significant growth in the amount of smart watches and fitness bands, such as the widely famous brands *Fitbit* and *Apple Watch*, which are able to track the number of steps, heart rate, sleep patterns, passive periods, etc. As well as there has been an increasing number of fitness applications for the smartphones with the same purpose. And these kinds of devices have started being used as components of more complex systems that employ many sensors to perform in-depth activity recognition for scenarios such as healthcare, in [4], or persuasive technology for healthy behavior in [5], among many others.

Therefore, the combination of these ideas converge in the focus of this project, which is explained deeply in the following sections.

III. Proposed Approach

Right from the start, the first approach was to carry out the activities identification by the smartphone's own accelerometer, through an external available application that could access this data. But we noticed that we would need to generate a lot of data to get proper results, so we contemplated choosing something similar but that has already available data. Then we ran into Strava, which, as it was introduced in section I, is a social network application for tracking human exercise, and it could provide us interesting features to consider. Ourselves and some of our friends are regular users, so we were able to take advantage of the past records. Furthermore, this application accepts data recorded through smartwatches or through smartphones; and of course we have smartphones, but also a

compatible smartwatch with Strava, so we could even record new data according to our needs, to get enough samples for each activity class. It was important to count with a smartwatch because its records includes a key feature that the smartphones can not, that is the heart rate. In the following sections it is detailed how we managed these lacks.

A. Data handling

The data from Strava corresponds to "Training Center XML" (TCX) file for each activity record; so first of all, we had to convert it to manageable data type. For this, we implemented a TCX parser to convert them to CSV files, extracting the following features for each activity duration instant:

- Time
- Latitude value
- Longitude value
- Altitude value
- Heart rate
- Speed
- Run cadence

Once it was accomplished, we had to analyse and summarize all the data provided for each activity to a single entry for the data set, but first we would like to squeeze the most out of the available information to extract more features, since not all of them are available for all the activities records. As we mentioned before, the heart rate feature is only available for those activities recorded by a smartwatch; the run cadence is only available for the activities recorded as running in the application; and the speed is sometimes unavailable. To fill these gaps, we have searched in each class if there are any value for these critical features, if so, we computed the mean of all the available ones, and it was added as the value for those void of the same class. Moreover, we implemented an additional program to translate the latitude and longitude values, together the time references, into speed and acceleration, which are very interesting features for our purpose. And afterwards, a second extra program was needed to compute statistical information for each activity feature, specifically:

- Maximum value
- Minimum value
- Mean
- Quantiles (3)

Those provide us with 32 features for each entry in the dataset, and a total of 258 entries. In addition, the final classes/labels considered in the data set are:

- Walk
- Run
- Bike
- Vehicle

They are not the ambitious ones of the beginning, when we would like to distinguish between mountain and road bike, and between some motor vehicles such as car and bus or metro. But eventually we collect the enough data to cover the enumerated ones.

B. Software

With the existing data set, we can proceed to the development of classifiers. This was carried out by Colab, which is a free Jupyter notebook environment that runs entirely in the cloud, so a shared project can be simultaneously edited by a team members. And we take advantage of the wonderful machine learning and pattern recognition libraries offered by Python, more concretely:

- Seaborn
- Skalearn

IV. Experimentation

A good practice to start to deal with data sets is to ensure that all their containing information is useful; otherwise time could be wasted analysing features that do not help to distinguish anything. Then we performed two methods in order to avoid this to happen, by applying what it is known as dimensionality reduction:

- **PCA**: it is a linear transformation, rotation and translation, of data from a high dimensional space to a lower one, reducing the number of features of it. This allows to reduce them in cases where some of the variables are highly correlated. To perform PCA, it is necessary to scale the data before, to obtain more balance representation for each of the directions. In order to determine the number of principal component features, it is necessary to reach an accumulative variance explained around the 95%. We achieved this value with a reduction from 32 to 8 features.
- **Correlation**: it is a feature selection method where the data highly correlated with each other is deleted from the data set (i.e. the one with a correlation factor greater than an empirical threshold). This algorithm allows us to understand better the data, keeping the features that are more significant to describe the general behavior. As in the PCA, it is also necessary to scale the data. The reduction in this case was quite similar, from 32 to 10 features.

Comparing both techniques applied to the data set, with the correlation method it is possible to understand what is more interesting about the data. Instead, the PCA is the most widely used and recommended for this kind of applications; but it does not return which are the most significant features of the data but a linear combination of them.

Once a lower dimensional space is obtained, the data is split in two sets to avoid overfitting:

- **Training set**: it is used to fit the model, accounts from 70% of the data.
- **Test set**: it is used to validate the model, accounts from 30% of the data.

To build a proper classifier, the following typical supervised and unsupervised techniques were considered:

- **Linear Discriminant Analysis (LDA)**: it is a supervised probabilistic classifier with a linear decision boundary.

- **Quadratic Discrimant Analysis (QDA)**: it is a supervised probabilistic classifier with a quadratic decision boundary.
- **Gaussian Naive Bayes (GNB)**: it is a supervised probabilistic classifier which assumes univariate normal distribution for each feature.
- **K-Nearest Neighbours (KNN)**: it is an unsupervised classifier which assign the most frequent nearest among samples label to an unlabeled observation.
- **Decision Trees (DT)**: it is a non-parametric supervised classifier. Tree models where the target variable can take a discrete set of values are called classification trees.
- **Random Forests (RF)**: it is a supervised learning algorithm, the ”forest” it builds, is an ensemble of decision trees.
- **Gradient Boosting (GB)**: it is a supervised machine learning technique that aggregates an ensemble of weak individual models to obtain more accurate final model.

With both sets commented above, all these classifiers have been trained and validated. Thus, in order to assess their performances, two main evaluations were carried out:

- **Confusion Matrix**: it is a matrix that shows the amount of hits of the model in a numerical and visual way, where the columns correspond to the true labels and the rows to the predicted ones.
- **Classification report**: it is a summary of standard parameters (precision, recall, f1-score and accuracy) to get information about the performance, for each class and also in an overall view.

The Fig. 1 correspond to the confusion matrices for each classifying model and for both PCA (left, red colors) and correlation (right, blue colors) methods. Whereas the following table summarizes the test set classification results with the average of all the classes' precision performances for each dimensionality reduction method:

TABLE I
GENERAL PERFORMANCES

Dim. Reduction	**Classifiers**						
	LDA	***QDA***	***GNB***	***KNN***	***DT***	***RF***	***GB***
PCA	92%	61%	81%	98%	92%	96%	95%
Correlation	69%	77%	71%	90%	89%	89%	90%

As it can be seen, PCA method obtained slightly better performance values for almost all classifiers evaluation than correlation one, which is understandable owing to the fact that, as it was previously commented, the features given by the PCA are combinations of some others to optimize the classification. Regarding the best-performing classifiers, K Nearest Neighbours and Gradient Boosting have the best marks for both dimensionality reduction methods, but also Decision Tree and Random Forests highlighted by their performance in both cases. They coincide with those classifying models with better performance in prior similar studies, so we can assume that

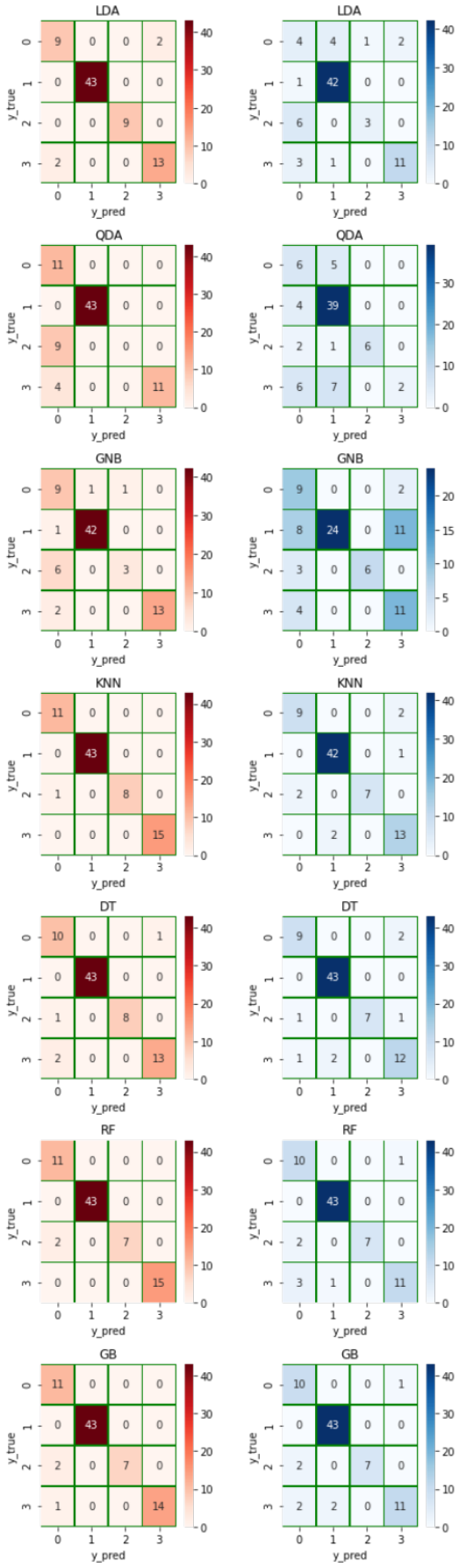

Fig. 1. Confusion matrices (left PCA, right Correlation).

this type of data fits better with random distributions rather than with conventional ones.

Besides, the Fig. 2, the Fig. 3 and the Fig. 4 were added to show the best-performing classifiers' data scatter distribution over the remaining features after the PCA.

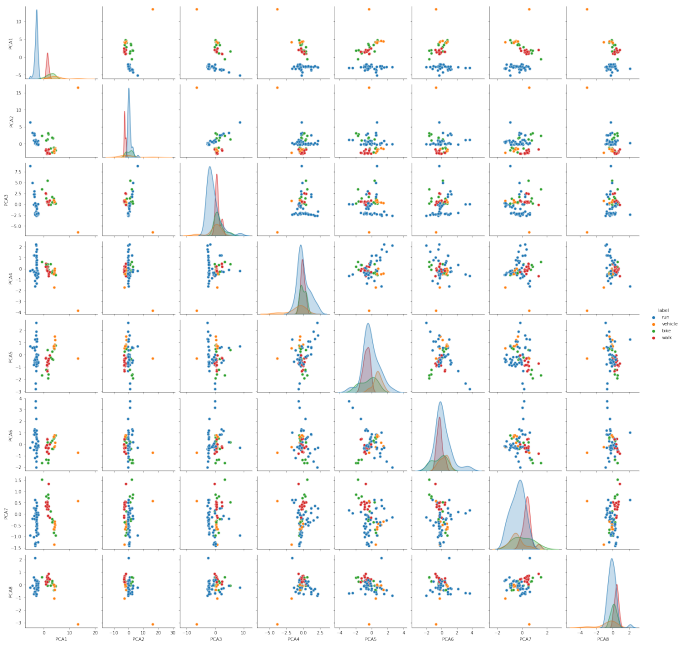

Fig. 2. KNN predicted data pairplot (PCA dimensionality reduction).

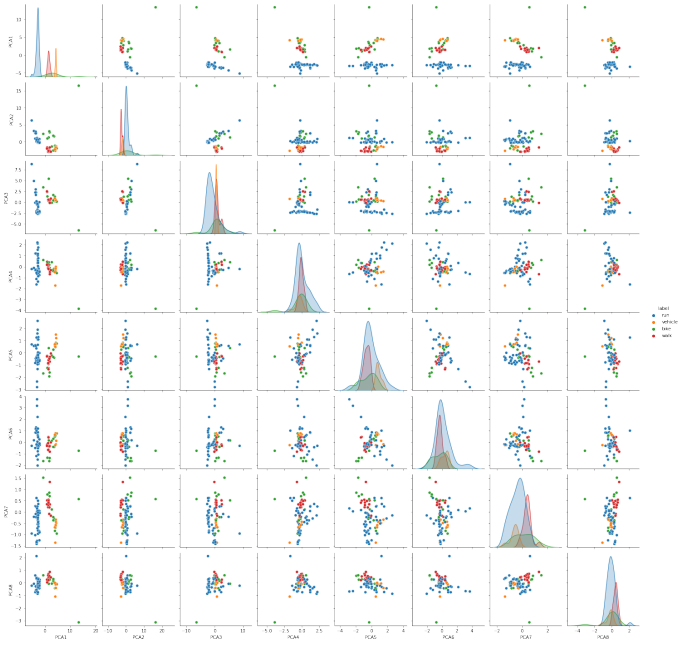

Fig. 3. RF predicted data pairplot (PCA dimensionality reduction).

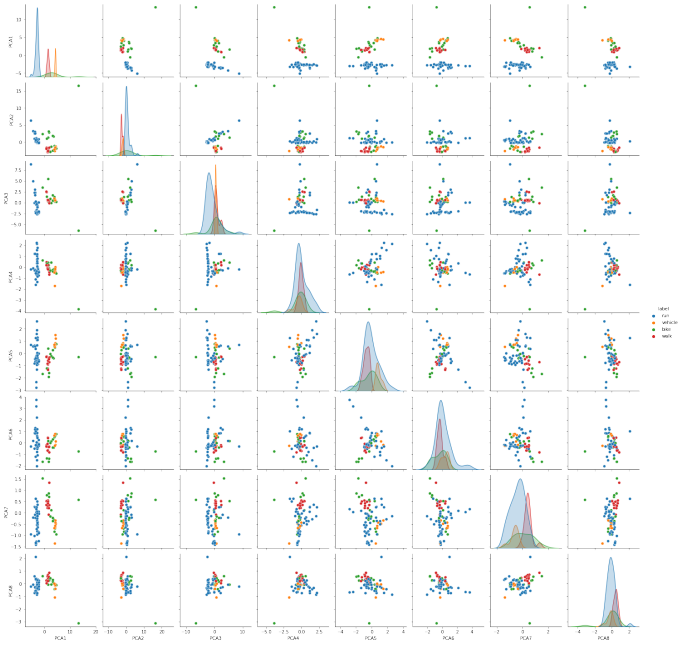

Fig. 4. GB predicted data pairplot (PCA dimensionality reduction).

V. Conclusions

To sum up, we can affirm that thanks to the knowledge acquired during the course to which this project belongs, we have been able to elaborate various functional and quite accurate classifier models for a data set also created and managed by ourselves.

In terms of results, as it was already mentioned above, the best-performing dimensionality reduction method was the PCA against the correlation one, although the second one provides more information about the most describing features, which in this case corresponded to:

- The maximum altitude.
- The maximum heart rate.
- The maximum default speed.
- The minimum default speed.
- The minimum run cadence.
- The minimum speed computed by the GPS data.
- The minimum acceleration computed by the GPS data.
- The mean acceleration computed by the GPS data.
- The 2nd quantile for the acceleration computed by the GPS data.
- The 3rd quantile for the acceleration computed by the GPS data.

In parallel, the best classifiers, with yields between 90-100% on accuracy, were:

- K-Nearest Neighbours (KNN)
- Decision Trees (DT)
- Random Forests (RF)
- Gradient Boosting (GB)

REFERENCES

[1] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, Michael L. Littman: Activity Recognition from Accelerometer Data. In: Department of Computer Science, Rutgers University.

[2] Bao, L., Intille, S.S.: Activity recognition from userannotated acceleration data. In: Pervasive 2004 pp. 1–17 (2004).

[3] Pekka Siirtola and Juha Roning: Recognizing Human Activities Userindependently on Smartphones Based on Accelerometer Data. Department of Computer Science and Engineering, University of Oulu, Finland.

[4] De, D.; Bharti, P.; Das, S.K.; Chellappan, S. Multimodal wearable sensing for fine-grained activity recognition in healthcare. IEEE Internet Comput. 2015, 19, 26–35.

[5] Fritz, T.; Huang, E.M.; Murphy, G.C.; Zimmermann, T. Persuasive technology in the real world: A study of long-term use of activity sensing devices for fitness. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14), Toronto, ON, Canada, 26 April–1 May 2014; ACM: New York, NY, USA, 2014; pp. 487–496.