Project Report

Udacity Machine Learning Engineer Nanodegree
Capstone Project

Jakaria Rabbi

Definition

Project Overview

Human Activity Recognition is the problem to classify human day to day activity using smartphone sensors data. Data continuously generated from the accelerometer and gyroscope and these data are very useful to predict our activities such as walking or standing. There are lots of datasets and ongoing research on this topic. In a survey paper [1], I have seen some works with wearable sensor data and prediction with machine learning techniques. Wearable devices can predict a large range of activities by using data from various sensors. Deep learning models are being used to predict various human activities [2]. Nowadays people use smartphones almost all the time and use many wearable devices. Through these devices, physical and mental health can be monitored by predicting human activity, and nowadays it is an efficient, cheap, and safe way to do this as the covid19 pandemic is ongoing.

Problem Statement

I have selected a dataset from the UCI machine learning repository [3] to calculate the accuracy of three machine learning models and to perform some statistical significance tests. I used a Support Vector Machine, Logistic Regression, and Neural Network with a hidden layer to predict the activity of humans from mobile data. Here the human activities are classified into six categories: walking, walking upstairs, walking downstairs, sitting, standing and laying. I ran experiments to see the performance of the models on the basis of classification results and tried to get the highest possible accuracy from these models. The sensor data in the dataset are collected from two

different sensors and can predict the six different activities. In this project, I tried to get the highest possible performance by these algorithms by parameter tuning, cross-validation and finally comparing the result with two statistical significance tests to get the winner algorithm.

Evaluation Metric

The following evaluation metric is used:

Accuracy classification score:
Accuracy Classification score is defined as the number of correct predictions divided by the total number of predictions. This metric is very important as we want to know whether a human is walking or sitting correctly. This is the main metric to distinguish between our models.

Analysis

Data Exploration & Exploratory Visualization

The dataset is taken from UCI machine learning repository [3]. The dataset contains information from 30 volunteers within age range: 19-48. Each volunteer performs six activities:

- Walking
- Walking upstairs
- Walking downstairs
- Sitting
- Standing
- Laying

Hence, the dataset has six labels to predict. Dataset consists of 561 feature vectors with time and frequency domain variables. These features come from the following data:

- Gravitational acceleration for x, y and z axes
- Body acceleration data files for x, y and z axes
- Body gyroscope data files for x, y and z axes

I also have the ID of an individual volunteer for each record. There are a total 10299 records and training and test dataset splits are 70%-30%. The 6 classes are converted to numerical values sequentially: [1, 2, 3, 4, 5, 6].

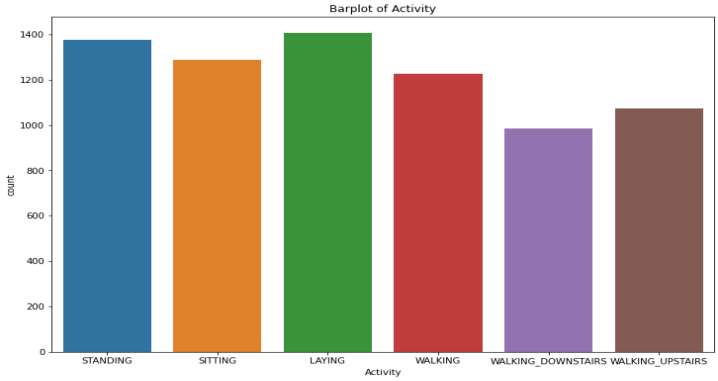

Figure 1. Different human activities with their count in training data [4].

Figure 1 shows the total number of different human activities in training data and figure 2 shows the pattern of two types of activity (static: sitting, standing and laying – dynamic: walking, walking_downstairs and walking_upstairs) using tBodyAccMag-mean() feature which is taken from accelerometer of smartphone. It is clear that two types of activities are easily separable. There are no missing and duplicate value.

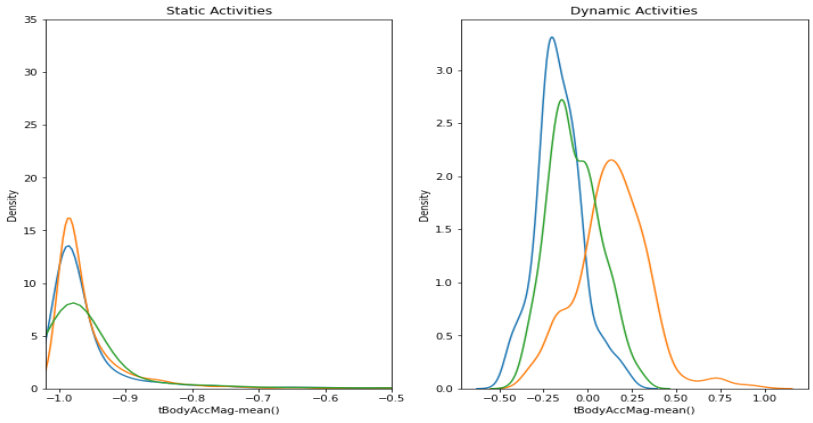

Figure 2. Static and dynamic activity pattern [4].

Algorithms & Techniques

Support Vector Machine (SVM) is used as the first algorithm. SVM is a discriminative classifier defined by a separating hyperplane. If training data is fed to the SVM, it gives an optimal hyperplane which categorizes new examples. In two-dimensional space, this hyperplane is simply a line dividing a plane into two parts where in each class lay in either side [5]. There are variants of SVM, one is Support Vector Classifier (SVC) for more than two classes. In this project, I used that with 'linear', 'rbf' [6] and 'sigmoid' kernel. SVM usually give good accuracy when the number of features is large. In our dataset, the number of features is large, hence I selected this algorithm.

Artificial Neural Network (ANN) with a single hidden layer and the sigmoid activation function is used as the second algorithm. Adam, a stochastic gradient-based optimizer [7], is used for weight optimization. The maximum epoch is 200 and the dataset is shuffled between each epoch. The ANN is selected because it is currently used in many learning problems. I would like to see if it can reach up to the expectations.

Benchmark Models

I used **Logistic Regression** (LR) as a benchmark model and it focuses on maximizing the probability of the data. This is a basic model without much complexity like neural networks. The farther the data lies from the separating hyperplane (on the correct side), the happier LR is [8]. I used the kernel trick and 'lbfgs' optimizer to compare the performance. Some previous research has some good results with that combination of LR [9]. Hence, I used this algorithm as LR and its variants is a good choice for benchmarking.

Methodology

Data preprocessing

Dataset has train and test portion. There are no duplicate and missing value. Train and test data are stored in the data frame initially. I tried to remove some features to reduce the feature space. I tried to select features by computing ANOVA F-value [10] for the dataset. I selected top 100, 200, 400

and 500 features with top ANOVA F-values. But each time I increased the features, the accuracy increased significantly for all the three algorithms. Therefore, I used all the features for training and testing.

The learning is divided into two parts. Initially, the best parameters for all the algorithms is learned. Then, in the second step, the algorithms are compared. The dataset is evaluated by K-fold cross-validation set. I used k=5 for cross-validation. The data splitting is done by stratified sampling.

Implementation:

For finding best hyperparameters, the cross-validation dataset is used for an exhaustive grid search. This cross-validation learned models for all combinations of the listed parameters. Table 1 (for SVC), Table 2 (for ANN) and Table 3 (for LR) have the parameter list that I used to get the best combination of parameters. The parameters are chosen from a range of value. Some parameters are taken as a good selection because they work well in many problems. Accuracy is used as a scoring method. I used python Scikit-learn library for the implementation. The details explanation of all parameters can be found here: [11] [12] [13].

Parameters	Values	Description
Kernel	[Linear, RBF, Sigmoid]	Specifies the kernel type to be used in the algorithm
C	[0.1, 0.5, 1, 2, 5, 10, 100]	Penalty parameter C of the error term.

Table 1. Parameters for SVC

Parameters	Values	Description
Hidden layer sizes	[(10,), (50,), (100,)]	Number of hidden nodes
Alpha	[1e-4, 1e-3, 1e-2]	L2 regularization parameter
Learning rate	[1e-3, 1e-2, 1e-1]	Step size
Beta1	[0.1, 0.5, 0.9]	Adam specific parameter
Beta2	[0.1, 0.5, 0.9]	Adam specific parameter

Table 2. Parameters for ANN

Parameters	Values	Description
Regularizer	[L1, L2]	Specifies the type of regularization
C	[0.1, 0.5, 1, 2, 5, 10, 100]	Penalty parameter C of the error term.

Table 3. Parameters for LR

Refinement

After finding the best parameters, I evaluated the three algorithms on test data. Test data is divided into 10 random sets and every set consist 50% of the data. Hence, I have 10 runs for every algorithm and calculated the mean and variance for these runs. The learning process to find the best parameters used only training data. Hence, I got an unbiased estimation for cross algorithm comparisons.

Best parameters for the three algorithms are found after exhaustive grid search. It took almost 8+ hours to find the parameters. All combination of the given parameters is used to find the best parameters After 5-fold cross validation (stratified), I got the following parameters:

Best parameters for SVC: [C: 1.0, Kernel: Linear**]**
Best parameters for ANN: [Alpha: 0.01, Beta1: 0.9, Beta2: 0.9, Hidden layer sizes: (50,), Learning rate: 0.001**]**
Best parameters for LR: [C: 2, Regularizer: L1**]**

Refinement with Statistical Significance Test

I used hypothesis testing for this project. A hypothesis is tested by measuring and examining a random sample of the population being analyzed. A random population sample is used to test two different hypotheses: the null hypothesis and the alternative hypothesis. The null hypothesis is usually a hypothesis of equality between population parameters; e.g., a null hypothesis may state that the population mean return is equal to zero. The alternative hypothesis is effectively the opposite of a null hypothesis; e.g., the population mean return is not equal to zero. Thus, they are

mutually exclusive, and only one can be true. However, one of the two hypotheses are always true. All hypotheses are tested using a four-step process:

1. The first step is for the analyst to state the two hypotheses so that only one can be right.

2. The next step is to formulate an analysis plan, which outlines how the data are evaluated.

3. The third step is to carry out the plan and physically analyze the sample data.

4. The fourth and final step is to analyze the results and either reject the null hypothesis, or state that the null hypothesis is plausible, given the data [14].

The algorithms are compared using Welch's t-test [15] method because the variances of the algorithms are different. I can find output from t-test that shows whether the algorithms have significant changes in their performance measurements. The t test tells us how significant the differences between groups are; in other words, it lets us know if those differences (measured in means) could have happened by chance. The two-tailed t-test is done pairwise for every algorithm with H_0: $\mu_0 = \mu_1$ and $\alpha = 0.05$ (significance level) [16] to produce a ranking between the three algorithms and for finding the winner algorithm. Here, H_0 is the null hypothesis and μ_0 and μ_1 are the means of two groups of population. Also, the algorithms are compared using 5 times 2-fold cross-validated paired t-test [17] as suggested by researchers [18].

Experimental Results

I used 5-fold stratified cross-validation (stratified) and grid search to find best hyper parameters. Then run the models with best parameters on test data and finally run statistical significance test.

Model Evaluation & Validation

Using the derived best parameters, best model for each algorithm is formulated. Then the models are evaluated on the test data. Test data is divided into 5 random fold (stratified) and each fold contains the whole dataset. Then using the data, I got the average accuracies for the three different models.

Model	Average Accuracy
SVC	96.33%
ANN	93.8%
LR	96.01%

Table 4. Accuracies for all algorithms

Class	Precision	Recall	F1-score
1	0.90	0.97	0.94
2	0.96	0.89	0.92
3	1.00	1.00	1.00
4	0.95	1.00	0.97
5	0.99	0.98	0.98
6	0.98	0.95	0.97

Table 5. Average scores per class for SVC

Class	Precision	Recall	F1-score
1	0.86	0.97	0.91
2	0.96	0.86	0.91
3	1.00	0.97	0.99
4	0.92	0.99	0.95
5	0.98	0.94	0.96
6	0.95	0.95	0.93

Table 6. Average scores per class for ANN

Class	Precision	Recall	F1-score
1	0.89	0.98	0.93
2	0.98	0.86	0.91
3	1.00	1.00	1.00
4	0.95	1.00	0.97
5	1.00	0.97	0.99
6	0.97	0.95	0.96

Table 7. Average scores per class for LR

In Table 4, we can see the accuracies for the algorithms. ANN performs good but has less accuracies compare to other two algorithms. From table 5, 6 and 7 we can get the average precision, recall and F1 score per class. There are 6 classes and all scores per class is in the table. From these tables, we also understand that SVC and LR performs better than ANN. I also tried multiple hidden layers with more hidden nodes, but accuracy dropped. Hence, I did not include that model for comparison. I used the kernel trick implementation for logistic regression, but the accuracy is not significantly good, and it takes more time compare to simple LR. Therefore, for simplicity, I did not include kernels in the LR. Both SVC and LR have closer accuracy and precision-recall scores per class. But we cannot take a decision without statistical significance test. In the next section, I discussed the statistics.

Statistics for Justification

Test data is divided into 10 random fold (stratified) and each fold contains 50% of the data. Then I run the three models with best parameters. After getting the result from 10 runs for each of the three models, I ran two tailed Welch's t-test on the results. I got the result for the three models and shown in the table 8.

Models	SVC (t-value | p-value)		ANN (t-value | p-value)		LR (t-value | p-value)	
SVC (t-value | p-value)	0	1	7.44	6.71e-07	2.31	.0324
ANN (t-value | p-value)	-7.44	6.71e-07	0	1	-5.88	1.42e-05
LR (t-value | p-value)	-2.31	.0324	5.88	1.42e-05	0	1

Table 8. Welch's t-test values for the three models

From Table 8, we understand that SVC is the winner algorithm. Our hypothesis (H_0: $\mu_0 = \mu_1$) has $\alpha = 0.05$ and from this table we see that every p-value is less than that. Therefore, we can reject the hypothesis. So, we can conclude that the results are significantly different. Hence, SVC is the winner algorithm here.

For confirmation about the winner algorithm, I also tested with 5 by 2-fold Cross Validation t paired test. In many discussion and research paper, I found this test and many researchers recommend this test. In this test, the whole dataset (training and test data) is divided into two

random folds and run 5 times. Then the results are collected, and t statistic is calculated. In table 9, the statistic is shown. It is clearly visible that the hypothesis is rejected every time and results are significantly different according to the test.

Models	SVC (t-value | p-value)		ANN (t-value | p-value)		LR (t-value | p-value)	
SVC (t-value | p-value)	0	1	2.72	4.1e-02	3.38	.0195
ANN (t-value | p-value)	-4.65	5.5e-03	0	1	-6.11	1.7e-03
LR (t-value | p-value)	-3.81	.0124	2.73	4.0e-02	0	1

Table 9. 5 by 2-fold Cross Validation t paired test

In the table 9, the t-value| p-value pair is different for same pair if the order is different. Because 5 times 2-fold cross validation makes the outcome different. From the table we can conclude that the SVC algorithm is the winner for this dataset.

Conclusion

In the conclusion, it is evident that SVC performs better than the other two algorithms, though LR is the closer in case of accuracy. ANN cannot compete with SVC for this dataset. In most of the classification problems ANN and deep ANN performs well. For this dataset, deep neural network architectures or convolutional neural network might increase the performance. I think, I will explore the path in near future. I also want to explore the performance of other machine learning algorithms in the future. In this experiment, I understand that simple machine learning algorithms can do well with proper parameter tuning and I can determine the significance of the result by statistic.

References

[1] O. D. Lara and M. A. Labrador, "A Survey on Human Activity Recognition using Wearable Sensors," in IEEE Communications Surveys & Tutorials, vol. 15, no. 3, pp. 1192-1209, Third Quarter 2013. doi: 10.1109/SURV.2012.110112.00192

[2] T. Plötz and Y. Guan, "Deep Learning for Human Activity Recognition in Mobile Computing," in Computer, vol. 51, no. 5, pp. 50-59, May 2018. doi: 10.1109/MC.2018.2381112

[3] https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones

[4] https://www.kaggle.com/vikashrajluhaniwal/eda-all-classification-algorithms-with-96-acc

[5] https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72

[6] https://en.wikipedia.org/wiki/Radial_basis_function_kernel

[7] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization," CoRR, vol. abs/1412.6980, 2014.

[8] http://www.cs.toronto.edu/~kswersky/wp-content/uploads/svm_vs_lr.pdf

[9] G. C. Cawley and N. L. C. Talbot, Efficient approximate leave-one-out cross-validation for kernel logistic regression, Machine Learning, vol, 71, no. 2-3, pp. 243--264, June 2008.

[10] https://en.wikipedia.org/wiki/F-test

[11] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

[12] https://scikit-learn.org/stable/modules/neural_networks_supervised.html

[13] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[14] https://www.investopedia.com/terms/h/hypothesistesting.asp

[15] https://en.wikipedia.org/wiki/Welch%27s_t-test

[16] https://en.wikipedia.org/wiki/Null_hypothesis

[17] http://rasbt.github.io/mlxtend/user_guide/evaluate/paired_ttest_kfold_cv/

[18] Dietterich TG (1998) Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Comput* 10:1895–1923.