# **Ensemble Model**

Krishna Kumar Veeraputhiran

Grand Canyon University

DSC 540 – O500 – Machine Learning for Data Science

Dr. Aiman Darwiche

September 29,2021

# **Ensemble Model**

In this technical report, we will see the implementation of Ensemble methods for classification of Physical activities from wrist accelerometry dataset and compare it with both standalone classification base models and custom Ensemble model. In our case the standalone models are K-Nearest Neighbor (KNN) classification, Classification and Regression Decision Tree (CART) and Support Vector Machine (SVM) classification. For our custom Ensemble model we have combined the three classification model using Voting technique. Finally the convention model that we have used here for our analysis are Random Forest, Adaboosting and Bagging Ensemble models.

**Dataset**

The dataset that we have used for our report is a real time data that has been collected from accelerometers used by 9 subjects doing 18 different physical activities. The data has three inertial measurements (IMU from hand, chest and ankle) and a heart rate monitor for 10 hrs. The dataset has some missing data due to either using wireless sensors or due to the hardware setup. The activity id is the label for each dataset indicating the readings performing these activities. The dataset is downloaded from the UCI repository (<https://archive.ics.uci.edu/ml/datasets/PAMAP2+Physical+Activity+Monitoring>).

**Key Decisions**

Before we go into the modeling the data, here are some of the key points that have been decided for our research. Due to limitation in computation power the training data has been reduced to 0.5% of the original dataset. Using a higher volume dataset on the models was causing memory issue and leading to failures. Linear interpolations have been performed on the Heart rate attribute and for other attributes the missing values are defaulted to 0. The Heart Rate and all the IMU variables are used as inputs for our model.

**Data Preprocessing**

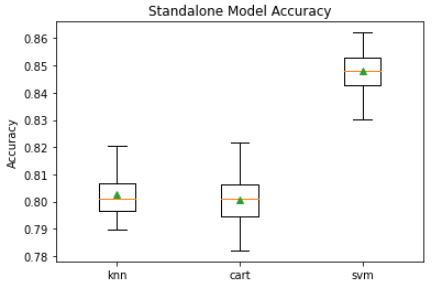
Before we use the dataset for the ensemble model, the data has been preprocessed to make sure the missing data are all addressed. Linear interpolation has been used to fill the missing data for Heart rate attribute. The pandas interpolate function is used with method defaulted to linear and a limit of 10 (consecutive number of Nan to be filled). Also the Nan from other attributes are defaulted to 0 due to time constraint. The pandas fillna function is used to default all the Nan to 0. The output variable is the activity that need to be predicted using the other Heart rate and IMU variables. The training and test data is split using the function train\_test\_split from the sklearn package. As mentioned earlier the training data is selected as 0.5% of the input dataset.

**Model Evaluation**

In our model comparison method we will be comparing the standalone classification model with both custom ensemble and conventional ensemble methods. The custom ensemble method is the combination of the standalone base classification model using the weighted voting method. All the models are being evaluated using the python cross validation function for accuracy. All the models are evaluated using the same set of dataset and their accuracy scores are plotted against each other.

**Standalone Classification Models**

For our research we have considered KNN, CART and SVM models on the training dataset. All these models are evaluated for accuracy. As mentioned above the cross validation function from sci-kit learn package is used to evaluate the accuracy score of all these models. Below is the plot for all these models.



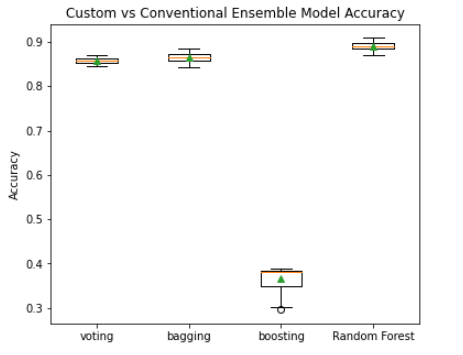
Out of all the three standalone models, SVM had a better accuracy score of 84.8%

**Custom Ensemble Model**

All the three standalone models are combined using the Voting ensemble technique to create our custom ensemble model. We use ‘hard’ voting here, where it uses predicted class labels for majority rule voting. Uniform weightage has been given to all the three standalone models while creating the ensemble model. The overall accuracy of this custom voting ensemble model showed a greater accuracy than the standalone models. The accuracy of the custom voting ensemble model was at 85.7% which was higher than the SVM model.

Conventional Ensemble Models

Finally three different conventional ensemble models were used on the training dataset namely Random Forest, Bagging and Adabossting. These models were then compared with the custom ensemble model for accuracy. Below is the comparison of both the custom and the conventional ensemble models.



We can see that the Random forest ensemble model has the highest accuracy at 89.1% and both Random Forest and Bagging models have better accuracy than the custom voting ensemble model. However the boosting model had a relatively lower accuracy rate.

**Conclusion**

In our research we were able to compare the core models used by Chowdury et al. for the Accelerometry data comparison. However due to both time and processing power constraint we were not able to follow each and every steps the authors have used. Based on the data selection and model usage we can still say that both our result and the authors result coincide making Random Forest the Conventional Ensemble model for the dataset. However the author was able to show that the custom Ensemble model of the Weighted Voting method had the best accuracy compared to all other models. As mentioned earlier some of the key decisions made for our research has caused our outcome to be different than the authors.

# **References**

CHOWDHURY, A. L. O. K. K. U. M. A. R., TJONDRONEGORO, D. I. A. N., CHANDRAN, V. I. N. O. D., & TROST, S. T. E. W. A. R. T. G. (2017). Ensemble methods for classification of physical activities from wrist accelerometry. *Medicine & Science in Sports & Exercise*, *49*(9), 1965–1973. https://doi.org/10.1249/mss.0000000000001291

*sklearn.model\_selection.cross\_val\_score*. scikit. (n.d.). Retrieved September 29, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.model\_selection.cross\_val\_score.html.