[Team 32] Terrain Identification from Time Series Data

CHAITANYA RAJEEV
Email: crajeev@ncsu.edu

SAI SASHANK NAYABU
Email: snayabu@ncsu.edu

LAXMI AISHWARYA THELA
Email: lthela@ncsu.edu

I. Methodology

The goal of this project is to find the type of terrain a person is walking on. Accelerometers measure linear acceleration (specified in mV/g) along X, Y, Z axis. A gyroscope measures angular velocity (specified in mV/deg/s) along X, Y, Z axis. Both gyroscope and accelerometer were connected to the knee of the subjects. Each subject was made to walk on the terrains. Terrains were of different types such as grass, solid ground, marble, tile, upstairs, and downstairs.

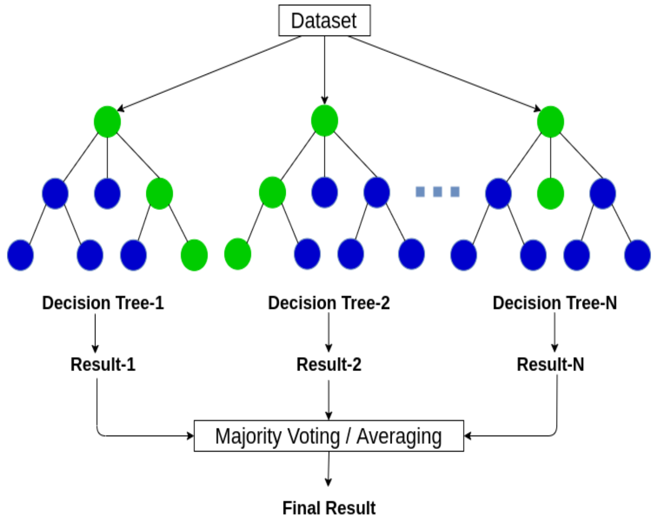
For the baseline model, a Random Forest was used to predict the type of terrain. To build the model we used Python language and different libraries like NumPy, Pandas, Scikit-Learn, Matplotlib. NumPy was used to transform the data. Pandas are used for preprocessing. Matplotlib is used to create plots. In Scikit learn, we used train_test_split to split the data into a training and validation set.

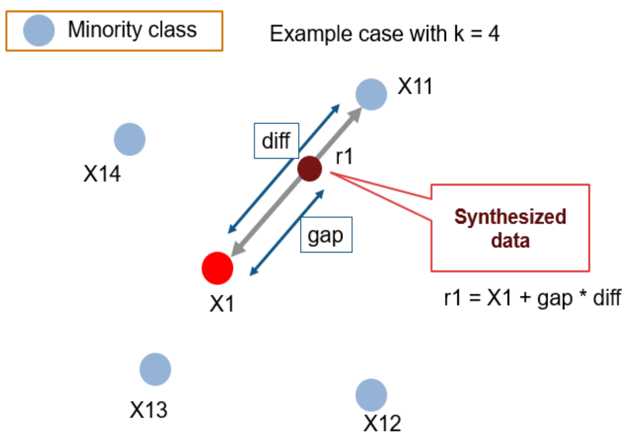
II. Model Training and Selection

A) Model training:

The information was processed in the following ways. The given accelerometer and gyroscope (x values) values were sampled at a rate of 40 HZ whereas the labels (y values) which indicated the type of terrain the subject was walking on, were sampled at 10 HZ. We up-sampled the y labels to match the sampling frequency of x values and then down-sampled the predicted y values to match the sampling frequency of y labels(given timestamps in y_time). All the subjects x and y values were merged to create the final training and validation sets.

Random forest is the first model we used. It is an ensemble technique that uses a large number of decision trees. Each tree generates an output class, and the resultant class is the one with the most votes. We used stratified sampling as our sampling technique. Data is divided into smaller homogenous groups in Stratified Sampling. SMOTE is the second sampling technique we implemented (Synthetic Minority Oversampling Technique). SMOTE is used when there is a severe class imbalance. SMOTE works by finding examples in the feature space that are close together, drawing a line in the feature space between the examples, and drawing a new sample at a point along that line. We used K-nearest neighbors for up sampling and down sampling techniques. K-nearest neighbors is a classification algorithm. We had used KNN for its simplicity and time efficiency.





B) Model Selection:

We implemented Support Vector Machines, Naïve Bayes Classifier, Random Forest and out of these Random Forest yielded us the best results. We did not include the results from the other models due to space constraints.

Table comparing performance of Hyper Parameters:

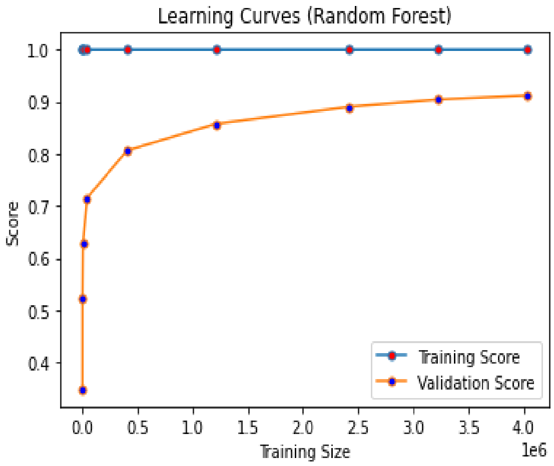
Number of trees	Max Features	Validation Accuracy	Training Accuracy	F1 score
1	Sqrt	0.78	0.91	0.77
1	Log2	0.78	0.92	0.77

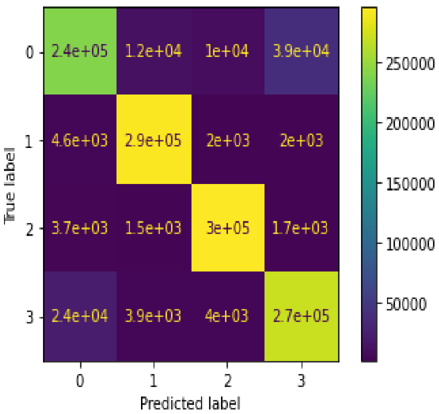
1	None	0.81	0.93	0.81
5	Sqrt	0.87	0.98	0.87
5	Log2	0.87	0.98	0.86
5	None	0.88	0.98	0.88
15	Sqrt	0.89	0.99	0.89
15	Log2	0.90	0.99	0.89
15	None	0.90	0.99	0.90
30	Sqrt	0.90	0.99	0.90
30	Log2	0.90	0.99	0.90
30	None	0.91	0.99	0.90
50	Sqrt	0.91	0.99	0.91
50	Log2	0.91	0.99	0.91
50	None	0.91	0.99	0.91

We used n_parameters= 30 and max_features=None as it gave us better results and there isn't much difference when compared to n_parameters=50

III. EVALUATION

Training, Validation scores and Confusion Matrix for Random Forest are shown in the plot.

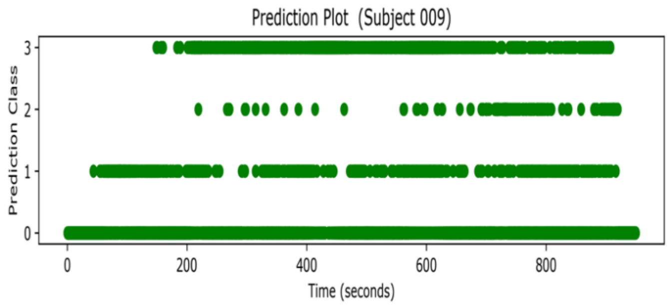


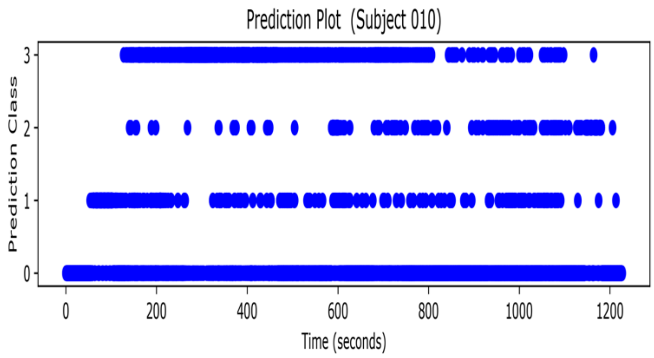


The Accuracy, Recall, and F1-Score determined for each class predicted using Random Forest are shown in the table.

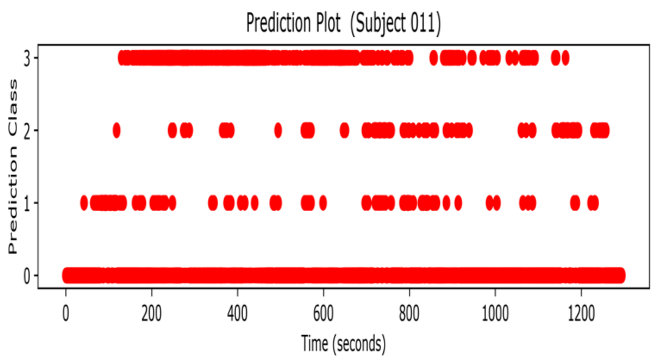
Class Score (Random) Forest	0	1	2	3	Average
Precision	0.88	0.94	0.94	0.86	0.90
Recall	0.79	0.97	0.97	0.85	0.91
F1 score	0.83	0.95	0.91	0.90	0.90
Accuracy	0.79	0.97	0.97	0.85	0.91

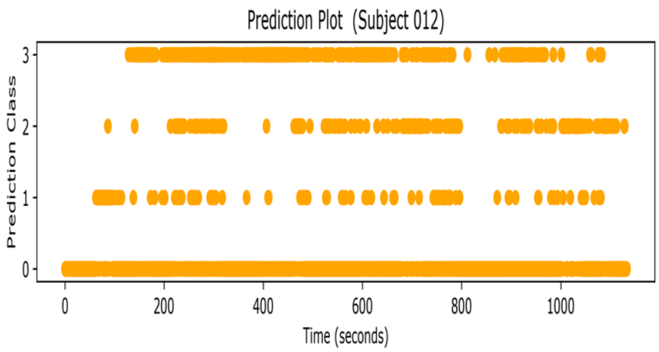
The following are the illustrations of our predictions on test set.





\





REFERENCES

[1] https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

[2] https://iopscience.iop.org/article/10.1088/1674-4527/9/2/011/meta