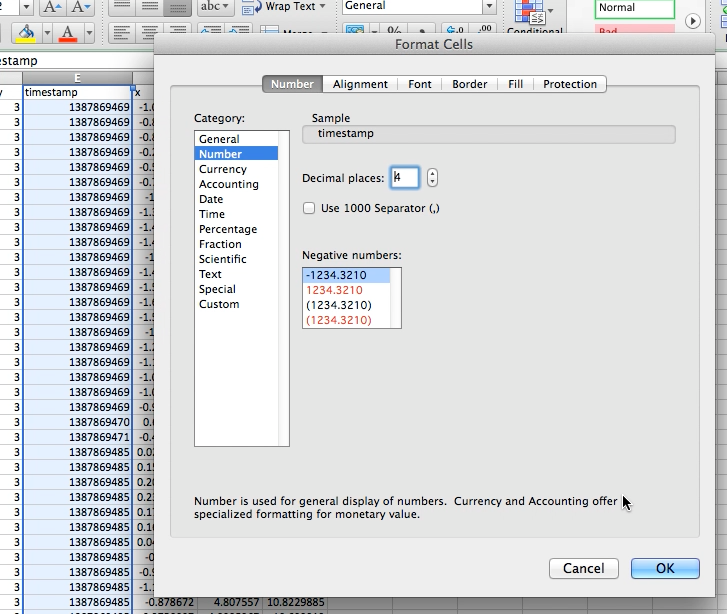
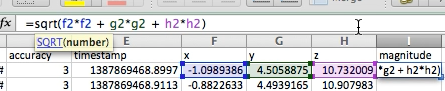
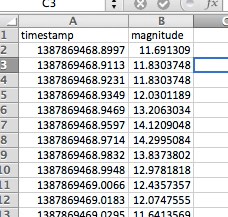
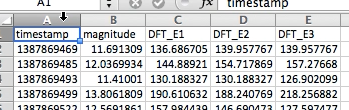
**Data Processing using Excel**

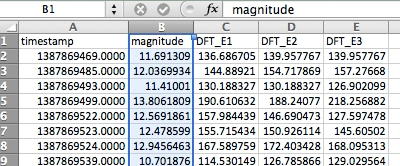
We use Excel to perform various transformations, cleaning, and join operations.

The “scripts” directory contains the python scripts to calculate the DFT and labeling the services.

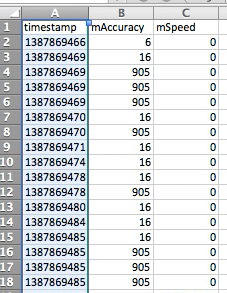
The “Rscripts” directory contains the R scripts for the SVM training and testing implementation.

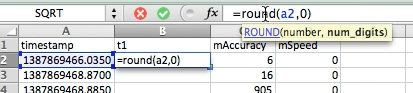
Steps:

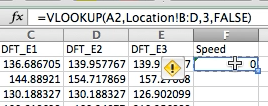
1. Merge the LocationProbe.csv file from each day into a single file and save it as LocationProbe.csv
2. Processing AccelerometerSensor.csv
   1. Open the file in Excel
   2. Transform the timestamp column to have 4 decimal places
   3. Add a new column - Magnitude based on the squared sum of x, y, and z coordinate values.
   4. Extract the timestamp and Magnitude values into a new file to be used as input for DFT calculation. Name it ‘forDFT.csv’
3. Invoke DFT python script. The output is a new file with timestamp, magnitude and DFT coefficients for 1Hz, 2Hz, and 3Hz. Rename the output file as AccewithDFT.csv
4. Transform the timestamp column format to have 4 decimal places in the AccewithDFT.csv



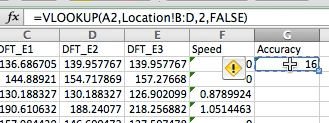
1. Add a new worksheet in AccewithDFT.csv and name it “Location”
2. Open the merged LocationProbe.csv file and copy 3 columns – “Timestamp”, “mSpeed” and “mAccuracy” contents into the “Location” worksheet

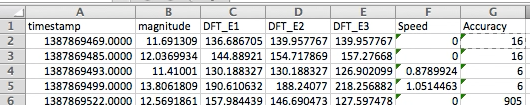
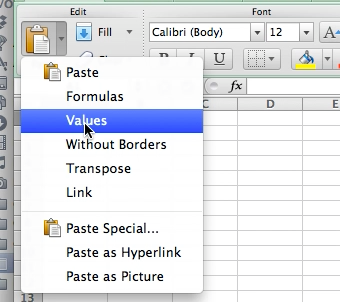


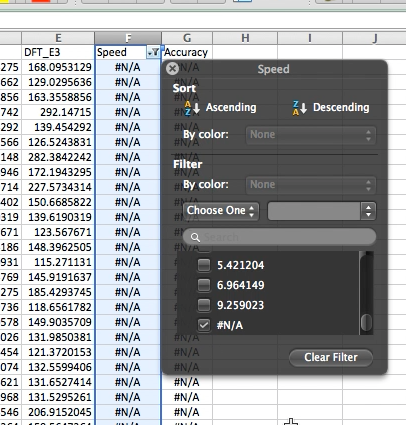
1. Format the timestamp column in the “Location” worksheet to have 4 decimal places
2. Add a new column “t1” to contain rounded values from the timestamp column. We use a *=ROUND(A2,0)* formulae. Repeat it for all rows.
3. Going back the first worksheet containing the accelerometer and DFT data, we add a new column – Speed.
4. The speed column values are fetched from the “Location” worksheet, for the corresponding match in timestamp values. This is the join operation that we perform in Excel.
5. Add a formula =VLookUp() to fetch values from the “speed” column in the “Location” worksheet by matching “timestamp” column in the Acceleration worksheet and “t1” column in the “Location” worksheet.



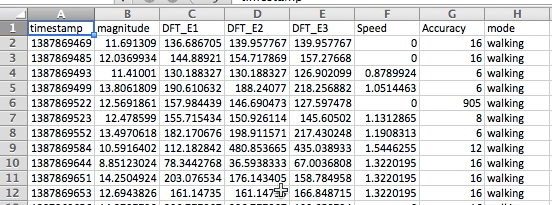
1. Repeat this formulae for the entire speed column
2. Add a new column “Accuracy”
3. Add a formula =VLookUp() to fetch values from the “accuracy” column in the “Location” worksheet by matching “timestamp” column in the Acceleration worksheet and “t1” column in the “Location” worksheet.



1. Repeat this formulae for the entire accuracy column
2. Now our columns should be in the order Timestamp, Magnitude, DFT\_E1, DFT\_E2, DFT\_E3, Speed, Accuracy.
3. We copy all rows and used the “paste special” feature to paste only the values into a new file. This is used to remove all references and formula mapping that existed in the previous worksheet.
4. Add a filter on the “Speed” column and select only “NA” values to be displayed



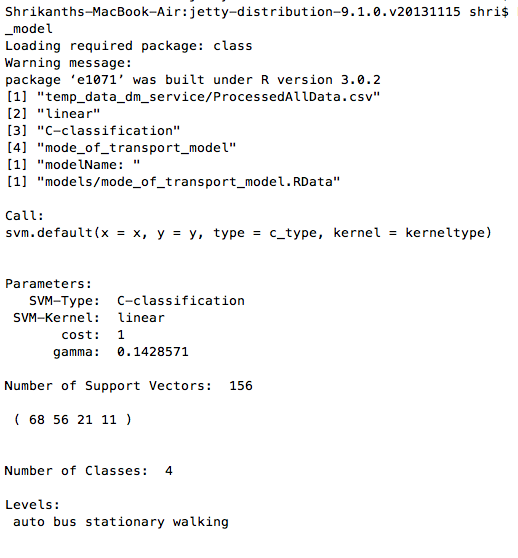
1. Delete all “NA” rows. These are rows for which data could not be joined between the LocationProbe and AccelerometerSensor files.
2. Remove all filters and export worksheet as a CSV file. Lets name it “AcceJoinedData.csv”
3. Invoke the addLabel Python script and use “AcceJoinedData.csv” file in the script as input.
4. The output of the script is another CSV file with the “label” column appended to our joined CSV file. Let us name the resultant file “ProcessedData.csv”



1. Open the “ProcessedData.csv” and apply filters on the “label” column. Select all “NA” rows and delete them.
2. Save this file as ProcessedAllData1.csv
3. Invoke SVM:
4. If this is the first file, then we directly train our model using it as the training set. From the command line, we invoke the R-script svmTraining.R with the file path and other parameters as below

*Rscript svmTraining.R <processed\_Data\_file> <kernel> <classification\_type> <model\_file\_name>*

Eg. *Rscript svmTraining.R acee\_combined.csv linear C-classification mode\_of\_transport\_model*



1. If this is not the first file, we first invoke the SVM\_test script and evaluate our model for the accuracy.
2. Now merge this file with the previous training data and train the SVM model again.

*Rscript Rscripts/svmTesting.R <test\_Data\_Set> <model\_file\_path> <prediction\_out\_file> <matrix\_file\_path>*

*Eg. Rscript Rscripts/svmTesting.R Data2.csv models/mode\_of\_transport\_model.RData predict.csv matrix.csv*