CSC6515 – Machine Learning for Big Data

Project

Professor: Dr. Stan Matwin

TA (Professor): Dr. Amilcar Soares Jr.

Date of Submission: Dec 11, 2017

Submitted by:

Hardik Galiawala

B00777450

Faculty of Computer Science

Dalhousie University

Halifax, Nova Scotia

1. Objective:

In this project, we have practiced feature engineering by developing certain metrics from the given raw data. After feature engineering, we performed two types of classification namely, flat and hierarchical classification using k-fold cross validation. We have also performed statistical significance test to evaluate the results.

2. Data:

The data is provided in a CSV format, which is a GPS trajectory dataset. This dataset was collected in (Microsoft Research Asia) Geolife project by 182 users for 3 years [1]. A GPS trajectory (here) consists of the information of latitude and longitude per time interval along with the mode of transportation of a person. The types of transportation are bus, car, walk, taxi, subway, train, motorcycle and run. But we have ignored the motorcycle and run mode of transportation. Here, the mode of transportation is a class which we need to predict. The rest of the columns can be used to construct features required for the classification models.

3. Task Explanation:

The project tasks are broadly divided into two parts.

A. Feature engineering:

As the given data itself has very few features i.e., columns which can be used to model our data using any classifier, we need to create some features. The project has specific requirements regarding the computation of point features such as distance travelled (Haversine distance in meters), speed (m/s), acceleration (m/s^2) and bearing.

After this step we are asked to group the trajectories by user id and day. The main idea behind this requirement is to reset the features to zero if a day or user is changed. The next step involves creating a sub-trajectory which is a grouping based on the class of the transportation mode. While creating these sub-trajectories we need to perform two computations. The first one involves discarding the trajectories with less than ten trajectory points. The second involves applying statistical functions such as minimum, maximum, mean, median and standard deviation for each feature.

Finally, we need to explore similarities or significant differences between these modes of transportations.

B. Hierarchical classification:

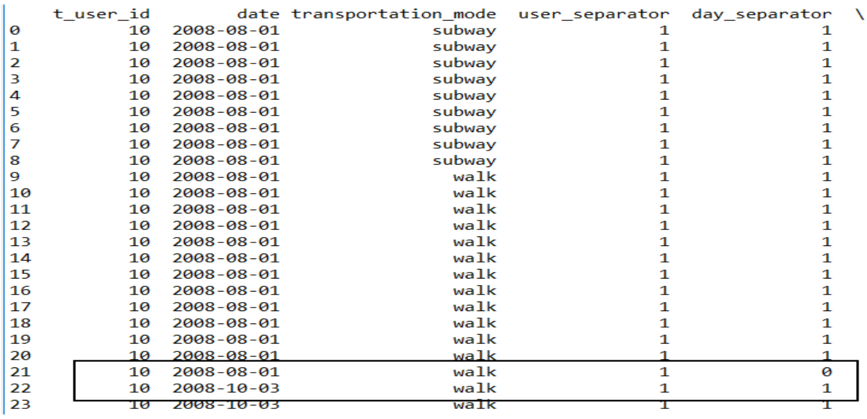
In 3(A), we have explored the data pattern. Based on this pattern, we need to propose a hierarchy to classify the data.

After proposing the hierarchy, we need to perform the data classification with a flat structure and hierarchical structure. But we also need to choose two different classifiers for each structure of classification. The classifiers need to be implemented using a ten-fold cross-validation with stratification. The results must be compared and reported using an accuracy metric along with a significance test for each classifier.

4. Part – A [Feature Engineering]

4.1. Point features computation:

The point features are calculated by computing current row and next row of the raw data (geolife.csv). These features include distance travelled (Haversine distance in meters), speed (m/s), acceleration (m/s^2) and bearing.



```
    t_user_id        date transportation_mode  user_separator  day_separator  \
0          10  2008-08-01              subway               1              1
1          10  2008-08-01              subway               1              1
2          10  2008-08-01              subway               1              1
3          10  2008-08-01              subway               1              1
4          10  2008-08-01              subway               1              1
5          10  2008-08-01              subway               1              1
6          10  2008-08-01              subway               1              1
7          10  2008-08-01              subway               1              1
8          10  2008-08-01              subway               1              1
9          10  2008-08-01                walk               1              1
10         10  2008-08-01                walk               1              1
11         10  2008-08-01                walk               1              1
12         10  2008-08-01                walk               1              1
13         10  2008-08-01                walk               1              1
14         10  2008-08-01                walk               1              1
15         10  2008-08-01                walk               1              1
16         10  2008-08-01                walk               1              1
17         10  2008-08-01                walk               1              1
18         10  2008-08-01                walk               1              1
19         10  2008-08-01                walk               1              1
20         10  2008-08-01                walk               1              1
21         10  2008-08-01                walk               1              0
22         10  2008-10-03                walk               1              1
23         10  2008-10-03                walk               1              1
```

As shown with a black rectangular box, we can easily identify that column "day_separator" becomes zero for the last row of a particular day for the same user. In the same way, the column "user_separator" is set to zero for the last row of a user. These columns will be involved in the calculation of the point features. Thus, we can understand that for the marked first row all point features will be zero.

4.2. Create sub-trajectory groups by class:

We have used pandas groupby function to aggregate data per t_user_id, date and transportation_mode. While grouping we have calculated aggregated values (e.g., min, max, mean, median and standard deviation) for each point feature. Also added a count column, which will count the number of sub-trajectories per group. This will help in selecting the sub-trajectories with greater than 10 trajectory points.

4.3. Exploring the data:

In order to, analyse the current structure of the data, we have used a bar plot. We have used seaborn library to generate the bar plots.

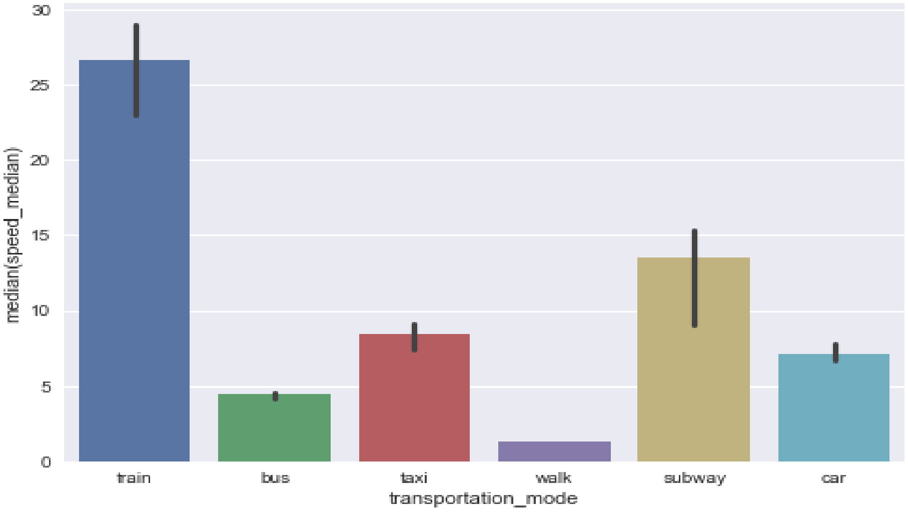


Figure 4.3.1. Transportation mode v/s Median of Speed

The above figure represents median of the speed_median column from the aggregated data frame. It is easily understood that we have median of speed_median on the Y-axis and the classes i.e., transportation_mode on the X-axis. This plot gives us a clear idea that subway and train are very different and lie in the same group since they are the only modes of transportation that has values greater than ten units (here). It also makes sense logically to consider in the same group as they come under the railway transportation. The rest of them come under on road transportation. But if we observe closely, we can further classify these values into 2 different classes i.e., transportation by wheels (car, bus & taxi) and without wheels (walk).

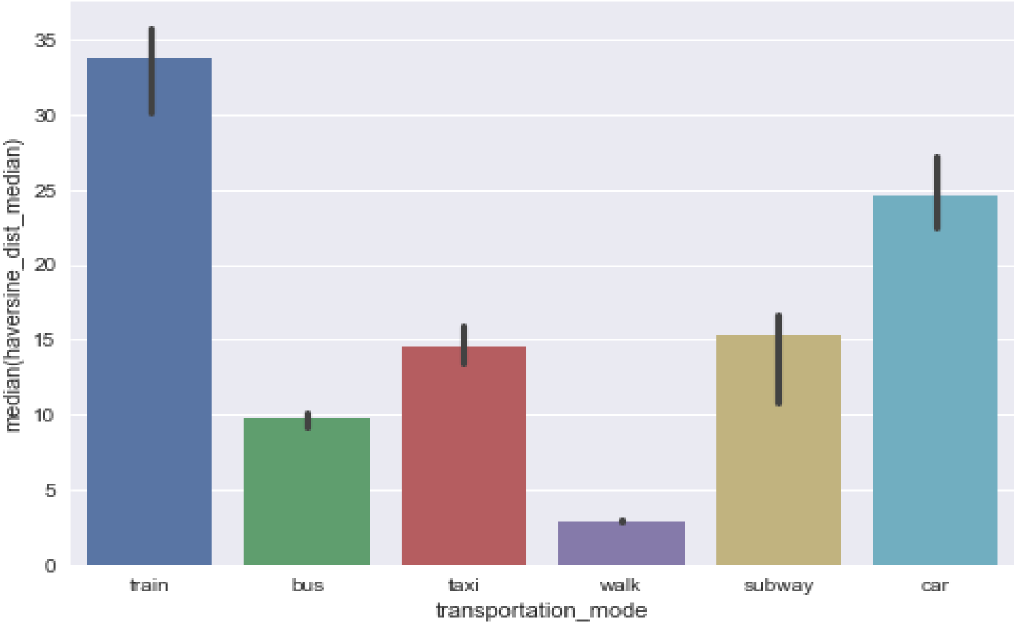


Figure 4.3.2. Transportation mode v/s Median of Distance covered

The above plot makes it very clear for us to differentiate the walk class from the other three classes under "On Road" category.

5. Part – A [Hierarchical Classification]

5.1. Proposing a hierarchy:

As we discussed in **4.3**, we it makes sense to consider a structure as show in the figure below.

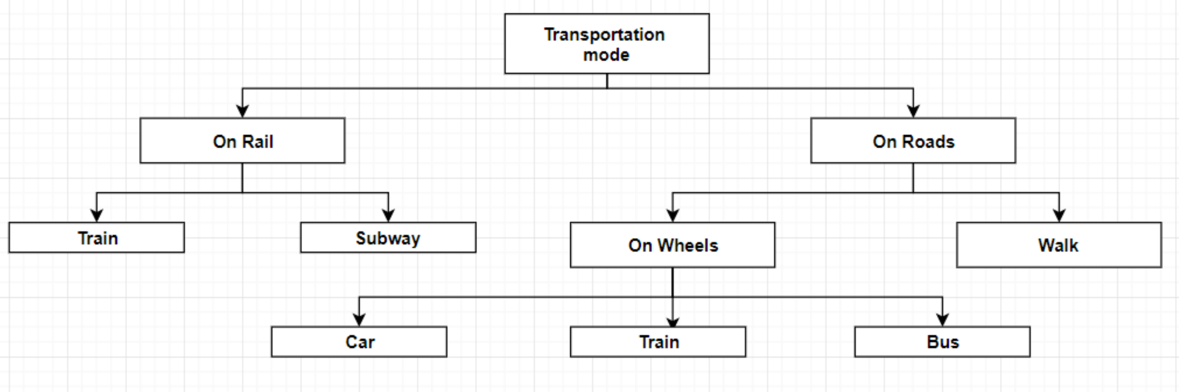


Figure 5.1.1. Hierarchy tree structure

From *Figure 4.3.1.*,we have considered 10 as a threshold unit for splitting classes into "On Rail" and "On Roads". The transportation modes having median greater than 10 units are taken under "On Rail" and less than 10 are taken under "On Roads". Thus, "On Rails" consists of Train and Subway which act as the leaf nodes. But for "On Roads", we can easily conclude after looking at the figure *Figure 4.3.2.* that the walk mode of transportation has least distance covered. This gives us our next level of classification and we split "On Roads" into "On Wheels" and walk (which is a leaf node). "On Wheels" finally consists of Car, Train and Subway.

Thus, this is how the hierarchy looks like which can be easily seen in the *Figure 5.1.1.*

5.2. Implementation of the proposed hierarchy:

In this step we have implemented the above proposed hierarchy with two classifiers. We have used Decision Trees and Random Forest (estimator = 20) in both the cases i.e., in the flat structure as well as hierarchical structure.

The final layer of the hierarchical structure consists of the basic 6 classes.

5.3. Performing Evaluation:

	Model	
	Decision Tree	**Random Forest**
Flat Structure	74	82
Hierarchical structure	91	94

Figure 5.3.1. Accuracy-Table

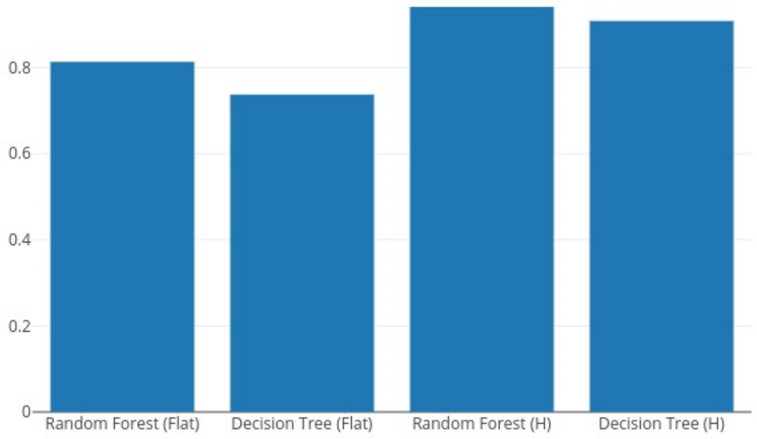
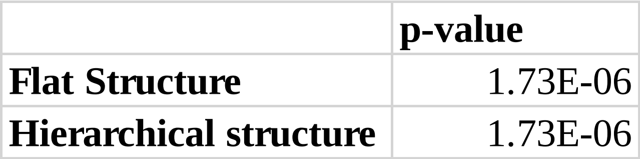


Figure 5.3.2. Accuracy-Bar

From the figure *Figure 5.3.1. and Figure 5.3.2.* we can understand that we can achieve the best accuracy after implementing Hierarchical structure. It is also understood that Random Forest is generally a better classifier (here) as compared to Decision Tree in both the cases regardless the structure implemented. We have implemented this by applying 10 fold cross validation method.



	p-value
Flat Structure	1.73E-06
Hierarchical structure	1.73E-06

Figure 5.3.3. Statistical test

Also, we performed t-test analysis and we understood that both the p-values are less than 0.05 (alpha). This brings us to the conclusion that the results are not dependent on each other.

6. References:

[1] https://www.microsoft.com/en-us/download/details.aspx?id=52367&from=https%3A%2F%2Fresearch.microsoft.com%2Fen-us%2Fdownloads%2Fb16d359d-d164-469e-9fd4-daa38f2b2e13%2F