10-718 Data Analysis Course (Spring 2019)
Assignment 2

Andrew ID: iapostol
Name: Ifigeneia Apostolopoulou
1 late day submission

1 Methods

1.1 Feature Extraction

1.1.1 Location Features

We first compute the popularity of each taxi zone: we count the number of trip records in the dataset which have this zone as either pickup or dropoff location. Subsequently, we discretize with 10 levels the raw number of trips per zone. This is done in the file *zone_popularity.py*. The popularity of the taxi zones is shown in Figure 1.

Subsequently, in order to deal with the discrete areas, we construct a graph $\mathcal{G}$ so that there is an edge between two areas iff they are adjacent. For each one of the taxi-zones (Manhattan, Brooklyn, Queens, Staten Island, Bronx), we create a binary cluster-tree: at distance $h = 1$ from the root, each zone will be divided in two sub-zones, at distance $h = 2$ from the root, each zone will be divided in four sub-zones (by splitting the sub-zones of level 1 in two halves etc). This is necessary and dictated by the sparsity of the data in order to avoid overfitting.

5in

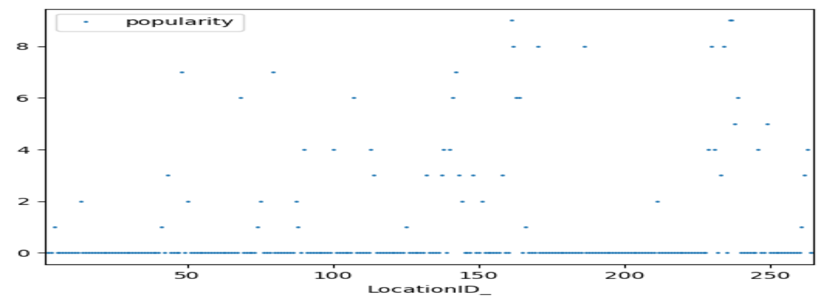

Figure 1: Level of popularity for each taxi zone.

As shown in Figure 2, for many areas there are no or only very few trip records.

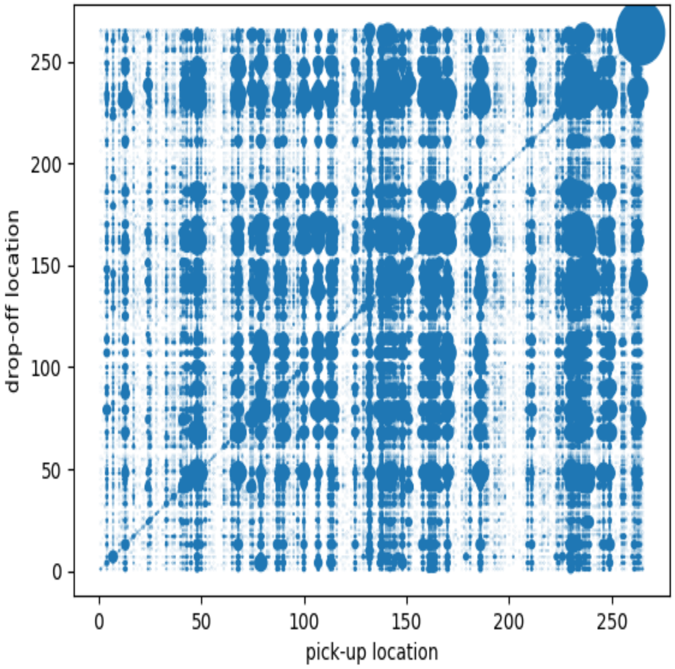

Figure 2: Data Sparsity for each pair of locations.

Therefore, information should be gathered from ”neighboring” (in terms of spatial distance of the pick-up and drop-off locations) trips. Define the trip as $\mathbf{t} = \{o, d, s, t\}$, where o the origin location, d the destination location, s the pickup datetime and t the trip time. Similar trips should fall under the same sub-cluster of its features o and d given a precision (height in the cluster-tree) h. Moreover, for certain locations, the broader area in which it lies captures sufficient information for the trip duration while other more significant factors (traffic effect) should be prioritized for a given model complexity. The clustering of the zones will be accomplished by the graph partition of $\mathcal{G}$. In order to account for the popularity (in terms of the number of pick-ups and drop-offs) of each area, a weight which equals the total number of pick-ups and drop-offs will be assigned to each node/vertice in $\mathcal{G}$. This can be beneficial so that finer discretization is achieved for more popular destinations. The computation of the popularity for each subzone is crucial because each cluster will contain less popular areas and therefore, larger spatial accuracy is achieved for them. Therefore, the clustering of the discretized areas amounts to a graph-partitioning for a graph with weighted nodes and unweighted edges. This can be solved by the method *part_graph* of the *METIS* library for a multi-constrained graph. It should be noted, that the clustering of the discretized zones incurs an overhead only at the pre-rocessing step and not during the training of the model. This overhead is compensated by controlling the complexity of the learned model. Given the computed cluster-tree of the taxi zones $\mathit{CTree}(s)$, where $s = Manhattan, Brooklyn, Queens, Staten, Bronx$, the catgorical feature $O(\mathbf{t}, h)$ for $h = 1, 2, \ldots$, indicates the sub-zone of the zone at height h of $\mathit{CTree}(s)$ in which the origin location o lies and, respectively $D(\mathbf{t}, h)$ indicates the sub-zone of the zone at height h of $\mathit{CTree}(s)$ in which the destination location d lies. Let H be the finest degree of discretization that will be considered (or equivalently the height in the cluster tree of the taxi zones).

Given $\mathcal{G}$, the distance (in terms of the length of the shortest path between two areas) is also computed. The graph construction, the clustering of the areas and the computation of the distance is done in the file *graph_part.py*. As we will see in a following section, from the importance of the features a cluster tree of depth $H = 6$ suffices to capture the precision of popular areas.

Given that the *scikit-learn* library cannot support categorical features, a one-hot representation is required for the clusters of each level in the cluster tree. At the first level of the cluster tree, the categorical feature can take 5 different values (to account for the 5 districts), at the second level it can take $5 * 2$ (since there are 2 sub-clusters for each district) and so-on. In total, $5 * (1 + 2 + 4 + 8 + 16 + 32 + 64)$ one-hot cluster features are required. In order to reduce the number of features, we keep only the 50 most popular clusters (100 features in total for the clustering information of the dropoff and pickup location): we sum the total number of trip records of the areas that fall into a cluster and we keep the clusters with the highest score. It should be mentioned that due to the unbalanced dataset, a cluster at a lower level can be more popular compared to a cluster at a higher level although it contains less areas. This is because its areas are more popular. This is done in the file *location_features.py*.

To account for the precise location, we use 80 one-hot features for the 40 most popular locations (40 for whether the pickup location belongs to one of the most popular areas and 40 for whether the dropoff location belongs to one of the most popular areas). Finally, we keep the pickup and dropoff location as two integer features (which are *not* treated by *scikit-learn* as categorical data). Therefore, an ordering between the location values is assumed during the construction of the tree which does not seem to be detrimental to the learning. However, preserving a one-hot representation of the popular areas does indeed give a better model (as opposed to a model which uses only two integer location features). This is corroborated by the feature importance of the learned model given in a following section.

1.2 Spatio-Temporal features and Traffic Effects

We also define traffic features, motivated by the fact that there is variance in the trip duration for each weekday or hour of the day, see Figures 3 and 4.

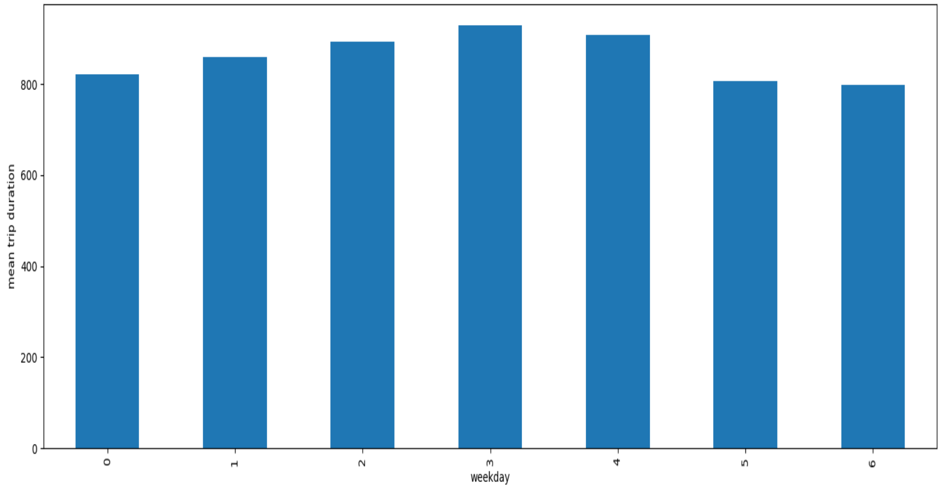

Figure 3: Average Trip Duration per weekday.

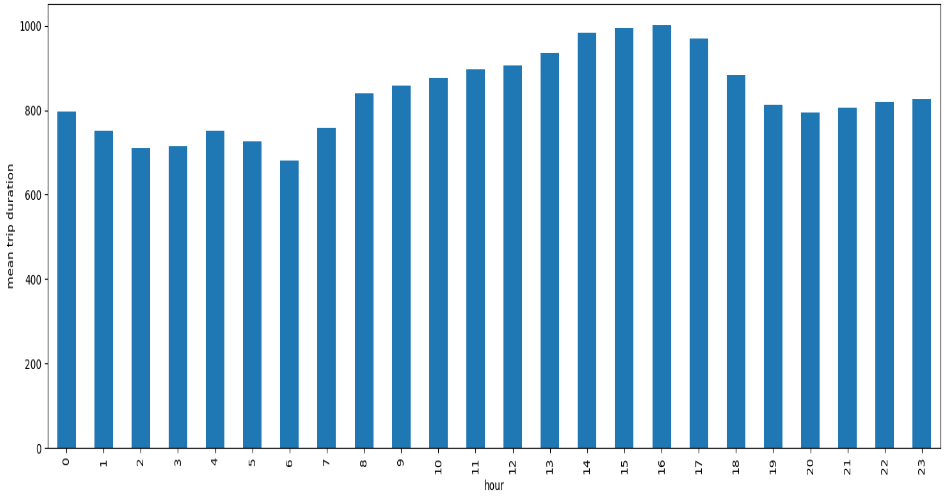

Figure 4: Average Trip Duration per weekday.

However, some areas are not affected by the traffic as we can see in Figure 5. Therefore, a spatio-temporal analysis is encouraged.

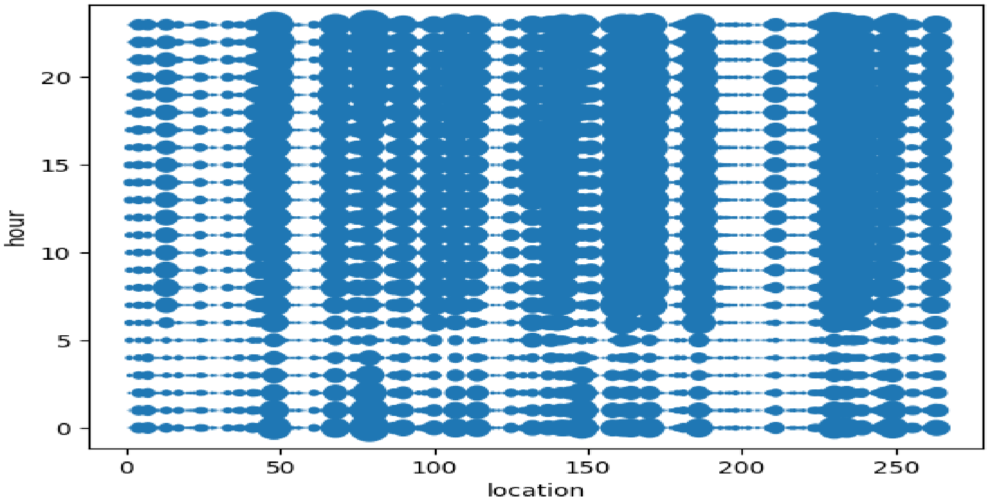

Figure 5: Traffic Information per location and hour of the day.

We compute the number of trips for each pair of location and hour of the day. We then discretize the number of trips with 10 levels. We do the same for each pair of location and day of the week. This is done in the *traffic_binarization.py* file.

We add two integer features for the traffic level at the pickup-location: one for the hour of the start of the trip and one for its day of the week. We have an additional feature for the traffic at the drop-off location on the day of the week. Since the duration of the trip is not known, it proves to be helpful to keep 5 traffic features for the drop-off location at an hour level: one for the pickup hour, and 4 for each one of the next four hours. It would be useful to account for the traffic effects across the whole path of the route, something that is left as an improvement of the model. To roughly account for the these effects, we keep as features the day of the week and the pick-up hour. Their importance is not negligible since they probably cover traffic effects for the whole route, which are not covered by the traffic features defined for the pickup and dropoff locations. Similarly, the traffic features are still important despite the inclusion of the hour and the day of the week, since they account for spatio-temporal and not purely temporal effects. The first part of the file *split_features_learn.py* computes the integer traffic features for each trip record.

1.3 Proposed Model

We employ the random forest regressor `RFR` of the *scikit-learn* library with the features described in the previous section to solve the regression problem. In Table 1, we formally provide the features described in the previous sections.

In Table 2, we describe the parameters of the `RFR` that have to be tuned.

Table 1: Explanatory Variables of the proposed model.

Symbol	Type	Description
$O(\mathbf{t}, h, l)$	Binary	$True$ if the origin location of trip $\mathbf{t}$ is l
$D(\mathbf{t}, h, l)$	Binary	$True$ if the destination location of trip $\mathbf{t}$ is l
$CO(\mathbf{t}, h, c)$	Binary	$True$ if the sub-cluster of the origin location at precision level h of trip $\mathbf{t}$ is c
$CD(\mathbf{t}, h, c)$	Binary	$True$ if the sub-cluster of the destination location at precision level h of trip $\mathbf{t}$ is c
OTrafficHour($\mathbf{t}$)	Integer	Hourly traffic effects in the pick-up location
DTrafficHour($\mathbf{t}, \delta t$)	Integer	Hourly traffic effects at temporal distance δt from the trip's start time in the dropoff location
OTrafficWeek($\mathbf{t}$)	Integer	Weekly traffic effects in pick-up location of trip $\mathbf{t}$
DTrafficWeek($\mathbf{t}$)	Integer	Weekly traffic effects in the drop-off location of trip $\mathbf{t}$
$Oid(\mathbf{t})$	Integer	Id of the pickup discretized area
$Did(\mathbf{t})$	Integer	Id of the dropoff discretized area
$d(\mathbf{t})$	Double	Distance of the trip
$hour(\mathbf{t})$	Integer	Pickup hour of the trip
$weekday(\mathbf{t})$	Integer	Pickup weekday of the trip

In order to adjust these parameters, we consider the last 5 million entries of the dataset given as a validation set. We compute the loss function on the validation dataset for different values of parameters and we keep those which yield the smallest loss function. Especially for H, C, T and L which affect the number of features of the `RFR`, we also get the feature importance for the learned `RFR`s and we note the threshold value after which the corresponding features become uninformative. The final values used are given in Table 3.

Table 2: Model Parameters

Parameter Symbol	Value
H	6
C	Number of popular clusters
L	Number of popular locations
T	Maximum temporal distance (in terms of hours) from the pickup hour for consideration of the traffic effects in the dropoff location
B	Number of base-learners (decision regression trees)
J	Tree depth of the base-learners
F	Ratio of feature sampling
S	Ratio of dataset sampling

Table 3: Model Parameters Selection

Parameter Symbol	Description
H	Height/Precision of the cluster-tree of the discretized taxi-zones
C	50
L	40
T	9
B	100
J	30
F	0.5
S	0.7

2 Results

In Table 4, we see the parameters of the various learned random forest regressors (RFRs), their performance and their learning time.

Table 4: Model and Learning Parameters, Root Mean Squared Log Error and Learning Time.

Fraction of Data	Number of Estimators	Tree Depth	Number of Features	Validation RMSLE Error	Learning Time (sec)
0.2	10	20	0.5	0.343	431.25
0.2	10	30	0.5	0.339	457.87
0.2	10	40	0.5	0.341	470.98
0.2	10	50	0.5	0.341	574.01
0.2	100	30	0.5	0.338	2854.17
0.4	10	30	0.5	0.337	755.84
0.4	10	60	0.5	0.337	1298.17
0.4	100	30	0.5	0.336	3429.35
0.4	100	60	0.5	0.337	3451.96
0.6	10	30	0.5	0.335	1015.31
0.6	100	30	0.5	0.335	4969.45
0.7	10	30	0.5	0.335	1349.71
0.7	100	30	0.5	0.334	5254.66
0.8	10	30	0.5	0.335	2267.71
0.8	100	30	0.5	0.334	7028.733

The first column refers to the fraction of the trip entries in the dataset that was used for the learning. The second and third column refers to the number of trees and their maximum depth in the RFR. The number of features randomly selected for the construction of the tree is given in the forth column. The loss function (RMLSE) on a validation set with the last 5 million entries of the given dataset achieved by each RFR is given in the fifth column. The learning time is given in the last column. This time does not include the feature construction.

We observe that the use of more ensemble estimators boosts the performance of the RFR. However, larger tree depth deteriorates its performance. This is probably due to overfitting. Finally, use of more data does not lead to further improvement. Hence, the final model is trained with 70% of the available trip records. We note though that due to the careful selection of the features, onl 20% and very fast learning can give decent accuracy. In Listing 1, we get the feature importance in the learned regressor. The distance is the most important factor; this facts motivates us for use of exact longtitude and latitude coordinates instead of the path length in the graph of the discretized areas. We observe that both the traffic at the pickup hour and the pickup hour are important: the one accounts for spatio-temporal correlations while the other for the traffic effects on the whole route. The same holds for the weekday and traffic at the pickup and dropoff location on the weekday. The binary features pickup_level_x_y refer to whether the pickup location belongs to the cluster y at level x. The last binary features refer to the whether or not the pickup or dropoff location is one of the 40 most popular. For example the area 138 as a pickup location is an important factor.

Listing 1: Feature importance.

```
Feature Importance
distance 0.497724
traffic_pickup_hour 0.016545
pickup_hour 0.064218
weekday 0.033537
traffic_dropoff_hour 0.005819
traffic_dropoff_p1_hour 0.009822
traffic_dropoff_p2_hour 0.007905
traffic_dropoff_p3_hour 0.003235
traffic_dropoff_p4_hour 0.006556
traffic_dropoff_p5_hour 0.006509
traffic_dropoff_p6_hour 0.004409
traffic_dropoff_p7_hour 0.002378
traffic_dropoff_p8_hour 0.001978
traffic_pickup_weekday 0.011629
traffic_dropoff_weekday 0.003332
PULocationID 0.009881
DOLocationID 0.023166
pickup_level_1_3 0.032815
pickup_level_2_3 0.001774
pickup_level_2_4 0.001950
pickup_level_3_3 0.001915
pickup_level_3_4 0.001086
pickup_level_3_6 0.000890
pickup_level_3_5 0.001097
pickup_level_4_4 0.000952
pickup_level_4_6 0.000398
pickup_level_4_10 0.000373
pickup_level_4_8 0.000773
pickup_level_4_9 0.000982
pickup_level_4_5 0.000952
pickup_level_4_3 0.000805
pickup_level_4_7 0.001301
pickup_level_5_6 0.001229
pickup_level_5_5 0.000264
pickup_level_5_18 0.000260
pickup_level_5_10 0.000255
pickup_level_5_4 0.000381
pickup_level_5_13 0.000603
pickup_level_5_14 0.000341
pickup_level_5_9 0.000300
pickup_level_5_11 0.000505
pickup_level_5_16 0.000342
pickup_level_5_8 0.000351
pickup_level_5_7 0.000334
pickup_level_1_5 0.071367
pickup_level_5_15 0.000612
pickup_level_5_17 0.000271
pickup_level_6_10 0.000327
pickup_level_6_34 0.000284
pickup_level_7_25 0.000153
pickup_level_6_18 0.000149
pickup_level_7_12 0.000167
pickup_level_6_8 0.000163
pickup_level_6_7 0.000181
pickup_level_7_11 0.000194
pickup_level_6_9 0.000170
pickup_level_2_7 0.003193
pickup_level_6_6 0.000405
pickup_level_7_36 0.000080
pickup_level_6_24 0.000080
pickup_level_6_15 0.000342
pickup_level_7_22 0.000335
pickup_level_6_17 0.000135
```

```
pickup_level_7_24 0.000131
pickup_level_3_11 0.002790
pickup_level_6_33 0.000161
pickup_level_7_49 0.000168
dropoff_level_1_3 0.040387
dropoff_level_2_3 0.001795
dropoff_level_2_4 0.002806
dropoff_level_3_3 0.002070
dropoff_level_3_4 0.001940
dropoff_level_3_6 0.000708
dropoff_level_3_5 0.001062
dropoff_level_4_4 0.000850
dropoff_level_4_6 0.000842
dropoff_level_4_10 0.000253
dropoff_level_4_8 0.000857
dropoff_level_4_9 0.000555
dropoff_level_4_5 0.000834
dropoff_level_4_3 0.000928
dropoff_level_4_7 0.000954
dropoff_level_5_6 0.000791
dropoff_level_5_5 0.000087
dropoff_level_5_18 0.000562
dropoff_level_5_10 0.000177
dropoff_level_5_4 0.000341
dropoff_level_5_13 0.000320
dropoff_level_5_14 0.000291
dropoff_level_5_9 0.000163
dropoff_level_5_11 0.000441
dropoff_level_5_16 0.000267
dropoff_level_5_8 0.000394
dropoff_level_5_7 0.000257
dropoff_level_1_5 0.018811
dropoff_level_5_15 0.000437
dropoff_level_5_17 0.000230
dropoff_level_6_10 0.000164
dropoff_level_6_34 0.000265
dropoff_level_7_25 0.000096
dropoff_level_6_18 0.000109
dropoff_level_7_12 0.000127
dropoff_level_6_8 0.000128
dropoff_level_6_7 0.000091
dropoff_level_7_11 0.000076
dropoff_level_6_9 0.000207
dropoff_level_2_7 0.004858
dropoff_level_6_6 0.000386
dropoff_level_7_36 0.000062
dropoff_level_6_24 0.000051
dropoff_level_6_15 0.000270
dropoff_level_7_22 0.000265
dropoff_level_6_17 0.000121
dropoff_level_7_24 0.000111
dropoff_level_3_11 0.002122
dropoff_level_6_33 0.000081
dropoff_level_7_49 0.000079
pickup_zone_161 0.000155
pickup_zone_237 0.000164
pickup_zone_236 0.000192
pickup_zone_170 0.000080
pickup_zone_162 0.000342
pickup_zone_230 0.000141
pickup_zone_186 0.000170
pickup_zone_234 0.000223
pickup_zone_48 0.000151
pickup_zone_79 0.000186
pickup_zone_142 0.000302
pickup_zone_163 0.000373
pickup_zone_239 0.000203
pickup_zone_141 0.000147
pickup_zone_107 0.000173
pickup_zone_68 0.000180
pickup_zone_164 0.000264
pickup_zone_238 0.000423
pickup_zone_249 0.000168
pickup_zone_229 0.000229
pickup_zone_231 0.000202
pickup_zone_263 0.000088
pickup_zone_100 0.000278
pickup_zone_138 0.012390
pickup_zone_90 0.000119
pickup_zone_140 0.000155
pickup_zone_246 0.000183
pickup_zone_113 0.000165
pickup_zone_132 0.018274
pickup_zone_233 0.000147
pickup_zone_137 0.000159
pickup_zone_148 0.000325
pickup_zone_114 0.000156
```

```
pickup_zone_43 0.000270
pickup_zone_262 0.000387
pickup_zone_158 0.000189
pickup_zone_143 0.000186
pickup_zone_50 0.000216
pickup_zone_144 0.000180
pickup_zone_13 0.000220
dropoff_zone_161 0.000095
dropoff_zone_237 0.000128
dropoff_zone_236 0.000083
dropoff_zone_170 0.000060
dropoff_zone_162 0.000257
dropoff_zone_230 0.000110
dropoff_zone_186 0.000084
dropoff_zone_234 0.000165
dropoff_zone_48 0.000204
dropoff_zone_79 0.000079
dropoff_zone_142 0.000565
dropoff_zone_163 0.000215
dropoff_zone_239 0.000119
dropoff_zone_141 0.000073
dropoff_zone_107 0.000092
dropoff_zone_68 0.000092
dropoff_zone_164 0.000139
dropoff_zone_238 0.000234
dropoff_zone_249 0.000066
dropoff_zone_229 0.000157
dropoff_zone_231 0.000107
dropoff_zone_263 0.000072
dropoff_zone_100 0.000195
dropoff_zone_138 0.002277
dropoff_zone_90 0.000068
dropoff_zone_140 0.000079
dropoff_zone_246 0.000142
dropoff_zone_113 0.000097
dropoff_zone_132 0.020073
dropoff_zone_233 0.000087
dropoff_zone_137 0.000086
dropoff_zone_148 0.000201
dropoff_zone_114 0.000075
dropoff_zone_43 0.000231
dropoff_zone_262 0.000177
dropoff_zone_158 0.000110
dropoff_zone_143 0.000109
dropoff_zone_50 0.000123
dropoff_zone_144 0.000136
dropoff_zone_13 0.000179
```

3 Model Improvements

We suggest and explain the following improvements:

1. Currently, we consider one graph per taxi-zone for Manhattan, Brooklyn, Queens, Staten Island and Bronx according to the given discretization scheme. The areas are connected only by one edge between two subareas in these areas which are chosen by inspection of the maps, when the distance of two subareas which belong to different taxi districts have to be computed. A finer graph construction would include many edges between the 5 taxi districts, and one large common graph would be clustered. The use of one large graph instead of 5 could also give more accurate discrete distances between the subzones and more accurate location subclusters. For example, with one large graph two subareas could be belong to the same cluster even if they belong to different taxi-zones if the popularity of the subareas would be more evenly distributed between the nodes of the cluster. This improvement is justified by the fact that both the distance and cluster features are very informative (and more informative than the weekday, pickup hour or the exact location id): for example see the features *pickup_level*_1_3,

$pickup\_level\_5\_6$, $dropoff\_level\_2\_7$ etc.

2. According to Listing 1, the distance is the most critical factor. Therefore, exact longtitude and latitude coordinates of the origin and destination locations could further improve the performance of the model. However, it should be noted that the spatial discretization is still required for clustering the taxi zones.

3. The traffic at an hourly and weekday level are also important factors for the problem, however they only account for the congestion in the pickup and dropoff location. This could be problematic if both the pickup and dropoff locations are not popular destinations but the route between them traverses popular areas. Further analysis of the graph to compute the shortest path between any pair of two areas and a total traffic score which sums the traffic level across the path could rectify this.

4. Another model modification would be to solve a weighted regression problem. This is justified by Figure 2. For some pairs of locations, there are a lot of available trip records even if the pickup hour is the same. Therefore, the dataset allows us to consider factors other than spatial or temporal such as the driver habit. To this end, the average trip duration can be computed for a given pair of locations and pickup hour (or a range of pickup hours) and the distance of a record from the ”average trip” could be used to give us the weight/ credibility of each datapoint: entries with larger variance are given less weight, since it is more likely that they constitute outliers or that the trip duration was affected by external factors. Inversely, entries close to the ”average trip” are given a higher weight.

5. A more sophisticated base model could be used. We currently use a random forest regressor `RFR`. The advantage is that the individual estimators can be learned in parallel and this fact renders the choice of a random regressor very attractive given the large size of the dataset. However, a gradient-boosting regression tree `GBRT` would be preferable if focus has to be placed not on the learning speed but the accurcacy of the model.

The model could be further augmented if external sources of information are added in the pipeline. For example,

1. Weather conditions on the days and at the hours of the trip records. Extreme weather conditions could justify outlier trip durations and mediate overfitting effects.

2. Traffic light information in the discretized areas such as the average waiting time in each subzone. This information could be crucial and significantly differentiate trip records even if their end or intermediate locations fall into the same spatial clusters in case the average waiting time is significantl different.

3. Routing information for example from Google maps. The route that drivers may chose between two locations can be different at a different time of the day for the same pair of dropoff and pickup location. This external factor decouples the distance from the traffic information (which is computed given the shortest path in the constructed graph).