NYU Shanghai - NYU Stern

Final Project_Rongjia_Jing

SHBI-GB 7311 B1: Machine Learning for Business (Summer 2021)

1 Prediction Task

(a) Initial Data Pre-processing

I carry out some basic analysis about the raw data in dataframe_train.csv, and here the results:

```
> str(train_df)
'data.frame':	509604 obs. of  25 variables:
 $ courier_id           : int  10007871 10007871 10007871 10007871 10007871 10007871 10007871 10007871 10007871 10007871 ...
 $ wave_index           : int  0 0 0 0 0 0 1 1 1 1 ...
 $ tracking_id          : num  2.1e+18 2.1e+18 2.1e+18 2.1e+18 2.1e+18 ...
 $ courier_wave_start_lng: num  122 122 122 122 122 ...
 $ courier_wave_start_lat: num  39.1 39.1 39.1 39.1 39.1 ...
 $ action_type          : Factor w/ 2 levels "DELIVERY","PICKUP": 2 1 2 1 2 1 2 1 2 1 ...
 $ date                 : int  20200201 20200201 20200201 20200201 20200201 20200201 20200201 20200201 20200201 20200201 ...
 $ group                : num  2.02e+16 2.02e+16 2.02e+16 2.02e+16 2.02e+16 ...
 $ level                : int  3 3 3 3 3 3 3 3 3 3 ...
 $ speed                : num  4.75 4.75 4.75 4.75 4.75 ...
 $ max_load             : int  11 11 11 11 11 11 11 11 11 11 ...
 $ weather_grade        : Factor w/ 4 levels "Bad Weather",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ aoi_id               : Factor w/ 34912 levels "0001e4c643b3623dea2a0e9bce7d15ad",..: 17519 17519 25009 25009 15901 15901 9497 9497 15901 15901 ...
 $ shop_id              : Factor w/ 11193 levels "00009e27a7938a119afe10d36649fa1d",..: 6464 6464 5986 5986 6398 6398 8792 8792 10652 10652 ...
 $ id                   : int  120 121 122 123 124 125 126 127 128 129 ...
 $ source_type          : Factor w/ 3 levels "ASSIGN","DELIVERY",..: 1 3 2 3 2 3 1 3 2 3 ...
 $ source_tracking_id   : num  2.1e+18 2.1e+18 2.1e+18 2.1e+18 2.1e+18 ...
 $ source_lng           : num  122 122 122 122 122 ...
 $ source_lat           : num  39.1 39.1 39.1 39.1 39.1 ...
 $ target_lng           : num  122 122 122 122 122 ...
 $ target_lat           : num  39.1 39.1 39.1 39.1 39.1 ...
 $ grid_distance        : num  377 780 550 707 770 ...
 $ expected_use_time    : int  804 298 545 341 166 315 537 759 434 529 ...
 $ urgency              : int  1246 1246 2462 1205 1882 1045 2194 757 2553 998 ...
 $ hour                 : int  11 11 11 11 11 11 12 12 12 12 ...
> summary(train_df)
   courier_id          wave_index      tracking_id       courier_wave_start_lng courier_wave_start_lat   action_type          date               group          
 Min.   : 10007871   Min.   : 0.0    Min.   :2.1e+18   Min.   :119.9          Min.   :36.06          DELIVERY:254802   Min.   :20200201   Min.   :2.020e+16  
 1st Qu.: 10697343   1st Qu.: 1.0    1st Qu.:2.1e+18   1st Qu.:121.4          1st Qu.:39.12          PICKUP  :254802   1st Qu.:20200209   1st Qu.:2.020e+16  
 Median :111751082   Median : 2.0    Median :2.1e+18   Median :121.5          Median :39.16                            Median :20200216   Median :2.020e+17  
 Mean   : 81512553   Mean   : 2.4    Mean   :2.1e+18   Mean   :121.5          Mean   :39.18                            Mean   :20200215   Mean   :1.496e+17  
 3rd Qu.:118760809   3rd Qu.: 4.0    3rd Qu.:2.1e+18   3rd Qu.:121.6          3rd Qu.:39.22                            3rd Qu.:20200222   3rd Qu.:2.020e+17  
 Max.   :125996858   Max.   :16.0    Max.   :2.1e+18   Max.   :122.3          Max.   :39.71                            Max.   :20200227   Max.   :2.020e+18  
                                                                                                                                                             
     level           speed          max_load                  weather_grade                                  aoi_id      
 Min.   :0.000   Min.   :3.009   Min.   : 1.00   Bad Weather         :   250   d85359523f72551e00a84203526763ea:  1646  
 1st Qu.:2.000   1st Qu.:4.868   1st Qu.: 8.00   Normal Weather      :385684   a6b4e84b85a0f916af1878a663adcc44:   894  
 Median :3.000   Median :5.458   Median : 9.00   Slightly Bad Weather: 57710   7604775a6af51891221a504623faccd7:   670  
 Mean   :2.607   Mean   :5.348   Mean   : 8.98   Very Bad Weather    : 65960   e9aa84196fa1300e2d1db6d179bd440d:   586  
 3rd Qu.:3.000   3rd Qu.:5.779   3rd Qu.:10.00                                 2dd7c1333118eebbece72e0bb52316b:   558  
 Max.   :3.000   Max.   :6.943   Max.   :19.00                                 69436d4ae309d5078cc59b68964d9671:   552  
                                                                               (Other)                         :504698  
                             shop_id             id            source_type      source_tracking_id   source_lng       source_lat       target_lng       target_lat   
 406a47750b2960d4666f4dc63f704d9f:  4494   Min.   :     0   ASSIGN  : 76069   Min.   :2.1e+18    Min.   :119.9   Min.   :36.06   Min.   :121.1   Min.   :38.83  
 8944ec8db309614c49fc787d3ba12f44:  2448   1st Qu.:127401   DELIVERY:178733   1st Qu.:2.1e+18    1st Qu.:121.4   1st Qu.:39.12   1st Qu.:121.4   1st Qu.:39.12  
 99a98a05589466aeafd178494ba439cc:  2004   Median :254802   PICKUP  :254802   Median :2.1e+18    Median :121.5   Median :39.16   Median :121.5   Median :39.16  
 4f0c5ad2934f0b4c88a8cec1d22d0e2c:  1970   Mean   :254802                     Mean   :2.1e+18    Mean   :121.5   Mean   :39.18   Mean   :121.5   Mean   :39.18  
 89436019672a6cf266544739e1d29c23:  1882   3rd Qu.:382202                     3rd Qu.:2.1e+18    3rd Qu.:121.6   3rd Qu.:39.22   3rd Qu.:121.6   3rd Qu.:39.22  
 61be8c5f24588a313c738cc8e68f60a5:  1702   Max.   :509603                     Max.   :2.1e+18    Max.   :122.3   Max.   :39.71   Max.   :122.3   Max.   :39.70  
 (Other)                         :495104                                                                                                                       
 grid_distance    expected_use_time    urgency              hour      
 Min.   :     0   Min.   :   1.0    Min.   :-340771   Min.   : 6.00  
 1st Qu.:   330   1st Qu.: 189.0    1st Qu.:    859   1st Qu.:12.00  
 Median :   869   Median : 354.0    Median :   1752   Median :14.00  
 Mean   :  1078   Mean   : 441.7    Mean   :   1572   Mean   :14.48  
 3rd Qu.:  1572   3rd Qu.: 584.0    3rd Qu.:   2590   3rd Qu.:17.00  
 Max.   :429173   Max.   :9246.0    Max.   :  11345   Max.   :23.00  
```

The selected features are 'action_type', 'level', 'weather_grade', 'source_type', 'courier_wave_start_lng', 'courier_wave_start_lat', 'speed', 'max_load', 'source_lng', 'source_lat', 'target_lng', 'target_lat', 'grid_distance', 'urgency', 'hour', 'expected_use_time'.

Especially, 'action_type', 'level', 'weather_grade', 'source_type' and 'hour' are converted into factor variables.

It is worth noticed that the maximum data in the 'grid_distance' column is 429,173, which is extremely large. So it could be an outlier of the dataset. With further examination, I decided to exclude those with 'grid_distance' over 10,000 (99.999% percentile), and thus 6 records are removed in this step.

```
 grid_distance
 Min.   :     0
 1st Qu.:   330
 Median :   869
 Mean   :  1078
 3rd Qu.:  1572
 Max.   :429173
```

```
> quantile(train_df$grid_distance,0.99999) # 10298.84
 99.999%
10298.84
```

(b) Baseline Model

I choose the LASSO model as baseline model. I tuned the lambda parameter with cross-validation and the best lambda here is 0.25.

Under this setting, the out-of-sample MAE is 217.399.

```
> lasso_cv
glmnet

356718 samples
    37 predictor

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 285375, 285374, 285373, 285375, 285375
Resampling results across tuning parameters:

  lambda  RMSE      Rsquared   MAE
  0.00    333.6335  0.3233115  218.1886
  0.05    333.6335  0.3233115  218.1886
  0.10    333.6335  0.3233115  218.1886
  0.15    333.6335  0.3233115  218.1886
  0.20    333.6335  0.3233115  218.1886
  0.25    333.6335  0.3233115  218.1886
  0.30    333.6343  0.3233095  218.1935
  0.35    333.6356  0.3233061  218.2006
  0.40    333.6370  0.3233024  218.2070
  0.45    333.6385  0.3232986  218.2126
  0.50    333.6401  0.3232947  218.2171
  0.55    333.6418  0.3232907  218.2245
  0.60    333.6438  0.3232862  218.2321
  0.65    333.6459  0.3232813  218.2399
  0.70    333.6481  0.3232761  218.2477
  0.75    333.6505  0.3232706  218.2556
  0.80    333.6530  0.3232648  218.2637
  0.85    333.6556  0.3232589  218.2717
  0.90    333.6584  0.3232526  218.2797
  0.95    333.6613  0.3232459  218.2876
  1.00    333.6643  0.3232390  218.2948

Tuning parameter 'alpha' was held constant at a value of 1
MAE was used to select the optimal model using the smallest value.
The final values used for the model were alpha = 1 and lambda = 0.25.
> MAE(predict.test.lasso,Y.test)
[1] 217.399
```

(c) XGBT model

Here I choose the XGBT model with the following setting:

```
> xgbt_cv = xgb.cv(data=xgb_train, nrounds = 30, early_stopping_rounds = 5, nfold=3, metrics = 'mae', showsd=FALSE,
+                  max_depth = 10, eta = 0.1, gamma = 0.001,lambda=1,colsample_bynode=0.8,
+                  objective = "reg:squarederror")
[1]     train-mae:398.834930    test-mae:398.901530
Multiple eval metrics are present. Will use test_mae for early stopping.
Will train until test_mae hasn't improved in 5 rounds.
```

```
[20]    train-mae:178.551575    test-mae:184.978633
[21]    train-mae:178.197021    test-mae:184.914007
[22]    train-mae:177.936295    test-mae:184.947902
[23]    train-mae:177.765045    test-mae:185.055191
[24]    train-mae:177.568807    test-mae:185.189946
[25]    train-mae:177.410822    test-mae:185.356496
[26]    train-mae:177.271057    test-mae:185.522593
Stopping. Best iteration:
[21]    train-mae:178.197021+0.438993   test-mae:184.914007+0.706461
```

The top 10 important feature are listed below:

```
> xgb.importance(model=model.xgbt)
                   Feature         Gain        Cover    Frequency
 1:          grid_distance 3.773221e-01 3.370794e-01 0.1152407084
 2:      action_typePICKUP 2.055346e-01 8.091756e-02 0.0122848591
 3:                urgency 1.604185e-01 2.031351e-01 0.1440508855
 4:      source_typePICKUP 1.019926e-01 5.149850e-02 0.0055500125
 5:    source_typeDELIVERY 7.400758e-02 4.282060e-02 0.0109753056
 6:                  speed 1.154826e-02 5.463878e-02 0.1077575455
 7:             target_lat 1.071900e-02 2.465145e-02 0.0442130207
 8: courier_wave_start_lng 9.273082e-03 2.399885e-02 0.1002120229
 9:             target_lng 7.955823e-03 1.879644e-02 0.0460838114
10: courier_wave_start_lat 7.373466e-03 2.021436e-02 0.0923547019
```

The out-of-sample MAE is 184.0564, which is much better than the baseline model.

```
> MAE(predict.test.xgb,Y.test)
[1] 184.0564
```

(d) Further Feature Engineering

After revisiting the selected features, it is noticeable that we have coordinates for courier, source and target respectively. Courier and target coordinates are listed in the top 10 important features.

It is common that given the similar distance, the traveling time for urban areas and suburban areas is different, mostly because of the traffic condition. So here I would like to introduce a feature that addresses this fact.

I decide to do clustering on 'source_lng', 'source_lat', 'target_lng', 'target_lat' to find similar group of paths and use the clustering label to replace these 4 features. Here I use K-means with 10 clusters to process data.

After introducing the cluster feature, the model performance is shown as below:

```
> xgbt_cv2 = xgb.cv(data=xgb_train_cluster, nrounds = 30, early_stopping_rounds = 5, nfold=3, metrics = 'mae', showsd=FALSE,
+                   max_depth = 10, eta = 0.1, gamma = 0.001,lambda=1,colsample_bynode=0.8,
+                   objective = "reg:squarederror")
[1]     train-mae:398.869578    test-mae:398.913981
Multiple eval metrics are present. Will use test_mae for early stopping.
Will train until test_mae hasn't improved in 5 rounds.
```

```
[20]	train-mae:178.929647	test-mae:185.267482
[21]	train-mae:178.600901	test-mae:185.218389
[22]	train-mae:178.380574	test-mae:185.280477
[23]	train-mae:178.192932	test-mae:185.392176
[24]	train-mae:178.053197	test-mae:185.550608
[25]	train-mae:177.943594	test-mae:185.740041
[26]	train-mae:177.859919	test-mae:185.955729
Stopping. Best iteration:
[21]	train-mae:178.600901+0.131627	test-mae:185.218389+0.232353
```

The top 10 important feature are listed below:

```
> xgb.importance(model=model.xgbt.cluster)
                          Feature         Gain        Cover    Frequency
 1:                 grid_distance 3.708499e-01 3.374039e-01 0.1357917570
 2:             action_typePICKUP 2.238553e-01 8.342827e-02 0.0114657577
 3:                       urgency 1.898435e-01 2.128496e-01 0.1639913232
 4:             source_typePICKUP 7.621764e-02 5.405970e-02 0.0073132941
 5:           source_typeDELIVERY 6.905214e-02 4.048982e-02 0.0117136659
 6:         courier_wave_start_lat 1.705635e-02 5.416478e-02 0.1453982027
 7:         courier_wave_start_lng 1.469215e-02 4.793360e-02 0.1510381159
 8:                         speed 1.334843e-02 5.448932e-02 0.1349240781
 9:                      max_load 6.027507e-03 4.073797e-02 0.0625968392
10:     weather_gradeVery.Bad.Weather 2.871189e-03 1.926675e-02 0.0152463588
```

The out-of-sample MAE is now 184.3148.

```
> MAE(predict.test.xgb.cluster,Y.test.cluster)
[1] 184.3148
```

(e) Model Interpretation

In the last model, the top 10 important features contain 'grid_distance', 'action_type', 'urgency', 'source_type', 'courier_wave_start_lng/lat', 'speed', 'max_load' and 'weather_grade'.

The 'grid_distance', 'action_type', 'urgency' are of the greatest importance when predicting the expected time for next action.

From this perspective, it is reasonable for Eleme delivery service to assign delivery task based on distance between courier, urgency of task, courier location, courier current status, and courier current location.

2 Casual Inference Task

(a) Initial Data Pre-processing and Balance Check

Using OLS approach to do Balance check on individuals in the Portland area:
P-value =0.68 >0.05 , so the distribution for individuals in the Portland area of treatment group is not significantly difference from that of the control group.

```
> summary(OLS_portland)#p-value:0.68,cannot reject H0, balance

Call:
lm(formula = portland ~ treatment + numhouse_1 + numhouse_2,
    data = df_all)

Residuals:
    Min      1Q  Median      3Q     Max
-0.3419 -0.3419 -0.3404  0.6581  0.7134

Coefficients: (1 not defined because of singularities)
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.341928   0.002312 147.891   <2e-16 ***
treatment   -0.001474   0.003571  -0.413     0.68
numhouse_1  -0.053836   0.004153 -12.963   <2e-16 ***
numhouse_2         NA         NA      NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4693 on 74919 degrees of freedom
Multiple R-squared:  0.002357,  Adjusted R-squared:  0.00233
F-statistic: 88.49 on 2 and 74919 DF,  p-value: < 2.2e-16
```

Balance check on 5 variables(birthyear/gender/selfsign/visitED/ num_visit_pre_cens_ed):

For example, the variable birthyear_list:

```
Call:
lm(formula = birthyear_list ~ treatment + numhouse_1 + numhouse_2,
    data = df_port)

Residuals:
     Min       1Q   Median       3Q      Max
-23.7815 -10.2784   0.6208  10.6208  19.7216

Coefficients: (1 not defined because of singularities)
             Estimate Std. Error   t value Pr(>|t|)
(Intercept) 1968.2784     0.1021 19282.841   <2e-16 ***
treatment      0.1008     0.1600     0.630    0.529
numhouse_1     0.4023     0.1948     2.065    0.039 *
numhouse_2         NA         NA        NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.04 on 24643 degrees of freedom
Multiple R-squared:  0.000216,  Adjusted R-squared:  0.0001348
F-statistic: 2.662 on 2 and 24643 DF,  p-value: 0.06985
```

```
summary(OLS_birthyear)# p-value: 0.529, cannot reject H0
summary(OLS_female) #p-value:0.132, cannot reject H0
summary(OLS_selfsign) #p-value:0.383, cannot reject H0
summary(OLS_visitED) #p-value:0.583, cannot reject H0
summary(OLS_numED)#p-value:0.979,cannot reject H0
```

Since P-value >0.05, the distribution for birthyeargender/selfsign/visitED/ num_visit_pre_cens_ed of treatment group is not significantly different from that of the control group.

(b) Causal Effect of Being Selected by Lottery

Model:
enrolled ~ treatment+numhouse_1+numhouse_2
+birthyear_list +gender +selfsign +visit_ED+ num_visit_pre_cens_ed

Label:
Enrolled in any Medicaid program

Features:
being selected by the lottery (Treatment)
number of people in household(numhouse_1+numhouse_2)
year of birth(birthyear_list)
female(gender)
Signed up self for lottery (selfsign)
Any ED visit(visit_ED)
Number of ED visits (num_visit_pre_cens_ed)

```
Call:
lm(formula = enroll ~ treatment + numhouse_1 + numhouse_2 + birthyear_list +
    gender + selfsign + visit_ED + num_visit_pre_cens_ed, data = df_port)

Residuals:
     Min       1Q   Median       3Q      Max
-0.68288 -0.26560 -0.15623 -0.02827  0.97900

Coefficients: (1 not defined because of singularities)
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)           -1.2806510  0.4262336  -3.005  0.00266 **
treatment              0.2462176  0.0054343  45.308  < 2e-16 ***
numhouse_1             0.0119097  0.0087667   1.359  0.17431
numhouse_2                    NA         NA      NA       NA
birthyear_list         0.0006611  0.0002164   3.055  0.00226 **
gender                 0.0858633  0.0052614  16.320  < 2e-16 ***
selfsign               0.0599551  0.0115226   5.203 1.97e-07 ***
visit_ED               0.0500367  0.0071574   6.991 2.80e-12 ***
num_visit_pre_cens_ed  0.0123082  0.0017758   6.931 4.28e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4089 on 24626 degrees of freedom
  (12 observations deleted due to missingness)
Multiple R-squared:  0.09665,   Adjusted R-squared:  0.0964
F-statistic: 376.4 on 7 and 24626 DF,  p-value: < 2.2e-16
```

The average treatment effect (ATE) of being selected by the lottery on being enrolled in any Medicaid program is 0.246.

Appendix: Code

```
### Question 1
library('xgboost')
train_df = read.csv("dataframe_train.csv")

# 1a. Initial Data Pre-processing¶
### EDD
str(train_df)
head(train_df)
summary(train_df)
quantile(train_df$grid_distance,0.99999) # 10298.84
subset(train_df,grid_distance>10000)

# remove outlier with grid_distance > 6000
df_filter = subset(train_df,grid_distance<=10000)

process_data_train = function(df){
  select_cols_train = c('action_type','level','weather_grade','source_type',
                 'courier_wave_start_lng','courier_wave_start_lat',
                 'speed','max_load',
                 'source_lng','source_lat','target_lng','target_lat',
                 'grid_distance','urgency','hour','expected_use_time')
  df$action_type = as.factor(df$action_type)
  df$level = as.factor(df$level)
  df$weather_grade = as.factor(df$weather_grade)
  df$source_type = as.factor(df$source_type)
  df$hour = as.factor(df$hour)
  result_df = subset(df,select=select_cols)
  return(result_df)
}

df_select_train = process_data_train(df_filter)
str(df_select_train)
summary(df_select_train)

df.dataframe = data.frame(model.matrix(~.,df_select_train[,1:15]))
df.dataframe$expected_use_time = df_select_train$expected_use_time

# 1b. Baseline
library('glmnet')
library('caret')
set.seed(100)
training.rows <- sample(1:nrow(df.dataframe), nrow(df.dataframe)*0.7)
df.train = df.dataframe[training.rows,]
df.test = df.dataframe[-training.rows,]
X.train = as.matrix(df.train[,1:37])
Y.train = df.train$expected_use_time
X.test = as.matrix(df.test[,1:37])
Y.test = df.test$expected_use_time
```

```
set.seed(100)
trControl_baseline <- trainControl(method  = "cv", number  = 5)
lasso_cv<- train(expected_use_time~., method = "glmnet",trControl=trControl_baseline,
             tuneGrid = expand.grid(alpha=1,lambda = seq(0,1,0.05)),
             metric='MAE',data = df.train)
lasso_cv

lasso_model = glmnet(X.train, Y.train,family="gaussian", alpha=1, lambda=0.25)
lasso_model
predict.test.lasso = predict(lasso_model,newx=X.test)
MAE(predict.test.lasso,Y.test)

# 1c. XGBT model
xgb_train = xgb.DMatrix(data = X.train, label = Y.train)
xgb_test = xgb.DMatrix(data = X.test, label = Y.test)

set.seed(100)
xgbt_cv = xgb.cv(data=xgb_train, nrounds = 30, early_stopping_rounds = 5, nfold=3, metrics = 'mae',
showsd=FALSE,
             max_depth = 10, eta = 0.1, gamma = 0.001,lambda=1,colsample_bynode=0.8,
             objective = "reg:squarederror")
xgbt_cv
set.seed(100)
model.xgbt = xgboost(data = xgb_train,nrounds = 21, max_depth = 10, eta = 0.1,
                gamma = 0.001,lambda=1,colsample_bynode=0.8,
                objective = "reg:squarederror",
                eval_metric='mae')
xgb.importance(model=model.xgbt)
predict.test.xgb = predict(model.xgbt,X.test)
MAE(predict.test.xgb,Y.test)

# 1d. Further Feature Engineering
# clustering based on source_lat,source_lng,target_lat,target_lng
set.seed(100)
cluster.df.train = X.train[,c('source_lat','source_lng','target_lat','target_lng')]
plot(cluster.df.train[,1:2])
plot(cluster.df.train[,3:4])

zmeans <- apply(cluster.df.train,2,mean) #1:row; 2:column,normalize the whole dataset
zsds <- apply(cluster.df.train,2,sd)
Cluster_nor <- scale(cluster.df.train,center = zmeans, scale = zsds)
cluster_k = kmeans(Cluster_nor, centers = 10)

# add cluster label to all records and remove 'source_lat','source_lng','target_lat','target_lng'
library(flexclust)
cluster.kcca = as.kcca(cluster_k, Cluster_nor)

cluster.df.all = df_select_train[,c('source_lat','source_lng','target_lat','target_lng')]
zmeans.all <- apply(cluster.df.all,2,mean) #1:row; 2:column,normalize the whole dataset
zsds.all <- apply(cluster.df.all,2,sd)
Cluster_nor_all <- scale(cluster.df.all,center = zmeans.all, scale = zsds.all)
```

```
loc_Clusters_all = as.factor(predict(cluster.kcca, newdata = Cluster_nor_all))

select_cols_train_2 = c('action_type','level','weather_grade','source_type',
              'courier_wave_start_lng','courier_wave_start_lat',
              'speed','max_load',
              'grid_distance','urgency','hour','expected_use_time')
df_select_train_cluster = df_select_train[,select_cols_train_2][,1:11]
df_select_train_cluster$cluster = loc_Clusters_all
df.dataframe.cluster = data.frame(model.matrix(~.,df_select_train_cluster))
df.dataframe.cluster$expected_use_time = df_select_train$expected_use_time

df.train.cluster = df.dataframe.cluster[training.rows,]
df.test.cluster = df.dataframe.cluster[-training.rows,]
X.train.cluster = as.matrix(df.train.cluster[,1:42])
Y.train.cluster = df.train.cluster$expected_use_time
X.test.cluster = as.matrix(df.test.cluster[,1:42])
Y.test.cluster = df.test.cluster$expected_use_time

xgb_train_cluster = xgb.DMatrix(data = X.train.cluster, label = Y.train.cluster)
xgb_test_cluster = xgb.DMatrix(data = X.test.cluster, label = Y.test.cluster)

set.seed(100)
xgbt_cv2 = xgb.cv(data=xgb_train_cluster, nrounds = 30, early_stopping_rounds = 5, nfold=3, metrics =
'mae', showsd=FALSE,
              max_depth = 10, eta = 0.1, gamma = 0.001,lambda=1,colsample_bynode=0.8,
              objective = "reg:squarederror")
xgbt_cv2
set.seed(100)
model.xgbt.cluster = xgboost(data = xgb_train_cluster,nrounds = 21, max_depth = 10, eta = 0.1,
                 gamma = 0.001,lambda=1,colsample_bynode=0.8,
                 objective = "reg:squarederror",
                 eval_metric='mae')
xgb.importance(model=model.xgbt.cluster)
predict.test.xgb.cluster = predict(model.xgbt.cluster,X.test.cluster)
MAE(predict.test.xgb.cluster,Y.test.cluster)

# 1e. Predict Result
test_df =  read.csv("dataframe_test.csv")
regression_csv = read.csv("Regression.csv")

process_data_test = function(df){
  df$action_typePICKUP = 1-df$action_type_DELIVERY
  select_cols_test = c('action_typePICKUP','level','weather_grade','source_type',
              'courier_wave_start_lng','courier_wave_start_lat',
              'speed','max_load',
              'source_lng','source_lat','target_lng','target_lat',
              'grid_distance','urgency','hour')
  df$level = as.factor(df$level)
  df$weather_grade = as.factor(df$weather_grade)
  df$source_type = as.factor(df$source_type)
  df$hour = as.factor(df$hour)
```

```
  result_df = subset(df,select=select_cols_test)
  return(result_df)
}
df_select_result = process_data_test(test_df)
cluster.df.test = df_select_result[,c('source_lat','source_lng','target_lat','target_lng')]
zmeans.test <- apply(cluster.df.test,2,mean) #1:row; 2:column,normalize the whole dataset
zsds.test <- apply(cluster.df.test,2,sd)
Cluster_nor_test <- scale(cluster.df.test,center = zmeans.test, scale = zsds.test)
loc_Clusters_test = as.factor(predict(cluster.kcca, newdata = Cluster_nor_test))

select_cols_test_2 = c('action_typePICKUP','level','weather_grade','source_type',
                'courier_wave_start_lng','courier_wave_start_lat',
                'speed','max_load',
                'grid_distance','urgency','hour')
df_select_result_cluster = df_select_result[,select_cols_test_2]
df_select_result_cluster$cluster = loc_Clusters_test
df.result.cluster = data.frame(model.matrix(~.,df_select_result_cluster))
df.result.cluster$hour7 = 0
X.result.cluster = as.matrix(subset(df.result.cluster,select=colnames(X.test.cluster)))
pred.result = predict(model.xgbt.cluster, X.result.cluster)
regression_csv$expected_use_time = as.numeric(pred.result)

write.csv(regression_csv,file='Jing_Rongjia_Final_Project.csv',row.names = FALSE)
result  = read.csv("Jing_Rongjia_Final_Project.csv")
head(result)


### Question 2
# 2a. Balance check
library(foreign)
df1 = read.dta("oregonhie_descriptive_vars.dta")
head(df1)
df2 = read.dta("oregonhie_stateprograms_vars.dta")
head(df2)
df3 = read.dta("oregonhie_ed_vars.dta")
head(df3)

## Balance check on ED sample
# merge 3 df into all-df (N= 74922)
df3$label = 1
df_m2 = merge(df1, df2,
            by.x = "person_id",
            by.y = "person_id",
            all.x = T,
            all.y = F)
df_all = merge(df_m2, df3,
             by.x = "person_id",
             by.y = "person_id",
             all.x = T,
             all.y = F)
```

```
# Initial Data Pre-processing
df_all = transform(df_all, portland = ifelse(is.na(label),0,1))
df_all = transform(df_all, numhouse_1 = ifelse(numhh_list=="signed self up + 1 additional person",1,0))
df_all = transform(df_all, numhouse_2 = ifelse(numhh_list=="signed self up + 2 additional person",1,0))
df_all$treatment = ifelse(df_all$treatment=="Selected",1,0)

#OLS regression
OLS_portland = lm(portland ~treatment+numhouse_1+numhouse_2, data = df_all)
summary(OLS_portland)#p-value:0.68,cannot reject H0, balance

## Balance check on 5 variables
# merge 3 df into df-portland (N=24646)
df_m1 = merge(df3, df2,
          by.x = "person_id",
          by.y = "person_id",
          all.x = T,
          all.y = F)
df_port = merge(df_m1, df1,
            by.x = "person_id",
            by.y = "person_id",
            all.x = T,
            all.y = F)

# Initial Data Pre-processing
df_port = transform(df_port, numhouse_1 = ifelse(numhh_list=="signed self up + 1 additional
person",1,0))
df_port = transform(df_port, numhouse_2 = ifelse(numhh_list=="signed self up + 2 additional
person",1,0))
df_port = transform(df_port, gender = ifelse(female_list=="0: Male", 0, 1))
df_port = transform(df_port, selfsign = ifelse(self_list=="Signed self up", 1, 0))
df_port = transform(df_port, visit_ED = ifelse(any_visit_pre_ed=="Yes", 1, 0))
df_port$treatment = ifelse(df_port$treatment=="Selected",1,0)

#OLS regression
OLS_birthyear = lm(birthyear_list ~ treatment+numhouse_1+numhouse_2, data = df_port)
OLS_female = lm(gender ~ treatment+numhouse_1+numhouse_2, data = df_port)
OLS_selfsign = lm(selfsign ~ treatment+numhouse_1+numhouse_2, data = df_port)
OLS_visitED = lm(visit_ED ~ treatment+numhouse_1+numhouse_2, data = df_port)
OLS_numED = lm(num_visit_pre_cens_ed ~ treatment+numhouse_1+numhouse_2, data = df_port)
summary(OLS_birthyear)# p-value: 0.529, cannot reject H0
summary(OLS_female) #p-value:0.132, cannot reject H0
summary(OLS_selfsign) #p-value:0.383, cannot reject H0
summary(OLS_visitED) #p-value:0.583, cannot reject H0
summary(OLS_numED)#p-value:0.979,cannot reject H0

### 2b.Casual Effect
df_port = transform(df_port, enroll = ifelse(ohp_all_ever_firstn_30sep2009=="Enrolled", 1, 0))

OLS_enroll = lm(enroll ~ treatment+numhouse_1+numhouse_2
            +birthyear_list +gender +selfsign +visit_ED+ num_visit_pre_cens_ed, data = df_port)
summary(OLS_enroll) #ATE =0.2462176
```