Taxi Deployment

Table of Contents

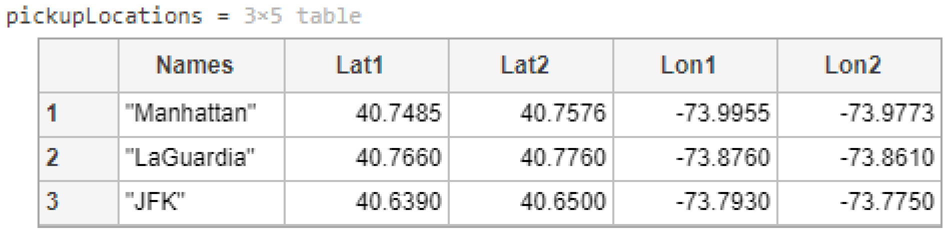
Data Extraction and Introduction 1
Data Exploration and Partitioning 2
    Visualization and Analysis 3
    Separate Test Data 4
Models Training and Validation 4
    Preprocessing 4
    Model Training 5
Model Testing and Evaluation 8
    Testing 8
    Evaluation 9
Model Application, Results, and Analysis 11
    Apply Model 11
    Use the Results to Allocate Fleet 20

Data Extraction and Introduction

To get started, navigate to the Taxi Project folder, and run the script **generateTaxiPickupTable.mlx**.

Note it may take a while to finish.

This will create (and save) two tables: `pickupLocations` and `taxiPickups.` A preview and description of each table is given below.



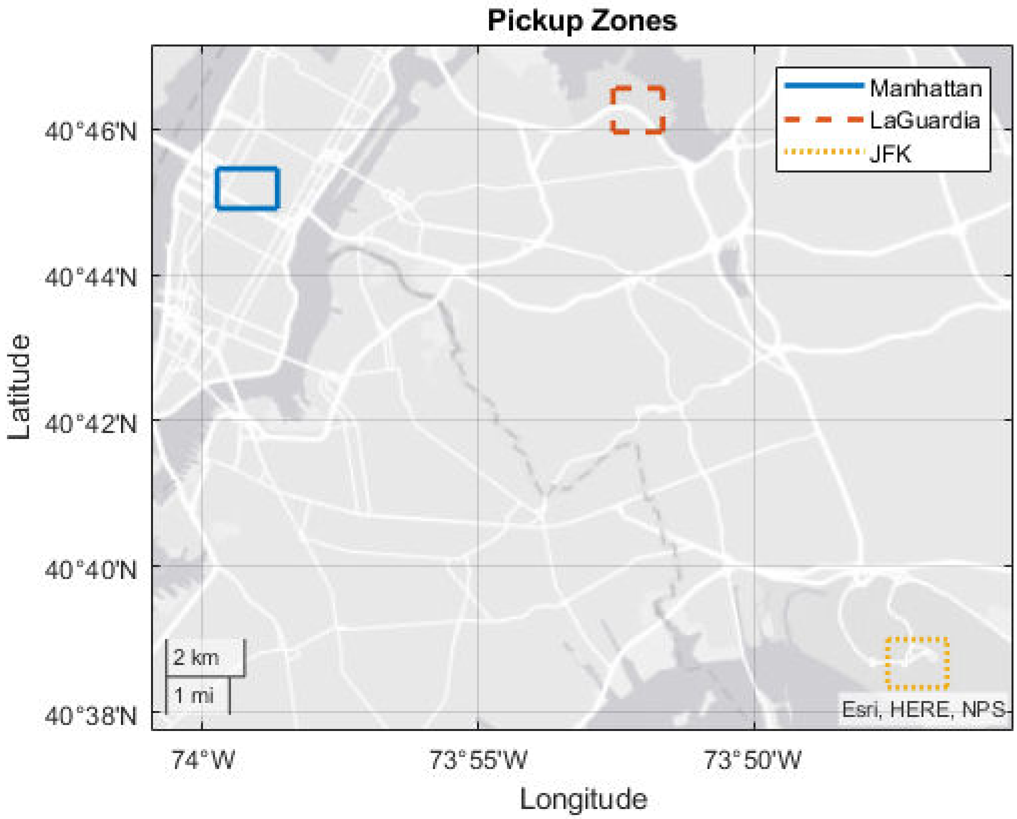
pickupLocations = 3×5 table

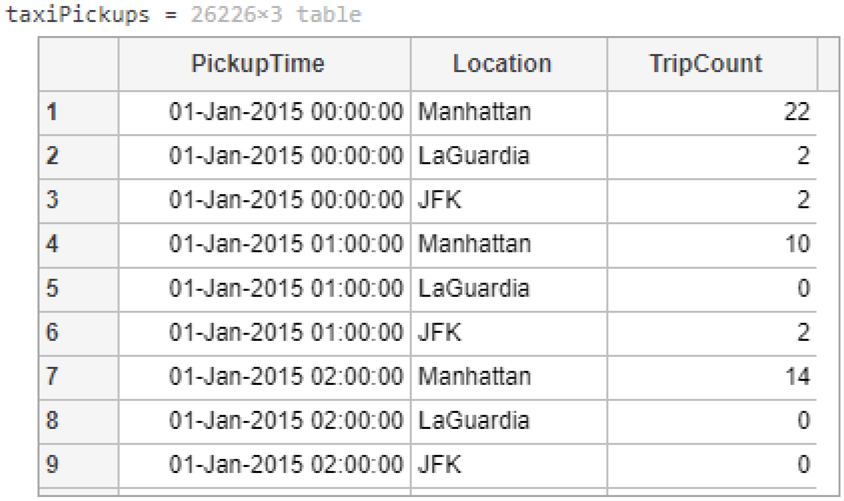
	Names	Lat1	Lat2	Lon1	Lon2
1	"Manhattan"	40.7485	40.7576	-73.9955	-73.9773
2	"LaGuardia"	40.7660	40.7760	-73.8760	-73.8610
3	"JFK"	40.6390	40.6500	-73.7930	-73.7750

The table `pickupLocations` gives the latitude and logitude bounds for three pickup zones:

- **Manhattan**: here meaning an area of high taxi traffic surrounding Penn Station, Grand Central Station, and the Port Authority Bus Terminal
- **LaGuardia**: meaning an area surrounding LaGuardia airport
- **JFK**: similarly meaning an area surrounding JFK airport.

The zones are shown below for reference.





```
taxiPickups = 26226×3 table
```

	PickupTime	Location	TripCount
1	01-Jan-2015 00:00:00	Manhattan	22
2	01-Jan-2015 00:00:00	LaGuardia	2
3	01-Jan-2015 00:00:00	JFK	2
4	01-Jan-2015 01:00:00	Manhattan	10
5	01-Jan-2015 01:00:00	LaGuardia	0
6	01-Jan-2015 01:00:00	JFK	2
7	01-Jan-2015 02:00:00	Manhattan	14
8	01-Jan-2015 02:00:00	LaGuardia	0
9	01-Jan-2015 02:00:00	JFK	0

The table `taxiPickpus` represents the number of pickups in your data over one hour intervals of 2015 in the zones defined in `pickupLocations`. The start of each hour is specified in the `PickupTime` datetime variable, the zone is specified by the `Location` categorical variable, and the number of pickups is given by the `TripCount`.

Data Exploration and Partitioning

The script **generateTaxiPickupTable.mlx** in the previous section saved copies of the tables to a MAT file named **taxiPickupData.mat**. To avoid having to run the script again if you continue working after clearing your workspace, the code below loads the saved data.

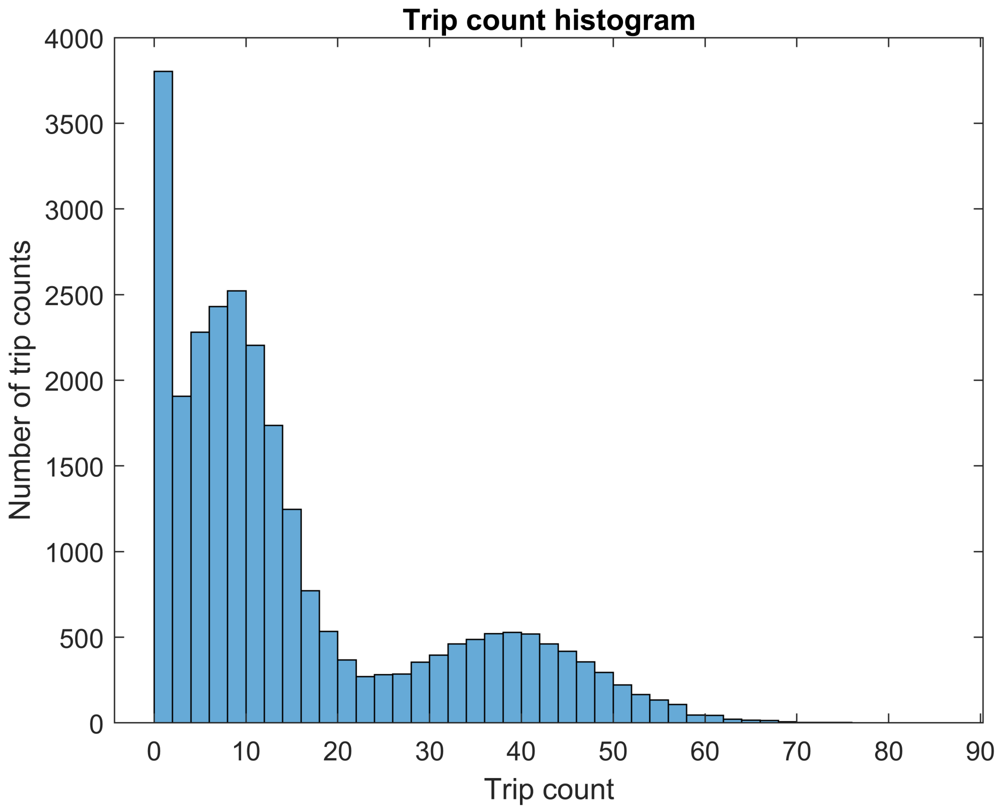
```
% run 'Module 3'\'Taxi Project'\generateTaxiPickupTable.mlx
clear
% Make sure to follow the instructions in the previous section
if ~isempty(which('-all','taxiPickupData.mat'))
    load taxiPickupData.mat
else
    error("The file taxiPickupData.mat is not found on the MATLAB path. Add it to the path or
end
```

Visualization and Analysis

Analyze the data in the `taxiPickups` table. At a minimum, provide a visualization of the distribution (histogram) of the response variable `TripCount`, as well as a box plot for `TripCount` grouped by `Location`.

```
histogram(taxiPickups.TripCount)

title('Trip count histogram')
xlabel('Trip count')
ylabel('Number of trip counts')
```



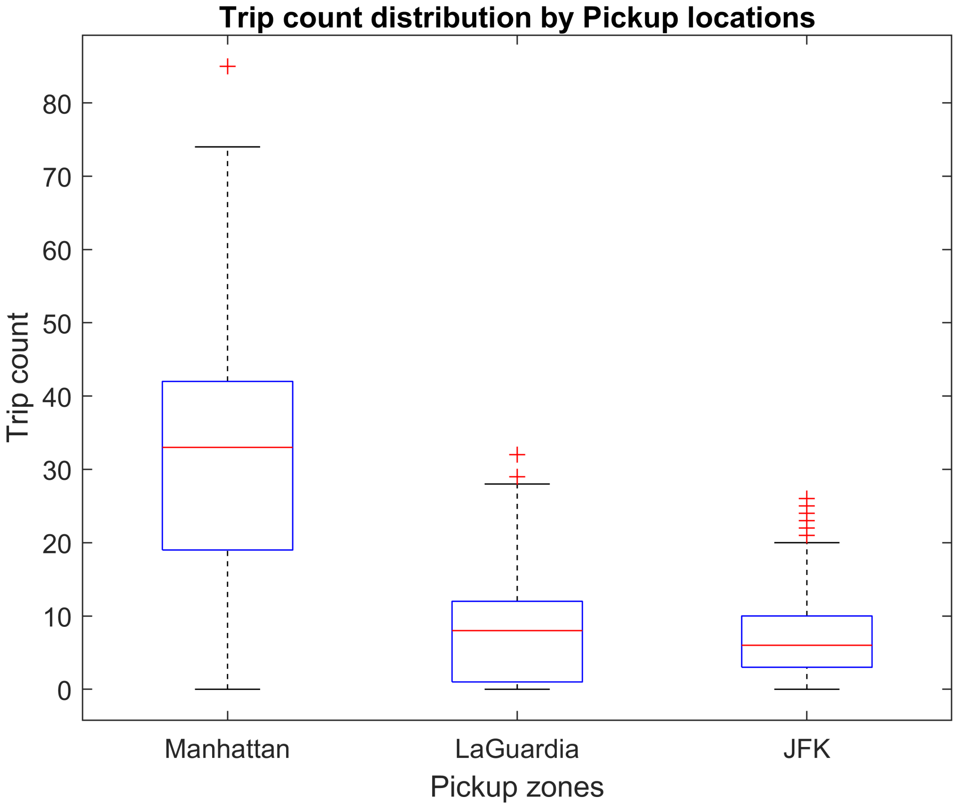
```
grpLocation = groupsummary(taxiPickups,"Location","sum","TripCount")
```

grpLocation = 3×3 table

	Location	GroupCount	sum_TripCount
1	Manhattan	8739	269250
2	LaGuardia	8739	66729

	Location	GroupCount	sum_TripCount
3	JFK	8739	56970

```
boxplot(taxiPickups.TripCount,taxiPickups.Location)
title('Trip count distribution by Pickup locations')
xlabel('Pickup zones')
ylabel('Trip count')
```



Separate Test Data

Use cvpartition to separate 20% of the data set for testing later on, and create the training data. Ensure your results are repeatable by setting the random number generator seed to 10. Provide your code.

```
rng(10)
taxiPartitions = cvpartition(height(taxiPickups),"HoldOut",0.2);

taxiPickupsTrain = taxiPickups(training(taxiPartitions),:);
taxiPickupsTest = taxiPickups(test(taxiPartitions),:);
```

Models Training and Validation

Preprocessing

As the focus of this course is machine learning, we've provided a function to do some feature engineering for you. Use **providedPreprocessing.mlx** to add the following features to your training/validation data set:

- `TimeOfDay` (numerical)
- `DayOfWeek` (categorical)
- `DayOfMonth` (numerical)
- `DayOfYear` (numerical)

For example:

```
taxiPickupsTrain = providedPreprocessing(taxiPickupsTrain)
```

```
taxiPickupsTrain = providedPreprocessing(taxiPickupsTrain);
taxiPickupsTest = providedPreprocessing(taxiPickupsTest);
% regressionLearner
```

Model Training

Train models to predict `TripCount` using the processed test/validation data. **Report your validation approach and validation $RMSE$ for your best model**. Your goal will be to get a validation $RMSE$ at or below 4.9. Include code so that your script can reproduce your final model, including the model training. **If using the app, export the training function** and include a correct call to it in your script. Note you do not need to include the generated training function code itself in your script, just a correct call to it. For example, if you'd trained a tree model in the app and exported the training function, you could include:

```
[modelStruct, validationRMSE] = trainRegressionModel(taxiPickupsTrain)
myModel = modelStruct.RegressionTree
```

```
[modelStruct, validationRMSE] = trainRegressionModel(taxiPickupsTrain)
```

```
modelStruct = struct with fields:
            predictFcn: @(x)ensemblePredictFcn(predictorExtractionFcn(x))
     RequiredVariables: {'DayOfMonth'  'DayOfWeek'  'DayOfYear'  'Location'  'TimeOfDay'}
    RegressionEnsemble: [1×1 classreg.learning.regr.RegressionBaggedEnsemble]
                 About: 'This struct is a trained model exported from Regression Learner R2021b.'
           HowToPredict: 'To make predictions on a new table, T, use: ↲  yfit = c.predictFcn(T) ↲replacing 'c' with t
validationRMSE = 4.6200
```

```
myModel = modelStruct.RegressionEnsemble
```

```
myModel = 
  RegressionBaggedEnsemble
           PredictorNames: {'Location'  'TimeOfDay'  'DayOfWeek'  'DayOfMonth'  'DayOfYear'}
             ResponseName: 'Y'
    CategoricalPredictors: [1 3]
        ResponseTransform: 'none'
          NumObservations: 20974
               NumTrained: 206
                   Method: 'Bag'
             LearnerNames: {'Tree'}
     ReasonForTermination: 'Terminated normally after completing the requested number of training cycles.'
                  FitInfo: []
       FitInfoDescription: 'None'
           Regularization: []
                FResample: 1
                  Replace: 1
         UseObsForLearner: [20974×206 logical]
    Properties, Methods
```

The code below shows Regression model function after using Regression Learner app:

```
function [trainedModel, validationRMSE] = trainRegressionModel(trainingData)
% [trainedModel, validationRMSE] = trainRegressionModel(trainingData)
% Returns a trained regression model and its RMSE. This code recreates the
% model trained in Regression Learner app. Use the generated code to
% automate training the same model with new data, or to learn how to
% programmatically train models.
%
%  Input:
%      trainingData: A table containing the same predictor and response
%       columns as those imported into the app.
%
%  Output:
%      trainedModel: A struct containing the trained regression model. The
%       struct contains various fields with information about the trained
%       model.
%
%      trainedModel.predictFcn: A function to make predictions on new data.
%
%      validationRMSE: A double containing the RMSE. In the app, the Models
%       pane displays the RMSE for each model.
%
% Use the code to train the model with new data. To retrain your model,
% call the function from the command line with your original data or new
% data as the input argument trainingData.
%
% For example, to retrain a regression model trained with the original data
% set T, enter:
%   [trainedModel, validationRMSE] = trainRegressionModel(T)
%
% To make predictions with the returned 'trainedModel' on new data T2, use
%   yfit = trainedModel.predictFcn(T2)
%
% T2 must be a table containing at least the same predictor columns as used
% during training. For details, enter:
%   trainedModel.HowToPredict

% Auto-generated by MATLAB on 05-Feb-2023 21:54:17


% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'Location', 'TimeOfDay', 'DayOfWeek', 'DayOfMonth', 'DayOfYear'};
predictors = inputTable(:, predictorNames);
response = inputTable.TripCount;
isCategoricalPredictor = [true, false, true, false, false];

% Train a regression model
% This code specifies all the model options and trains the model.
template = templateTree(...
    'MinLeafSize', 1, ...
```

```
            'NumVariablesToSample', 3);
        regressionEnsemble = fitrensemble(...
            predictors, ...
            response, ...
            'Method', 'Bag', ...
            'NumLearningCycles', 206, ...
            'Learners', template);

        % Create the result struct with predict function
        predictorExtractionFcn = @(t) t(:, predictorNames);
        ensemblePredictFcn = @(x) predict(regressionEnsemble, x);
        trainedModel.predictFcn = @(x) ensemblePredictFcn(predictorExtractionFcn(x));

        % Add additional fields to the result struct
        trainedModel.RequiredVariables = {'DayOfMonth', 'DayOfWeek', 'DayOfYear', 'Location', 'TimeOfDay'};
        trainedModel.RegressionEnsemble = regressionEnsemble;
        trainedModel.About = 'This struct is a trained model exported from Regression Learner R2021b.';
        trainedModel.HowToPredict = sprintf('To make predictions on a new table, T, use: \n  yfit = c.predictFcn(T) \n

        % Extract predictors and response
        % This code processes the data into the right shape for training the
        % model.
        inputTable = trainingData;
        predictorNames = {'Location', 'TimeOfDay', 'DayOfWeek', 'DayOfMonth', 'DayOfYear'};
        predictors = inputTable(:, predictorNames);
        response = inputTable.TripCount;
        isCategoricalPredictor = [true, false, true, false, false];

        % Perform cross-validation
        partitionedModel = crossval(trainedModel.RegressionEnsemble, 'KFold', 5);

        % Compute validation predictions
        validationPredictions = kfoldPredict(partitionedModel);

        % Compute validation RMSE
        validationRMSE = sqrt(kfoldLoss(partitionedModel, 'LossFun', 'mse'));
```

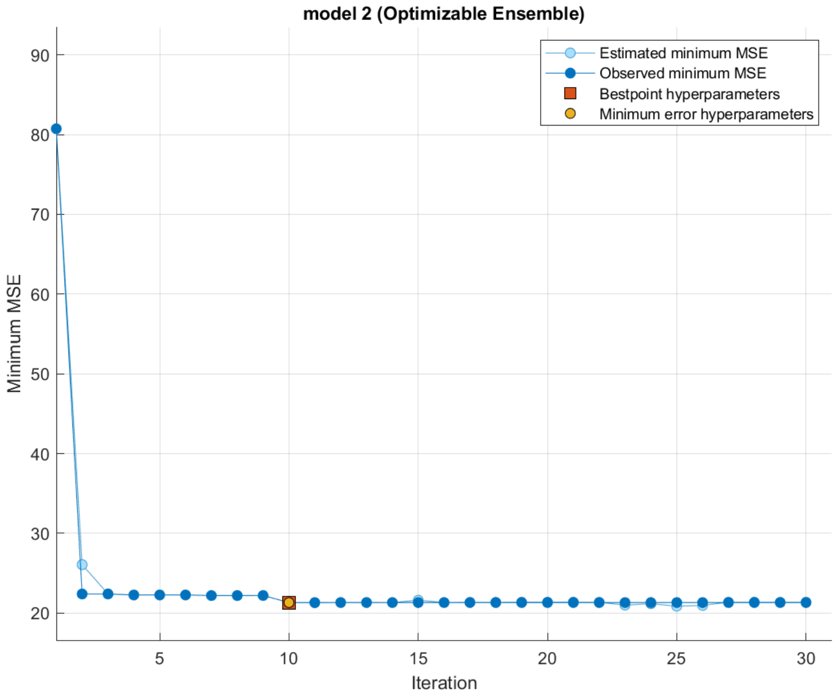
Validation $RMSE$:

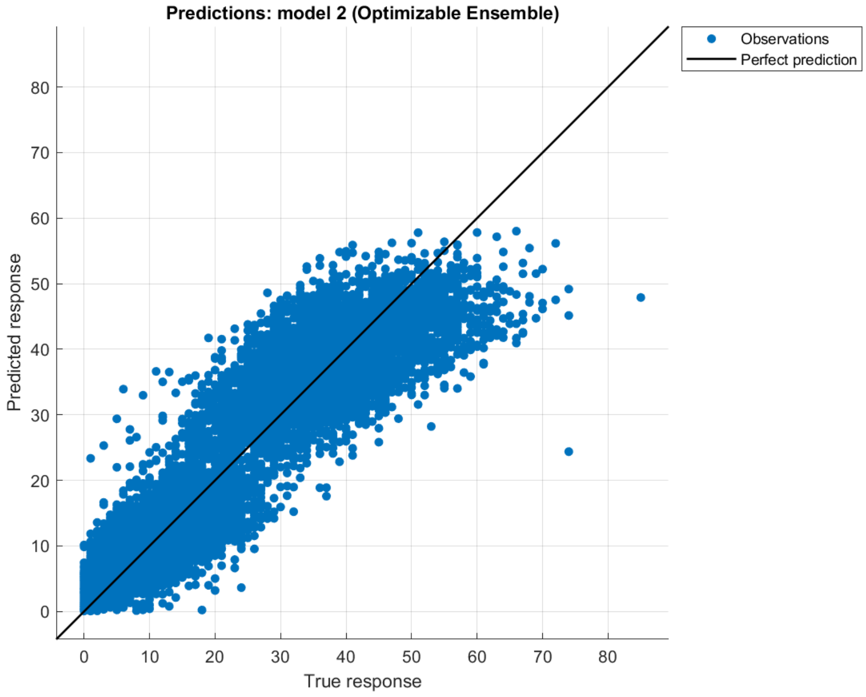
```
validationRMSE
```

```
validationRMSE = 4.6200
```

Validation Method: Cross-validation, 5 folds

The figures below shows parameter optimization of bagged ensemble model using Regression Learner App.





Model Testing and Evaluation

Testing

Preprocess the test data as needed, and use it to test your best model. Provide your code and report at least the $RMSE$ and R^2. You will need to achieve a test $RMSE$ at or below 4.9 to receive full points here.

```
rmse(yActual,yPredict), Rsq(yActual,yPredict)
```

```
ans = 1×1 table
```

	rmse
1	4.5576

```
ans = 1×1 table
```

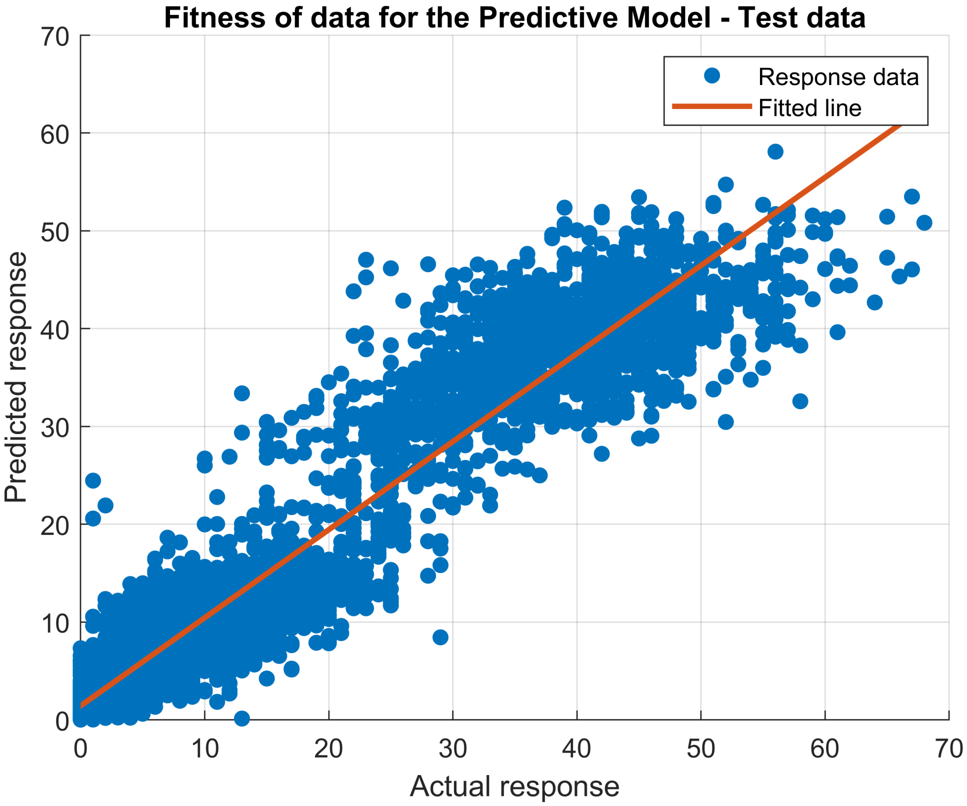
	Rsq
1	0.9073

Evaluation

Discuss the results from training and testing. How well did your model generalize to new data? Include at least a plot of the residuals, and discuss your observations from the plot(s).

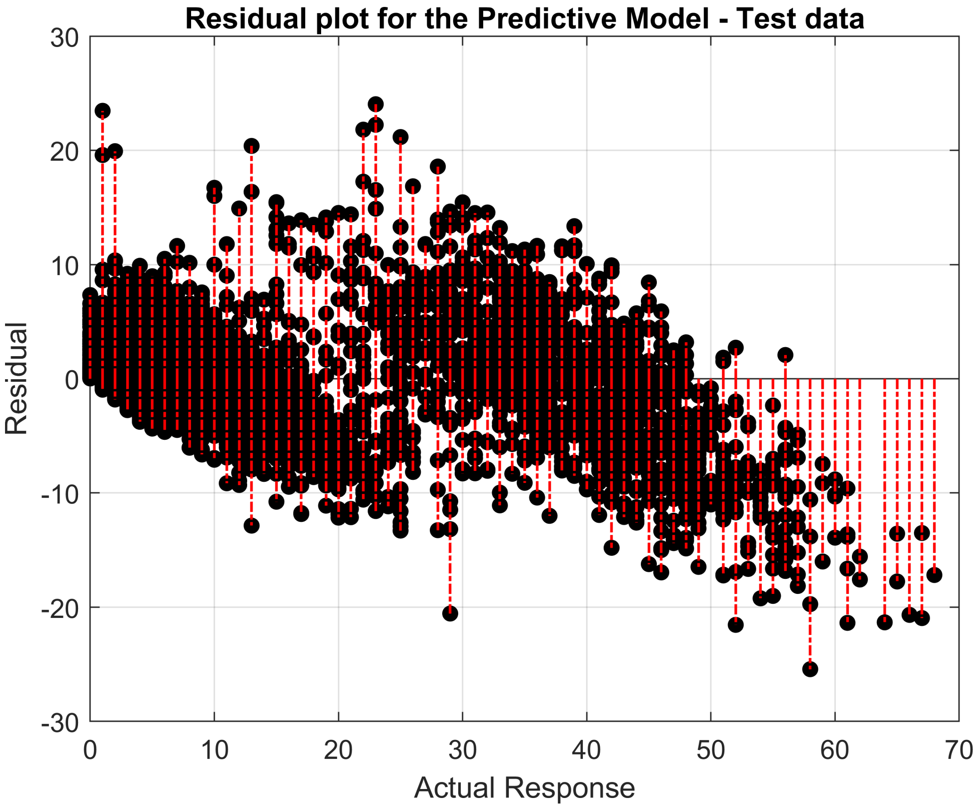
```
yPredict = predict(myModel,taxiPickupsTest);
yActual = taxiPickupsTest.TripCount;
yLine = polyval(polyfit(yActual,yPredict,1),yActual);
clf, hold on
scatter(yActual,yPredict,"filled","o")
plot(yActual,yLine,LineWidth=2)
hold off

grid on
legend({'Response data','Fitted line'})
title('Fitness of data for the Predictive Model - Test data')
xlabel('Actual response')
ylabel('Predicted response')
```



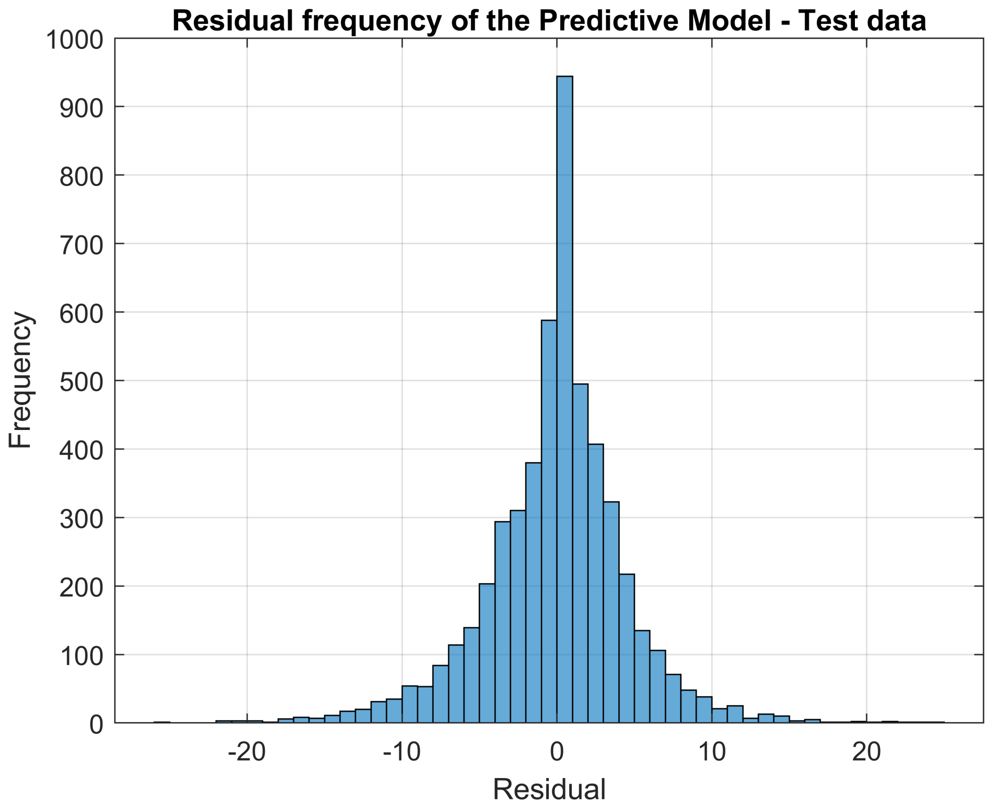
```
stem(yActual,yPredict-yActual,'LineStyle','-.','LineWidth',1,'color','red',...
     'MarkerFaceColor','black','MarkerEdgeColor','none')
```

```
grid on
title('Residual plot for the Predictive Model - Test data')
xlabel('Actual Response')
ylabel('Residual')
```



```
histogram(yPredict-yActual)

grid on
title('Residual frequency of the Predictive Model - Test data')
xlabel('Residual')
ylabel('Frequency')
```



The model predicts accurately within specification using the test data.

Model Application, Results, and Analysis

Apply Model

As the focus of this course is machine learning, we've provided some skeleton code in this section. Below, we create a new table of staring features for 2016.

Note that you will need to use the variable names provided in comments or make your own additional edits to the code.

```
taxiPickups2016 = table;
taxiPickups2016.PickupTime = taxiPickups.PickupTime + years(1);
taxiPickups2016.Location = taxiPickups.Location;
taxiPickups2016 = providedPreprocessing(taxiPickups2016);
% Display only the first 8 rows of the table
head(taxiPickups2016)
```

```
ans = 8×6 table
```

	PickupTime	Location	TimeOfDay	DayOfWeek	DayOfMonth	DayOfYear
1	2016-01-01 05:...	Manhattan	5.8200	Friday	1	1
2	2016-01-01 05:...	LaGuardia	5.8200	Friday	1	1
3	2016-01-01 05:...	JFK	5.8200	Friday	1	1
4	2016-01-01 06:...	Manhattan	6.8200	Friday	1	1

	PickupTime	Location	TimeOfDay	DayOfWeek	DayOfMonth	DayOfYear
5	2016-01-01 06:...	LaGuardia	6.8200	Friday	1	1
6	2016-01-01 06:...	JFK	6.8200	Friday	1	1
7	2016-01-01 07:...	Manhattan	7.8200	Friday	1	1
8	2016-01-01 07:...	LaGuardia	7.8200	Friday	1	1

Choose your favorite day in 2016 and edit the variable `myDay` below which will be used to extract that day from the table.

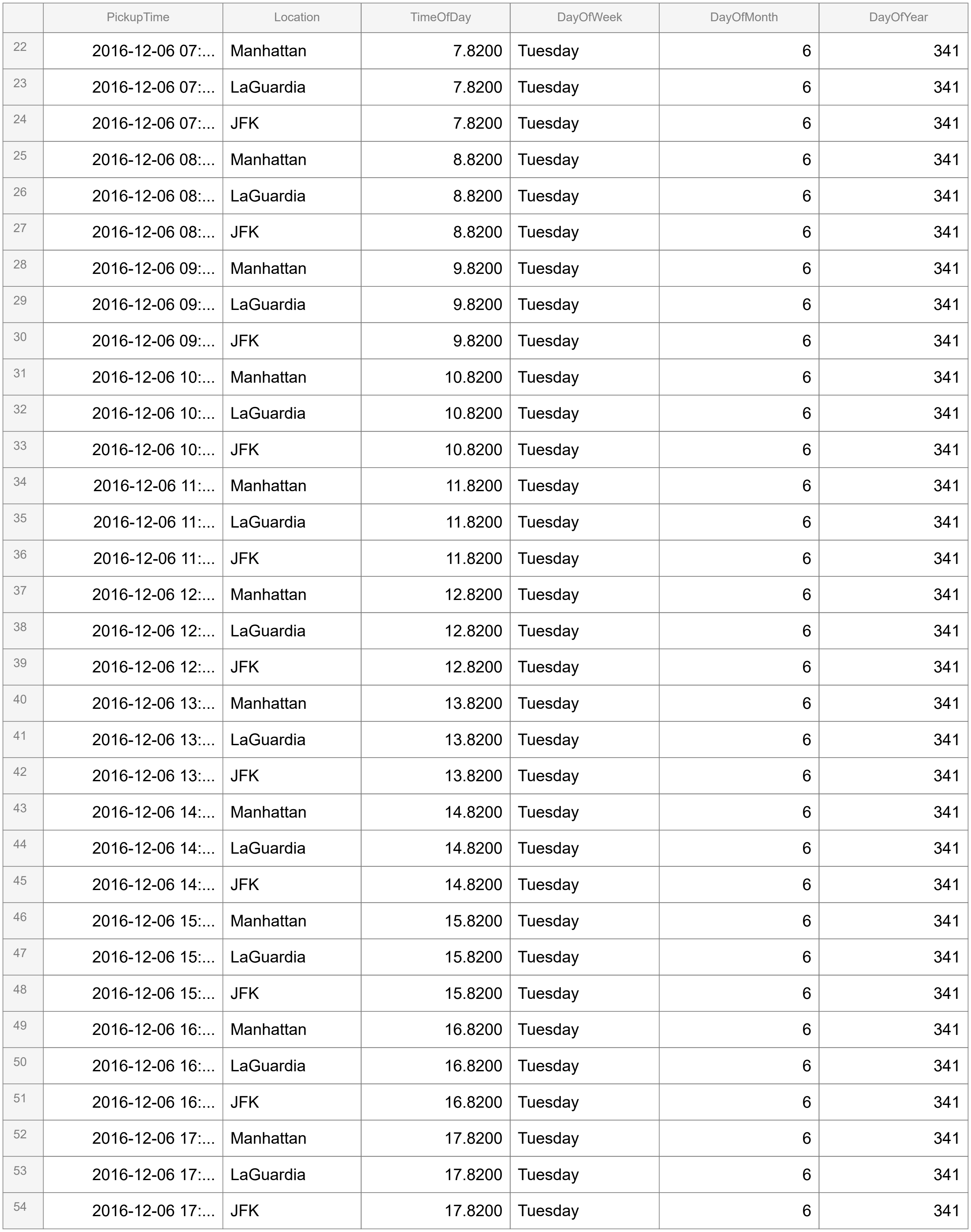
```
myDay = datetime("2016-12-6")
```

```
myDay = datetime
   06-Dec-2016
```

```
taxiPickupsMyDay = taxiPickups2016(day(taxiPickups2016.PickupTime,"dayofyear") == day(myDay,"da
```

```
taxiPickupsMyDay = 72×6 table
```

	PickupTime	Location	TimeOfDay	DayOfWeek	DayOfMonth	DayOfYear
1	2016-12-06 00:...	Manhattan	0.8200	Tuesday	6	341
2	2016-12-06 00:...	LaGuardia	0.8200	Tuesday	6	341
3	2016-12-06 00:...	JFK	0.8200	Tuesday	6	341
4	2016-12-06 01:...	Manhattan	1.8200	Tuesday	6	341
5	2016-12-06 01:...	LaGuardia	1.8200	Tuesday	6	341
6	2016-12-06 01:...	JFK	1.8200	Tuesday	6	341
7	2016-12-06 02:...	Manhattan	2.8200	Tuesday	6	341
8	2016-12-06 02:...	LaGuardia	2.8200	Tuesday	6	341
9	2016-12-06 02:...	JFK	2.8200	Tuesday	6	341
10	2016-12-06 03:...	Manhattan	3.8200	Tuesday	6	341
11	2016-12-06 03:...	LaGuardia	3.8200	Tuesday	6	341
12	2016-12-06 03:...	JFK	3.8200	Tuesday	6	341
13	2016-12-06 04:...	Manhattan	4.8200	Tuesday	6	341
14	2016-12-06 04:...	LaGuardia	4.8200	Tuesday	6	341
15	2016-12-06 04:...	JFK	4.8200	Tuesday	6	341
16	2016-12-06 05:...	Manhattan	5.8200	Tuesday	6	341
17	2016-12-06 05:...	LaGuardia	5.8200	Tuesday	6	341
18	2016-12-06 05:...	JFK	5.8200	Tuesday	6	341
19	2016-12-06 06:...	Manhattan	6.8200	Tuesday	6	341
20	2016-12-06 06:...	LaGuardia	6.8200	Tuesday	6	341
21	2016-12-06 06:...	JFK	6.8200	Tuesday	6	341



	PickupTime	Location	TimeOfDay	DayOfWeek	DayOfMonth	DayOfYear
22	2016-12-06 07:...	Manhattan	7.8200	Tuesday	6	341
23	2016-12-06 07:...	LaGuardia	7.8200	Tuesday	6	341
24	2016-12-06 07:...	JFK	7.8200	Tuesday	6	341
25	2016-12-06 08:...	Manhattan	8.8200	Tuesday	6	341
26	2016-12-06 08:...	LaGuardia	8.8200	Tuesday	6	341
27	2016-12-06 08:...	JFK	8.8200	Tuesday	6	341
28	2016-12-06 09:...	Manhattan	9.8200	Tuesday	6	341
29	2016-12-06 09:...	LaGuardia	9.8200	Tuesday	6	341
30	2016-12-06 09:...	JFK	9.8200	Tuesday	6	341
31	2016-12-06 10:...	Manhattan	10.8200	Tuesday	6	341
32	2016-12-06 10:...	LaGuardia	10.8200	Tuesday	6	341
33	2016-12-06 10:...	JFK	10.8200	Tuesday	6	341
34	2016-12-06 11:...	Manhattan	11.8200	Tuesday	6	341
35	2016-12-06 11:...	LaGuardia	11.8200	Tuesday	6	341
36	2016-12-06 11:...	JFK	11.8200	Tuesday	6	341
37	2016-12-06 12:...	Manhattan	12.8200	Tuesday	6	341
38	2016-12-06 12:...	LaGuardia	12.8200	Tuesday	6	341
39	2016-12-06 12:...	JFK	12.8200	Tuesday	6	341
40	2016-12-06 13:...	Manhattan	13.8200	Tuesday	6	341
41	2016-12-06 13:...	LaGuardia	13.8200	Tuesday	6	341
42	2016-12-06 13:...	JFK	13.8200	Tuesday	6	341
43	2016-12-06 14:...	Manhattan	14.8200	Tuesday	6	341
44	2016-12-06 14:...	LaGuardia	14.8200	Tuesday	6	341
45	2016-12-06 14:...	JFK	14.8200	Tuesday	6	341
46	2016-12-06 15:...	Manhattan	15.8200	Tuesday	6	341
47	2016-12-06 15:...	LaGuardia	15.8200	Tuesday	6	341
48	2016-12-06 15:...	JFK	15.8200	Tuesday	6	341
49	2016-12-06 16:...	Manhattan	16.8200	Tuesday	6	341
50	2016-12-06 16:...	LaGuardia	16.8200	Tuesday	6	341
51	2016-12-06 16:...	JFK	16.8200	Tuesday	6	341
52	2016-12-06 17:...	Manhattan	17.8200	Tuesday	6	341
53	2016-12-06 17:...	LaGuardia	17.8200	Tuesday	6	341
54	2016-12-06 17:...	JFK	17.8200	Tuesday	6	341

	PickupTime	Location	TimeOfDay	DayOfWeek	DayOfMonth	DayOfYear
55	2016-12-06 18:...	Manhattan	18.8200	Tuesday	6	341
56	2016-12-06 18:...	LaGuardia	18.8200	Tuesday	6	341
57	2016-12-06 18:...	JFK	18.8200	Tuesday	6	341
58	2016-12-06 19:...	Manhattan	19.8200	Tuesday	6	341
59	2016-12-06 19:...	LaGuardia	19.8200	Tuesday	6	341
60	2016-12-06 19:...	JFK	19.8200	Tuesday	6	341
61	2016-12-06 20:...	Manhattan	20.8200	Tuesday	6	341
62	2016-12-06 20:...	LaGuardia	20.8200	Tuesday	6	341
63	2016-12-06 20:...	JFK	20.8200	Tuesday	6	341
64	2016-12-06 21:...	Manhattan	21.8200	Tuesday	6	341
65	2016-12-06 21:...	LaGuardia	21.8200	Tuesday	6	341
66	2016-12-06 21:...	JFK	21.8200	Tuesday	6	341
67	2016-12-06 22:...	Manhattan	22.8200	Tuesday	6	341
68	2016-12-06 22:...	LaGuardia	22.8200	Tuesday	6	341
69	2016-12-06 22:...	JFK	22.8200	Tuesday	6	341
70	2016-12-06 23:...	Manhattan	23.8200	Tuesday	6	341
71	2016-12-06 23:...	LaGuardia	23.8200	Tuesday	6	341
72	2016-12-06 23:...	JFK	23.8200	Tuesday	6	341

Now, use your best model to predict `TripCount` on the day you've chosen and add it to the table.

```
taxiPickupsMyDay.TripCount = predict(myModel,taxiPickupsMyDay);
```

Again, to focus on machine learning, we have provided the necessary table manipulations and calculations below to use your model predictions to give the predicted fraction of trips happening in each hour on your selected day.

Uncomment below once you have defined the table `taxiPickupsMyDay` above.

```
taxiPickupsMyDayTotals = groupsummary(taxiPickupsMyDay,"PickupTime","sum","TripCount");
taxiPickupsMyDay = join(taxiPickupsMyDay,taxiPickupsMyDayTotals,"RightVariables","sum_TripCount
```

```
taxiPickupsMyDay = 72×8 table
```

...

	PickupTime	Location	TimeOfDay	DayOfWeek	DayOfMonth	DayOfYear
1	2016-12-06 00:...	Manhattan	0.8200	Tuesday	6	341
2	2016-12-06 00:...	LaGuardia	0.8200	Tuesday	6	341
3	2016-12-06 00:...	JFK	0.8200	Tuesday	6	341
4	2016-12-06 01:...	Manhattan	1.8200	Tuesday	6	341

	PickupTime	Location	TimeOfDay	DayOfWeek	DayOfMonth	DayOfYear
5	2016-12-06 01:...	LaGuardia	1.8200	Tuesday	6	341
6	2016-12-06 01:...	JFK	1.8200	Tuesday	6	341
7	2016-12-06 02:...	Manhattan	2.8200	Tuesday	6	341
8	2016-12-06 02:...	LaGuardia	2.8200	Tuesday	6	341
9	2016-12-06 02:...	JFK	2.8200	Tuesday	6	341
10	2016-12-06 03:...	Manhattan	3.8200	Tuesday	6	341
11	2016-12-06 03:...	LaGuardia	3.8200	Tuesday	6	341
12	2016-12-06 03:...	JFK	3.8200	Tuesday	6	341
13	2016-12-06 04:...	Manhattan	4.8200	Tuesday	6	341
14	2016-12-06 04:...	LaGuardia	4.8200	Tuesday	6	341
15	2016-12-06 04:...	JFK	4.8200	Tuesday	6	341
16	2016-12-06 05:...	Manhattan	5.8200	Tuesday	6	341
17	2016-12-06 05:...	LaGuardia	5.8200	Tuesday	6	341
18	2016-12-06 05:...	JFK	5.8200	Tuesday	6	341
19	2016-12-06 06:...	Manhattan	6.8200	Tuesday	6	341
20	2016-12-06 06:...	LaGuardia	6.8200	Tuesday	6	341
21	2016-12-06 06:...	JFK	6.8200	Tuesday	6	341
22	2016-12-06 07:...	Manhattan	7.8200	Tuesday	6	341
23	2016-12-06 07:...	LaGuardia	7.8200	Tuesday	6	341
24	2016-12-06 07:...	JFK	7.8200	Tuesday	6	341
25	2016-12-06 08:...	Manhattan	8.8200	Tuesday	6	341
26	2016-12-06 08:...	LaGuardia	8.8200	Tuesday	6	341
27	2016-12-06 08:...	JFK	8.8200	Tuesday	6	341
28	2016-12-06 09:...	Manhattan	9.8200	Tuesday	6	341
29	2016-12-06 09:...	LaGuardia	9.8200	Tuesday	6	341
30	2016-12-06 09:...	JFK	9.8200	Tuesday	6	341
31	2016-12-06 10:...	Manhattan	10.8200	Tuesday	6	341
32	2016-12-06 10:...	LaGuardia	10.8200	Tuesday	6	341
33	2016-12-06 10:...	JFK	10.8200	Tuesday	6	341
34	2016-12-06 11:...	Manhattan	11.8200	Tuesday	6	341
35	2016-12-06 11:...	LaGuardia	11.8200	Tuesday	6	341
36	2016-12-06 11:...	JFK	11.8200	Tuesday	6	341
37	2016-12-06 12:...	Manhattan	12.8200	Tuesday	6	341

	PickupTime	Location	TimeOfDay	DayOfWeek	DayOfMonth	DayOfYear
38	2016-12-06 12:...	LaGuardia	12.8200	Tuesday	6	341
39	2016-12-06 12:...	JFK	12.8200	Tuesday	6	341
40	2016-12-06 13:...	Manhattan	13.8200	Tuesday	6	341
41	2016-12-06 13:...	LaGuardia	13.8200	Tuesday	6	341
42	2016-12-06 13:...	JFK	13.8200	Tuesday	6	341
43	2016-12-06 14:...	Manhattan	14.8200	Tuesday	6	341
44	2016-12-06 14:...	LaGuardia	14.8200	Tuesday	6	341
45	2016-12-06 14:...	JFK	14.8200	Tuesday	6	341
46	2016-12-06 15:...	Manhattan	15.8200	Tuesday	6	341
47	2016-12-06 15:...	LaGuardia	15.8200	Tuesday	6	341
48	2016-12-06 15:...	JFK	15.8200	Tuesday	6	341
49	2016-12-06 16:...	Manhattan	16.8200	Tuesday	6	341
50	2016-12-06 16:...	LaGuardia	16.8200	Tuesday	6	341
51	2016-12-06 16:...	JFK	16.8200	Tuesday	6	341
52	2016-12-06 17:...	Manhattan	17.8200	Tuesday	6	341
53	2016-12-06 17:...	LaGuardia	17.8200	Tuesday	6	341
54	2016-12-06 17:...	JFK	17.8200	Tuesday	6	341
55	2016-12-06 18:...	Manhattan	18.8200	Tuesday	6	341
56	2016-12-06 18:...	LaGuardia	18.8200	Tuesday	6	341
57	2016-12-06 18:...	JFK	18.8200	Tuesday	6	341
58	2016-12-06 19:...	Manhattan	19.8200	Tuesday	6	341
59	2016-12-06 19:...	LaGuardia	19.8200	Tuesday	6	341
60	2016-12-06 19:...	JFK	19.8200	Tuesday	6	341
61	2016-12-06 20:...	Manhattan	20.8200	Tuesday	6	341
62	2016-12-06 20:...	LaGuardia	20.8200	Tuesday	6	341
63	2016-12-06 20:...	JFK	20.8200	Tuesday	6	341
64	2016-12-06 21:...	Manhattan	21.8200	Tuesday	6	341
65	2016-12-06 21:...	LaGuardia	21.8200	Tuesday	6	341
66	2016-12-06 21:...	JFK	21.8200	Tuesday	6	341
67	2016-12-06 22:...	Manhattan	22.8200	Tuesday	6	341
68	2016-12-06 22:...	LaGuardia	22.8200	Tuesday	6	341
69	2016-12-06 22:...	JFK	22.8200	Tuesday	6	341
70	2016-12-06 23:...	Manhattan	23.8200	Tuesday	6	341

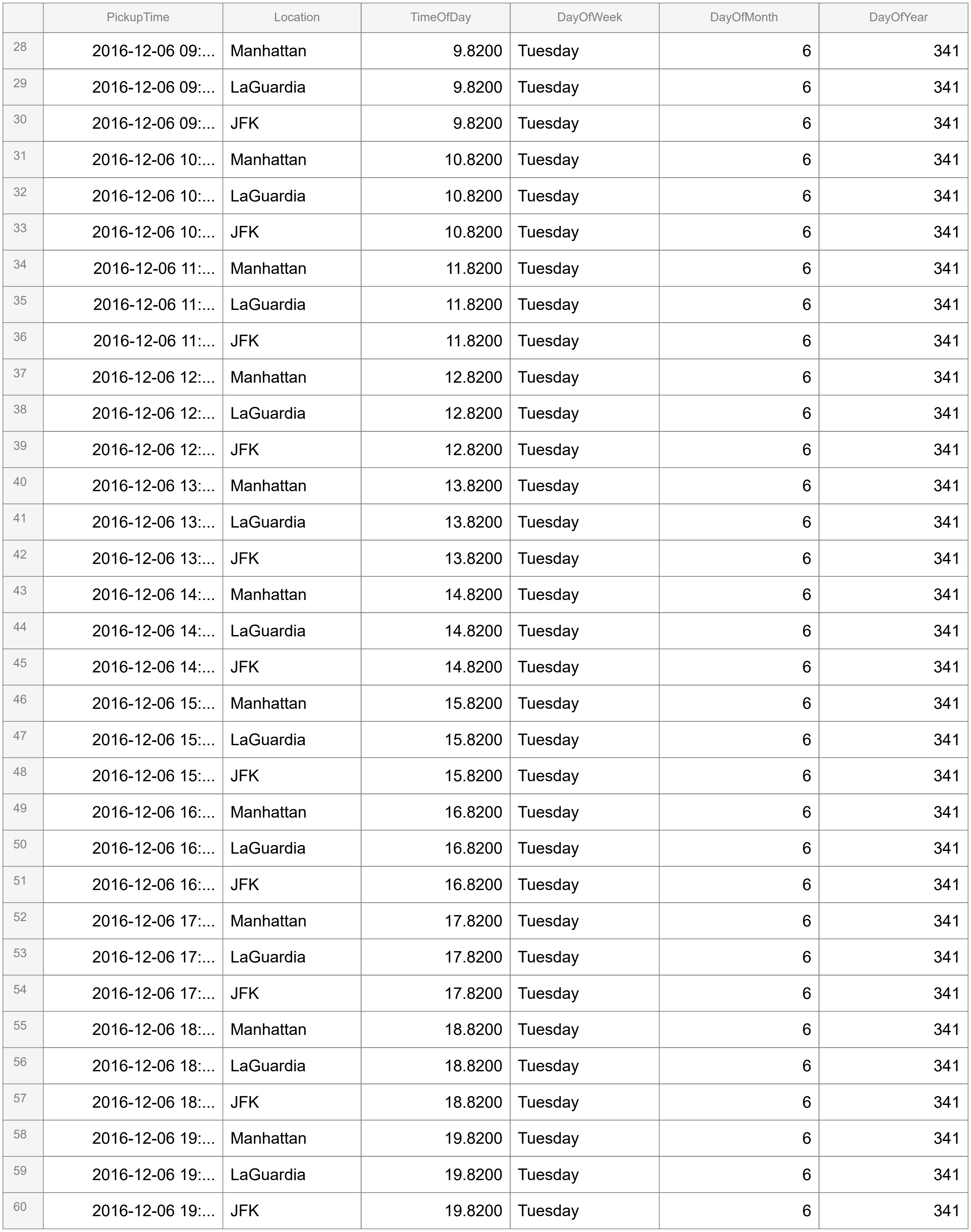
	PickupTime	Location	TimeOfDay	DayOfWeek	DayOfMonth	DayOfYear
71	2016-12-06 23:...	LaGuardia	23.8200	Tuesday	6	341
72	2016-12-06 23:...	JFK	23.8200	Tuesday	6	341

```
taxiPickupsMyDay.PickupFraction = taxiPickupsMyDay.TripCount./taxiPickupsMyDay.sum_TripCount
```

```
taxiPickupsMyDay = 72×9 table
```

...

	PickupTime	Location	TimeOfDay	DayOfWeek	DayOfMonth	DayOfYear
1	2016-12-06 00:...	Manhattan	0.8200	Tuesday	6	341
2	2016-12-06 00:...	LaGuardia	0.8200	Tuesday	6	341
3	2016-12-06 00:...	JFK	0.8200	Tuesday	6	341
4	2016-12-06 01:...	Manhattan	1.8200	Tuesday	6	341
5	2016-12-06 01:...	LaGuardia	1.8200	Tuesday	6	341
6	2016-12-06 01:...	JFK	1.8200	Tuesday	6	341
7	2016-12-06 02:...	Manhattan	2.8200	Tuesday	6	341
8	2016-12-06 02:...	LaGuardia	2.8200	Tuesday	6	341
9	2016-12-06 02:...	JFK	2.8200	Tuesday	6	341
10	2016-12-06 03:...	Manhattan	3.8200	Tuesday	6	341
11	2016-12-06 03:...	LaGuardia	3.8200	Tuesday	6	341
12	2016-12-06 03:...	JFK	3.8200	Tuesday	6	341
13	2016-12-06 04:...	Manhattan	4.8200	Tuesday	6	341
14	2016-12-06 04:...	LaGuardia	4.8200	Tuesday	6	341
15	2016-12-06 04:...	JFK	4.8200	Tuesday	6	341
16	2016-12-06 05:...	Manhattan	5.8200	Tuesday	6	341
17	2016-12-06 05:...	LaGuardia	5.8200	Tuesday	6	341
18	2016-12-06 05:...	JFK	5.8200	Tuesday	6	341
19	2016-12-06 06:...	Manhattan	6.8200	Tuesday	6	341
20	2016-12-06 06:...	LaGuardia	6.8200	Tuesday	6	341
21	2016-12-06 06:...	JFK	6.8200	Tuesday	6	341
22	2016-12-06 07:...	Manhattan	7.8200	Tuesday	6	341
23	2016-12-06 07:...	LaGuardia	7.8200	Tuesday	6	341
24	2016-12-06 07:...	JFK	7.8200	Tuesday	6	341
25	2016-12-06 08:...	Manhattan	8.8200	Tuesday	6	341
26	2016-12-06 08:...	LaGuardia	8.8200	Tuesday	6	341
27	2016-12-06 08:...	JFK	8.8200	Tuesday	6	341



	PickupTime	Location	TimeOfDay	DayOfWeek	DayOfMonth	DayOfYear
28	2016-12-06 09:...	Manhattan	9.8200	Tuesday	6	341
29	2016-12-06 09:...	LaGuardia	9.8200	Tuesday	6	341
30	2016-12-06 09:...	JFK	9.8200	Tuesday	6	341
31	2016-12-06 10:...	Manhattan	10.8200	Tuesday	6	341
32	2016-12-06 10:...	LaGuardia	10.8200	Tuesday	6	341
33	2016-12-06 10:...	JFK	10.8200	Tuesday	6	341
34	2016-12-06 11:...	Manhattan	11.8200	Tuesday	6	341
35	2016-12-06 11:...	LaGuardia	11.8200	Tuesday	6	341
36	2016-12-06 11:...	JFK	11.8200	Tuesday	6	341
37	2016-12-06 12:...	Manhattan	12.8200	Tuesday	6	341
38	2016-12-06 12:...	LaGuardia	12.8200	Tuesday	6	341
39	2016-12-06 12:...	JFK	12.8200	Tuesday	6	341
40	2016-12-06 13:...	Manhattan	13.8200	Tuesday	6	341
41	2016-12-06 13:...	LaGuardia	13.8200	Tuesday	6	341
42	2016-12-06 13:...	JFK	13.8200	Tuesday	6	341
43	2016-12-06 14:...	Manhattan	14.8200	Tuesday	6	341
44	2016-12-06 14:...	LaGuardia	14.8200	Tuesday	6	341
45	2016-12-06 14:...	JFK	14.8200	Tuesday	6	341
46	2016-12-06 15:...	Manhattan	15.8200	Tuesday	6	341
47	2016-12-06 15:...	LaGuardia	15.8200	Tuesday	6	341
48	2016-12-06 15:...	JFK	15.8200	Tuesday	6	341
49	2016-12-06 16:...	Manhattan	16.8200	Tuesday	6	341
50	2016-12-06 16:...	LaGuardia	16.8200	Tuesday	6	341
51	2016-12-06 16:...	JFK	16.8200	Tuesday	6	341
52	2016-12-06 17:...	Manhattan	17.8200	Tuesday	6	341
53	2016-12-06 17:...	LaGuardia	17.8200	Tuesday	6	341
54	2016-12-06 17:...	JFK	17.8200	Tuesday	6	341
55	2016-12-06 18:...	Manhattan	18.8200	Tuesday	6	341
56	2016-12-06 18:...	LaGuardia	18.8200	Tuesday	6	341
57	2016-12-06 18:...	JFK	18.8200	Tuesday	6	341
58	2016-12-06 19:...	Manhattan	19.8200	Tuesday	6	341
59	2016-12-06 19:...	LaGuardia	19.8200	Tuesday	6	341
60	2016-12-06 19:...	JFK	19.8200	Tuesday	6	341

	PickupTime	Location	TimeOfDay	DayOfWeek	DayOfMonth	DayOfYear
61	2016-12-06 20:...	Manhattan	20.8200	Tuesday	6	341
62	2016-12-06 20:...	LaGuardia	20.8200	Tuesday	6	341
63	2016-12-06 20:...	JFK	20.8200	Tuesday	6	341
64	2016-12-06 21:...	Manhattan	21.8200	Tuesday	6	341
65	2016-12-06 21:...	LaGuardia	21.8200	Tuesday	6	341
66	2016-12-06 21:...	JFK	21.8200	Tuesday	6	341
67	2016-12-06 22:...	Manhattan	22.8200	Tuesday	6	341
68	2016-12-06 22:...	LaGuardia	22.8200	Tuesday	6	341
69	2016-12-06 22:...	JFK	22.8200	Tuesday	6	341
70	2016-12-06 23:...	Manhattan	23.8200	Tuesday	6	341
71	2016-12-06 23:...	LaGuardia	23.8200	Tuesday	6	341
72	2016-12-06 23:...	JFK	23.8200	Tuesday	6	341

```
taxiPickupsMyDayFractions = unstack(taxiPickupsMyDay,"PickupFraction","Location","GroupingVaria
```

```
taxiPickupsMyDayFractions = 24×4 table
```

	PickupTime	Manhattan	LaGuardia	JFK
1	2016-12-06 00:...	0.6815	0.0266	0.2919
2	2016-12-06 01:...	0.9117	0.0070	0.0813
3	2016-12-06 02:...	0.9351	0.0060	0.0589
4	2016-12-06 03:...	0.9040	0.0068	0.0892
5	2016-12-06 04:...	0.7997	0.0146	0.1857
6	2016-12-06 05:...	0.8462	0.0079	0.1459
7	2016-12-06 06:...	0.7624	0.1268	0.1108
8	2016-12-06 07:...	0.7207	0.2003	0.0790
9	2016-12-06 08:...	0.7128	0.2366	0.0506
10	2016-12-06 09:...	0.6891	0.2454	0.0655
11	2016-12-06 10:...	0.7091	0.2108	0.0801
12	2016-12-06 11:...	0.6789	0.2154	0.1057
13	2016-12-06 12:...	0.6682	0.2023	0.1295
14	2016-12-06 13:...	0.6783	0.1936	0.1282
15	2016-12-06 14:...	0.6171	0.2185	0.1644
16	2016-12-06 15:...	0.5830	0.2327	0.1843
17	2016-12-06 16:...	0.6173	0.2303	0.1524
18	2016-12-06 17:...	0.6311	0.2318	0.1372

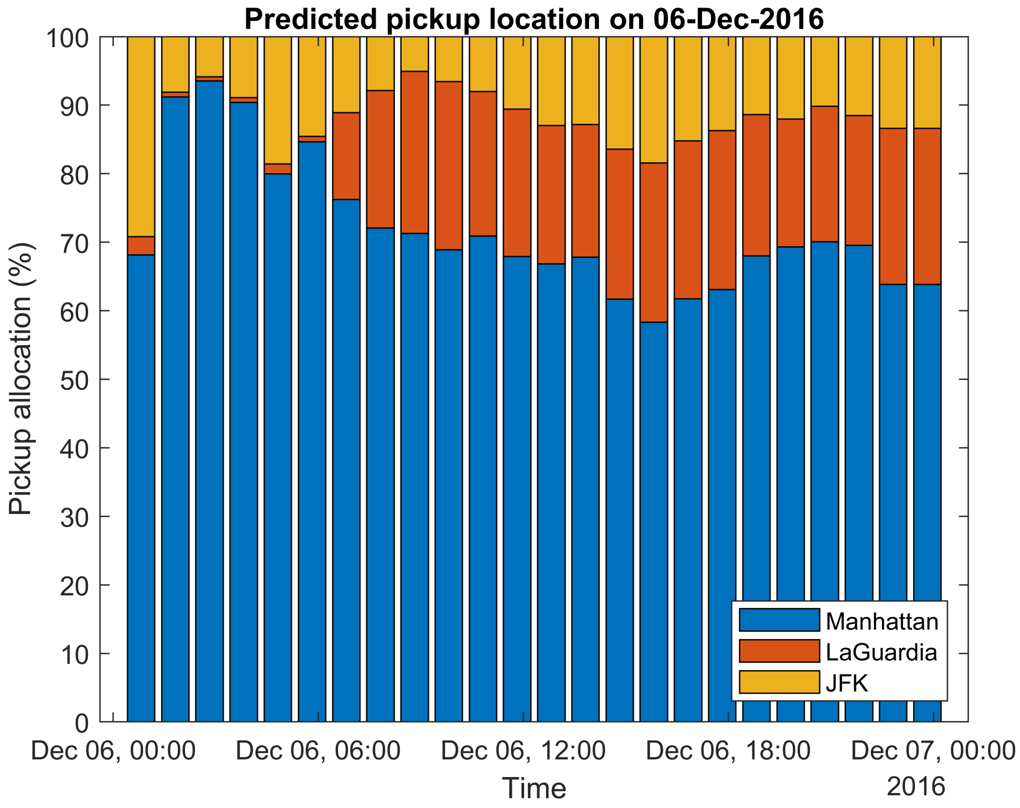
	PickupTime	Manhattan	LaGuardia	JFK
19	2016-12-06 18:...	0.6800	0.2063	0.1137
20	2016-12-06 19:...	0.6929	0.1869	0.1202
21	2016-12-06 20:...	0.7004	0.1978	0.1018
22	2016-12-06 21:...	0.6956	0.1894	0.1150
23	2016-12-06 22:...	0.6383	0.2276	0.1340
24	2016-12-06 23:...	0.6383	0.2276	0.1340

Use the Results to Allocate Fleet

Now it is time to present to Mr. Walker. Discuss the results you were able to obtain, and provide recommendations how you would allocate the fleet of taxis on the chosen day. Provide your reasoning, and also present your case using at least one visualization, e.g. a stacked bar plot.

```
bar(taxiPickupsMyDayFractions.PickupTime,table2array(taxiPickupsMyDayFractions(:,2:end))*100,'

ylim([0 100])
xlabel('Time')
ylabel('Pickup allocation (%)')
title('Predicted pickup location on '+string(myDay))
legend(taxiPickupsMyDayFractions.Properties.VariableNames(2:end),Location="southeast")
```



1. Manhattan taxi pickups is highest all the time

2. Taxi pickups for JFK start reducing after 7:00 AM compares to LaGurdia.