

# Flight Ticket Price Analysis with Linear Regression

*Jun Cai, Vikas Boddu*

*February 11, 2016*

This document describes an implementation of Mapreduce instance with Hadoop doing ticket price analysis on the OTP (On-Time Performance) dataset. The purpose of the application is to perform a simple linear regression on the ticket price. The features we used are distance and air time. The label is average ticket price. For each carrier which is active in 2015, we finished two simple linear regression for it. One between distance and average ticket price. The other one between air time and average ticket price.

## Implementation

### Data Normalization

In order to get a good regression results, the features should be normalized. We normalized the data so that they have zero mean and one variance. A R script is used to go over all the data and find out the mean and standard deviation of the distance as well as the air time. Then these information is provided to Mapper along with the dataset. Mapper will normalize the features in map phase.

### Model and Normal Equation

For a simple linear regression, there is only one explanatory variable which in our case is either distance or air time. So the model will simply be

$$y = \theta_0 + \theta_1 x$$

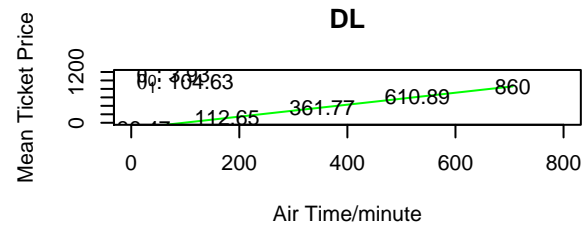
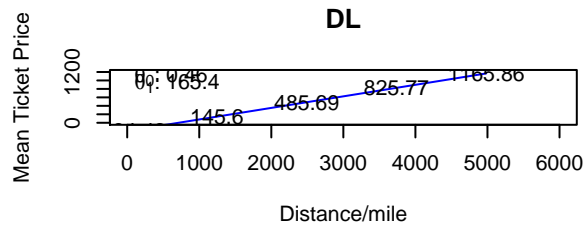
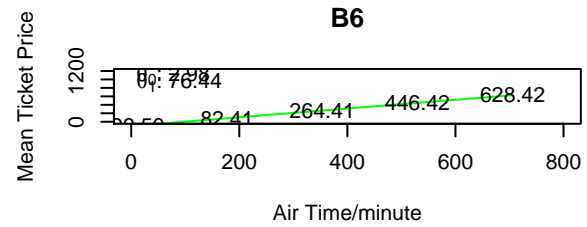
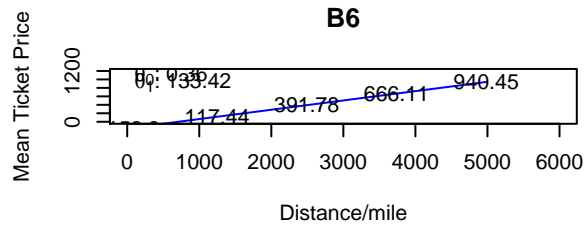
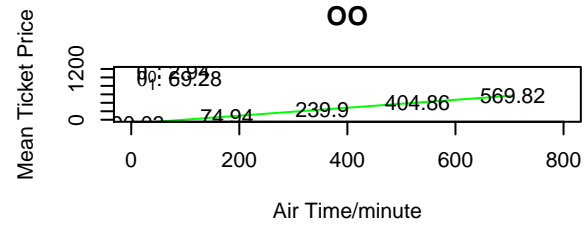
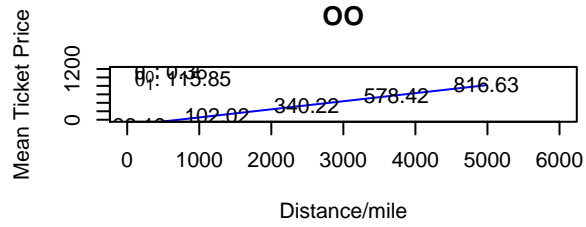
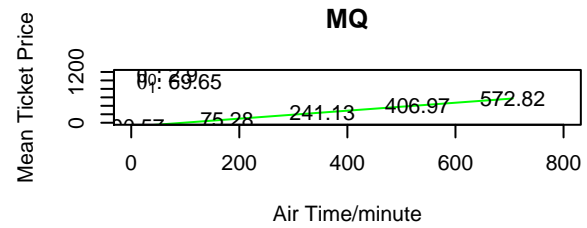
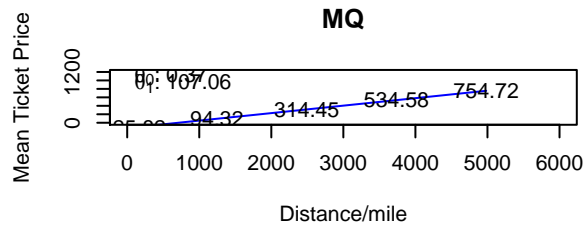
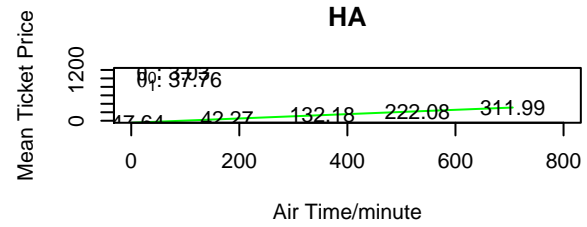
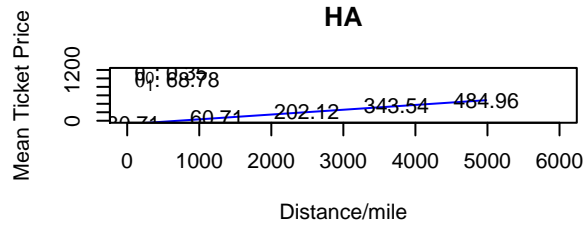
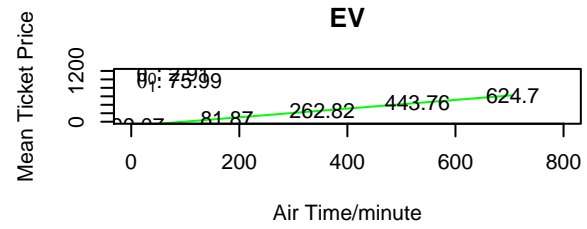
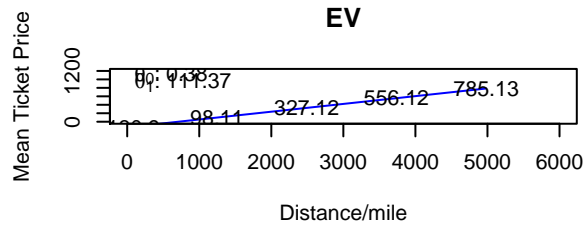
where  $y$  is the average ticket price and  $x$  is distance or air time of the flight.  $\theta_0$  and  $\theta_1$  are two model parameters which can be estimated from linear regression. In this implementation, we use normal equation to solve the linear regression problem.

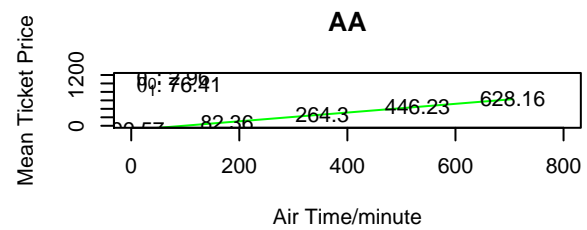
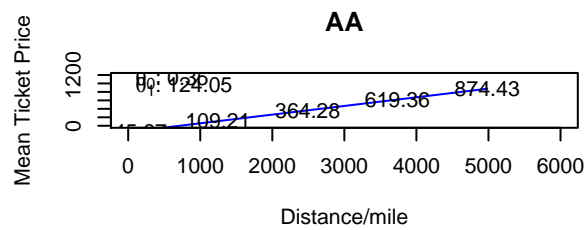
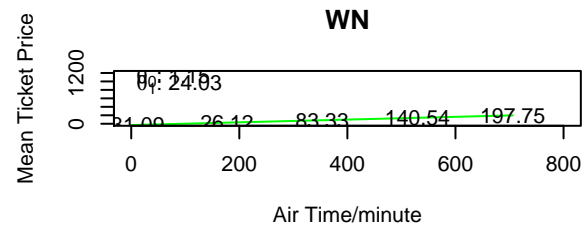
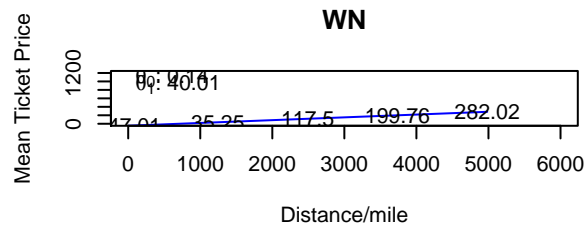
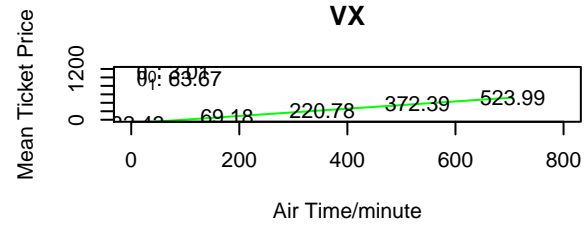
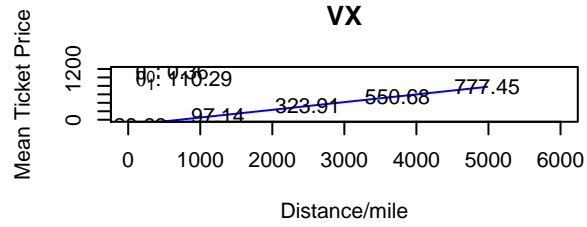
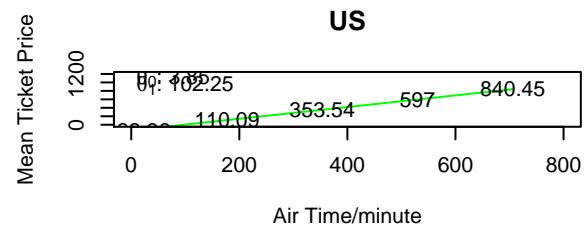
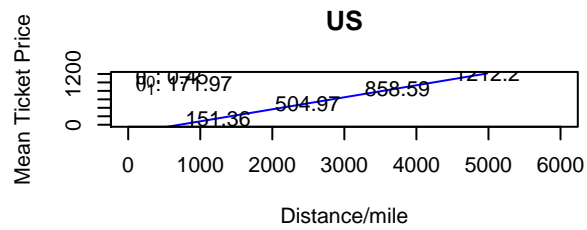
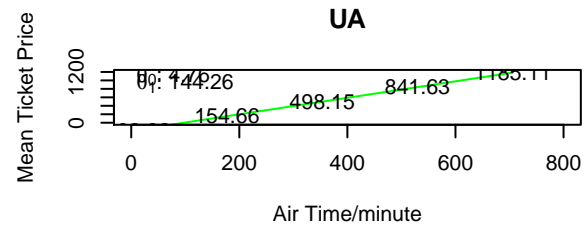
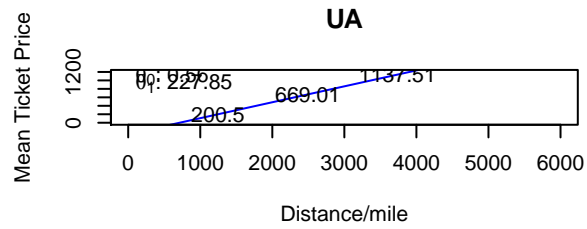
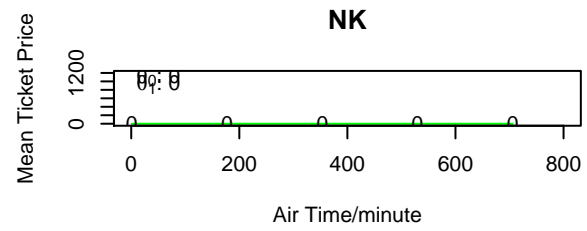
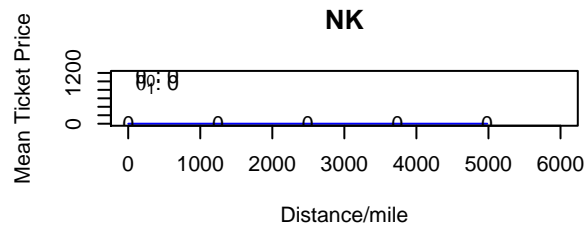
$$\theta = (X^T X)^{-1} X^T Y$$

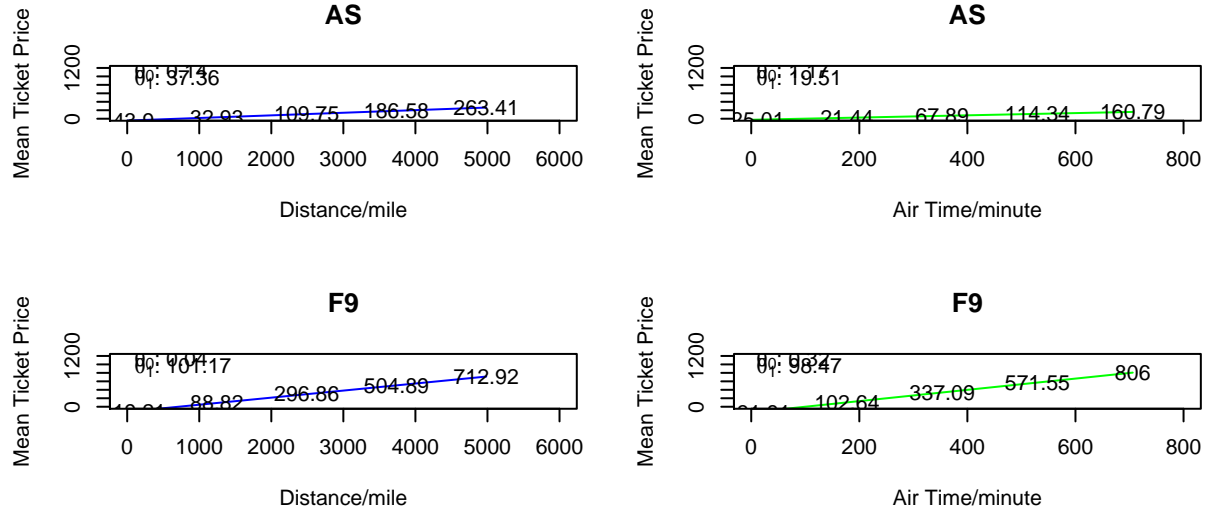
In order to calculate the normal equation in Mapreduce, we divide the computation into three steps: 1. Compute  $X_i^T X_i$  and  $X_i^T Y_i$  for each observation and combine the results for each carrier in the mapper. 2. Compute the actual  $X^T X$  and  $X^T Y$  in the reducer by summing up all  $X_i^T X_i$  and  $X_i^T Y_i$  for each carrier. 3. Compute  $\theta = (X^T X)^{-1} X^T Y$  for each carrier.

## Regression Results

Following is the regression results created by R.







Where the title of each plot is the carrier's name. The left column is for the regression using distance as the feature and the right column is for that using air time. Model parameters for each regression are also given in the plots as  $\theta_0$  and  $\theta_1$ .

## Conclusions

### Comparing the Features

From all the coefficients above, we can see that  $\theta_1$  is always much larger than the  $\theta_0$  which means the features always contribute much more to the result rather than the constant does. In order to find out which feature is better for predicting the average ticket price, we calculate the MSE for each regression. We think the distance is a better feature because for most carriers, the MSE of regression with distance is better than that with air time. So the distance can give us a more accurate prediction of the ticket price.

Carriers	MSE for Regression with Distance	MSE for Regression with Air Time
EV	119416.625	108508.326
HA	151595.237	177144.706
MQ	114820.834	104138.244
OO	117750.206	105643.573
B6	201054.329	240822.089
DL	303487.411	337597.071
NK	274066.014	274066.014
UA	587859.835	755985.014
US	300724.435	339529.608
VX	301370.967	388672.229
WN	24717.913	25185.605
AA	204990.616	238818.965
AS	42374.216	50237.479
F9	27895.279	28162.002

### Cheapest Carrier

We think the cheapest carrier is AS since it has the smallest model parameters among all the carriers.