Machine Learning for Predicting Flight Ticket Prices

André Cibils,
Ecole Polytechnique Fédérale de Lausanne

ABSTRACT: By using machine learning and in particular linear and non linear regression, we predict flight tickets prices. For this purpose we are using data collected during the past 6 months. We explore multiple features such as the time before the flight, the month of the flight and route of the flight to arrive to an acceptable prediction. The trend is that ticket prices are minimum roughly two months and one year before the day of the flight.

1 . INTRODUCTION

The main purpose of this paper is to find a common trend in the evolution of the prices of flight tickets. In this context, the use of machine learning is highly appropriate as a lot of data has been crawled during the past six months.

The mechanisms used by the companies to change the prices of flights tickets are not transparents. Many parameters are not available, such as the number of remaining places in the airplane. We can expect the price to go up when we reach the date of the flight, but also that the ticket will cost more in the summer or in december (due to the vacations) than in october or november.

For this problem, the start was to use simple models such as linear regression and multilinear regression to find a common trend as a first solution, and in a second time more complex models like the ridge regression in order to improve the quality of the results.

Multiple feature will be explored to improve the model, the principal one being the time remaining before the flight. The month of the flight, the route (ie from where to where is the flight), the duration of the flight are some examples of explored features.

2 . MATERIALS AND METHODS

2.1 Data collection

The datas are separated in two sets. For each of this sets, everyday at 10:00, a crawler extract data from https://wizzair.com/en-GB/TimeTable. For the first set, the crawling began the 09/11/2015 and it gathered informations of eight different routes. The second set has data only from the 04/01/2015 but has information about twelve others routes. The dataset are merged once the useful data have been parsed.

Each day, the crawler will go through the entire timetable and gather information in a JSON format. The information hold in those JSON is multiple, and the first step is to extract the useful data from these.

The parser extracted relevant information from the JSON and wrote them as csv. Depending on the stage of the project, the information is for instance the time before the flight (Unix time), the day of the flight, the route, the month.

No identification number has been assigned to a flight since there are not two plane going off at the exact same time.

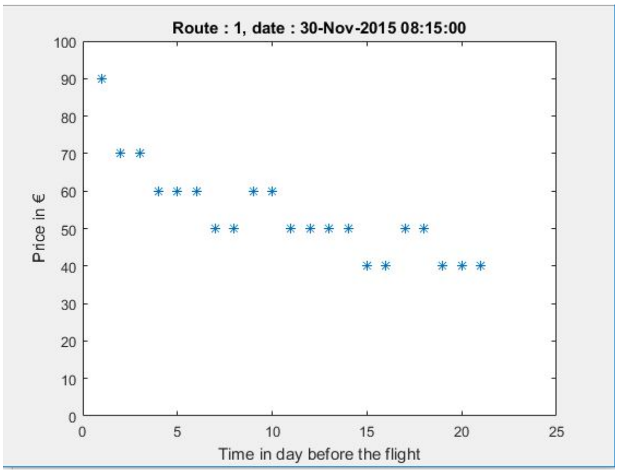

Fig1 : Example of data for a given flight date

2.2 Linear & Multilinear Regression

The goal here is to find a valid estimation of the common trend by using linear & multilinear regression. Our hypothesis is really simple at first : it takes the date before the flight as input and give the expected price as output, more formally hθ(x) = θ * x, where θ is a vector representing the parameters discovered during the training phase and x is the vector representing the features (Only the date before the flight here).

To start, we divide our data set into two sets, the training set and the test set (respectively 70% and 30%). The first one will be used to train our program and the second one to verify the validity of the hypothesis on new, fresh, unused data.

In order to proceed, we will define a cost function J(θ)

$$J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

m is the number of our training examples, x are our features and y our observations. This cost function is the least-square estimation and force the hypothesis h(x) to be as close as possible to our training example. Indeed, the cost function punishes heavily big differences.

Our aim is to find the minimum of this function ie. the parameters θ such that J(θ) is as small as possible. To find this minimum, we use the gradient descend algorithm.

```
Repeat {
    $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \ldots, \theta_n)$
} (simultaneously update for every $j = 0, \ldots, n$)
```

Let α is the learning step, carefully chose. A too big learning step will lead to non convergence of the algorithm, but a too small one will make the program really slow.

The gradient descend algorithm does not give the absolut minimum but only a local one, which is fine in practice. We stop the algorithm when |θ_new – θ_old|<threshold.

Once our parameters are settled and clear we proceed to test our hypothesis with the test set.

2.3 Ridge regression & Regularization

The next step is now to improve our model. To do so, we will use a technique called regularization. This is used to reduce the variance, ie the overfitting problem. Overfitting appears when our hypothesis fit to well to our training data. To add regularization in our model, we need to add a term to our costfunction :

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2$$

This pose a problem, which is finding the right λ. For this, we will redivide our data in 3 sets this time. To make the model more robust, the k-fold method be used (Fig 2).

20% of the data will be put aside and called, once again, test set. We will loop 10 time on the 80% data remaining, each time we select 80% of this data to be our training set, and 20% to be our cross validation set.

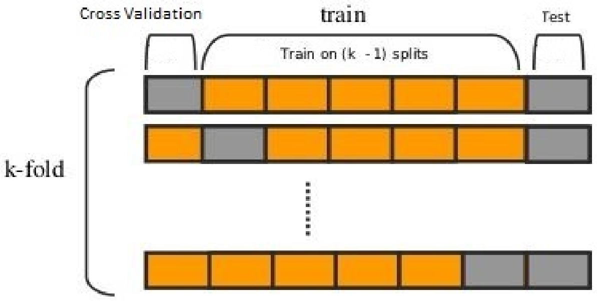

Fig 2 :K-fold technique

The Cross Validation Set will be used to select λ : for each λ, we find the optimized θ and we compute the cost associated with those parameters (the data used to calculate this cost are the data from the cross validation set : like the test set, this data haven't been used to train our program).

Then, we select the better one by simply taking those who have the lowest cost.

We end up with 10 set of parameters, which we can compare with one another to find some unexcpected behavior.

2.4 Explored features

In order to have a strong and complete model, we will have to choose the right features.

The most obvious and important one is the time before the flight. By augmenting the complexity of the hypothesis, we can change the shape of our trend, but we need to be careful to not have a too big complexity as it would lead to overfitting. We can use the test set or the cross validation set to select the right number of degree for the polynomial equation.

Other features considered are the month were the flight take off, the duration of the flight, the route of the flight, if the flight take off during the week or the week end. After that, we need to analyse the impact of theses parameters.

3 . RESULT

3.1 Useful parameters

As we can see in Fig 3, we can see that too much complexity will lead to a drastic augmentation of our error.

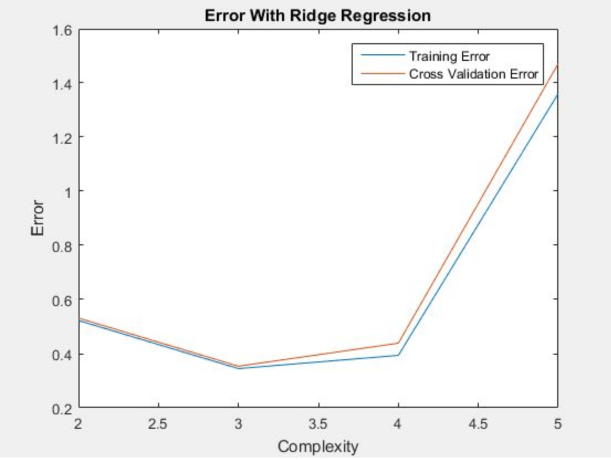

Fig 3 : TrainingError and CrossValidationError depending on the complexity

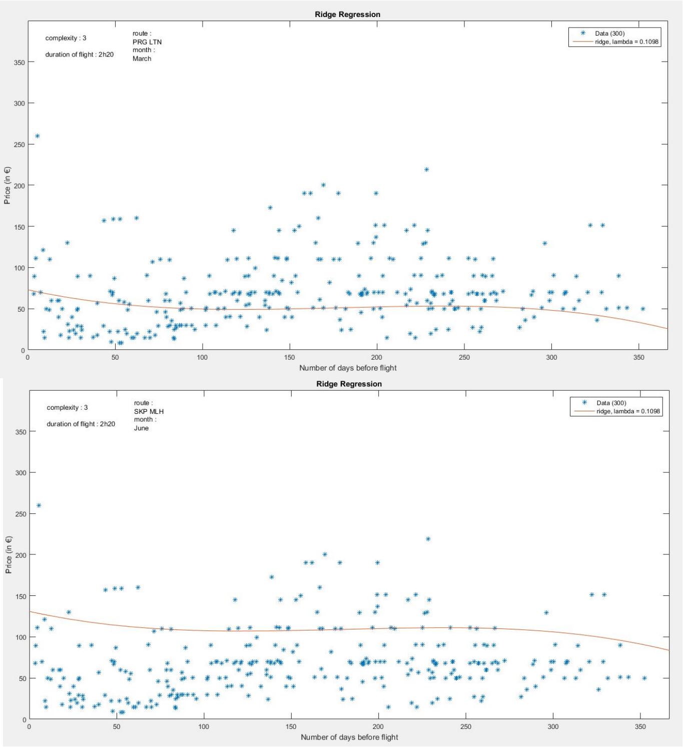

Fig 4 : Influence of the month & route parameters

The month of the flight and the route are very impactant parameters, as we can see in Fig 4

Another parameter considered was the duration of the flight. The associated θ has a value of 0.12, where the θ associated to the time before the date is 0.98. This show that the duration of the flight is not a significant parameter.

3.2 Results of the linear & multilinear regression

The results of this methods are roughly correct and doesn't suffer too much of overfitting. It points that the two key moment to buy a ticket is two months ago or one year ago, without considering anything else than the number of day remaining before the flight (see Fig 5)

After the data got normalized, the training error is 0.0589 and the test error is 0.0632, which is 7% more.

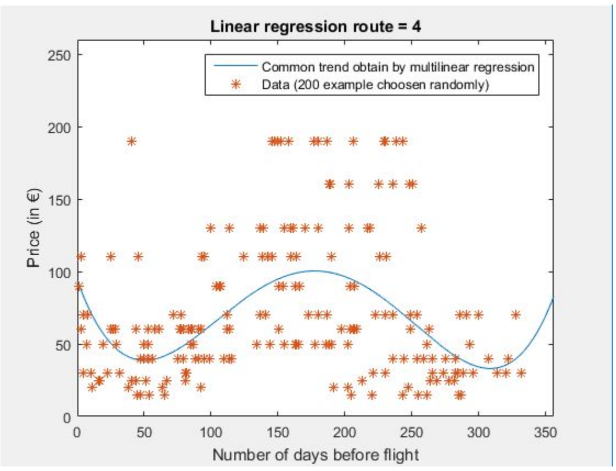

Fig 5 :Results of the multilinear regression,

$$h(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

3.3 Result of Ridge regression & Regularization

Ridge regression offers us a lot more of parameters to choose, and the results depends highly of this parameters. We used some tools to help us select those (see Fig 6)

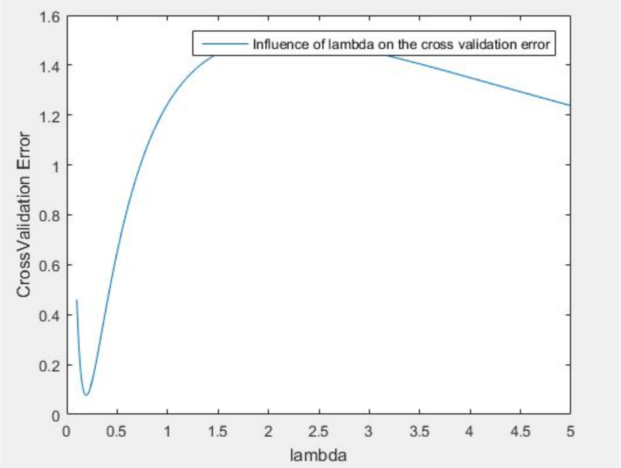

Fig 6 : Influence of λ in Cross Validation Error, with the optimized θ.

The trend obtained by the ridge regression is better than the one from the multilinear regression, the shape is a lot less eratic (see Fig 7).

The test error is lower using the same parameters : 0.015915 for the ridge regression against 0.0665 for the multilinear regression, which is 4.18 times more.

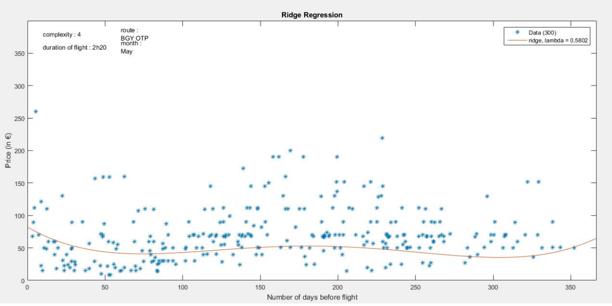

Fig 7 : Results of the ridge regression

4 . DISCUSSION

The results seems to validate our hypothesis, the trend found is correct. The best moment to buy a ticket is two months or one year before the date of the flight. We notice that the traject (route) or the month of the flight we are choosing have a huge influence on the price. Moreover, the duration of the flight is not significant.

In order to improve our result, we should take more parameter in sight, such as if the flight is during the week or the week end.

What we could as well is try to clusterize the routes : the idea here is to find multiple trends and associate each route to a trend (see Fig 8)

For this, we use an overlapping mixtures of Gaussian processes[ref]

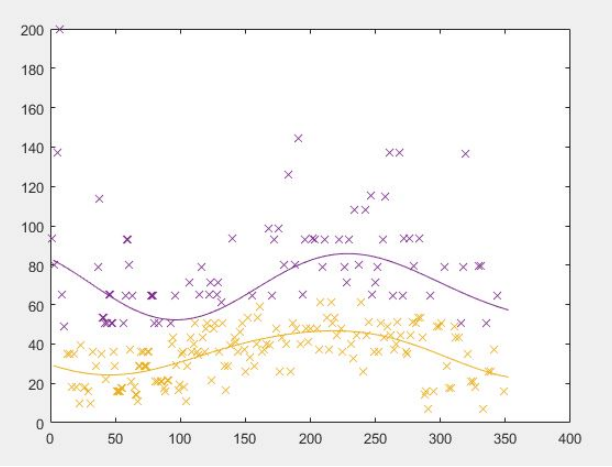

Fig 8 : Two different trend found in the data

5 . ACKNOLEDGMENTS

I would like to thanks in particular my project responsible, Mr. Igor Kulev for his precious advices and his guidance. He also provided me the data, I couldn't have make it without him.