Feature Engineering

* **Flight\_date**

The main difficulty of the challenge is the limited amount of information per flight (the data is "thin", it has few columns). In order to improve our score, we will look for new variables that correlate with our target. The first approach is that we can treat our data as a Time Series to see the impact of ‘flight\_date’. After a correlation and an autocorrelation test, we can see a pseudo seasonality in our data. To get benefits from this ascertainment, we decided to do some feature engineering on this variable by creating many new variables: we split our date into **“year”**, **“month”** and **“day”** (the year variable isn’t that important because our train data has only two years, 2011 and 2012, but our test has other ones, 2012 and 2013). Also, we generate other variables like, ”**julian**” (number of days between ‘2010-01-01’ and the given date), “**quarter**”, **“week”** and **“dayofweek”**, this last one turned to be very interesting for our model.

* **Distance**

Taking into consideration that the airline does not refund passengers that doesn’t show, we can make a hypothesis that expensive trips have a low no-show rate, in other way there is a dependency between no-show rate and ticket fares. After many attempts, we didn’t find the mean price for a specific trip and for a specific date, so we decided to use another approach. In a way or other there is a dependency between the ticket price and the distance of the trip. So, we used an API to determine the GPS coordinate (“**from\_longitude**”, “**from\_latitude**”, “**to\_longitude**” and “**to\_latitude**”) of each IATA and then calculate the distance (“**distance**”).

* **Fare**

For the same reasons that we considered for the distance variable, we decided to follow another approach to determine the mean price of ticket for the trip by quarter. And it’s the “**Fare**” variable.

Different approaches

In the model Notebook, we saw different approaches and different models.

Conclusion