Machine Learning Engineer Nanodegree

Capstone Proposal

Clement LEFEVRE July 10th 2017

Proposal

Predict the occupancy rate of Airbnb appartments in Berlin.

Domain Background

Sharing economy is booming thanks to the spread of internet. It can also redraw entire pieces of the economy, like the taxi industry with Uber and Lyft. For this capstone project, i will work on AirBnB, and focus on the appartments rented in Berlin, where i live. The AirBnB system has experienced a significant growth in the last 4 years there, and in some ways contributed to the shortage of housing in Berlin. Due to the historically relatively low rents in Berlin, local inhabitants are very sensitive to this topic, which is perceived as a further syndrom of gentrification. As a consequence, the local press often raises the issue of Airbnb (New Berlin ban for Airbnb), AIRBNB vs. BERLIN.

For the personal aspect along this topic, one appartment in the building i live in is rented full-time via AirBnb, generating disturbance for our neighborhood such as loud music, smoking in the stairs, and broken glass.

Problem Statement

From the aspects highlighted above, how can we predict the occupancy rate of an AirBnB appartment given the data publicly available ? The occupancy rate is defined as the proportion of booked nights over the next 6 months.

Datasets and Inputs

As the AirBnb data are not directly available already aggregated, i used the latest dataset (May 2017) from insideAirBnB which scraps for some cities the entire offers and make those datasets available.

The dataset for each city consists in a list of appartments with all features available using the regular AirBnB booking website, including the reviews of the past guests, the booking calendar.

- http://data.insideairbnb.com/germany/be/berlin/2017-05-08/data/listings.csv.gz :Detailed Listings data for Berlin.
- http://data.insideairbnb.com/germany/be/berlin/2017-05-08/data/calendar.csv.gz : Detailed calendar per appartment.
- http://data.insideairbnb.com/germany/be/berlin/2017-05-08/data/reviews.csv.gz : All reviews for all appartments.
- http://data.insideairbnb.com/germany/be/berlin/2017-05-08/visualisations/listings.csv : Summary of the appartments infos.
- http://data.insideairbnb.com/germany/be/berlin/2017-05-08/visualisations/reviews.csv : Summary Review data and Listing ID (to facilitate time based analytics and visualisations linked to a listing).

Solution Statement

Using the dataset mentioned above, we create a regression model that fits to the current occupancy rate. For each appartment, using the different features availables (price, number of reviews, content of reviews, location) we train a regression model on it that minimize the error between the predicted value and the observed one.

Benchmark Model

Predictive tools for the appartment occupancy are already available as commercial tool AirBnb Investment Explorer, but they lack the necessary transparency to understand the logic behind it. Thus we will stick on the linear regression as a benchmark to evaluate our own model performance.

Evaluation Metrics

As we are in the context of the regression problem, i will use the Root Mean Squared Error metrics (RMSE) of the occupancy rate as the main metrics.

The RMSE is defined as :

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y}_i)^2}$$

where y_i is the real occupancy rate and $\bar{y}_i$the predicted one.

Project Design

- The first step will consist in transforming the text reviews into usable features. By using TFIDF(term-frequency times inverse document-frequency) and filter on the most relevant words. Some dimensionality reduction via clustering might be necessary, given the number of features generated by this text transformation.
- Next, after a thorough **Exploratory data analysis**, we will process to **features selection** to filter only the ones that significantly explain the occupancy rate variability.
- Then, following the cross-validation / train-test methodology, we will implement differents regression algorithms (linear, Decision Tree family) and also give a try with Neural Networks using the python keras wrapper for Tensorflow.
- Finally, i will use the stacking technique to get the best predictor as a combination of the differents models.