Machine Learning For a House pricing Prediction Web Application

Alex Soudant

Ynov Ingésup M1,

20 Boulevard Général de Gaulle, 44200 Nantes

Correspondence: Alex Soudant. E-mail: alex.soudant@ynov.com

report on the 10th of February 2017

Our planned objectives were to clean the scraped informations and put images to the same format".

After using the scraper to scrap prices, features and images from four cities in France, I ran data quality checks to garanty that each scraped house advert had both images and a json file containing the wanted pieces of information on the property sale. These checks were integrated to the image reformating procedure (see imageTransformation.py). Image reformating consists in changing images to a gray scale, cropping and resizing the original images into a new set of images for each advert saved to a different location. The objective is to obtain derived images of size 56*56 that represents the double of the training set images we used during asignments from the deep learning course on Udacity (see tensorFlow notebooks folder on github). As I have much less images than during these exercices, double the size seems like a reasonable guess that I will still be able to perform analysis in the given time while retaining a good level of picture information.

While running image transformation, I also have set up the script to discard images that are corrupted, of too small a size to be transformed (like thumbnails) or do not proceed through reformating due to an unusual shape. After this step, I acknowledge a 15% loss in the original adverts number due to the cleaning and reformatting procedure.

Now that my images are in a decent shape for analysis, I imported the matching advert json files to the derived dataset (see jsonFileTransfert.py). This step allow me to keep all the wanted information at the same place without risk of mistaking folders when regrouping prices and images during further analysis.

Finally, I proceeded though the json files transformation to provide usable information (see jsonHandler.py). This step mainly consists in deleting unicode characters and keeping only numeric values from features such as price, surface, number of rooms and land size. I acknowledge that some prices are sporadiously missing but will not greatly affect the final number of adverts we can analyse later on with tensor flow (see AdvertsInfo.csv).

I am now happy with the shaping of my data to perform the analysis. I started the contruction of the tensors and will need a few hours more before starting testing a first linear regression model that will estimate sale prices by using solely images. The cost function will be based on minimizing residuals from least square estimation. I expect this first result will be poor but we can then observe if feeding more information to the algorithm will improve the result or try to build a logistic regression with categories of sale prices instead of

a continuous scale. I am currently still working over a jupyter notebook to test the tensors construction (see TensorFlowHousingImages.ipynbTensorFlowHousingImages.ipynb).

This first results are obtain only on Nantes city (see derivedImages folder on dropbox). However, raw data from Angers, Brest and Rennes will also be processed (see the folders named after Angers, Brest and Rennes). The scraper in javascipt can also be checked at the dropbox private link sent on Slack.