UNIVERSITÀ DI PISA

Artificial Intelligence and Data Engineering

Data Mining and Machine Learning

AirBnb Price Estimation using Machine Learning

Project Documentation

TEAM MEMBERS:
Edoardo Fazzari
Mirco Ramo

Academic Year: 2020/2021

Contents

1 — Introduction

Housing prices are an important reflection of the economy, and pricing a rental property on Airbnb, therefore, can be a challenging task for the owner as it determines the number of customers for the place. On the other hand, customers have to evaluate an offered price with minimal knowledge of an optimal value for the property.

1.1 Goals

The aim of this paper is to explain the choices and the strategies we adopted on the project and development of **AirBnb Price Estimator**, whose aim is to help owners to decide the most correct price for their *BnB*. In order to accomplish it, we started from web-scraped data, we performed all the preprocessing needed for having a suitable dataset and then we built several classifiers, using different strategies, in order to determine the one that predicts best the class attribute. All these classifiers have been tested using more than one method and the analysis of the results guided us in the choice of the best classifier. Since the class attribute is numeric, we had two possible choices:

1. Discretize the attribute, choosing the most appropriate algorithm
2. Keep it numeric, using regression algorithms for the classification purposes

The first approach is surely easier, but it would not be as helpful as the second one for our application purposes: suggesting a precise value to an owner will give him/her a more accurate advice rather than a range.

The regression model that generalizes best the class feature has then been chosen as the "heart" of AirBnB Price Estimator: the application asks users to input the required fields that correspond to the attributes needed by the classificatory. On these fields bases, it simply outputs to the user the suggested price for night.

1.2 Initial Dataset

The fee of a real estate is strictly related to the city where it is located, thus would force us to recompute for every different city each step we make over and over, wasting time. Thus we decide to consider only one location (i.e, city), big enough to have a huge number of records. We chose *New York City*. The dataset, related to all the **BnB** situated in *NYC*, is taken from `http://insideairbnb.com/get-the-data.html` [1]. The scraping has been performed the 11th December 2020 (the scraped data has been made available by third parties).

The *initial dataset* is composed by 44667 instances and 74 columns. In order not to confuse the reader with useless information, the attribute list is not here reported, however on the following chapter regarding preprocessing we justify in detail the actions performed on the data. For now, the most important things to know is that:

1. The dataset is composed by various mixed features regarding the estate, the host and he geographical position
2. The source file has been published without respecting exactly the *csv* format, thus some frameworks and *csv* f handlers are not able to parse it

[1] In the case of the dataset form the website inside the *listing.csv* file, related to NYC, is updated, we stored the dataset used on Google Drive. It can be downloaded at: `https://drive.google.com/file/d/12ZA4Jo-MgQGGjnNVMUdDom8USHXBagr_/view?usp=sharing`

3. The initial data is very dirty: there are missing values, redundant attributes, lists of strings embedded in a single column, pointless features and so on. For all of these problems, a suitable solution has been provided and it is fully reported on the next chapter.

2 — Preprocessing

2.1 What can be preprocessed now?

Since we have to deal with classification problems, before preprocessing data we must be sure not to apply supervised filters on the whole dataset. If we need supervised filters, we must split the dataset in training set and test set before applying them. However, in our case there is no need for a supervised filters but attribute selection, so we performed all the unsupervised preprocessing operations at the beginning, postponing the attribute selection after the training/test split.

2.2 Data Cleaning and Reduction

Since the Weka framework was not able to parse correctly the source file, for the following operations we used *Microsoft Excel* and *Apple Numbers*.

2.2.1 Removing Noisy and Irrelevant attributes

The first preprocessing operation is the attribute reduction, indeed we decided to remove all the features that are not domain-specific, nor useful for classification purposes. Among them we deleted:

1. **ID**s: *Bnb_ID, scrape_id, host_id*
2. **URL**s: *listing_url, picture_url, host_url, host_thumbnail_url, host_picture_url*
3. Scraping informations like *last_scraped_date, calendar_last_scraped*
4. Useless information on the rent for classification purposes like *name, description, first_review_date*
5. Useless information about the host like *host_name, host_location* (it is not the place in which the rent is, but where the host lives), *host_about, host_has_profile_pic*

2.2.2 Removing Redundant Attributes

The next step was the reduction of attributes explicitly redundant. For all of these we did not compute the χ^2 test because of the explicit correlation between the features

1. *host_neighbourhood and neighbourhood* **w.r.t.** the couple *neighbourhood_cleansed, neighbourhood_group_cleansed*
2. *host_total_listings_count* completely equal to *host_listing_count*
3. *host verification* **w.r.t.** *host_identity_verified*
4. *minimum_minimum_nights, minimum_maximum_nights, maximum_minimum_night, maximum_maximum_nights, minimum_nights_avg* and *maximum_nights_avg* are redundancies of the *attributes minimum_nights* and *maximum_nights*
5. *review_scores_accuracy, review_scores_cleanliness, review_scores_checkin, review_scores_communications, review_scores_location, review_scores_value* are rounded values that have been more precisely combined in another already-existing attribute that is *review_scores_rating*
6. *calculated_host_listings_count* (+ category) are redundant attributes of *listings_count* valid only for their respective category

2.2.3 Additional Removed Features

Eventually, the other columns that have been discarded are:

1. Attributes that strongly depend from the instant in which the scraping has been performed like *has_availability_now, availability_30, availability_60, availability_90, number_of_reviews_ltm, number_of_reviews_l30*
2. Empty attributes like *license*
3. Others, like *latitude* and *longitude* that are not more useful than the easier data on the neighborhood

3 — Classification

3.1 Strategies

Come dividiamo train e test, quali classificatori scegliamo, quali algoritmi di attribute selection

3.1.1 Train and Test Splitting

3.1.2 Chosen Classifiers

3.2 Building Classification Models

Metterci anche il codice

3.2.1 Procedure

3.2.2 Implementation

3.3 Performance Evaluation and Effects od Attribute Selection

Metterci codice, screenshots, procedure di valutazione e un SACCO DI ROBA

3.4 Conclusions

4 — AirBnb Price Estimator App

4.1 Introduction

4.2 Functional Requirements

4.3 Application Guide