Group Project

A case study for creating a service for Airbnb hosts in Athens

October 2022

Overview

This document describes the scope of the case study project that will be undertaken by the different teams. The project involves building a model that predicts the prices of an Airbnb listing in Athens. Your team is asked to explore the given data, process them as you see fit and build a ML model.

An API will be implemented to provide a machine friendly access and management of the data. Also, a web-based user-interface will be built that will allow a user to interact and perform various actions. Finally, a presentation will be given with all the work, decisions, implementations and results that have taken place.

Table of Contents

1. Introduction

This group project aims at encouraging students to apply the knowledge and experience learned in the class towards a real-life business intelligence system.

You are employed as a Data Scientist at Airbnb, a company that is involved in short-term rentals. Airbnb wants to create a service for hosts with top-rated undervalued listings that will suggest they increase their prices. Your team is tasked with building a POC for this service. You have to (a) create and train a model that will predict the price of a listing, given its attributes, (b) deploy the model as an API, and (c) implement a web-based user-interface (UI).

In terms of data content, you are provided with Airbnb data for the region of Athens, where the POC will take place. The data includes information about the **listings** (neighborhood, amenities, bedrooms, and bathrooms, etc.) and **ratings** for those listings.

Your project will focus on the above three tasks. The steps you should follow regarding the data flow, the modelling process and the resource management are up to you. Your code needs to be well documented and organized.

2. Project Scope and Deliverables

The main objective of this project is to make a model that predicts the **price** of a listing, given its attributes.

Several subtasks can be spawned from this objective. The main categories are:

a. **Explore** the given data. See what they describe and gather valuable insights about their properties.
b. **Preprocess** the data so that they can be used for predicting the listing price.
c. **Model** the data through the sklearn estimators.
d. Create an API that will allow users to utilize the model and access statistics
e. Create a front-end application that the user will be able to test the proposed model.

Your **project deliverables** which will support the objectives are identified as deliverables **D01-D05** in the following sections. You will collect all deliverables and submit them as your **project portfolio** work.

3. Overview of Project Work

For running this project, you are advised to frequently meet as a team, and discuss and agree on your implementation plan and actions. This means that you must end up with a clear understanding of

a. the roles and responsibilities of the team members
b. the project requirements
c. the data requirements
d. the way you will run your project
e. the tools you will use for the technical work
f. the tools you will need for the running of your team
g. the deliverables of your work

You will use some of the above decision content in the deliverables outlined next.

4. Data Description

The dataset is provided in three files called *listings.csv, calendar.csv* and *ratings.csv*.

The first contains nominal information about the listings, like its neighborhood, its description, amenities, bedrooms, bathrooms and more. Some are useful, some not so much. These are in a very raw form and need to be processed in order to be used by the model.

The other two contain the calendar bookings for the next year and user ratings for the listings. It is not mandatory to use these files in your analysis, nor is it self-evident how to incorporate this information in your prediction model. Should any team, though, have any idea on how to utilize this information, feel free.

Note: The dataset is provided by Airbnb on a Creative Commons CC0 1.0 Universal (CC0 1.0) "Public Domain Dedication" license, so it is free to use in this project.

5. Detailed Objectives

5.1 Exploratory Data Analysis

Investigate your data. Try to get a feel of the dataset and decide on the preprocessing steps you may need to perform in the next step. The following questions can guide you through this exploration.

1. How many samples and features does each file have?
2. What are the types of your features?
3. Are there any missing values? If yes, how many and how many rows are affected?
4. How many listings per neighborhood are there?
5. How many listings per room type are there?
6. How many listings per room number are there?
7. What is the distribution of listings per host? What are the most listings that a single host has?
8. When was the first host registered?
9. What year had the most hosts registered?
10. How many identified hosts are there? What is their percentage over all hosts?
11. What are the top-20 most common amenities provided by the hosts?
12. What is the distribution of price for each room type?
13. How many ratings do I have?
14. Do all listings have ratings?

Additionally, we encourage you to perform **your own exploration** on the dataset and identify anything you find interesting.

D01: In a GitHub repo push a notebook containing any Exploratory Data Analysis (EDA) you performed.

5.2 Preprocessing

In this step you must bring the dataset in a format understandable by most machine learning algorithms. Some steps you might want to consider:

- Handling missing values in the dataset.
- Encoding categorical features.
- Scaling the features.
- Cleaning erroneous values.
- Handling outliers.
- Feature selection/extraction.

Note 1: Not all of these steps are mandatory. You should do what you think better suits your needs.

Note 2: The same preprocessing steps followed during training should also be implemented for the model that serves the requests.

D02: Push a notebook or well-structured script showing the preprocessing steps as you applied them.

5.3 Modelling

This task is where you must build a model that accurately predicts the price of a given new listing. The metrics you should use for evaluating the results are **Mean Absolute Error**, **Mean Absolute Percentage Error** and any other way you see fit! For this task you should examine different models, while performing hyperparameter tuning for each. Feel free to try any idea or model you want!

In this task you should:

- Build a model that correctly predicts how long a patient will be hospitalized, according to the labels above.
- You can use any technique you want (heuristic, statistical, machine learning. etc.)

D03: Push a notebook or well-structured script showing the steps you followed to build the model.

5.4 Deploy the model as an API

Create an API that provides access to the Airbnb data. The goal of this task is to expose the processed data and the created models from the two previous tasks to the world via a RESTful API.

1. Develop an application which will provide two endpoints:

   - **/stats:** Provide meaningful aggregations based on the dataset. Some examples of meaningful statistics:
     - Plot a histogram of the values of a column (e.g., price ranges)
     - Plot a pie chart of the values of a column (e.g., room types)
     - Plot a bar chart with the average listing price per neighborhood.

     Hint 1: The above are just examples. You should plot the variables and relationships that you have found to be meaningful during your EDA.

     Hint 2: These plots need to be generated from the processed data! There is no point to plot categories containing typos, mistakes, etc.

     Hint 3: these figures are meant to be viewed by a human. Treat them as such. E.g., the categories in a pie chart should be understandable by a human; an encoded/scaled variable loses some of its interpretability.

   - **/models:** This endpoint should accept the data for an individual listing, in the format of the original file we provided to you and return the price.
     To do this you should:
     - Perform a check on the validity of the data provided. Has the client provided you with the right number of features for the model to make a prediction? Does each feature have a valid value?
     - Bring the data in a format understandable by the model. To do this you must process the data in the same way you did with the original dataset (cleaning, encoding, scaling).
     - Pass the processed sample to the model and return its prediction to the user.

2. Log to a file any incoming requests and system behavior you find relevant.

3. Response from /stats endpoint should comply with how you want to draw your charts in the web application. Your front-end should not do any processing on the data.

4. Sample /stats response based on examples:


```
{
  "stats":{
    "histogram_data":[
      {
        "price_range":"0-50",
        "count":2
      },
      {
        "price_range":"51-100",
        "count":143
      },
      {
        "price_range":"101-150",
        "count":28
      },
      {
        "price_range":"150-max",
        "count":28
      }
    ],
    "pie_chart_data":[
      {
        "room_type":"Private room",
        "count":150
      },
      {
        "room_type":"Entire home/apt",
        "count":50
      }
    ],
    "bar_char_data":[
      {
        "neighborhood":"Centrum-West",
        "avg_price":120.3
      },
      {
        "neighborhood":"Slotervaart",
        "avg_price":81
      }
    ]
  }
}
```

5. Sample /model request and response:

RQ:


```
{
  "id":2818,
  "name":"Quiet Garden View Room & Super Fast WiFi",
  "host_id":3159,
  "host_name":"Daniel",
  "neighbourhood_group":"Oostelijk",
  "neighborhood":"Buurt",
  "latitude":52.36435,
  "longitude":4.94358,
  "room_type":"Private room",
  "minimum_nights":3,
  "number_of_reviews":305,
  "last_review":"2022-08-30",
  "reviews_per_month":1.86
}
```


RS:


```
{
  "listing_info":{
    "reviews_per_month":1.86,
    "neighborhood":"Buurt",
    "room_type":"Private room",
    "neighbourhood_group":"Oostelijk"
  },
  "prediction":{
    "price":49.5
  }
}
```


Again, those are samples, and you should modify them according to your modeling. (e.g., request only the fields that you use for prediction and respond with transformed/processed fields.)

D04: A GitHub Repo with the code for implementing the API described and instructions on how this code will run.

5.5 Web APP

The goal of this task is to consume the RESTful API - developed in the previous step - and present it nicely in a web application. The web app will be responsive and viewable from any device (and screen size) with a modern browser. From a more technical view, the web app will be a SPA (single page application) based on the React library and its extensive ecosystem.

- Create the appropriate layout and components
- Create 2 routes / pages – one for the "**models**" and one for the "**stats**"

D05: A GitHub Repo with the code for implementing the described web app and instructions on how this code will run.

5.6 Deliver code

To sum up, we ask you to deliver a well-written, documented, and organized code. Some guidelines to follow:

- All code (notebooks and scripts) should be runnable.
- Deliverables in notebook format should not have dead cells, commented out code or anything code-related that is nonessential.
- We also ask for some markdown cells briefly explaining what is being done in the cells.
- Scripts should be clear and well documented. Avoid code duplication as much as possible.
- Include a requirements.txt detailing your project's requirements
- Include a readme briefly explaining how to run the code.

6. Project deliverables

You will need to push the deliverables **D01**-**D05** to the **main** branch of your team's private GIT repo. This branch should be as clean as possible, containing only the deliverables and no former experimentation!

You will also need to prepare a **presentation** on your project work. This is a presentation that you will give as a team at the end of the course.

Hint: keep your presentation at a high-level and entertaining. Say what you did, but not how you technically did it. The presentation is not a place to showcase your code.

More details on the content and a more personalized opinion on your presentation can also be provided during the Project Development sessions.