A good tourist city

Anthony Suárez

December 20th, 2020

I. Introduction

Tourism is an important economic income for cities. Some cities are attractive because of their beaches, their building, their history and much more. But there are also some cities which would not be nice to go for a vacation. In this project I will analyze the **possibility of predicting how attractive a given city could be and try to build a machine learning model that can do it**. This could help people know which city to visit for their next vacation and governments know how to make their cities more attractive.

II. Data

The data used for this project will be obtained from Foursquare using the Foursquare API. As many categories of data are too specific and many of them are unique in their country, I used the most general categories in the Foursquare Venue Category Tree.

In addition, the cities categorized as "good tourist cities" will be chosen based on articles on the internet like this one. Venues data obtained from these cities will be fed into the machine learning algorithm to create the model.

III. Methodology

There are many types of tourist destinations, as shown in this article, and some of them could be treated as a single venue by the Foursquare API which would not be good for our model. That is the reason this project is aimed at predicting good towns and cities, considering these tend to have a variety of venues we can work with,

So now, the first step will be choosing the cities I will be getting the data from. Let's start!

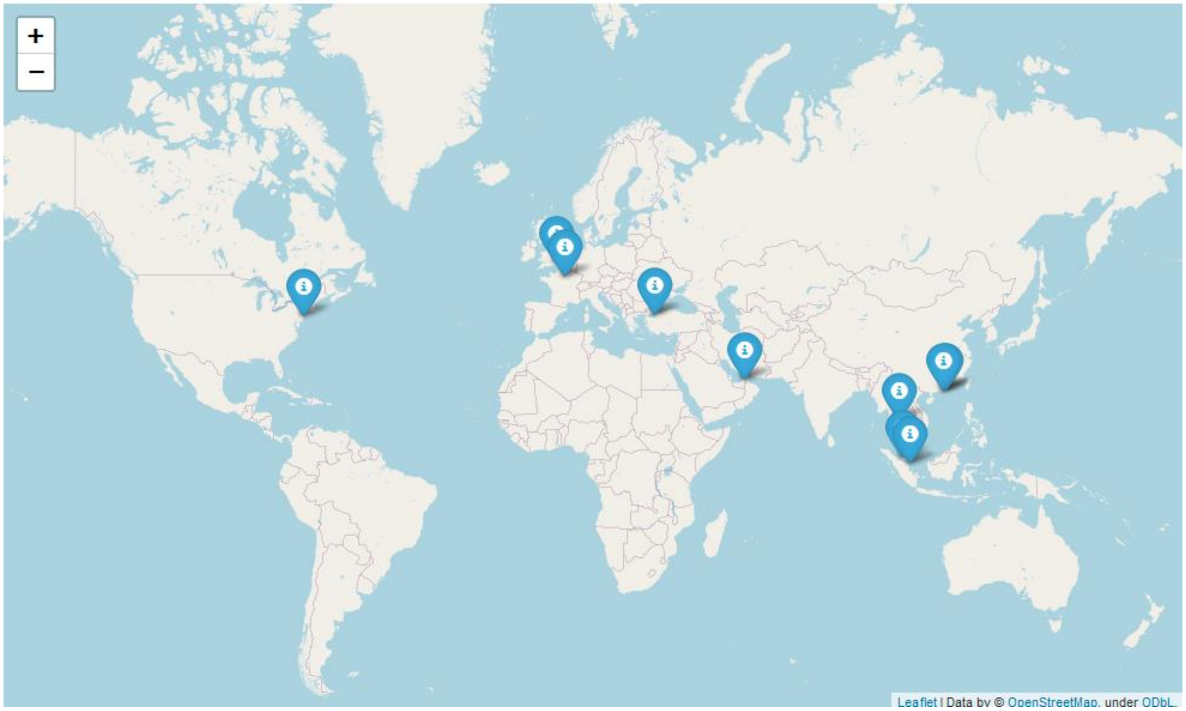
a) Choosing good tourist cities

A good start for choosing our cities is looking for the most popular destinations in the world. According to this Wikipedia article (on 12/14/2020), the 10 most visited cities are:

- Hong Kong, China

- Bangkok, Thailand
- London, United Kingdom
- Macau, China
- Singapore, Singapore
- Paris, France
- Dubai, United Arab Emirates
- New York City, United States
- Kuala Lumpur, Malaysia
- Istanbul, Turkey

After obtaining the coordinates for each of these cities, we can visualize them in a map:

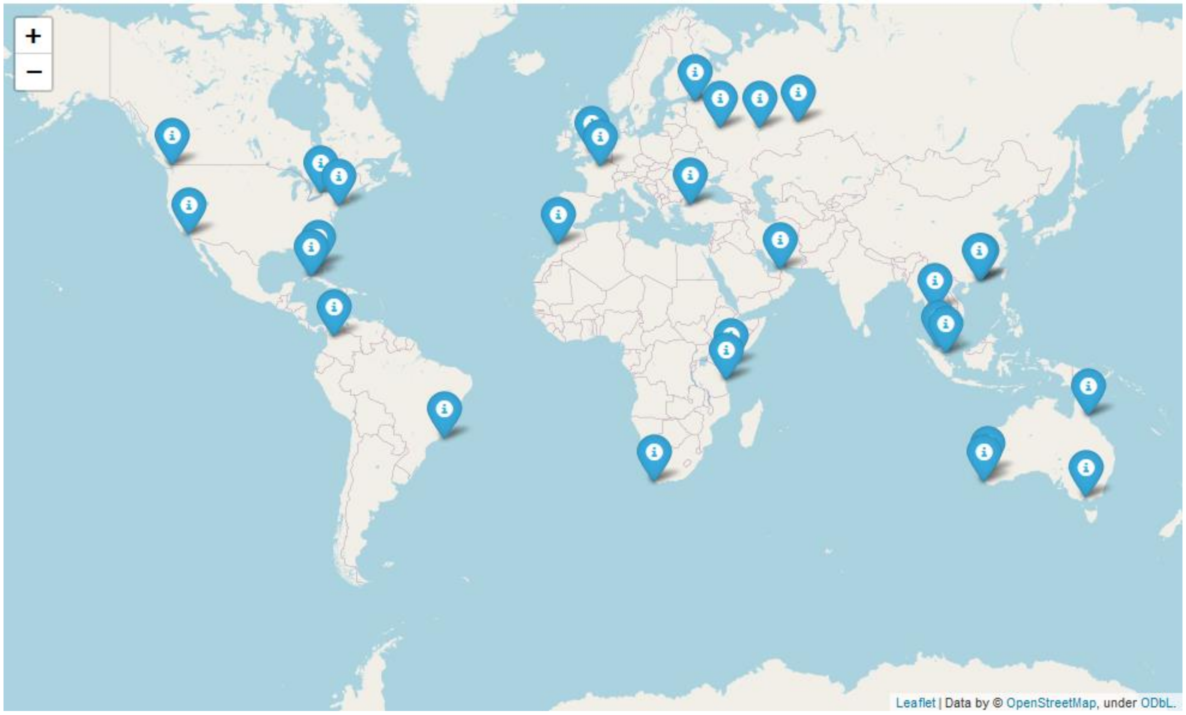


As we can see, most of the cities in our list are in Asia and Europe. But there are also nice tourist cities in North and South America, Africa, Russia and Australia. Let's fix that adding some more cities to our list:

- Latin America
    - Havana, Cuba
    - Medellin, Colombia
    - Rio de Janeiro, Brazil
- North America
    - Miami, USA
    - Los Angeles, USA
    - Toronto, Canada
    - Vancouver, Canada

- Africa
    - Cape Town, South Africa
    - Zanzibar City, Tanzania
    - Lamu, Kenya
    - Essaouira, Morocco
- Russia
    - Moscow, Russia
    - St Petersburg, Russia
    - Kazan, Russia
    - Yekaterinburg, Russia
- Australia
    - Perth, Australia
    - Margaret River, Australia
    - Melbourne, Australia
    - Port Douglas, Australia

The complete good-cities map now looks like this:



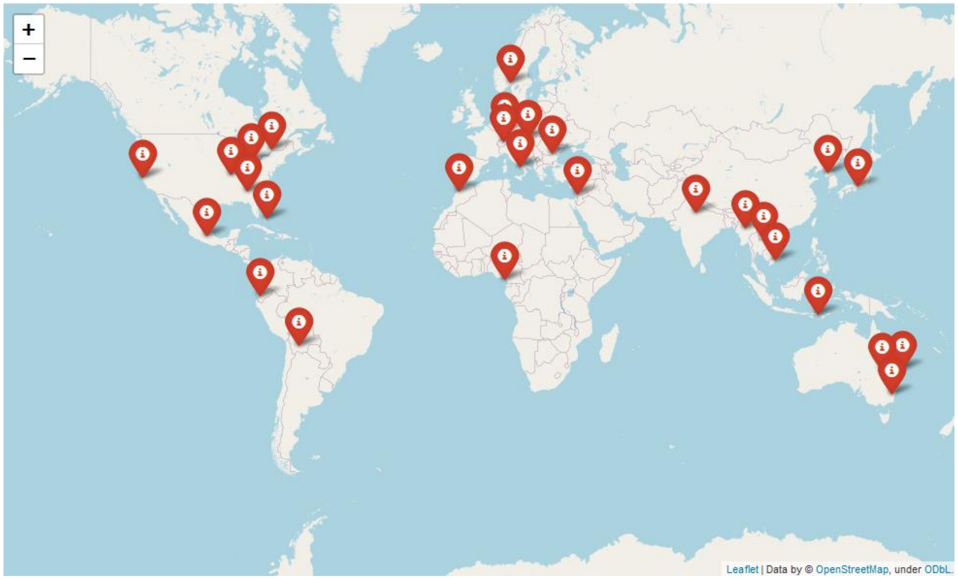
b) Choosing bad tourist cities

Of course, our algorithm not only needs good tourist cities, but it also needs bad ones. I will choose some from the following articles:

- https://journalistontherun.com/2016/01/04/15-worst-travel-destinations/
- https://www.mapquest.com/travel/the-worst-cities-to-visit-in-the-united-states/
- https://www.smartertravel.com/9-boring-cities-world/
- https://leaveyourdailyhell.com/2019/10/14/most-boring-cities-in-the-world/

The list of unattractive cities is the following:

- Cunnamulla, Australia
- Malabo, Equatorial Guinea
- Naples, Italy
- Potosí, Bolivia
- Flores, Indonesia
- Bratislava, Slovakia
- Mandalay, Myanmar
- Saigon, Vietnam
- Pyongyang, North Korea
- St Louis, United States
- Detroit, United States
- Oakland, United States
- Atlatna, United States
- Nagoya, Japan
- Casablanca, Morocco
- Ottawa, Canada
- Frankfurt, Germany
- Nassau, Bahamas
- Zurich, Switzerland
- Canberra, Australia
- Guayaquil, Ecuador
- Agra, India
- Brisbane, Australia
- Bucharest, Romania
- Haifa, Israel
- Mexico City, Mexico
- Oslo, Norway
- Vientiane, Laos

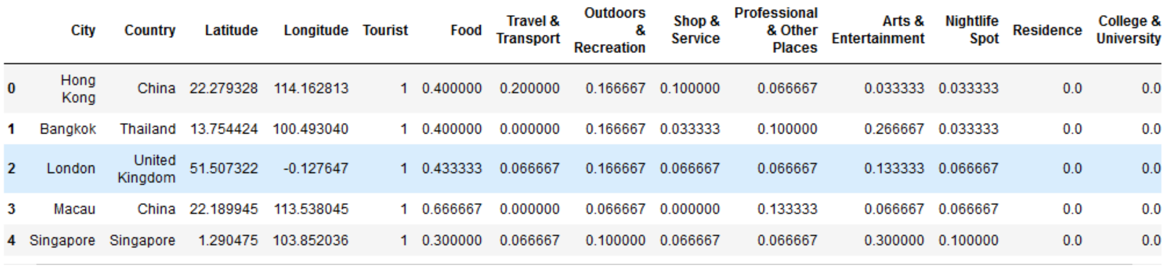


c) *Venue data*

Once the list of cities was completed, I proceeded to use the *Foursquare API* to get venue data from all the cities. The categories in which the venues were grouped are:

- Food
- Travel & Transport
- Outdoors & Recreation
- Shop & Service
- Professional & Other Places
- Arts & Entertainment
- Nightlife Spot
- Residence
- College & University

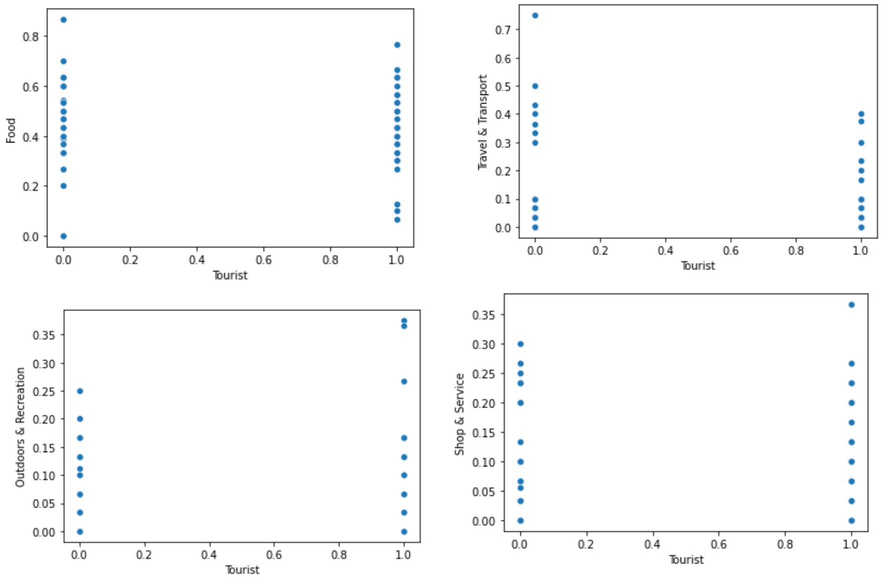
The head of the resulting DataFrame is the following:

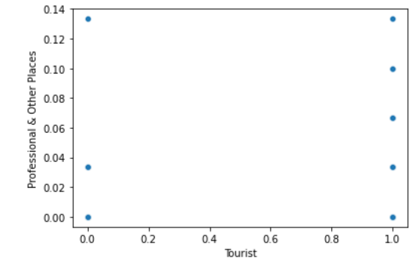


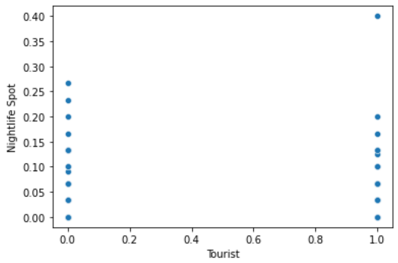
	City	Country	Latitude	Longitude	Tourist	Food	Travel & Transport	Outdoors & Recreation	Shop & Service	Professional & Other Places	Arts & Entertainment	Nightlife Spot	Residence	College & University
0	Hong Kong	China	22.279328	114.162813	1	0.400000	0.200000	0.166667	0.100000	0.066667	0.033333	0.033333	0.0	0.0
1	Bangkok	Thailand	13.754424	100.493040	1	0.400000	0.000000	0.166667	0.033333	0.100000	0.266667	0.033333	0.0	0.0
2	London	United Kingdom	51.507322	-0.127647	1	0.433333	0.066667	0.166667	0.066667	0.066667	0.133333	0.066667	0.0	0.0
3	Macau	China	22.189945	113.538045	1	0.666667	0.000000	0.066667	0.000000	0.133333	0.066667	0.066667	0.0	0.0
4	Singapore	Singapore	1.290475	103.852036	1	0.300000	0.066667	0.100000	0.066667	0.066667	0.300000	0.100000	0.0	0.0

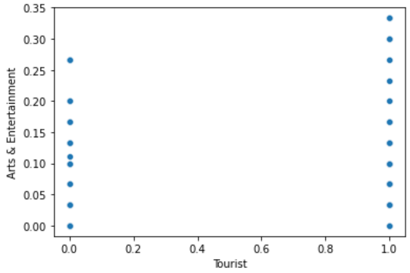
d) *Exploring the data*

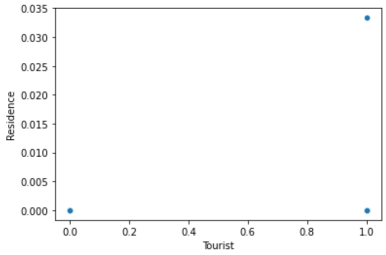
In this section I performed exploratory analysis on the data. The following graphics show the differences in venues between attractive and not-attractive cities:

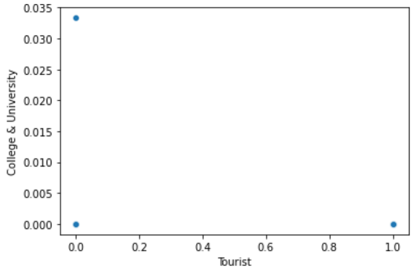










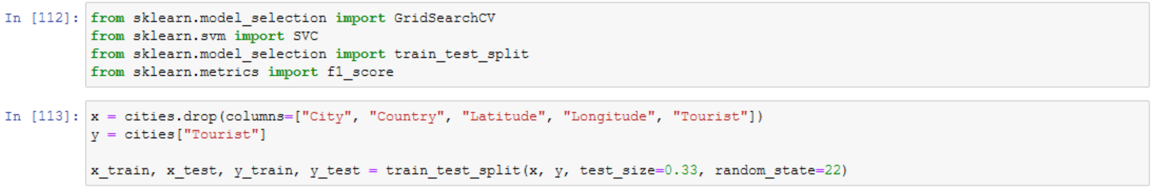


As you can see, the data between attractive and not-attractive cities is similar. This means that venues are not a good predictor of attractiveness. Keep this in mind. Next, I will train a machine learning model with the data.

e) Machine Learning Implementation

It's important to remember the objective of this project is finding out whether a city would be attractive for tourism or not. In other words, we are going to classify cities. There are a lot of classification algorithms, such as Support Vector Machines, Logistic Regression, Naive Bayes, and others.

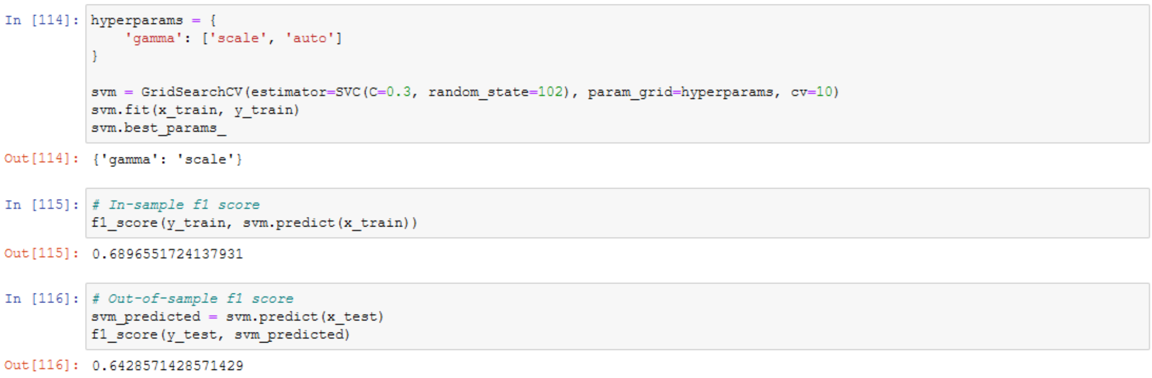
As our dataset is limited to only 57 cities, I will choose an algorithm that does not require much data to work, such as a Support Vector Machine. Also, I will use GridSearchCV to find the best hyperparameters for the model.



```
In [112]: from sklearn.model_selection import GridSearchCV
          from sklearn.svm import SVC
          from sklearn.model_selection import train_test_split
          from sklearn.metrics import f1_score
```

```
In [113]: x = cities.drop(columns=["City", "Country", "Latitude", "Longitude", "Tourist"])
          y = cities["Tourist"]

          x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.33, random_state=22)
```



```
In [114]: hyperparams = {
              'gamma': ['scale', 'auto']
          }

          svm = GridSearchCV(estimator=SVC(C=0.3, random_state=102), param_grid=hyperparams, cv=10)
          svm.fit(x_train, y_train)
          svm.best_params_
Out[114]: {'gamma': 'scale'}

In [115]: # In-sample f1 score
          f1_score(y_train, svm.predict(x_train))
Out[115]: 0.6896551724137931

In [116]: # Out-of-sample f1 score
          svm_predicted = svm.predict(x_test)
          f1_score(y_test, svm_predicted)
Out[116]: 0.6428571428571429
```

IV. Results

As result of this project, we obtained a SVM model that predicts whether a given city is attractive for tourism or not based on its venues. The model has an in-sample f1 score of 0.689, and an out-of-sample f1 score of 0.64. However, the model is unstable and has some issues that are going to be discussed in the following section.

V. Discussion

The model that was obtained as a result has an f1 score of 0.64, which I would not say is very good. In addition to that, it is too dependent on the random state of the functions used. If you change the random state of the train-test-split or the SVM instance, the scores will change dramatically, even getting to 0 sometimes. There are some possible reasons as to why this happens.

a) Issues

The first main issue is the chosen cities to train our algorithm. There is no objective way to measure the attractiveness of a city. This is subjective. The first measure I used to choose the cities was the amount of people visiting them, which doesn't imply they are attractive but that they are relevant.

The second issue is that venue data is really similar across all cities. Every city has restaurants, shops, parks, etc. Because our data was based just on the amount of venues of certain categories, it cannot be a good predictor.

The third issue is the categories of the venues. I chose the main branches of the category tree of Foursquare because it was the fastest way to format the data.

b) Possible solutions

I have come up with some solutions for the issues listed above. I wish I could've solve them in this same notebook, but time and money limitations won't allow me to do so.

1. Find a way to objectively measure the attractiveness of a city.
2. Obtain additional data that could be potential predictors, such as crime rates and venue user-ratings instead of amount.
3. Find better venue categories.

VI. Conclusion

The objective of this project was to find if it is possible to predict how attractive a city is for tourism based on its venues. After exploring the venue data of the dataset I concluded that venues were not a good predictor of the attractiveness of a city, statement that was later confirmed by the trained SVM model, which did not have high accuracy and was very sensitive to changes in random state.

However, the idea of the project is not bad and the results have uncovered hints on what kind of data can lead us to improvement in the future. I myself will be working on training a high accuracy model and I encourage the reader to do it too.