Alumno:	García González Aarón Antonio
Grupo:	3CV19
Unidad de Aprendizaje:	Data Mining
Profesor:	Zagal Flores Roberto Eswart
Parcial #3 Practica #8:	Aplicación de tareas de aprendizaje supervisado
Fecha:	20 de Junio de 2021

Propuesta de solución

A lo largo del semestre se ha trabajado sobre un proyecto semestral el cual se eligió tema “Airbnb” en CDMX, para hacer referencia con la práctica anterior, el nombre propuesto para este proyecto es “análisis de algunos factores de competitividad para prestadores de servicio Airbnb en ciudad de México”, mismo donde se concluyeron con algunos de los hallazgos más interesantes.

La propuesta del proyecto se basa en datos abiertos de Airbnb de CDMX (http://insideairbnb.com/get-the-data.html), el cual se solicitó a la compañía insideairbnb datos desde el 2019 a lo que va del 2021, me fueron otorgados unos días después, obtuve alrededor de 25 archivos .csv, los cuales mediante procesos de limpieza y transformación para concluir con la mezcla de todos los archivos, dando un total de más de 443,000 registros en la tabla de hechos final, posteriormente para esta practica al visualizar de manera gráfica la distribución, se muestran algunos outliers para el conjunto de datos para para algunas dimensiones a utilizar, a modo de no tener sesgo en el modelo a emplear, se omiten dichos valores no comunes, esto se hace mediante limites inferiores y superiores con ayuda de percentil 5% y percentil 95%, es decil nos quedamos con datos entre 5% y 95% de la muestra, al aplicar este filtro, obtenemos alrededor de 367,000 registros ya que se discrimino por múltiples dimensiones, aproximadamente se descarto el 17% del conjunto de datos limpios.

Este dataset final incluye datos como delegación donde se encuentra el AirBnB, identificador del host dentro de la aplicación, el precio, el número de días mínimos, el numero d días disponibles de manera anual, el numero d días que fue ocupado por el mes de datos, coordenadas geográficas, fecha de ultimo review, el total de reviews, etc.

Al final se aplica un modelo de aprendizaje de máquina, el cual se usa SVM, el cual se le dan 4 dimensiones de entrenamiento con el 80% de los datos, para llevar el proceso de aprendizaje con el 20% restante de los datos.

Desarrollo

Vamos a explicar brevemente el código utilizado:

Importar las librerías a utilizar, especialmente énfasis en las funcionalidades de scikitlearn

```
import os
import csv
import datetime
import pandas as pd
import numpy as np
from tabulate import tabulate

from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.svm import SVR
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import precision_score,accuracy_score
from sklearn.preprocessing import StandardScaler
```

Crear una función para omitir outliers dado el dataset, el atrbituo sobre el cual remover los mismos, los rangos de percentiles a considerar.

```
def removeOutliers(data, value, low_percentil, high_percentil):
    min_thresold = data[value].quantile(low_percentil)
    max_thresold = data[value].quantile(high_percentil)

    return data[(data[value] >= min_thresold) & (data[value] <= max_thresold)]


def main():
    path = "../../DataSet/project/TARGET/"
    main_file = "airbnb.csv"

    # Obtener datos de fuente original
    data_x = ["neighbourhood", "room_type", "minimum_nights",
"calculated_host_listings_count"]
    data_y = "price"
    data = data_x + [data_y]

    main_df = pd.read_csv(path + main_file, usecols=data)
    main_df = main_df.sample(frac = 1)
    main_df = main_df.head(-1)


    # Eliminar valores atipicos de algunas columnas
    data_to_put_off_outliers = ['price','calculated_host_listings_count',
'minimum_nights']

    for value in data_to_put_off_outliers:
        main_df = removeOutliers(main_df,value, 0.05, 0.95)
```

Separamos el conjunto de datos original en datos de entrenamiento y prueba mediante la función train_test_split(conjunto x, conjunto y, porcentaje de datos de prueba) de la librería de scikitlearn.

```
    # Separar datos en conjunto de entrenamiento y pruebas
    df_x = main_df[data_x]
```

```
    df_y = main_df[data_y]
    x_train, x_test, y_train, y_test = train_test_split(df_x, df_y, test_size = 0.2)
```

Es posible aplicar el modelo de SVM, también se puso haber utilizado algún tipo de regresión sin embargo la naturaleza de los datos no hace gran cohesión con el modelo.

```
    # Aplicar modelo de aprendizaje de maquina
    clf = SVR(C=1.0, epsilon=0.2)
    parametersSVM = {"C":  [1,10,100, 1000, 10000,100000],
              "gamma": [0.1,0.01,0.001,0.0001,1,10,100]}

    gs_clf = GridSearchCV(clf, parametersSVM, n_jobs=-1)
    gs_clf = gs_clf.fit(x_train,y_train)
    gs_clf.best_score_
    rbf_svc_tunning = gs_clf.best_estimator_

    y_svm2 = rbf_svc_tunning.fit(x_train, y_train)
    score2=rbf_svc_tunning.score(x_train, y_train)
    crossvalue = cross_val_score(rbf_svc_tunning, x_train, y_train, cv = 10)
    res2=rbf_svc_tunning.predict(x_test)

    # Presentacion de resultados
    table = []

    for index, item in enumerate(res2):
        #aux = []
        error = (abs(y_test.iloc[index] - item) / y_test.iloc[index]) * 100

        #aux.append(y_test.iloc[index], item, error)
        table.append([y_test.iloc[index], item, str(error) + " %"])

    print(tabulate(table, headers=['Real','Prediccion', 'Error' ],tablefmt="grid",
numalign="center"))

    print("----------------------------------------------------")
    print(rbf_svc_tunning.score(x_test, y_test))
    print("----------------------------------------------------")


if __name__ == "__main__":
    main()
```

```
(base) aarongarcia@Aarons-MacBook-Pro P8-Machine-Learning % python3 source.py
+--------+--------------+-----------------------+
|  Real  |  Prediccion  | Error                 |
+========+==============+=======================+
|  490   |   337.313    | 31.160614408780816 %  |
+--------+--------------+-----------------------+
|  330   |   445.023    | 34.85538778709421 %   |
+--------+--------------+-----------------------+
|  2113  |   645.641    | 69.44432160573933 %   |
+--------+--------------+-----------------------+
|  591   |   998.224    | 68.9042857105395 %    |
+--------+--------------+-----------------------+
|  725   |    732.8     | 1.075855429508418 %   |
+--------+--------------+-----------------------+
|  1195  |    1143.7    | 4.2927828481282955 %  |
+--------+--------------+-----------------------+
|  2000  |   1052.04    | 47.39799313570911 %   |
+--------+--------------+-----------------------+
|  609   |   691.304    | 13.514674470721252 %  |
+--------+--------------+-----------------------+
|  1595  |   985.306    | 38.22531473641988 %   |
+--------+--------------+-----------------------+
|  792   |    917.8     | 15.883867147297535 %  |
+--------+--------------+-----------------------+
|  406   |   609.792    | 50.19495203303723 %   |
+--------+--------------+-----------------------+
|  630   |     1276     | 102.53950710185184 %  |
+--------+--------------+-----------------------+
|  248   |    248.91    | 0.3670662456727314 %  |
+--------+--------------+-----------------------+
|  1358  |    696.73    | 48.69443505413517 %   |
+--------+--------------+-----------------------+
|  1250  |    1212.2    | 3.023966512144016 %   |
+--------+--------------+-----------------------+
|  656   |    622.8     | 5.060955273337316 %   |
+--------+--------------+-----------------------+
|  1368  |    1212.2    | 11.388887638092902 %  |
+--------+--------------+-----------------------+
|  345   |   392.699    | 13.825652710217048 %  |
+--------+--------------+-----------------------+
|  401   |   1052.04    | 162.35414894908175 %  |
+--------+--------------+-----------------------+
|  1100  |   1184.31    | 7.664966463700362 %   |
+--------+--------------+-----------------------+
|  408   |    399.8     | 2.009890110786458 %   |
+--------+--------------+-----------------------+
|  1406  |    1143.7    | 18.655672477605485 %  |
+--------+--------------+-----------------------+
```

1423	1052.04	26.068858939858202 %
1685	1184.31	29.71426521657543 %
994	794.088	20.111855863075334 %
340	499.2	46.823543818875315 %
1783	859.374	51.80179053274306 %
985	995.801	1.0965274449969638 %
937	472.766	49.544681900910994 %
607	341.2	43.78916089617357 %
379	492.8	30.02649941533604 %
529	406.279	23.198652563744457 %
1734	830.715	52.09254625426208 %
725	413.749	42.93120470907733 %
240	380.358	58.48247902367414 %
697	473.799	32.023071623227615 %
300	450.2	50.066655400965146 %
910	963.498	5.878927755447139 %
380	356.974	6.059399017362024 %
953	649.323	31.865361656267527 %
1106	671.018	39.32933765728364 %
2083	1239.25	40.50664640257198 %
467	392.699	15.910385042773273 %
1897	1300.13	31.463860800814565 %
996	1022.8	2.6907236186159813 %
550	1206.2	119.30908726922054 %
1856	1023.79	44.83888815500037 %
244	458.031	87.71755027322384 %
1750	762.293	56.44042235080731 %

1574	1203.47	23.5408897099779 %
1342	1212.2	9.672129872511988 %
460	458.801	0.26073846518927535 %
986	1215.05	23.230213584438125 %
1309	457.2	65.07258047468495 %
1612	949.2	41.1166260042525 %
432	386.669	10.493373210260291 %
1243	986.23	20.657262153000822 %
746	995.801	33.48536130472119 %
1516	893.117	41.087237191356664 %
487	493.199	1.27284702776144 %
2007	1053.62	47.50257656865215 %
996	437.598	56.06444596524194 %
339	458.801	35.33941093219155 %
800	839.698	4.962255593327626 %
697	521.464	25.184572848101787 %
321	820.479	155.60108588715062 %
724	566.778	21.715791014634675 %
1160	1281.81	10.500968330584486 %
229	326.89	42.74656080691071 %
459	452.8	1.3508269380925964 %
562	374.315	33.39597150286876 %
287	392.699	36.828746289285306 %
398	343.8	13.61811863213428 %
697	817.848	17.33823668115645 %

Conclusiones

Fue una práctica interesante, el proceso de aprendizaje que requieren los modelos de machine learning son costosos computacionalmente hablando, para este caso tomo más de 3 horas el entrenamiento, fue una interacción superficial con machine learning, también intente realizar predicción con otros modelos pero no se ajustaba bien debido a la naturaleza de los datos, el no considerar outliers fue un gran acierto que no había considerado en procesos anteriores,

Análisis de algunos* factores de competitividad para prestadores de servicio AirBnB en CDMX

-Aarón García

Objetivo

Dados los alcances de la unidad de aprendizaje, se busca aplicar la mayor cantidad de conocimientos adquiridos a lo largo del curso, centrado en datos abiertos de AirBnB y relacionarlo con datos demográficos, territoriales y de transporte en CDMX, donde al final del ejercicio se busca conocer que dimensiones tienen mayor peso o importancia en el éxito o fracaso de una oferta de AirBnB.

Preguntas de mineria a responder

- ¿El número de estaciones de metro y metrobus son un factor importante?
- ¿Qué servicio de transporte es más atractivo?
- ¿Importa el número de hoteles?¿La categoría (en estrellas) son un factor importante?
- ¿Delegaciones más atractivas?
- ¿Precios pico y promedio por alcaldía, temporada y año?
- ¿El número de noches mínimas importa?
- ¿Distribución de tipo de oferta por alcaldía y temporada del año?
- ¿El número de reviews y el ultimo review son factores importantes?
- ¿Dónde hay más y menos competencia para un oferente?

Entendimiento de los datos

Aarón Garcia <a… mar, 18 may. 00:47 (hace 13 días)

para data

Dear http://insideairbnb.com, I'm a college student in Mexico City (CDMX), I'm learning about data mining and one of my projects for this season needs data from some topic, I decided to use data from your site and I ask me if you could share me Mexico CDMX AirBnB Data from 2019 and 2020.

My purpose is only for school, I really hope you consider my request, greetings from Mexico.

--

Saludos y muchas gracias, excelente día.

"Vive como si fueras a morir mañana, aprende como si fueras a vivir siempre"

Aarón Antonio García González

Murray Cox <murray@insideairbn… 30 may. 2021 11:39 (hace 1 día)

para mí, data

Hi

Inside Airbnb is a mission driven activist project with the objective to:

> Provide free data that quantifies the impact of short-term rentals on housing and residential communities; and also provides a platform to support advocacy for policies to protect our cities from the impacts of short-term rentals.

After offering free data across multiple cities and dates for more than 5 years, it's no longer possible to sustain the project by doing so. The new policy is to offer a reasonable amount of free data (last 12 months) that meets the immediate needs of users the project was designed for; and all other data requests will be assessed based on: the amount of data being requested (across time and space); the intended data use; and the identity of the requestor. Requests for a large amount of data and/or of a use outside the mission of the project may be refused or asked to donate to help sustain the project.

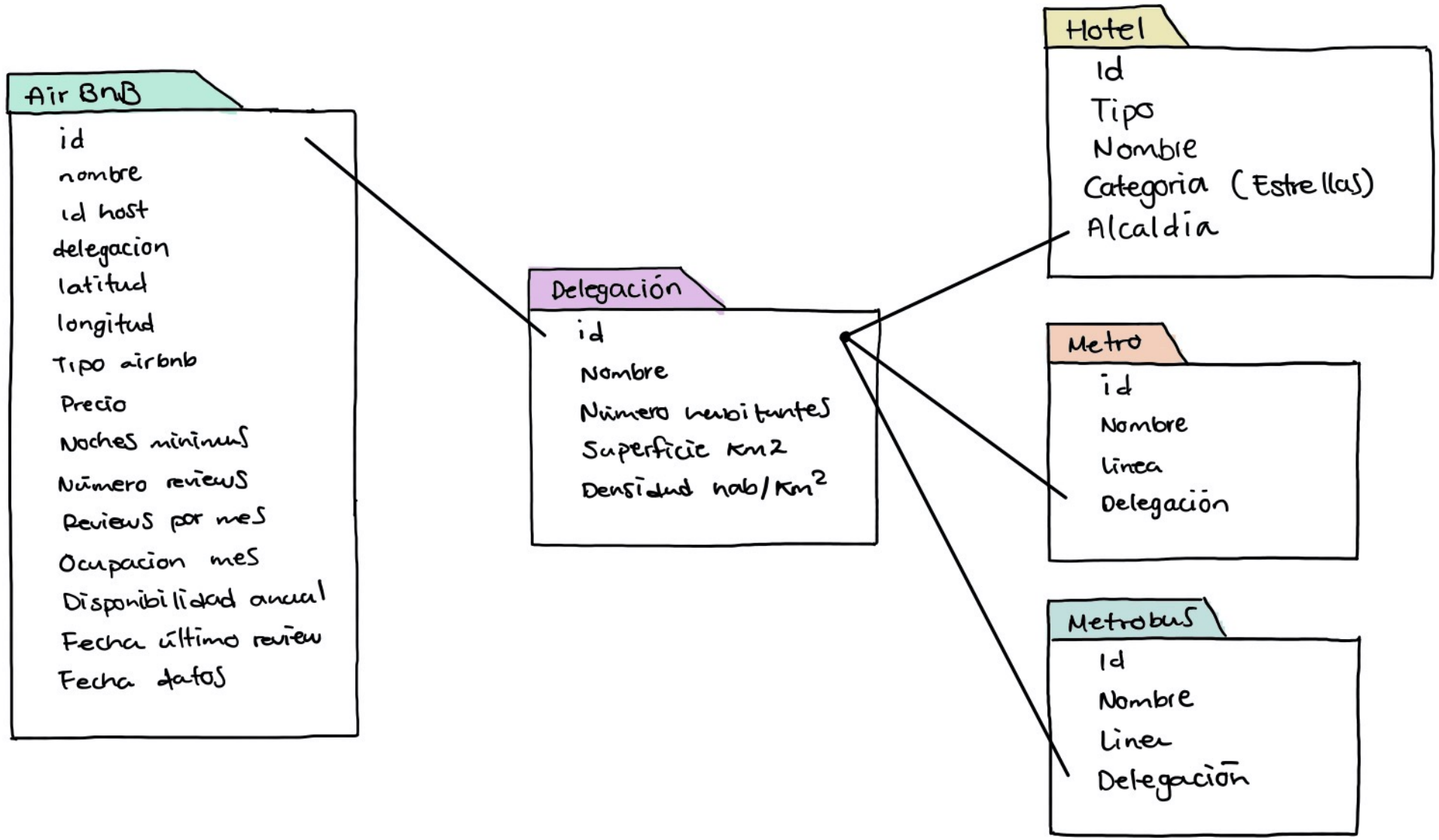
Your request (one city) is modest, and your area of study (learning about data mining) does not particularly align with the mission of the project.

Below are links to the data you requested.

I would also highly encourage you to make some type of donation to the project to help sustain it for others.

--

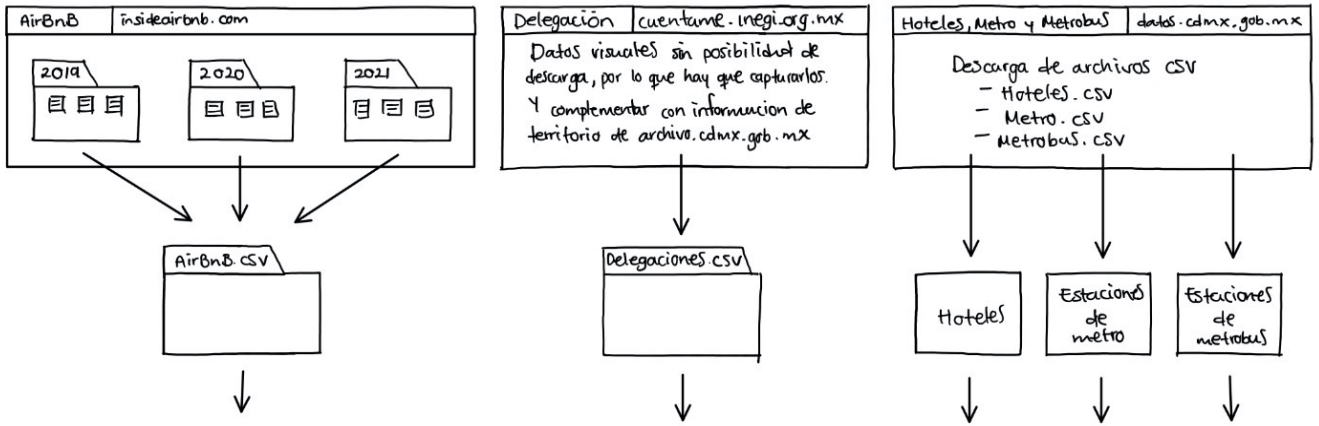
These links expire in 7 days, so please download the data before then.


Air BnB
id
nombre
id host
delegacion
latitud
longitud
Tipo airbnb
Precio
Noches minimus
Número reviews
Reviews por mes
Ocupacion mes
Disponibilidad anual
Fecha último review
Fecha datos
Delegación
id
Nombre
Número habitantes
Superficie km2
Densidad hab/km2
Hotel
Id
Tipo
Nombre
Categoria (Estrellas)
Alcaldia
Metro
id
Nombre
Linea
Delegación
Metrobus
Id
Nombre
Linea
Delegación

pandas

Microsoft® SQL Server®

Extracción

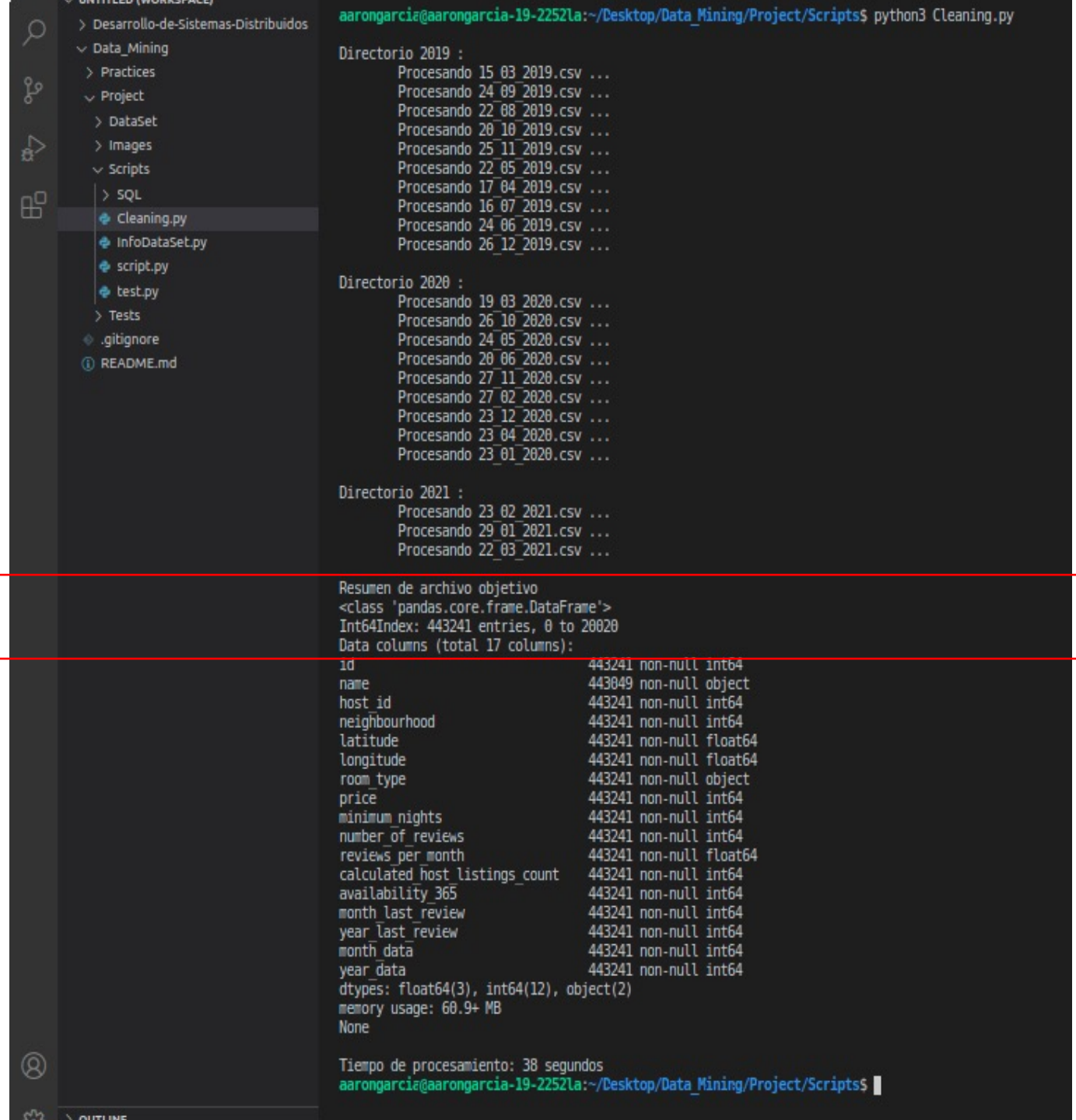


Transformación

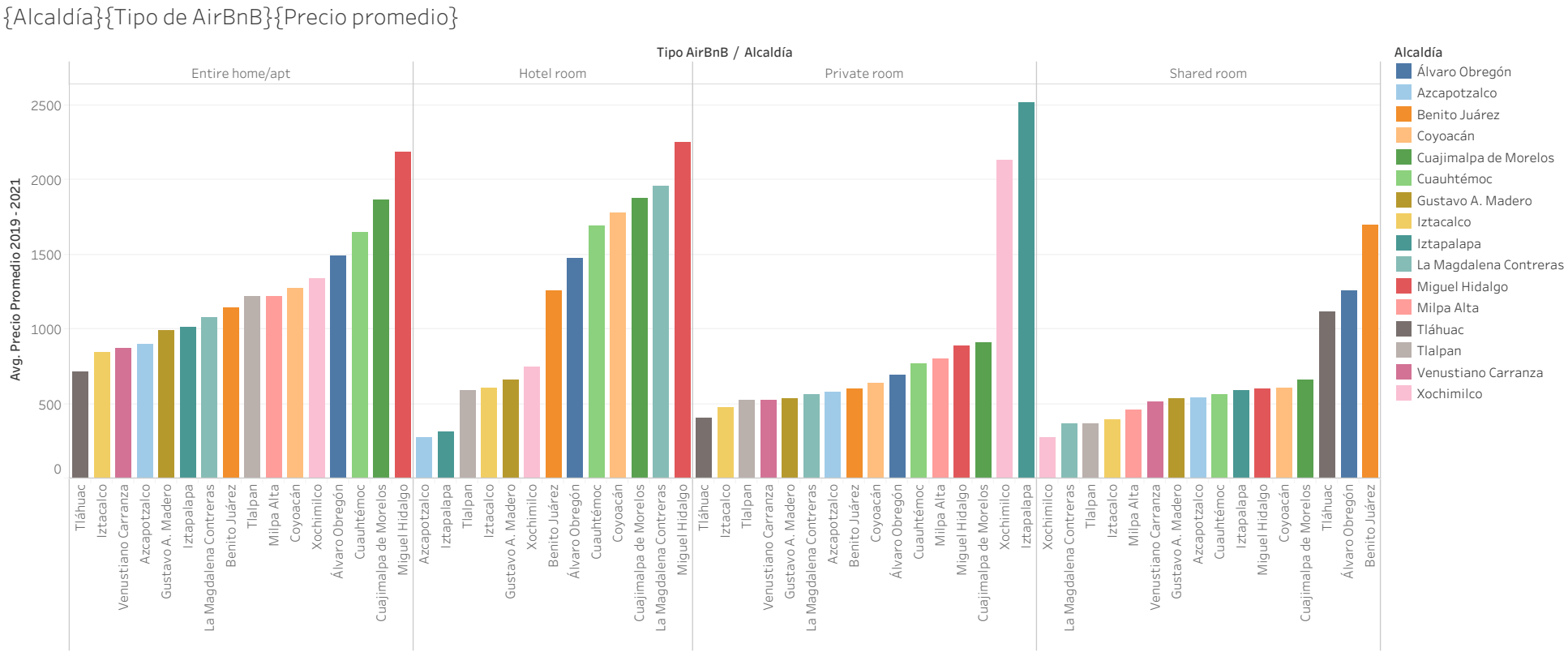
- Rellenar con fecha comodín a los registros que jamás han sido evaluados.
- Separar la fecha de último review en día, mes y año, donde únicamente mes y año serán columnas, posterior a ello eliminamos la fecha de review entera del data set.
- Obtener a partir de cada archivo.csv la fecha de dichos datos, donde el mes y año se agregan como columnas al data set.
- Eliminar el campo "neighbourhood_group" del dataset
- Cambiar el nombre de delegación o alcaldía por el id de alcaldía con base al archivo delegaciones.csv.
- Eliminar registros que incluyan uno o más campos vacíos o no válidos
- Combinar datos de ambas fuentes de tal manera que podamos identificar a cada delegación y además conocer su nombre, número de habitantes, extención superficial en kilometros cuadrados y densidad de población.
- Para hoteles
  - Eliminar columnas: calle_y_num, colonia y CP
  - Cambiar el nombre de delegación o alcaldía por el id de alcaldía con base al archivo Delegaciones.csv.
- Para estaciones de metro y metrobus
  - Eliminar columnas: geometry, stop_id, stop_code, stop_desc, stop_lust, stop_lon, trip_heads, agency, geopoint.
  - Agregar columna "delegacion", la cual se llena con el id de delegacion donde se encuentra la estacion, esto se busca estación a estación en la página del servicio.

Carga

Todos los archivos resultantes en el paso de transformación, resultan archivos .csv que son cargados en SQL Server para la minería siguiente.

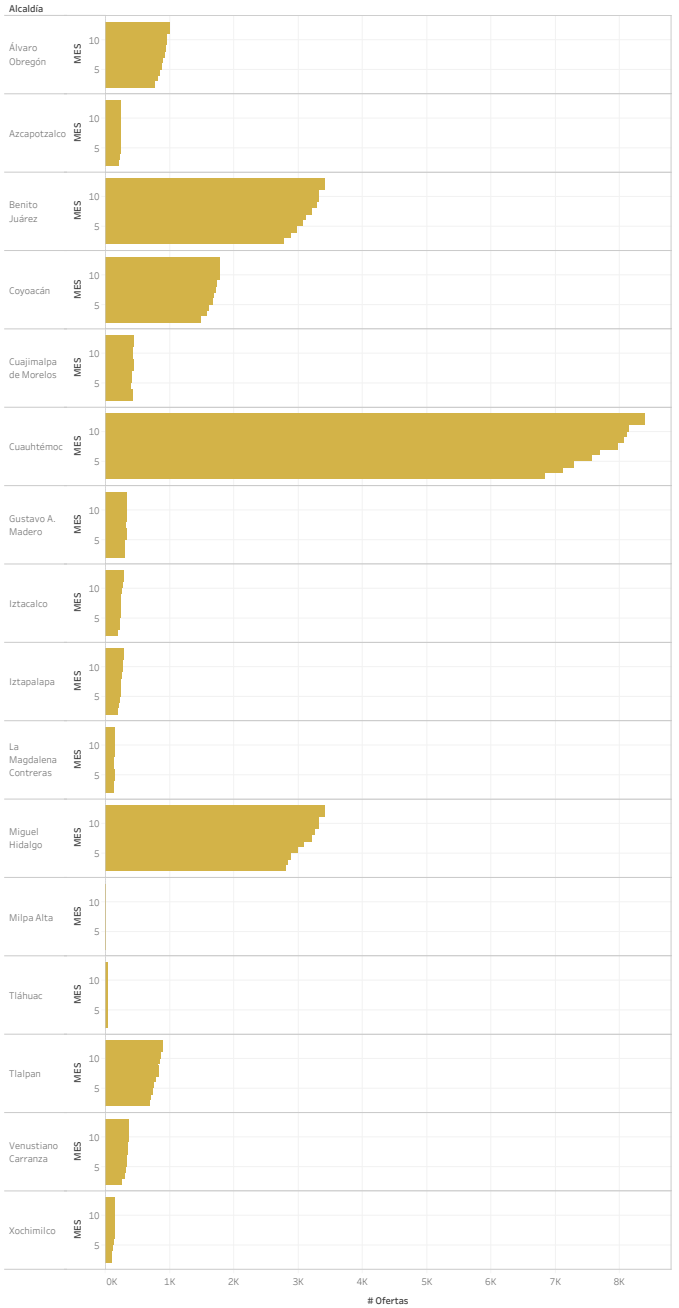

Desarrollo-de-Sistemas-Distribuidos
Data_Mining
Practices
Project
DataSet
Images
Scripts
SQL
Cleaning.py
InfoDataSet.py
script.py
test.py
Tests
.gitignore
README.md
OUTLINE
aarongarcia@aarongarcia-19-2252la:~/Desktop/Data_Mining/Project/Scripts$ python3 Cleaning.py
Directorio 2019 :
Procesando 15_03_2019.csv ...
Procesando 24_09_2019.csv ...
Procesando 22_08_2019.csv ...
Procesando 20_10_2019.csv ...
Procesando 25_11_2019.csv ...
Procesando 22_05_2019.csv ...
Procesando 17_04_2019.csv ...
Procesando 16_07_2019.csv ...
Procesando 24_06_2019.csv ...
Procesando 26_12_2019.csv ...
Directorio 2020 :
Procesando 19_03_2020.csv ...
Procesando 26_10_2020.csv ...
Procesando 24_05_2020.csv ...
Procesando 20_06_2020.csv ...
Procesando 27_11_2020.csv ...
Procesando 27_02_2020.csv ...
Procesando 23_12_2020.csv ...
Procesando 23_04_2020.csv ...
Procesando 23_01_2020.csv ...
Directorio 2021 :
Procesando 23_02_2021.csv ...
Procesando 29_01_2021.csv ...
Procesando 22_03_2021.csv ...
Resumen de archivo objetivo
<class 'pandas.core.frame.DataFrame'>
Int64Index: 443241 entries, 0 to 20020
Data columns (total 17 columns):
id 443241 non-null int64
name 443049 non-null object
host_id 443241 non-null int64
neighbourhood 443241 non-null int64
latitude 443241 non-null float64
longitude 443241 non-null float64
room_type 443241 non-null object
price 443241 non-null int64
minimum_nights 443241 non-null int64
number_of_reviews 443241 non-null int64
reviews_per_month 443241 non-null float64
calculated_host_listings_count 443241 non-null int64
availability_365 443241 non-null int64
month_last_review 443241 non-null int64
year_last_review 443241 non-null int64
month_data 443241 non-null int64
year_data 443241 non-null int64
dtypes: float64(3), int64(12), object(2)
memory usage: 60.9+ MB
None
Tiempo de procesamiento: 38 segundos
aarongarcia@aarongarcia-19-2252la:~/Desktop/Data_Mining/Project/Scripts$

Precio promedio por alcaldia

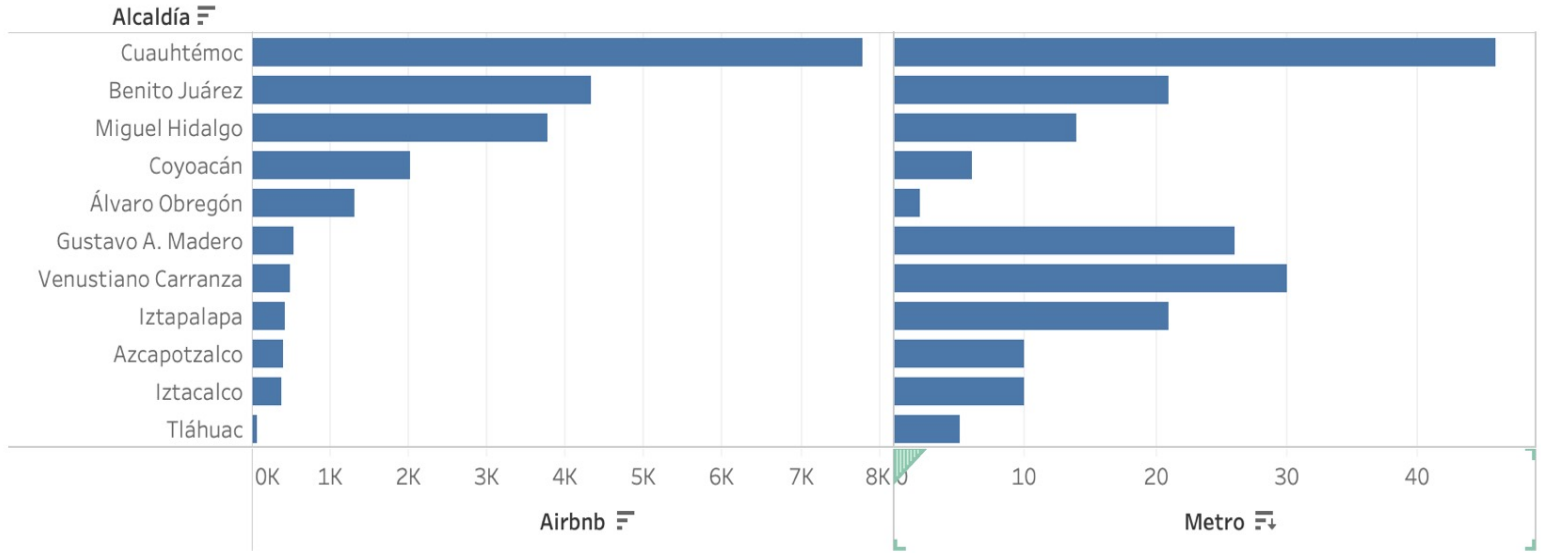


Desde el punto de vista de oferente, es una buena opción ofrecer AirBnB en las delegaciones tales como; Miguel Hidalgo, Cuajimalpa, Cuauhtémoc, Álvaro Obregón, Coyoacán y Xochimilco debido a predominar en precios altos.

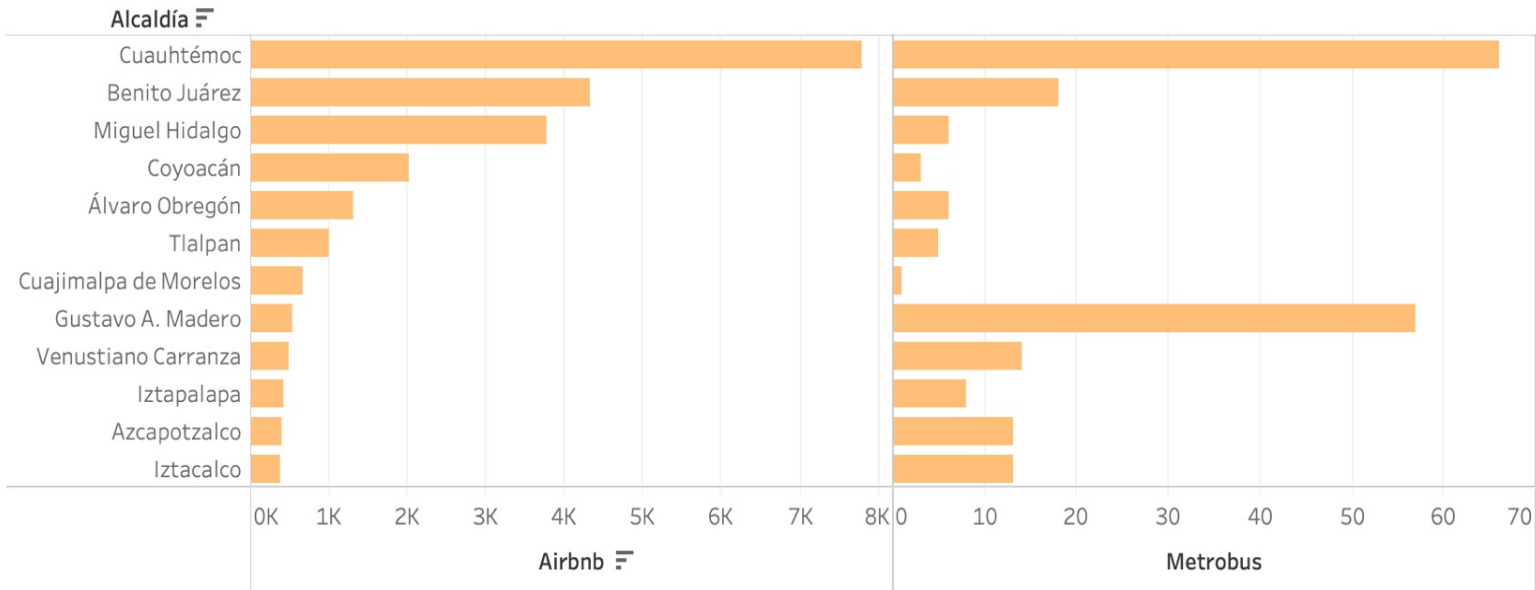
{Alcaldía}{Numero de AirBnB diferentes}{Mes} 2019



{Alcaldía}{Numero de AirBnB diferentes}{Numero de estaciones de metro}



{Alcaldía}{Numero de AirBnB diferentes}{Numero de estaciones de metrobus}



Un detalle muy importante, la alcaldía Cuauhtémoc siendo la mas atractiva y con mayor oferta, es de las que menor numero de días mínimos de estadía pide, mientras que las otras dos alcaldías mas exitosas en AirBnB (Benito Juárez y Miguel Hidalgo), son de las que mayor numero de días mínimos de estadía piden, además que los precios de estas 3 alcaldías son de los más altos de la ciudad, un contraste muy interesante.