به نام خدا

گزارش مربوط به دیتاست Airbnb

زهرا احمدنژاد

شماره دانشجویی:400422011

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

اول کتابخانه هایی که از آنها در اینجا استفاده میکنیم را ران میکنیم.

from google.colab import files

files.upload()

!mkdir -p ~/.kaggle

!cp kaggle.json ~/.kaggle/

!chmod 600 ~/.kaggle/kaggle.json

!kaggle datasets download -d dgomonov/new-york-city-airbnb-open-data

import zipfile

zip\_ref = zipfile.ZipFile('new-york-city-airbnb-open-data.zip', 'r')

zip\_ref.extractall('files')

zip\_ref.close()

!unzip \\*.zip  && rm \*.zip

بعد با این روش دیتاست مربوط را از کگل در کولب ایمپورت و دانلود میکنیم. چون فایل دانلود شده بصورت زیپ است بنابراین آنرا آنزیپ میکنیم.

#Read Dataset

ab\_nyc = pd.read\_csv('/content/files/AB\_NYC\_2019.csv')

در اینجا دیتاست را میخوانیم.

#checking type of every column in the dataset

ab\_nyc.dtypes

در اینجا نوع هر ستون در دیتاست را بررسی میکنیم.

ab\_nyc.isnull().sum()

به دنبال داده های از دست رفته یعنی نال میگردیم تا بتوانیم آنها را پاکسازی کنیم. برای اینکه پاکسازی داده انجام دهیم ابتدا نگاه میکنیم ببینیم چه ستونهایی دارای مقادیر از دست رفته یا نال هستند. آیا این ستونها اهمیت خاصی دارند یا خیر. سپس پاکسازی را انجام میدهیم.

ab\_nyc.drop(['host\_name','last\_review'], axis=1, inplace=True)

با استفاده از این تابع دو ستون که دارای مقدار نال هستند و اهمیت چندانی ندارند را حذف کرده. یعنی با حذف این دو ستون تغییر چندانی در دیتاست بوجود نیامده و بدون این مقادیر هم میتونیم به کار خود ادامه دهیم و دچار مشکل نمیشویم.

من ستون host name و last review را حذف میکنم جون به آنها نیازی ندارم و به جای این دو ستون میتوانم از ستون های دیگر مثل review permonth و id استفاده کنم.

#checking if there is any null left in 'review\_per\_month' column

ab\_nyc.fillna({'reviews\_per\_month':0}, inplace=True)

در اینجا ستون review permonth هم دارای مقدار نال است اما چون این ستون برای ما مهم است میخواهم آنرا نگه دارم و آنرا حذف نمیکنم. پس آنهایی که در هر ماه بازدید ندارد و مقدار ندارد یعنی نال است را 0 قرار میدهم. مثل این است که 0 تا دارد و به همان معنی است. پس اینها را با 0 پرکرده که نگه داریم و بعد از اینها اسفاده کنیم.

#Tedad agahi yekta

ab\_nyc['id'].nunique()

پس از اینکه پاکسازی انجام دادیم اول از همه نگاه میکنیم ببینیم کلا چند آگهی داریم.

برای پیدا کردن تعداد آگهی من از آی دی استفاده میکنم چون هر آی دی یک فرد است و تعداد آی دی میتواند نشان دهنده تعداد آگهی باشد. 48895 آی دی داریم که اگر این را برابر با تعداد آگهی یکتا قرار دهیم یعنی 48895 آگهی وجود دارد.

#Tedad agahi dar har mantaghe joghrafiayi

ab\_nyc.groupby('neighbourhood').count()['id'].sort\_values(ascending=False)

برای پیدا کردن تعداد آگهی در هر منطقه جغرافیایی من از neighbourhood و id استفاده میکنم. چون آی دی را در تابع قبل برابر با تعداد آگهی قرار دادم بنابراین برای پیدا کردن تعداد آگهی در هر منطقه جغرافیایی از تابع بالا استفاده میکنم تا تعداد آگهی در هر منطقه جغرافیایی را اینگونه پیدا کنم.

#Tedad agahi be ezaye har neighbourhood group

ab\_nyc.groupby('neighbourhood\_group').count()['id'].sort\_values(ascending = False)

در اینجا تعداد آگهی به ازای هر neighbourhood group و id را به دست آوردم. که تعداد آگهی در هر منطقه را به ما نشان میدهد. با استفاده از این تابع متوجه میشویم که منتهن و بروکلین دارای بیشترین تعداد آگهی هستند و این به این معنی است که منتهن و بروکلین دو مقصد پر مسافر هستند. پس از این دو به ترتیب کوئینز، برونکس و جزیره استاتن دارای بیشترین تعداد آگهی هستند.

plt.scatter(x=ab\_nyc['price'],y=ab\_nyc['neighbourhood\_group'])

این نمودار neighbourhood group و price را نشان میدهد. و میتوانیم با استفاده از این قیمت را در هر منطقه باهم مقایسه کنیم.

در منتهن و بروکلین و کوئینز قیمت 10000 هم وجود دارد. اما در برونکس و جزیره استاتن اینگونه نیست. در برونکس بالاترین قیمت حدود 3000 و در جزیره استاتن حدود 5000 است.

همانطور که مشاهده میشود در کوئینز قیمت از 3000 به 1000 تبدیل میشود یعنی بین 3000 و 10000 هیچ قیمتی وجود ندارد.

اما در منتهن و بروکلین قیمت به نسبت از توزیع خوبی برخوردار است.

#10 mahale bartar ke bishtarin tedad agahi ra darand

ab\_nyc.neighbourhood.value\_counts().head(10)

پس از بخش، 10 محله که بیشترین تعداد آگهی را داشتند پیدا کردم.

ویلیامزبورگ، بدفورد استیووزانت، هارلم و بوشویک 4 محله هستند که بیش از 2000 آگهی دارند. و محله هایی که پس از آنها هستند کمتر از 2000 و بیشتر از 1000 آگهی دارند.

#room\_type

ab\_nyc.room\_type.unique()

با استفاده ازین تابع نوع اتاق هایی که وجد دارند را به دست می آوریم.

سه نوع اتاق وجود دارد. اتاق مشترک، اتاق خصوصی و کل خانه.

plt.figure(figsize = (15,5))

plt.scatter(x=ab\_nyc['price'],y=ab\_nyc['room\_type'])

plt.xticks(rotation=45)

plt.show()

در این نمودار نیز قیمت را بر اساس نوع اتاق نشان میدهد.

اتاق مشترک نسبت به کل خانه و اتاق خصوصی دارای قیمت کمتری است. و توزیع قیمت در کل خانه نسبت به اتاق خصوصی بهتر است.

#gereftan 10 mahale bartar

sub\_7=ab\_nyc.loc[ab\_nyc['neighbourhood'].isin(['Williamsburg','Bedford-Stuyvesant','Harlem','Bushwick',

                 'Upper West Side','Hell\'s Kitchen','East Village','Upper East Side','Crown Heights','Midtown'])]

#estefade az catplot

viz\_3=sns.catplot(x='neighbourhood', hue='neighbourhood\_group', col='room\_type', data=sub\_7, kind='count')

viz\_3.set\_xticklabels(rotation=90)

اکنون محله و نوع اتاق را باهم ترکیب میکنیم. و از catplot برای اینکار استفاده میکنیم تا بتوانیم مقایسه و ترکیب چند ویژگی را باهم ببینیم.

در اینجا میبینیم که از بین سه نوع اتاقی که داشتیم اتاق مشترک به سختی موجود است نسبت به دو اتاق دیگر. و چون منتهن و بروکلین دو مقصد محبوب و پرسفر هستند و بیشترین لیست و آگهی موجود به آنها تعلق دارد از بین این 10 محله برتر تنها این دو محله نشان داده شده اند.

بدفورد استیوزانت و ویلیامزبورگ برای منطقه منتهن و هارلم برای بروکلین جزو محبوب ترینها هستند.

host\_house\_count

میخواهیم تعداد خانه های میزبانی که وجود دارد را به دست آوریم.

top\_host=ab\_nyc.host\_id.value\_counts().head(10)

top\_host

میخواهم ببینم به ازای هر host id چه تعدادی id دارم.

یعنی چه کسانی میزبان برتر هستند و چه تعداد آگهی دارند.

top\_host\_df=pd.DataFrame(top\_host)

top\_host\_df.reset\_index(inplace=True)

top\_host\_df.rename(columns={'index':'Host\_ID', 'host\_id':'P\_Count'}, inplace=True)

top\_host\_df

10 میزبان برتر و تعداد آگهی های هرکدام را به دست آوردیم.

با استفاده از این میتوانیم ببینیم که میزبان برتر 327 آگهی دارد و بیشترین استفاده ازین پلتفرم را نسبت به بقیه دارد.

viz\_1=sns.barplot(x="Host\_ID", y="P\_Count", data=top\_host\_df,

                 palette='Blues\_d')

viz\_1.set\_title('Hosts with the most listings in NYC')

viz\_1.set\_ylabel('Count of listings')

viz\_1.set\_xlabel('Host IDs')

viz\_1.set\_xticklabels(viz\_1.get\_xticklabels(), rotation=45)

با استفاده ازین تابع میزبان برتر و تعداد آگهی هایی که هرکدام دارند را روی نمودار نمایش دادیم.

محور x ، host id و محور y ، count of listing یعنی تعداد آگهی است.

#Kasani ke bishtarin tedad comment ro gereftan

hosts\_with\_high\_customers = ab\_nyc.groupby('host\_id').sum()['reviews\_per\_month'].reset\_index().sort\_values(by='reviews\_per\_month',ascending = False).iloc[:10]

hosts\_with\_high\_customers.head()

برای اینکه ببینم چه کسانی بیشترین تعداد کامنت را گرفتن از host id و review permonth استفاده میکنم.

میخواهم ببینم کدام میزبان در مجموع بیشترین بازدید را گرفته است.

high\_customers\_agahi = ab\_nyc[ab\_nyc['host\_id'].isin(hosts\_with\_high\_customers['host\_id'])]

high\_customers\_agahi.groupby(['host\_id','neighbourhood\_group']).count().sort\_values(by='id', ascending = False)['id']

بعد نگاه میکنم ببینم چه کسانی بیشترین بازدید و کامنت را داشتند و در چه منطقه و محله ای هستند. این به این معنی است که بیشترین مشتری را داشتند.

high\_customers\_agahi.groupby('neighbourhood\_group').count()['id'].reset\_index().sort\_values(by='id')

در کل دیتافریم اصلی را که نگاه کنیم بیشترین تعداد آگهی و مشتری را داشتند در منهتن و بروکلین و کوئینز بوده است.

high\_customers\_agahi.groupby(['neighbourhood\_group']).mean()['price']

بعد قیمت را در این سه منطقه که بیشترین مشتری را داشتند بررسی میکنیم که ببینیم چگونه است.

ab\_nyc[ab\_nyc['price'] < 1000].groupby(['neighbourhood\_group']).mean()['price']

در اینجا میبینیم که قیمت در دو ایالت از قیمت میانگین در آن دو محل پایینتر است. در نتیجه این میتواند یکی از دلایلی باشد که اینها مشتری بیشتری دارند.

high\_customers\_agahi.groupby(['room\_type']).count()['id']

در اینجا تفاوت خاصی مشاهده نمیکنیم.

high\_customers\_agahi.groupby(['minimum\_nights']).count()['id'].reset\_index().sort\_values(by='id',ascending=False)

یکبار این گروه را در نظر میگیریم که بیشترین تعداد مشتری را داشتند و یکبار هم کل را در نظر میگیریم که ببینیم اینها چه فرقی با آنها داشتند.

میخواهیم ببینیم جند شب ماندند که در دیتافریم اصلی 1 شب بوده اما در اینجا دو شب است.

ab\_nyc.groupby(['minimum\_nights']).count()['id'].reset\_index().sort\_values(by='id',ascending=False).head(5)

تابع نشان دهنده اینکه جند شب رزرو کردند در دیتافریم اصلی.

for column in ['price', 'minimum\_nights']:

    box\_and\_whisker(ab\_nyc, column)

در اینجا میگم من بر اساس یه منطقی outlayer ها را پیدا کرده و میریزم دور. منطق این بوده که میگویم من میرم به ازای هر ستون میانشو نگاه میکنم و میبینم که میانه آن سری اعداد کجا بوده. من میام میانه را نگاه میکنم و میبینم 25 درصد اعداد آن ستون در چه بازه ای هستند و 75 درصدش را هم میبینم. چارک اول و چارک سوم. بعد میگویم هر چیزی که کوچکتر از میانه و پایینتر از چارک اول بود را من دور میریزم چون معلوم است که خیلی عجیب کوچک هستند و هر چیزی که بزرگتر بود را هم دور میریزم چون معلوم است که الکی بزرگ است.

در اینجا گفته outlayer ها کجاست و آنها را کشیده.

در اینجا این نمودار میگوید من برای price این را کشیده ام که میانه آن 106 بوده یعنی میانه قیمتها 106 بوده .Q1 همان 25 درصد اول اعداد است و price آن 69 بوده. و q3 یا همان 75درصد 175 بوده

در اینجا یک مستطیل سمت چپ تشکیل شده که نزدیک 0 است و بقیه یک سری نقطه افتاده است به اینها outlayer میگوییم بخاطر اینکه از آن مستطیلی که ما میخواستیم همه چیزها در آن بیفتند دور افتاده و در نتیجه همه اینها outlayer هستند و باید اینها را دور بریزیم.

نمودار بعدی هم همینکار را انجام داده است. برای minimum nights هم همین کار را انجام میدهیم. میبینم میانه آن 3، q1 آن 1و q3 آن 5 است.یعنی مینیموم شبی که اینها میماندند، همه را که نگاه میکردیم 3 شب بوده، 25 درصد آنها 1 شب بوده، 75 درصد آن 5 شب بوده ولی در داده ها دیدیم که کسی بوده که مثلا 1000 شب هم داشته پس آن قطعا outlayer است.

for column in ['price', 'minimum\_nights']:

    ab\_nyc = remove\_all\_outliers(ab\_nyc, column)

    print(f"{column} has {count\_outliers(ab\_nyc, column)} outliers")

    box\_and\_whisker(ab\_nyc, column)

در اینجا outlayer ها را ریمو کرده با آن فانکشنهایی که در بالا تعریف کرده و گفته اینهایی که به درد نخور است را من دور میریزم و دوباره رسم میکنم و پس از این کارها این مستطیل آبی رسم شده است که نقطه عجیب و غریبی خارج از این مستطیل نیفتاده است.

import matplotlib.image as mpimg

city\_img=mpimg.imread('/content/files/New\_York\_City\_.png', 0.2)

plt.figure(figsize=(20, 10))

plt.imshow(city\_img, extent= [-74.244420, -73.712990, 40.499790, 40.913060], alpha=0.5)

sns.scatterplot(x='longitude',  y='latitude', hue = "room\_type", data=ab\_nyc, alpha = 0.6);

در اینجا از همه طول و عرض های جغرافیایی استفاده کرده و اینها رو بهش داده و گفته اینها را در این نقشه رسم کن.

و به نقشه هم طول و عرض جغرافیایی نیویورک را دادیم که بداند دقیقا کجاست و نقطه ها را در مقایسه با آن سر جای درست قرار دهد. و هر رنگ متعلق به این نقطه در سمت راست نوع اتاقش را نوشته است.

و اینکه کل خانه و اتاق خصوصی به نسبت دارای مشتری بیشتری است میتواند به این دلیل باشد که بیشتر در لبه ساحلی هستند.

اینا چیزایی بود که من میدونستم و هم از جاهای مختلف خونده بودم هم وقتی از دوستام پرسیدم راهنماییم کردن درباره این نقشه.