תרגיל מסכם –

מגישים :

ליאור יעקובוביץ – 207258625

אביב ארז – 211356027

לינק למקור הנתונים **-**<https://www.kaggle.com/datasets/joebeachcapital/airbnb/data>

תיאור טכני על המאגר-89 עמודות, 1048576 שורות, 1.94 גיגה.

תיאור תוכן המאגר –

מה יש במאגר?-

מערך הנתונים של קגל, "Airbnb Global Listings", מכיל מידע מפורט על רישומי Airbnb ברחבי העולם. נתונים אלה כוללים תכונות שונות כגון:

פרטי רישום:שם נכס, מיקום, סוג, סוג חדר, מחיר, מינימום לילות.

פרטי מארח: מזהה מארח, שם מארח וספירת רישום מארח.

זמינות: זמינות עבור תאריכים ספציפיים וחלונות הזמנה.

ביקורות: מספר הביקורות, ציוני הסקירה ותאריך הסקירה האחרון.

מיקום גיאוגרפי: קואורדינטות קו אורך ורוחב של הרישומים.

מקורות מידע-המידע במאגר מגיע מ-<http://insideairbnb.com/get-the-data.html> **.**

פירוט לגבי עמודות קריטיות במסד

**Property Type**- סוג מקום האירוח, האם זאת דירה, האם מדובר בבית, או אפילו מדובר בחדר בתוך בית עם שותפים נוספים. העמודה הזאת חשובה כי היא יכולה להעיד על ההבדלים במחירים בין סוגים שונים של מקומות אירוח, נוכל לראות בעזרת זאת היכן יש יותר בתים, היכן יש יותר דירות פרטיות, אפילו זה יכול להעיד על מקומות אירוח הקרובים לאזורים של צעירים (באזורים בהם יש יותר דירות שותפים).

**Bedrooms**- כמות חדרים במקום האירוח, יש לזה השפעה רבה בבחירות מקום אירוח לתיירים. ניתן ללמוד היכן יש מקומות אירוח עם יותר חדרים, היכן יש פחות.

**Price**- מחיר ההשכרה, מחיר השכרה שבועי\חודשי. ניתן ללמוד על כך הרבה פרטים, איך כמות החדרים משפיעה על המחיר, העיר, השכונה, סוג מקום ההארחה ועוד.

**Review Scores Rating**- הדירוג של בית ההארחה, אולי הפרמטר הכי קריטי למזמינים. ניתן לראות איך הביקורות משפיעות על המחיר. באילו מקומות ישנו ממוצע ציונים יותר טוב ומה זה מעיד על אותו מקום ועל האוכלוסייה המבקרת בו ועוד.

**Neighborhood**- השכונה בה בית האירוח שוכן, ניתן ללמוד על שכונות עם ביקוש גבוהה, שכונות יקרות יותר, יקרות פחות, אפשר להבין באילו שכונות יש ציונים גבוהים ובאילו נמוכים.

**City**- העיר בה בית האירוח שוכן, מאוד דומה למה שציינו לגבי השכונה, נוכל להסיק תובנות דומות רק לגבי ערים ולא שכונות.

נתונים חסרים בטבלה (יכלו להיות שימושיים)

**Listing Views** -מספר צפיות במודעה, מאפשר לדעת עד כמה הנכס פופולרי. ככל שיש לנכס יותר צפיות הוא יותר פופולרי - ניתן להסיק כך מה גורם לנכסים להיות פופולריים יותר.

**- Guest Profile Information**

מידע לגבי המתארחים, מידע לגבי מין, גיל, מדינת מוצא ועוד. היה ניתן ללמוד מכך הרבה לגבי הנכס ולהביןלמי הוא מתאים בדיוק ומדוע.

תיאור מערכת סכמטית תאורטית שאוספת מעכלת ומעבדת המידע לכדי צורתו הסופית-

שלב 1 : לכידה (איסוף נתונים)-

**·** איסוף נתונים ממקורות חיצוניים, כגון הורדות של API של Airbnb או של Kaggle .

**·** הפוך את איסוף הנתונים לאוטומטי לפי לוח זמנים קבוע (יומי/שבועי/חודשי).

**·** שמור נתונים גולמיים במערכת אחסון.

שלב 2: **Ingest** (טעינת נתונים)-

**·** העברת נתונים למערכת עיבוד מרכזית.

**·** נקה והכן נתונים: הסר כפילויות , טפל בערכים חסרים , תקן שגיאות.

**·** התאם מבנה נתונים והקצה מזהים ייחודיים.

שלב 3: אחסון נתונים (**Data Storage**)-

· אחסן נתונים מעובדים במסדי נתונים (SQL, NoSQL, Data Warehouse).

· להבטיח נתונים נקיים ומובנים באמצעות יצירת אינדקס לשאילתות יעילות.

· נהל גרסאות נתונים (מקוריות, לאחר ניקוי, לאחר עיבוד).

שלב 4: מחשוב (עיבוד נתונים)-

- בצע ניתוח נתונים מתקדם באמצעות שאילתות SQL , כלים של BI כמו (Tableau, Power BI, Looker)

- יצירת תובנות באמצעות אלגוריתמים של למידת מכונה.

מוצר סופי:

מידע מעובד לניתוח שוק, זיהוי מגמות ותובנות כלכליות.

זרימת נתונים לדוגמה:

1. הורד קבצי CSV מ- Kaggle מדי חודש.

2. העלאת קבצי CSV ל- Data Lake, בצע ניקוי ראשוני.

3. אחסן נתונים נקיים במחסן נתונים עם יצירת אינדקס.

4. ערוך ניתוח, צור דוחות וחפש תובנות באופן חזותי.

תיאור מילולי ופקודות ההרצה בשביל להעלות את הדאטה לתוך הביג דאטה **Warehouse** שבחרנו -

בחרנו להשתמש ב-Wearhouse של DuckDB אחסון ועיבוד נתונים.

פקודות הרצה-

**הגדרת שם לקובץ CSV:**

csv\_file\_name = airbnb-listings.csv'

בפקודה זו, אנו מגדירים את השם של קובץ ה-CSV שבו מצוינים הנתונים שיש לטעון ל-Warehouse.

חיבור ל-**Warehouse** של **DuckDB**: con = duckdb.connect()  
 בשורה זו אנו מתחברים ל-Warehouse של DuckDB באמצעות החיבור המוגדר במשתנה 'con'.

יצירת טבלה וטעינת הנתונים מהקובץ **CSV:**

con.execute("""

CREATE TABLE airbnb\_listings AS

SELECT \* FROM read\_csv\_auto(?,

types={

'Listing Url': 'VARCHAR',

'Scrape ID': 'VARCHAR',

'Last Scraped': 'DATE',

'Name': 'VARCHAR',

'Summary': 'VARCHAR',

'Space': 'VARCHAR',

'Description': 'VARCHAR',

'Experiences Offered': 'VARCHAR',

'Neighborhood Overview': 'VARCHAR',

'Notes': 'VARCHAR',

'Transit': 'VARCHAR',

'Access': 'VARCHAR',

'Interaction': 'VARCHAR',

'House Rules': 'VARCHAR',

'Thumbnail Url': 'VARCHAR',

'Medium Url': 'VARCHAR',

'Picture Url': 'VARCHAR',

'XL Picture Url': 'VARCHAR',

'Host ID': 'VARCHAR',

'Host URL': 'VARCHAR',

'Host Name': 'VARCHAR',

'Host Since': 'DATE',

'Host Location': 'VARCHAR',

'Host About': 'VARCHAR',

'Host Response Time': 'VARCHAR',

'Host Response Rate': 'FLOAT',

'Host Acceptance Rate': 'FLOAT',

'Host Thumbnail Url': 'VARCHAR',

'Host Picture Url': 'VARCHAR',

'Host Neighbourhood': 'VARCHAR',

'Host Listings Count': 'INTEGER',

'Host Total Listings Count': 'INTEGER',

'Host Verifications': 'VARCHAR',

'Street': 'VARCHAR',

'Neighbourhood': 'VARCHAR',

'Neighbourhood Cleansed': 'VARCHAR',

'Neighbourhood Group Cleansed': 'VARCHAR',

'City': 'VARCHAR',

'State': 'VARCHAR',

'Zipcode': 'VARCHAR',

'Market': 'VARCHAR',

'Smart Location': 'VARCHAR',

'Country Code': 'VARCHAR',

'Country': 'VARCHAR',

'Latitude': 'FLOAT',

'Longitude': 'FLOAT',

'Property Type': 'VARCHAR',

'Room Type': 'VARCHAR',

'Accommodates': 'INTEGER',

'Bathrooms': 'FLOAT',

'Bedrooms': 'INTEGER',

'Beds': 'INTEGER',

'Bed Type': 'VARCHAR',

'Amenities': 'VARCHAR',

'Square Feet': 'FLOAT',

'Price': 'FLOAT',

'Weekly Price': 'FLOAT',

'Monthly Price': 'FLOAT',

'Security Deposit': 'FLOAT',

'Cleaning Fee': 'FLOAT',

'Guests Included': 'INTEGER',

'Extra People': 'FLOAT',

'Minimum Nights': 'INTEGER',

'Maximum Nights': 'INTEGER',

'Calendar Updated': 'DATE',

'Has Availability': 'BOOLEAN',

'Availability 30': 'INTEGER',

'Availability 60': 'INTEGER',

'Availability 90': 'INTEGER',

'Availability 365': 'INTEGER',

'Calendar last Scraped': 'DATE',

'Number of Reviews': 'INTEGER',

'First Review': 'DATE',

'Last Review': 'DATE',

'Review Scores Rating': 'FLOAT',

'Review Scores Accuracy': 'FLOAT',

'Review Scores Cleanliness': 'FLOAT',

'Review Scores Checkin': 'FLOAT',

'Review Scores Communication': 'FLOAT',

'Review Scores Location': 'FLOAT',

'Review Scores Value': 'FLOAT',

'License': 'VARCHAR',

'Jurisdiction Names': 'VARCHAR',

'Cancellation Policy': 'VARCHAR',

'Calculated host listings count': 'INTEGER',

'Reviews per Month': 'FLOAT',

'Geolocation': 'VARCHAR',

'Features': 'VARCHAR'

}

) """, [csv\_file\_name])

בפקודה זו אנו מבצעים שני פעולות:

משתמשים בפונקציה 'read\_csv\_auto' שמאפשרת טעינה אוטומטית של קובץ CSV לתוך טבלה ב-DuckDB.

מציינים את שמות העמודות ואת סוגי הנתונים שלהם במאגר הנתונים (VARCHAR, INTEGER, DATE וכו') על מנת להגדיר את מבנה הטבלה.

באמצעות פקודות אלו, אנו מבצעים את הטעינה הראשונית של הנתונים מהקובץ CSV אל תוך מאגר הנתונים של DuckDB, ומאפשרים לנו להתחיל בעיבוד ובנית דוחות על הנתונים במערכת.

השאלות ששאלנו על הדטאה, מה אנחנו יכולים ללמוד מהם ואלו תובנות אנחנו רואים-

שאלה 1-

חישוב המחיר הממוצע ומספר הנכסים של **Airbnb** עבור כל עיר.

המטרה העיקריתשל Query 1 היא לספק תובנות לגבי המחיר הממוצע של דירות Airbnb ומספר הנכסים הזמינים בכל עיר. זה עוזר להבין את דינמיקת התמחור ואת גודל השוק בערים שונות.

שימוש:מארחים, מנהלי נכסים ומשקיעים יכולים להשתמש במידע זה כדי לקבל החלטות מושכלות לגבי אסטרטגיות תמחור, הזדמנויות השקעה ומיצוב שוק. ערים עם מחירים ממוצעים גבוהים יותר עשויות להצביע על ביקוש גבוה יותר או על מיקומי פרמיה, בעוד אלו עם יותר נכסים יכולים להציע שוק גדול יותר או תחרות גדולה יותר.

תובנות:לדוגמה, עיר עם מחיר ממוצע גבוה ומספר מתון של נכסים עשויה להיות שוק אטרקטיבי עבור נכסים יוקרתיים או השכרות יוקרה. לעומת זאת, ערים עם מחירים ממוצעים נמוכים יותר אך מספר רב של רישומים עשויים לשרת יותר את הנוסעים מודעים לתקציב או להציע פוטנציאל להשקעה בנכס בשל שיעורי תפוסה גבוהים.

שאלה 2 -

לחשב את מספר Airbnb רישומים ואת המחיר הממוצע על ידי שירותים.

המטרה העיקרית :שלשאילתה 2 נועדה לספק תובנות לגבי מספר הנכסים של Airbnb והמחירים הממוצעים שלהם בהתבסס על שירותים שונים. ניתוח זה עוזר להבין כיצד נוחיות משפיעה על רישום הפופולריות ,התמחור של נכסים שונים.

שימוש:מארחים, מנהלי נכסים ומשקיעים יכולים למנף נתונים אלה כדי לייעל את רישומי הנכסים, להתאים את השירותים לדרישות השוק, ולהגדיר אסטרטגיות תמחור תחרותיות. ההבנה אילו שירותים מושכים יותר הזמנות או מחייבים מחירים גבוהים יותר יכולה לשפר את רווחיות הנכס.

תובנות:לדוגמה, שירותים כגון "Wi-Fi", "בריכת שחייה" או "מרפסת" עשויים להשפיע באופן משמעותי על הפופולריות של הרישום ועל התמחור. נכסים המציעים שירותים רצויים עשויים למשוך יותר אורחים ולהצדיק תעריפי שכירות גבוהים יותר.

שאלה 3-

חישוב ממוצע ציוני הביקורת עבור כל סוג נכס הרשום ב- Airbnb.

המטרה של **QUERY** 3**-**היא לספק תובנות לגבי ציוני הסקירה הממוצעים בסוגי מאפיינים שונים. זה עוזר להבין את האיכות הנתפסת של סוגים שונים של מקומות אירוח זמין ב- Airbnb.

שימוש:בעלי נכסים, מנהלים ואורחים פוטנציאליים יכולים למנף מידע זה כדי לאמוד את איכות האירוח על פי סוג הנכס. הוא מסייע בקבלת החלטות בנוגע להעדפות ההזמנה ואסטרטגיות ניהול ההשכרה. ציונים גבוהים יותר בסקירה עשויים להצביע על שביעות רצון טובה יותר של הלקוחות ועזרה במשיכת אורחים נוספים.

תובנות:לדוגמה, סוגי נכסים עם ציוני ביקורת ממוצעים גבוהים יותר עבור ניקיון ומיקום עשויים להיות מועדפים על ידי מטיילים המחפשים נוחות ונוחות. מצד שני, הבנת ציונים נמוכים יותר יכולה להדגיש תחומים לשיפור כדי לשפר את שביעות רצון האורחים ואת ביצועי הנכס הכוללים.

שאלה 4-

חשב את המספר הכולל של ביקורות ואת ציון הביקורת הממוצע עבור כל שכונה הרשומה ב- Airbnb.

המטרה של **Query** 4 : היא לספק תובנות על הפופולריות ועל ציוני הסקירה הממוצעים של שכונות שונות. זה עוזר להבין אילו שכונות לקבל יותר ביקורות וכיצד הם מדורגים על ידי האורחים.

שימוש:בעלי נכסים, מנהלים ואורחים פוטנציאליים יכולים להשתמש במידע זה כדי להעריך את הפופולריות והמוניטין של השכונה. הוא מסייע בקבלת החלטות מושכלות לגבי השקעות בנכס, העדפות אורח ואסטרטגיות השכרה. שכונות עם ציוני ביקורת גבוהים יותר וסך כל הביקורות עשויות למשוך יותר אורחים ולדרוש מחירי שכירות גבוהים יותר.

תובנות:לדוגמה, שכונות עם מספר גבוה של ביקורות ותוצאות חיוביות של ביקורות ממוצעות עשויות להצביע על מקומות רצויים למטיילים. הבנת שכונות עם ציונים נמוכים יותר יכולה להדגיש תחומים לשיפור כדי לשפר את שביעות רצון האורחים ואת ביצועי הנכס.

שאלה 5-

אילו סוגי דירות הם בעלי המחיר הממוצע הגבוה ביותר בנוסף הממוצע הגבוה ביותר של חדרי שינה?

המטרה של **Query** 5 **:**היא לזהות סוגי נכסים אשר שולטים במחירים ממוצעים גבוהים יותר ומציעים אירוח גדול יותר, כפי שמצוין על ידי המספר הממוצע של חדרי השינה. זה עוזר להבין את פלח השוק עבור סוגי נכסים שונים על בסיס תמחור וגודל.

שימוש:סוכני נדל"ן, בעלי נכסים ומשקיעים יכולים להשתמש במידע זה כדי לאמוד את הביקוש והרווחיות של סוגי נכסים שונים. הוא מסייע בקבלת החלטות מושכלות לגבי רכישות נכסים, אסטרטגיות תמחור ודמוגרפיה ממוקדת.

תובנות:לדוגמה, סוגי נכסים עם מחירים ממוצעים גבוהים ומספר משמעותי של חדרי שינה עשויים להתאים למטיילים אמידים או משפחות המחפשים אירוח מרווח. לעומת זאת, סוגי נכסים עם מחירים נמוכים יותר ופחות חדרי שינה עשויים למשוך מטיילים בעלי מודעות לתקציב או אנשים המחפשים שטחי מגורים קטנים יותר.

שאילתה 6:

שאילתת SQL מעודכנת זו אינה כוללת שורות שבהן המחיר או מספר הביקורות הם Null (אינו תנאי NULL). בנוסף, היא מסננת רשימות שבהן המחיר הוא Null כדי להתמקד רק ברישומים עם מידע מחיר חוקי. זה מבטיח כי הניתוח מבוסס רק על נקודות נתונים מלאות, מתן תובנות משמעותיות יותר על הקשר בין המחיר לבין מספר הביקורות.

שימוש:על-ידי אי הכללת ערכי Null, השאילתה מסייעת להבין כיצד התמחור משפיע על פעילות הסקירה ללא הרעש של נקודות נתונים לא שלמות או חסרות.

תובנות: התוצאות יכולות לספק תובנות ברורות יותר על אסטרטגיות התמחור שמתאימות לספירות סקירה גבוהות יותר, מה שעשוי להצביע על נכסים רצויים יותר או תחרותיים יותר בשוק Airbnb.

השאילתות:

* "" -- Query 1: Average price and number of listings by city
* SELECT
* City,
* AVG(Price) AS Average\_Price,
* COUNT(\*) AS Number\_of\_Listings
* FROM airbnb\_listings
* GROUP BY City
* ORDER BY Average\_Price DESC;
* """

השאילתה שבוחרת שורות - SELECT City, AVG(Price) AS Average\_Price, COUNT(\*) AS Number\_of\_Listings

* "" -- Query 2: Number of listings and average price by amenities
* SELECT
* Amenities,
* COUNT(\*) AS Number\_of\_Listings,
* AVG(Price) AS Average\_Price
* FROM airbnb\_listings
* WHERE Amenities IS NOT NULL
* GROUP BY Amenities
* ORDER BY Number\_of\_Listings DESC;
* ""

השאילתה שבוחרת שורות - SELECT Amenities, COUNT(\*) AS Number\_of\_Listings, AVG(Price) AS Average\_Price

* “””-- Query 3: Average review scores by property type
* SELECT
* "Property Type",
* AVG("Review Scores Rating") AS Average\_Review\_Score,
* AVG("Review Scores Cleanliness") AS Average\_Cleanliness\_Score,
* AVG("Review Scores Location") AS Average\_Location\_Score
* FROM
* airbnb\_listings
* GROUP BY
* "Property Type"
* ORDER BY
* Average\_Review\_Score DESC;
* “””

השאילתה שבוחרת שורות - SELECT "Property Type", AVG("Review Scores Rating") AS Average\_Review\_Score, AVG("Review Scores Cleanliness") AS Average\_Cleanliness\_Score, AVG("Review Scores Location") AS Average\_Location\_Score

* """-- Query 4: Total number of reviews and average review score by neighborhood
* SELECT
* "Neighbourhood",
* SUM("Number of Reviews") AS Total\_Reviews,
* AVG("Review Scores Rating") AS Average\_Review\_Score
* FROM
* airbnb\_listings
* GROUP BY
* "Neighbourhood"
* ORDER BY
* Total\_Reviews DESC;
* """

השאילתה שבוחרת שורות - SELECT "Neighbourhood", SUM("Number of Reviews") AS Total\_Reviews, AVG("Review Scores Rating") AS Average\_Review\_Score

* """-- Query 5: Property types with highest average price and number of bedrooms
* SELECT
* "Property Type",
* AVG("Price") AS Average\_Price,
* AVG("Bedrooms") AS Average\_Bedrooms
* FROM
* airbnb\_listings
* GROUP BY
* "Property Type"
* ORDER BY
* Average\_Price DESC, Average\_Bedrooms DESC;
* """

השאילתה שבוחרת שורות - SELECT "Property Type", AVG("Price") AS Average\_Price, AVG("Bedrooms") AS Average\_Bedrooms

* """ -- Query 6: Relationship between price and number of reviews (excluding NULL values)
* SELECT
* Price,
* "Number of Reviews",
* COUNT(\*) AS Listings
* FROM
* airbnb\_listings
* WHERE
* Price IS NOT NULL
* AND "Number of Reviews" IS NOT NULL
* GROUP BY
* Price, "Number of Reviews"
* ORDER BY
* Price DESC, "Number of Reviews" DESC;
* """

השאילתה שבוחרת שורות - SELECT Price, "Number of Reviews", COUNT(\*) AS Listings

הסיפור שנספר - הבנת הדינמיקה של **Airbnb** תוך התחשבות בפרמטרים שונים

הסיפור שלנו מתמקד בחקר הדינמיקה של שוק Airbnb בערים שונות, ובחינת הגורמים המשפיעים על תמחור, שביעות רצון האורחים ופופולריות המודעות. אנו שואפים לספר סיפור מקיף על המערכת המהפכנית של Airbnb, תוך הדגשת מגמות ודפוסים מרכזיים העולים מתוך ניתוח הנתונים שלנו. בשלב הויזואליזציה ניתן גם להסיק מסקנות לגבי הנושאים שנעלה כאן.

שינויים בתמחור בין ערים:

נבדוק כיצד המחירים הממוצעים שונים בין ערים, אפשר לחשוף אילו מיקומים יקרים יותר או ידידותיים יותר לתקציב למטיילים. חלק זה של הסיפור יסופר באמצעות השאילתה והויזואליזציה הראשונה שלנו.

שביעות רצון האורחים לפי סוגי נכסים:

נבחן כיצד סוגי נכסים שונים מתפקדים במונחים של שביעות רצון האורחים. זה יחשוף האם סוגי נכסים מסוימים מקבלים באופן עקבי דירוגים גבוהים יותר. היבט זה של הסיפור מבוסס על השאילתה השלישית שלנו.

פופולריות ושביעות רצון בשכונות:

נבדוק אילו שכונות הן הפופולריות ביותר (במונחים של מספר הביקורות והציון הממוצע) וכיצד זה קשור לשביעות רצון האורחים. חלק זה של הסיפור מגיע מהשאילתה הרביעית שלנו.

השפעת סוג הנכס וכמות החדרים על המחיר:

האם נכס בעל חדרים רבים יעלה יותר? האם נכס בעל אופי מיוחד יעלה יותר? נרצה לבדוק עד כמה חזקה ההשפעה של סוג הנכס על המחיר, או שבכלל אין כל קשר בין מספר החדרים למחיר, תובנות לגבי הנושא יוכלו לעזור למשתמשים להחליט לגבי סוג הנכס אותו הם רוצים להשכיר, בנוסף יוכלו לעזור למשקיעים עתידיים להחליט איזה נכס לקנות להשקעה ולהעמידו להשכרה בפלטפורמה.

השיטות שנבחרו להקטנת הדאטה:

אגרגציה:

למה בחרנו בשיטה זו:

שאילתה 2: שיטה זו מאפשרת לנו לסכם את מספר הרישומים והמחיר הממוצע עבור כל סט ייחודי של שירותים, ומספקת תובנות לגבי האופן שבו שירותים משפיעים על מספר הרישומים והתמחור שלהם.

שאילתה 3: Aggregation עוזר לנו להבין ציוני חוות דעת ממוצעים עבור סוגי נכסים שונים, וחושף מגמות בשביעות רצון האורחים.

שאילתה 4**:** אנו משתמשים בפונקציות SUM ו- AVG כדי לקבל תצוגה מקיפה של כמות חוות דעת ושביעות רצון לפי שכונה.

שאילתה 5: Aggregation מאפשר לנו להשוות מחירים ממוצעים ומספר חדרי שינה בין סוגי נכסים שונים.

שאילתה 6**:** אנחנו משתמשים בפונקציית COUNT כדי להבין את התפלגות הרישומים על פני רמות מחיר ומספרי חוות דעת.

השיטה מאפשרת לנו לסכם מיליוני מודעות בודדות לסטטיסטיקות משמעותיות ברמת העיר. על ידי חישוב ממוצעים וספירות, אנו יכולים להשוות בין ערים בקלות מבלי להיות מוצפים בפרטים של מודעות בודדות. שיטת ההפחתה הזו שומרת על המידע החיוני (מגמות מחירים ונפחי מודעות) תוך שהיא הופכת את הנתונים לנוחים יותר לניתוח ולהמחשה.

**Filtering**

השתמשנו ב WHERE כדי לצמצם דאטה.

הסיבות שבחרנו בשיטה זו:

שאילתה 1: בשאילתה זו, הסינון מבטיח שאנחנו מנתחים רק רישומים עם נתוני מחיר וחוות דעת תקינים, ובכך משפר את הדיוק של הסטטיסטיקות שלנו ברמת העיר.

שאילתה 2: סינון לפי שירותים ומחירים שאינם חסרים (כלומר, יש להם ערך) מתמקד בניתוח שלנו בנקודות נתונים שלמות.

שאילתה 3: אנחנו מסננים ציוני חוות דעת שאינם חסרים כדי להבטיח שחישובי הממוצע שלנו יתבססו על חוות דעת אמיתיות.

שאילתות 4-6: בשאילתות אלו, פילטרים דומים מבטיחים שאנחנו עובדים עם נתונים תקינים ושלמים עבור כל ניתוח ספציפי.

ובסה"כ תפקידו:

וידוי נתונים שלמים ותקינים: אנחנו מוודאים שאנחנו עובדים עם נתונים שלמים ותקינים על ידי סינון רישומים חסרי מידע קריטי.

שיפור הדיוק: שיטה זו משפרת את דיוק הניתוח שלנו על ידי מניעת ערכים חסרים (ערכי Null) מעיוות התוצאות.

התמקדות בנתונים רלוונטיים: סינון נתונים מאפשר לנו להתמקד בנתונים הרלוונטיים ביותר לכל שאילתה ספציפית.

**Ordering and Limiting**

השתמשנו ב-ORDER BY וגם בLIMIT כדי להתמקד בנקודות הנתונים המשמעותיות ביותר.

מדוע בחרנו בשיטה זו:

סינון (מיון) הנתונים מאפשר לנו לזהות ולהתמקד בנקודות המידע המשמעותיות ביותר (לדוגמה, מחירים ממוצעים הגבוהים ביותר ומספר חדרי שינה).

זיהוי מגמות עיקריות: על ידי סידור נקודות נתונים בסדר מסוים (לדוגמה, מחיר גבוה לנמוך), נוכל לזהות בקלות דפוסים ומגמות מתעוררות. תצוגה ממוקדת זו יכולה להיות תובנה טובה במיוחד בעת ניתוח מאגרי נתונים גדולים, שם תובנות חשובות עלולות להיות קשות לזיהוי.

הדגשת חריגים: סינון נתונים יכול גם לעזור לנו לאתר חריגים, נקודות נתונים שסוטות באופן משמעותי מהנורמה. חריגים אלו עשויים להצדיק חקירה נוספת, וחשיפת ממצאים חשובים או עידוד הערכה מחדש של ההנחות שלנו.

קדימות למידע עבור הדמיה וניתוח: בעת הצגת נתונים באופן ויזואלי או ביצוע ניתוח מעמיק, קביעת קדימות למידע הרלוונטי ביותר היא חיונית. סידור הנתונים מאפשר לנו להציג את נקודות המידע הקריטיות ביותר מראש.

**Window Functions**

השתמשנו בשאילתה מספר 1. שימוש בפונקציות חלון כמו **PARTITION BY** ו-**OVER**.

למה בחרנו בשיטה זו:

שיטה זו מאפשרת לנו לחשב פונקציות כמו (AVG ו- COUNT) עבור כל עיר מבלי להשתמש ב- GROUP BY. פונקציות חלון (Window Functions) מאפשרות לנו לשמור על רמת הפרט של הנתונים המקוריים ועדיין לבצע פונקציות סוכמות. זה מאוד שימושי עבור ניתוחים מורכבים יותר.

## הסברים על סוגי התצוגות שנבחרו, למה נבחרו ומה ניתן לראות בהן:

א) מחיר ממוצע ומספר **listings** (מספר מקומות הלינה) לפי עיר:

* ויזואליזציה: תרשים עמודות למחיר הממוצע, וגרף קו למספר listings
* למה: זה משלב שני מדדים בתרשים אחד, ומאפשר השוואה קלה בין ערים שונות
* מה ניתן לראות: הבדלים במחירים בין ערים, קורלציה בין מחיר לכמות listings

ב) ציוני ביקורות לפי סוג נכס**:**

* ויזואליזציה: מפת חום (Heatmap)
* למה: מציגה ביעילות מספר ציוני ביקורות על פי סוגי נכסים
* מה ניתן לראות: דפוסים בשביעות רצון האורחים וגם אילו סוגי נכסים מצטיינים בתכונות ספציפיות

ג) סך כל הביקורות וציון ביקורות ממוצע לפי שכונה:

* ויזואליזציה: תרשים עמודות לסך כל הביקורות, וגרף קו לציון ביקורות ממוצע
* למה: מציג את הפופולריות (סך כל הביקורות) ואת שביעות הרצון (ציון ממוצע) בו-זמנית
* מה ניתן לראות: אילו שכונות הן המבוקרות ביותר ובעלות הדירוג הגבוה ביותר

ד) מחיר ממוצע וחדרים לפי סוג נכס:

* ויזואליזציה: תרשים פיזור עם קידוד צבע לפי סוג נכס
* למה: חושף קשרים בין מספר משתנים (מחיר, חדרי שינה, סוג נכס)
* מה ניתן לראות: כיצד סוגי נכסים שונים מתומחרים ביחס לגודל שלהם

## קוד פיתון ליישום **הDashboard** של **Streamlit**:

המבנה של הקוד הוא כדלקמן:

פונקציה ראשית (main()): מגדירה את עמוד ה-Streamlit ואת הניווט

פונקציות עמוד נפרדות:

* questions\_page(): מציגה שאלות ניתוח נתונים
* story\_page(): מספקת סקירה ותובנות
* sample\_rows\_page(conn): מציגה דוגמאות של נתונים גולמיים
* avg\_price\_by\_city\_page(conn): מחזיקה נתוני מחיר על פי העיר
* review\_scores\_by\_property\_type\_page(conn): מציגה ציוני ביקורות לפי סוג נכס
* reviews\_by\_neighborhood\_page(conn): מציגה פופולריות ודירוגים על פי שכונות
* price\_bedrooms\_by\_property\_type\_page(conn): מראה קשר בין מחיר לחדרים

הוראות הרצה

1. התקנות:

* הורדת מאגר הנתונים -

<https://www.kaggle.com/datasets/joebeachcapital/airbnb/data>

הקובץ נקרא - airbnb-listings.csv

* התקינו את החבילות הנדרשות באמצעות פקודת ה-pip -

***pip install -r requirements.txt***

קובץ ה-requirements.txt אמור להכיל את רשימת החבילות הנדרשות לתכנית זו.

2. הרצת **main** :

על מנת ליצור את הקובץairbnb\_queries.db הרץ את הmain באמצעות הIDE או באמצעות הטרמינל כך - *python main.py*

**3.** בדיקת מיקום קבצים:

ודאו שקובץ ה-Python ‏(airbnb\_dashboard.py) וקובץ מסד הנתונים של Airbnb ‏(airbnb\_queries.db) נמצאים באותו התיקייה.

4**.** פתיחת טרמינל:

פתחו חלון טרמינל או שורת פקודה ונווטו אל התיקייה שבה נמצאים שני הקבצים הנ"ל.

5. הפעלת התוכנית:

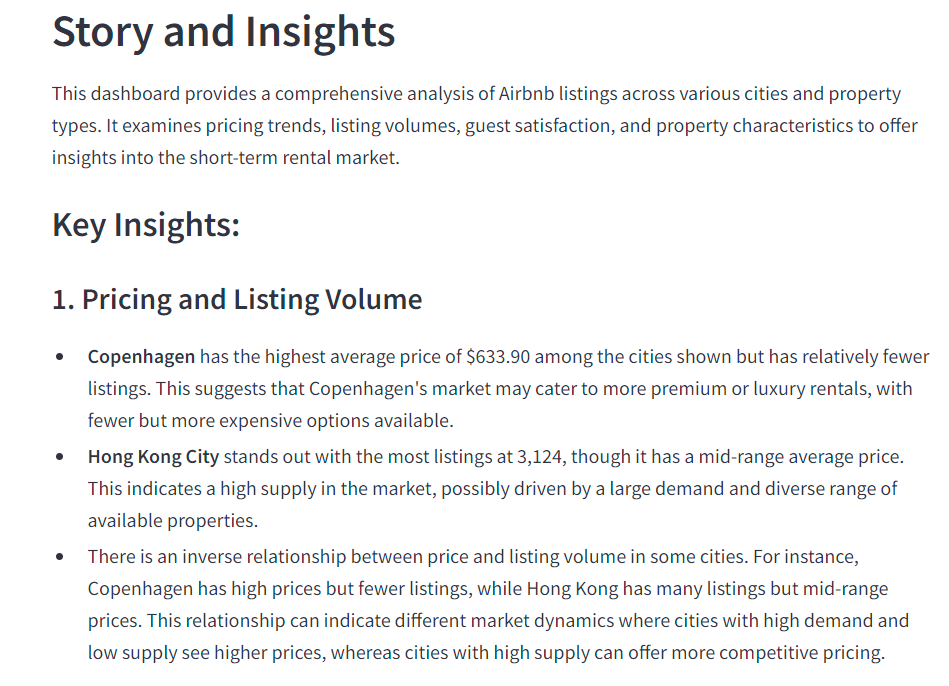
הריצו את הפקודה הבאה בחלון הפקודה:

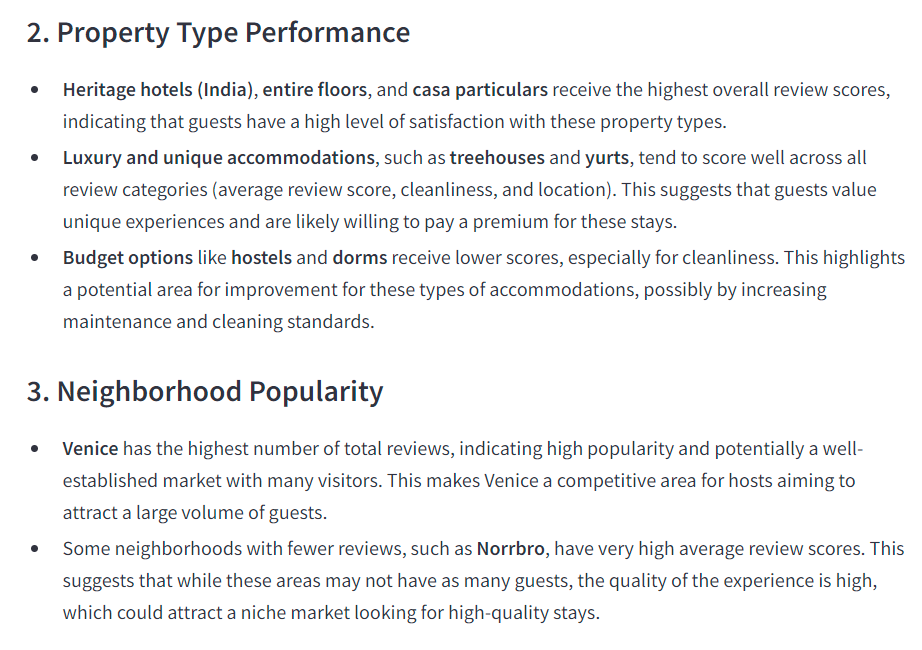
*streamlit run airbnb\_dashboard.py*

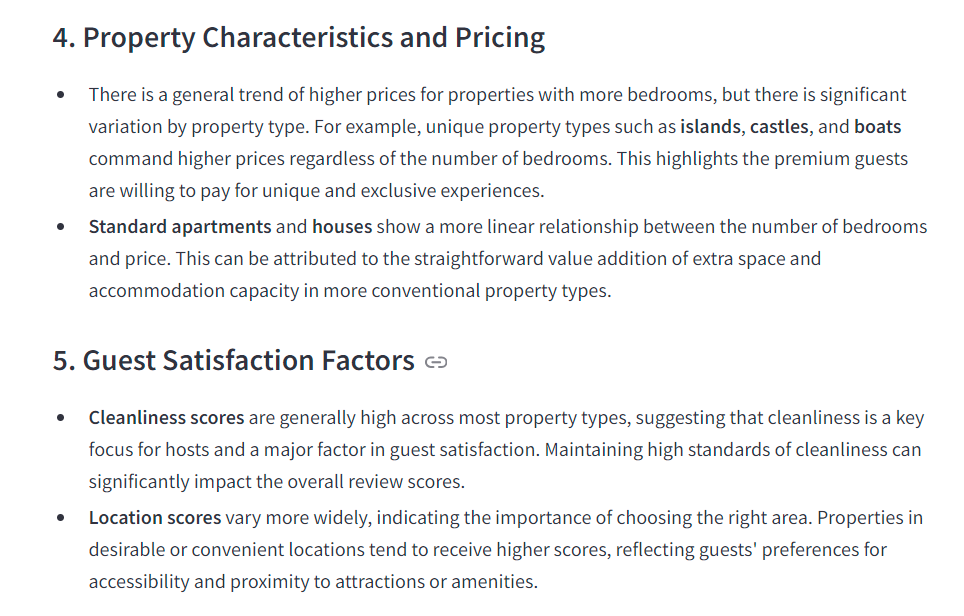
6. תצוגת **dashboard:**

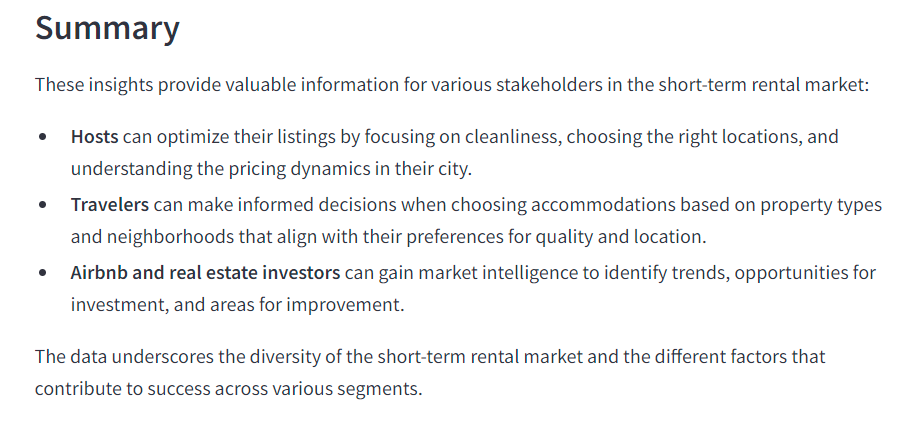
הלוח אמור להיפתח בדפדפן האינטרנט המוגדר כברירת מחדל.

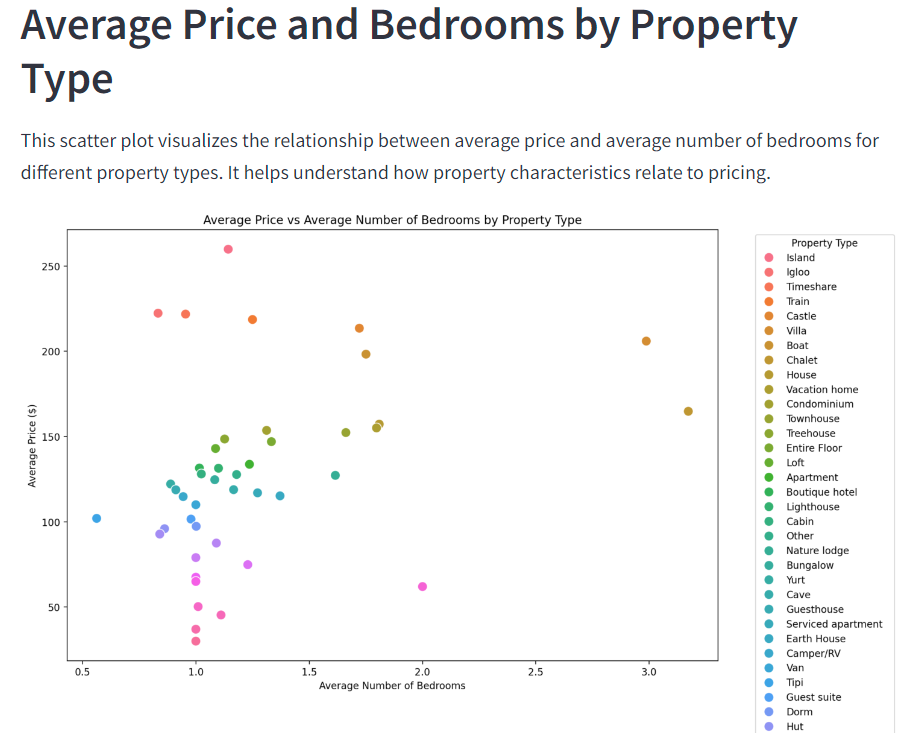
צילומי מסך מהדאשבורד

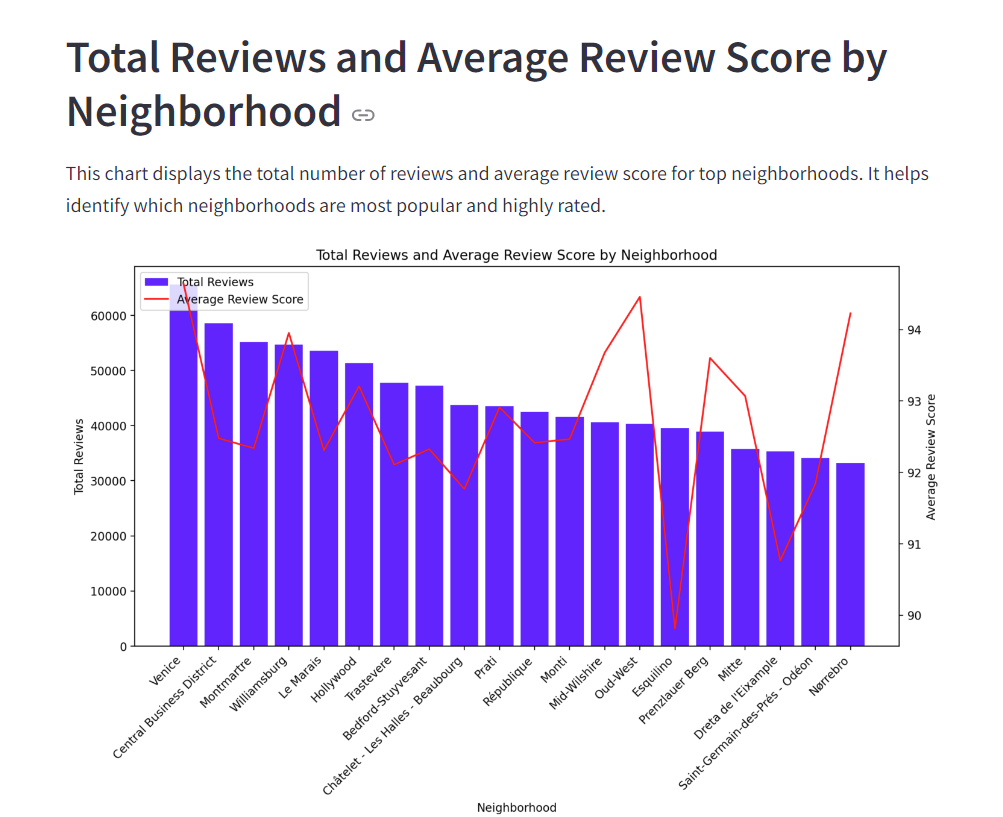
****

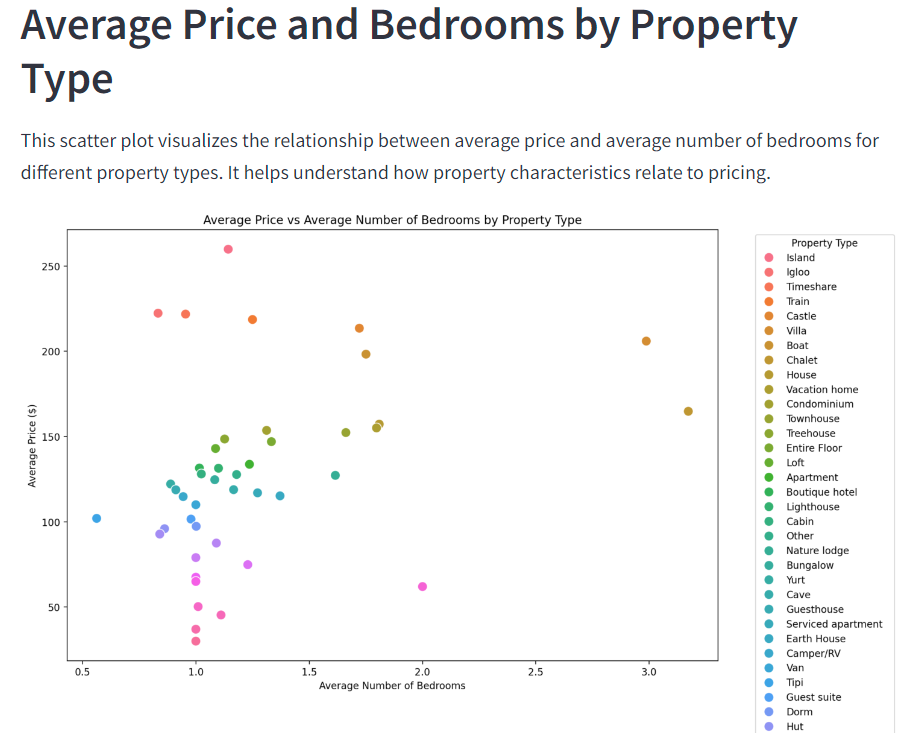


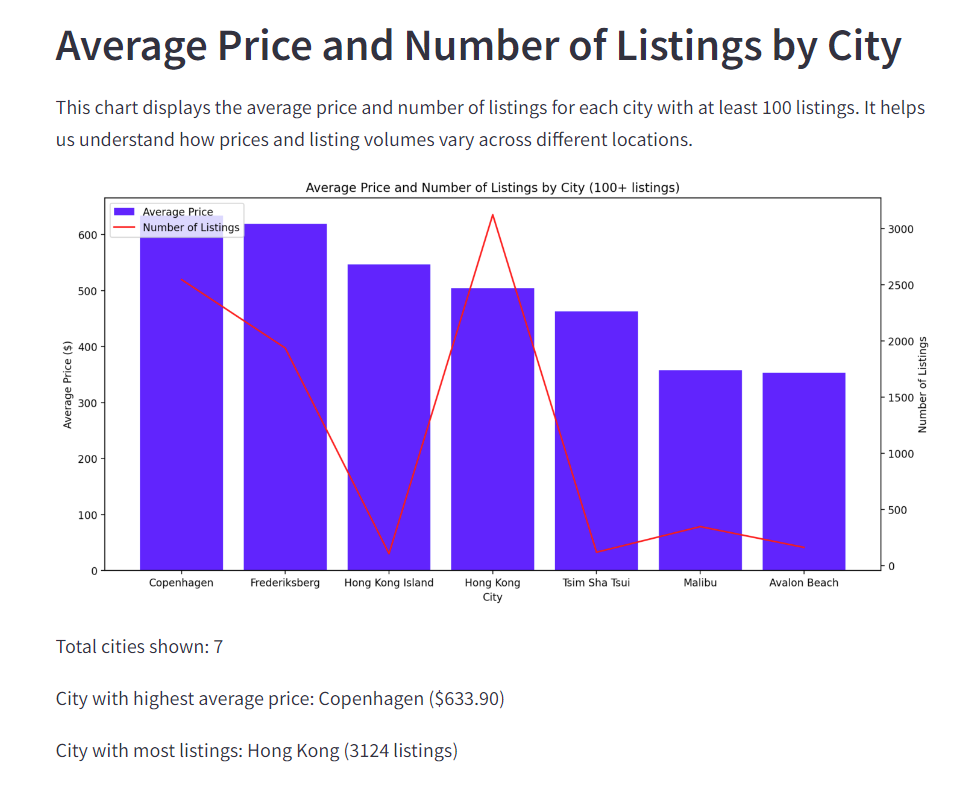
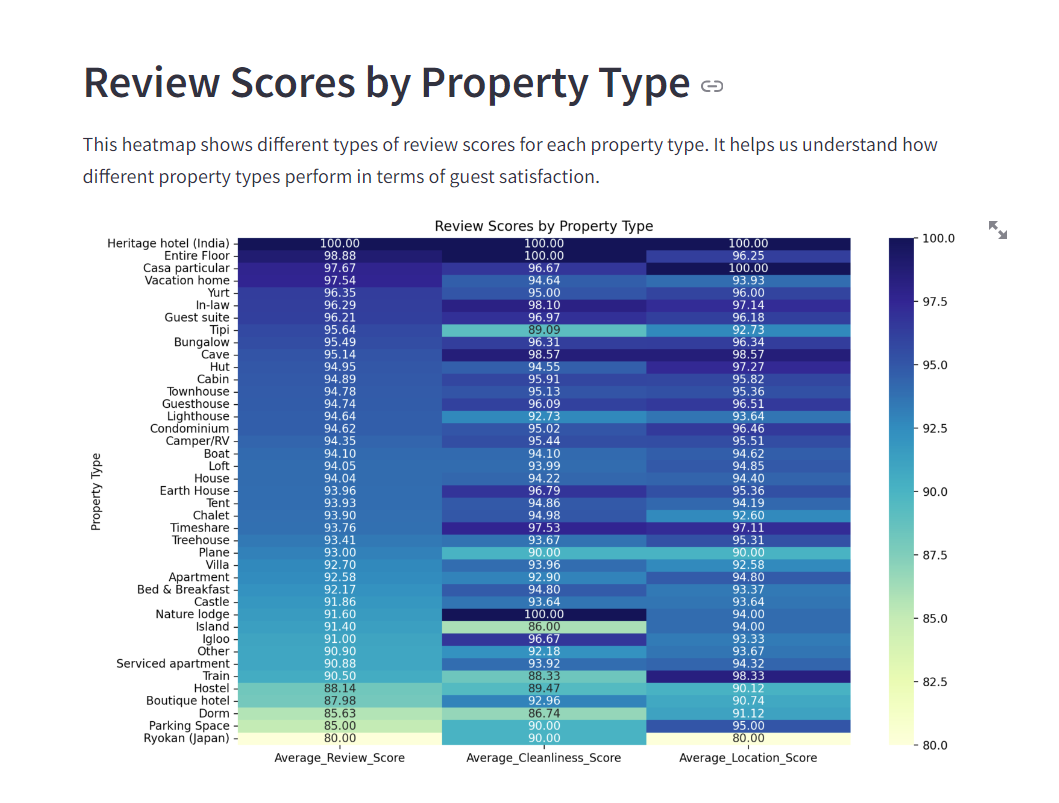
****

****







****

## תיאור קצר של התרגיל

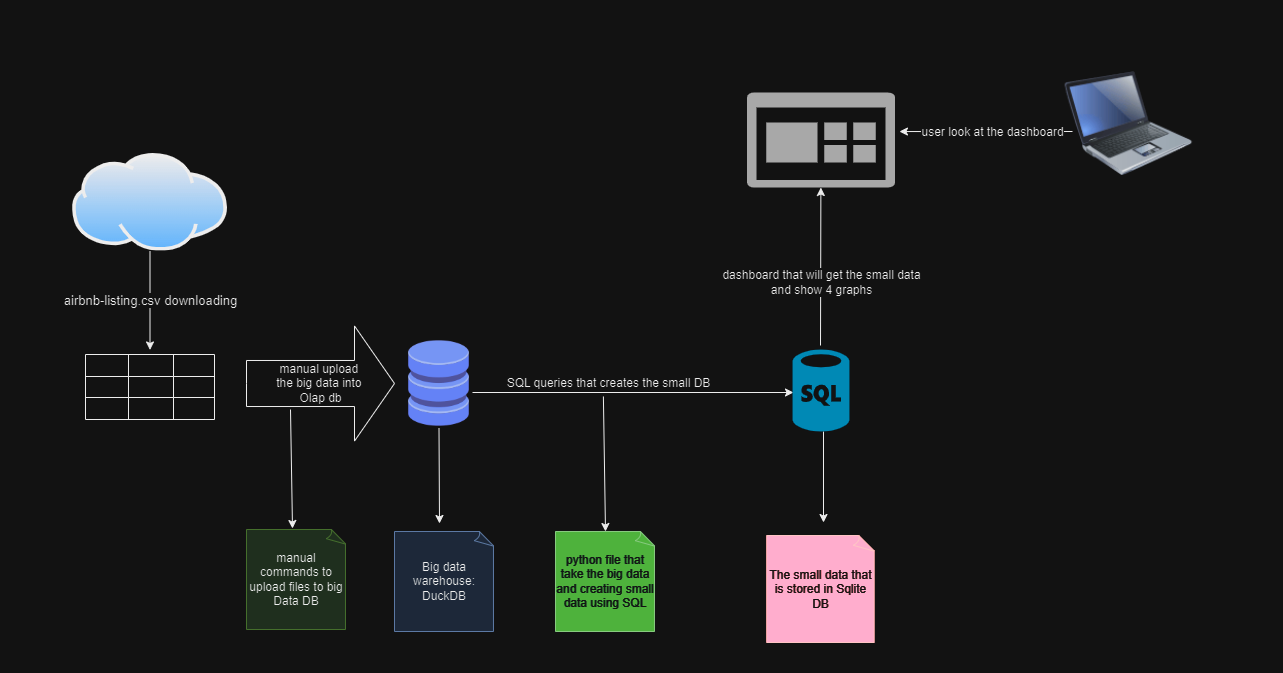
פרויקט זה מדגים תהליך מקיף של ניתוח ויזואליזציה של נתונים. ארכיטקטורת המערכת פועלת על פי מודל של שלוש שכבות: אחסון נתונים, עיבוד נתונים ותצוגה.

אחסון נתונים: אנו משתמשים ב-SQLite) airbnb\_queries.db) כמאגר הנתונים שלנו לאחסון תוצאות של שאילתות מעובדות. מאגר נתונים קל משקל כזה מאפשר אחזור נתונים יעיל ללא צורך בשרת מסד נתונים נפרד.

עיבוד נתונים: פייתון, לצד pandas, משמשת ליצירת מאגר הנתונים של SQLite וביצוע מניפולציות נוספות על הנתונים. שכבה זו פועלת כמתווכת בין הנתונים הגולמיים לשכבת הויזואליזציה.

תצוגה: Streamlit משמש כמסגרת ממשק הפרונט, ומספק ממשק אינטרנט אינטראקטיבי למשתמשים לחקור את הנתונים. Matplotlib, Seaborn משמשות ליצירת ויזואליזציות שונות בתוך אפליקציית Streamlit.

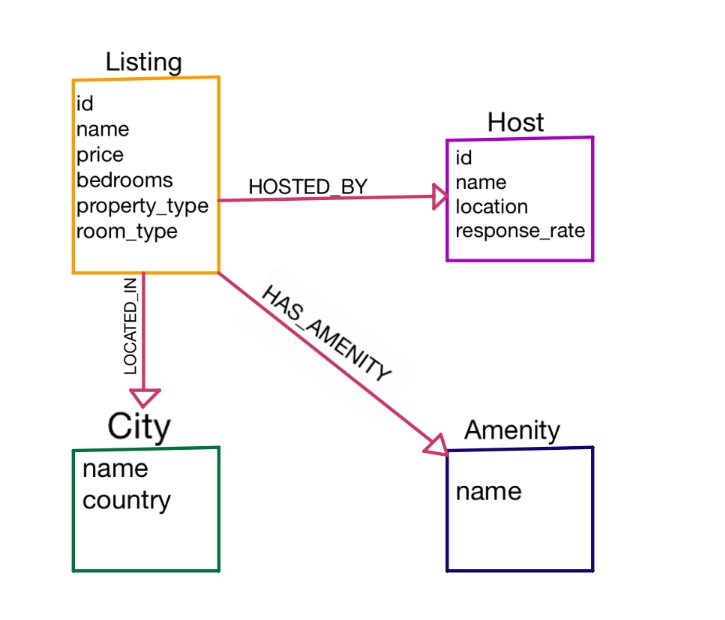
מבנה הקוד משקף ארכיטקטורה זו:

* **main.py** מכיל את השאילתות ויוצר את מאגר הנתונים airbnb\_queries.db
* **airbnb\_dashboard.py** קובץ ה-Python הראשי עבור הדשבורד של Streamlit
* **airbnb\_queries.db** מאגר נתונים SQLite המכיל תוצאות של בירורים מעובדים מראש

אחריות לעבודה:

ביצענו את כל שלבי העבודה יחד.

**חלק ג -**



סכמה זו מייצגת את ישויות הליבה במערך הנתונים של Airbnb. הצומת המרכזי הוא הרישום, שמחובר למארח שלו, לעיר בה הוא ממוקם והשירותים שהוא מציע. מבנה זה מאפשר שאילתה יעילה של מערכות יחסים, כגון מציאת כל הרישומים על ידי מארח, כל השירותים בעיר או כל המארחים במיקום ספציפי. המאפיינים בכל צומת מייצגים את המידע החשוב ביותר עבור אותה ישות, ומאפשרים ניתוח מפורט מבלי לסבך את המודל.