**פרויקט- אנליזה של ביג דאטה**

מאגר מידע: <https://www.kaggle.com/datasets/dilwong/flightprices/data>

מאגר המידע מכיל 27 עמודות, כ82 מיליון שורות בנפח כולל של כ31GB.  
כל שורה במאגר מתאר חיפוש טיסה באתר Expedia, עבור כל טיסה מתועד ID יחודי לטיסה, תאריך חיפוש, תאריך הטיסה, מוצא, יעד, זמן טיסה, מרחק טיסה, ימי טיסה, מידע אודות תעריף ותוספות לכרטיס, האם הטיסה ישירה, תעריף בסיס, מחיר מלא, מס' המושבים שנותרו בטיסה זו, חברת תעופה, מחלקה, סוג מטוס ועוד.

**פירוט עמודות חשובות:**

1. **legId** - מזהה ייחודי לכל טיסה.

משמש לזיהוי ולמעקב אחרי טיסות מסוימות.

1. **searchDate** - תאריך החיפוש באתר Expedia .

מאפשר ניתוח קשר בין זמן החיפוש למחיר הכרטיס.

1. **flightDate** - תאריך הטיסה.

מאפשר לנתח כיצד מחירי הכרטיסים משתנים ביחס לתאריך הטיסה ותאריך ההזמנה.

1. **startingAirport / destinationAirport**  - קודים של שדות התעופה המוצא והיעד (IATA).

מסייע בניתוח מסלולים בין שדות תעופה.

1. **totalTravelDistance** - המרחק הכולל של הטיסה במיילים.

עשוי להיות קשור למחיר הטיסה ולמשך הטיסה.

**מחירים ומחלקות:**

1. - **totalFare** המחיר הכולל של הכרטיס כולל מיסים ועמלות (בדולרים).
2. - **isNonStop**האם מדובר בטיסה ישירה ללא עצירות ביניים (Boolean).

עשוי להשפיע על נוחות הנוסע ועל המחיר.

**זמני טיסות:**

1. **segmentsDepartureTimeRaw / segmentsArrivalTimeRaw** - זמני המראה ונחיתה לכל קטע טיסה.

מאפשר לנתח שעות פופולריות (ואת ההשפעה של השעה על המחיר) ולחשב זמני טיסות בפועל.

1. – **travelDuration**משך הטיסה בפורמט שעות ודקות.

**מאפייני נוסעים ומטוסים:**

1. - **seatsRemaining** מספר המושבים שנותרו בטיסה.

עשוי להשפיע על המחיר עקב ביקוש והיצע.

על מנת להכניס את הדאטה לDUCKDB השתמשנו ביכולות של DUCKDB, מפאת גודל הדאטה, הקובץ נוצר רק אם הוא לא קיים, שכן מדובר בהמון זמן ריצה.

הקוד להלן:

if not file\_exists("database.duckdb"):  
 duckdb\_conn = duckdb.connect("database.duckdb")  
 duckdb\_conn.execute("CREATE TABLE main AS SELECT \* FROM read\_csv\_auto('itineraries.csv')")  
else:  
 duckdb\_conn = duckdb.connect("database.duckdb")

השאלות שנשאל על הדאטה והשאילתות SQL שנבצע עבורן:

1. כיצד משתנים מחירי הטיסות ביחס לזמן החיפוש מול מועד הטיסה?

duckdb\_conn.execute("""duckdb\_conn.execute("""  
 CREATE TABLE query1 AS  
 WITH time\_ranges AS (  
 SELECT \*,  
 (flightDate - searchDate) as days\_until\_flight  
 FROM main  
 )  
 SELECT   
 days\_until\_flight as days\_before\_flight,  
 ROUND(AVG(totalFare), 2) as avg\_fare,  
 COUNT(\*) as number\_of\_searches,  
 ROUND(MIN(totalFare), 2) as min\_fare,  
 ROUND(MAX(totalFare), 2) as max\_fare  
 FROM time\_ranges  
 GROUP BY days\_until\_flight  
 ORDER BY days\_until\_flight  
""")

1. האם משתלם יותר לטוס בטיסה ישירה או בטיסה עם עצירות (קונקשן) בהתחשב במרחקי טיסה שונים?

duckdb\_conn.execute("""  
CREATE TABLE query2 AS  
SELECT   
 CASE   
 WHEN totalTravelDistance < 500 THEN 'Short (<500 miles)'  
 WHEN totalTravelDistance < 1000 THEN 'Medium (500-1000 miles)'  
 ELSE 'Long (>1000 miles)'  
 END as distance\_category,  
 ROUND(AVG(CASE WHEN isNonStop THEN totalFare END), 2) as direct\_avg\_fare,  
 ROUND(AVG(CASE WHEN NOT isNonStop THEN totalFare END), 2) as connection\_avg\_fare,  
 COUNT(CASE WHEN isNonStop THEN 1 END) as direct\_flights\_count,  
 COUNT(CASE WHEN NOT isNonStop THEN 1 END) as connection\_flights\_count  
FROM main  
WHERE totalTravelDistance IS NOT NULL  
GROUP BY 1  
ORDER BY distance\_category;  
""")

1. איך שעת ההמראה משפיעה על המחיר, הזמינות ואופי הטיסה (ישירה\לא ישירה)?

duckdb\_conn.execute("""  
 CREATE TABLE query3 AS  
 WITH flight\_times AS (  
 SELECT \*,  
 CAST(SUBSTR(segmentsDepartureTimeRaw, 12, 2) AS INTEGER) as departure\_hour  
 FROM main  
 WHERE segmentsDepartureTimeRaw IS NOT NULL  
 )  
 SELECT   
 departure\_hour,  
 ROUND(AVG(totalFare), 2) as avg\_fare,  
 ROUND(AVG(seatsRemaining), 1) as avg\_seats\_remaining,  
 COUNT(\*) as number\_of\_flights,  
 ROUND(MIN(totalFare), 2) as min\_fare,  
 ROUND(MAX(totalFare), 2) as max\_fare,  
 SUM(isNonStop::INTEGER) \* 100.0 / COUNT(\*) as nonstop\_percentage  
 FROM flight\_times  
 GROUP BY departure\_hour  
 ORDER BY departure\_hour  
""")

1. מהם ימי השבוע היקרים והזולים ביותר לטיסה והאם יש הבדל בין טיסות ישירות לטיסות עם עצירות?

duckdb\_conn.execute("""  
 CREATE TABLE query4 AS  
 WITH day\_info AS (  
 SELECT   
 totalFare,  
 STRFTIME(flightDate::DATE, '%A') as flight\_day,  
 isNonStop  
 FROM main  
 )  
 SELECT   
 flight\_day,  
 ROUND(AVG(totalFare), 2) as avg\_fare,  
 ROUND(AVG(CASE WHEN isNonStop THEN totalFare END), 2) as avg\_nonstop\_fare,  
 ROUND(AVG(CASE WHEN NOT isNonStop THEN totalFare END), 2) as avg\_connection\_fare,  
 COUNT(\*) as number\_of\_flights  
 FROM day\_info  
 GROUP BY flight\_day  
 ORDER BY CASE flight\_day  
 WHEN 'Sunday' THEN 0  
 WHEN 'Monday' THEN 1   
 WHEN 'Tuesday' THEN 2  
 WHEN 'Wednesday' THEN 3  
 WHEN 'Thursday' THEN 4  
 WHEN 'Friday' THEN 5  
 WHEN 'Saturday' THEN 6  
 END  
""")

1. מהם הנתיבים (מוצא ויעד) של טיסות להם היו שינויי מחיר קיצוניים (מעל 20%) כלפי מעלה וכלפי מטה?

duckdb\_conn.execute("""  
 CREATE TABLE query5 AS  
 WITH price\_volatility AS (  
 SELECT   
 startingAirport,  
 destinationAirport,  
 flightDate,  
 AVG(totalFare) as avg\_fare,  
 LAG(AVG(totalFare)) OVER (  
 PARTITION BY startingAirport, destinationAirport  
 ORDER BY flightDate  
 ) as prev\_day\_fare  
 FROM main  
 GROUP BY startingAirport, destinationAirport, flightDate  
 )  
 SELECT   
 flightDate,  
 startingAirport,  
 destinationAirport,  
 avg\_fare,  
 prev\_day\_fare,  
 ((avg\_fare - prev\_day\_fare) / prev\_day\_fare \* 100) as daily\_change\_percent  
 FROM price\_volatility  
 WHERE ((avg\_fare - prev\_day\_fare) / prev\_day\_fare \* 100) > 20  
 ORDER BY daily\_change\_percent DESC;  
""")

**הסיפור שנרצה לספר:**

בעולם התיירות והטיסות, קיימים גורמים שונים המשפיעים על מחירי הטיסה. נרצה לנתח אותם ולהבין מהם הגורמים לכך ולזהות תנודות במחירי הטיסות.

הניתוח מתמקד בדפוסי המחירים של כרטיסי טיסה, תוך בחינה של קשרים בין גורמים שונים כמו הזמן בין ההזמנה עד הטיסה, מרחק הטיסה, זמני יציאה, וסוגי טיסות (ישירה או עם עצירות).

השאילתות מספקות תובנות ברורות שמראות איך ומתי המחירים משתנים ומהם הגורמים שמניעים את השינויים הללו.

בכדי לייצא את הדאטה מduckDB לsqlite חיברנו את הsqlite לduckDB, פתחנו קובץ חדש בפורמט המתאים ואליו העתקנו את הטבלאות אחת אחת

duckdb\_conn.execute("INSTALL sqlite;")  
duckdb\_conn.execute("LOAD sqlite;")  
duckdb\_conn.execute("ATTACH 'database.sqlite' AS sqliteDB (TYPE SQLITE);")  
  
duckdb\_conn.execute("CREATE TABLE sqliteDB.sample AS SELECT \* FROM sample")  
duckdb\_conn.execute("CREATE TABLE sqliteDB.query1 AS SELECT \* FROM query1")  
duckdb\_conn.execute("CREATE TABLE sqliteDB.query2 AS SELECT \* FROM query2")  
duckdb\_conn.execute("CREATE TABLE sqliteDB.query3 AS SELECT \* FROM query3")  
duckdb\_conn.execute("CREATE TABLE sqliteDB.query4 AS SELECT \* FROM query4")  
duckdb\_conn.execute("CREATE TABLE sqliteDB.query5 AS SELECT \* FROM query5")