**פרויקט ביג דאטה**

**מגישים : אסף חכמון (205364409) ועומר שמעוני (318201597)**

1. לינק לדאטה: <https://www.kaggle.com/datasets/kaushiksuresh147/bitcoin-tweets>
2. במאגר יש כ- 4 מיליון שורות , 13 עמודות. גודל המאגר הוא כ-2 ג'יגה
3. המאגר כולל ציוצים מהרשת החברתית טוויטר בנושא ביטקוין , כולל מספר עוקבים של המצייץ , מיקום שלו , מספר חברים, מאיזה מכשיר צייץ (מחשב , אייפון , אנדרואיד , ווב וכו')
4. קובץ bitcoin\_tweets\_small
5. תיאור הפקודות המעבירות את המאגר לduckdb :
6. **יבוא הספרייה**: import duckdb
7. התחברות למסד הנתונים : conn = duckdb.connect('db\_file.duckdb')

כאן מתבצע חיבור למסד נתונים מסוג **DuckDB** בקובץ בשם db\_file.duckdb. אם הקובץ לא קיים, הוא יווצר.

1. **יצירת טבלה והכנסת נתונים**:

conn.execute("""

CREATE TABLE IF NOT EXISTS bitcoin\_tweets AS

SELECT \* FROM read\_csv\_auto('Bitcoin\_tweets.csv', sample\_size=-1, ignore\_errors=true)

""")

שורת קוד זו עושה כמה דברים חשובים:

* בודקת אם קיימת כבר טבלה בשם bitcoin\_tweets.
* אם הטבלה **לא קיימת**, היא יוצרת אותה באמצעות **טעינת הנתונים ישירות מקובץCSV** (Bitcoin\_tweets.csv)
* השימוש ב - read\_csv\_auto מאפשר ל DuckDB לזהות באופן אוטומטי את סוגי הנתונים בעמודות.
* sample\_size=-1 גורם ל - DuckDB לקרוא את כל הקובץ, ולא רק דגימה ממנו, לפני שהוא מכריע מה סוג הנתונים בכל עמודה
* ignore\_errors=true מבטיח שהתהליך ימשיך גם אם יש שורות פגומות בקובץ (תוך סינונן).

6+7+8.

* 1. השאילתה:

CREATE TABLE IF NOT EXISTS locations\_most\_tweets AS

SELECT

user\_location,

COUNT(\*) AS tweet\_count

FROM bitcoin\_tweets

WHERE user\_location IS NOT NULL AND TRIM(user\_location) != '' -- Remove empty locations

GROUP BY user\_location

ORDER BY tweet\_count DESC;

**מה השאילתה עושה?**

* מחשבת את מספר הציוצים מכל מיקום(user\_location) .
* מסננת ערכים ריקים או מיקומים לא תקינים.
* מסדרת את התוצאות כך שהמיקומים עם הכי הרבה ציוצים יופיעו ראשונים.

**מה אנו מנסים ללמוד?**

* מהם המיקומים שבהם מצייצים הכי הרבה על ביטקוין?
* היכן מתבצע הכי הרבה דיון בנושא ביטקוין?

---

* 1. השאילתה:

CREATE TABLE IF NOT EXISTS TWEETS\_PER\_DAY AS

SELECT

STRFTIME('%Y-%m-%d', TRY\_CAST(date AS TIMESTAMP)) AS tweet\_date,

COUNT(\*) AS tweet\_count

FROM bitcoin\_tweets

WHERE TRY\_CAST(date AS TIMESTAMP) IS NOT NULL -- Exclude bad dates

GROUP BY tweet\_date

ORDER BY tweet\_date DESC;

**מה השאילתה עושה?**

* מחשבת את מספר הציוצים בכל יום .(date)
* ממירה את תאריך הציוץ לפורמט YYYY-MM-DD
* מסננת תאריכים לא תקינים.

**מה אנו מנסים ללמוד?**

* מגמות יומיות בציוצים על ביטקוין.
* האם יש ימים ספציפיים שבהם יש עלייה חדה במספר הציוצים?

---

* 1. השאילתה:

CREATE TABLE IF NOT EXISTS bull\_mention\_by\_location AS

SELECT

user\_location,

COUNT(\*) AS bull\_mention\_count

FROM bitcoin\_tweets

WHERE user\_location IS NOT NULL AND LOWER(text) LIKE '%bull%'

GROUP BY user\_location

ORDER BY bull\_mention\_count DESC;

**מה השאילתה עושה?**

* בודקת כמה ציוצים בכל מיקום מכילים את המילה **"bull"** (שמשמעותה שוק שורי - כלומר במגמת עלייה).
* מסדרת את התוצאות לפי מספר ההופעות.

**מה אנו מנסים ללמוד?**

* מהם המיקומים שבהם מדברים הכי הרבה על שוק שורי?
* האם יש הבדל בין אזורים שונים בגישה שלהם לשוק?

---

* 1. השאילתה:

CREATE TABLE IF NOT EXISTS bear\_mention\_by\_location AS

SELECT

user\_location,

COUNT(\*) AS bear\_mention\_count

FROM bitcoin\_tweets

WHERE user\_location IS NOT NULL AND LOWER(text) LIKE '%bear%'

GROUP BY user\_location

ORDER BY bear\_mention\_count DESC;

**מה השאילתה עושה?**

* בודקת כמה ציוצים בכל מיקום מכילים את המילה **"bear"** (שמשמעותה שוק דובי - כלומר במגמת ירידה).
* מסדרת את התוצאות לפי מספר ההופעות.

**מה אנו מנסים ללמוד?**

* מהם המיקומים שבהם מדברים הכי הרבה על שוק דובי?
* האם יש אזורים מסוימים שבהם יותר פסימיים לגבי ביטקוין.

---

* 1. השאילתה:

CREATE TABLE IF NOT EXISTS tweets\_per\_month\_compare\_price AS

SELECT t1.month , t2.tweet\_count , t1.avg\_btc\_price

FROM BTC\_PRICE\_PER\_MONTH AS t1 INNER JOIN

TWEETS\_PER\_MONTH AS t2

ON t1.month = t2.month

ORDER BY t2.tweet\_count DESC;

**מה השאילתה עושה?**

* משווה בין מספר הציוצים בכל חודש למחיר הביטקוין הממוצע באותו חודש.
* משתמשת ב - INNER JOIN כדי לקשר בין טבלאות המחירים (BTC\_PRICE\_PER\_MONTH) לטבלאות הציוצים .(TWEETS\_PER\_MONTH)
* מסדרת את החודשים עם הכי הרבה ציוצים בראש הרשימה.

**מה אנו מנסים ללמוד?**

* האם יש מתאם בין מספר הציוצים למחיר הביטקוין.
* האם עלייה בהתעניינות בטוויטר משפיעה על מחיר הביטקוין (או להפך).
  1. השאילתה:

CREATE TABLE IF NOT EXISTS compare\_bull\_bear\_count\_locations AS

SELECT

l.user\_location,

l.tweet\_count AS total\_tweets,

b.bear\_mention\_count,

bu.bull\_mention\_count

FROM locations\_most\_tweets AS l

LEFT JOIN bear\_mention\_by\_location AS b ON l.user\_location = b.user\_location

LEFT JOIN bull\_mention\_by\_location AS bu ON l.user\_location = bu.user\_location

ORDER BY total\_tweets DESC;

**מה השאילתה עושה?**

* משווה בין מספר הציוצים הכולל בכל מיקום (total\_tweets) לבין מספר הפעמים שהוזכרו **"bull"** (=שוק שורי)או **"bear"** (=שוק דובי).
* מחברת את טבלת locations\_most\_tweets עם bear\_mention\_by\_location עםbull\_mention\_by\_location באמצעות LEFT JOIN.

**מה אנו מנסים ללמוד?**

* מהם המיקומים שבהם יש הכי הרבה שיח על ביטקוין.
* איזה מיקומים נוטים להיות יותר **שוריים** (bull) ואיזה יותר **דוביים** (bear).
* האם יש מדינות/ערים שבהן יש גישה אופטימית או פסימית יותר כלפי ביטקוין.

**סיכום כללי**

השאילתות בודקות **מגמות ומאפיינים** של הציוצים על ביטקוין:

1. **מיקום הציוצים** – איפה יש הכי הרבה דיונים על ביטקוין?
2. **מגמות בזמן** – איך משתנה מספר הציוצים לאורך זמן (יומי / חודשי)?
3. **רגשות השוק** – באילו מקומות מדברים יותר על שוק שורי (bull) או דובי ?(bear)
4. **השוואה למחיר הביטקוין** – האם יש קשר בין מספר הציוצים לשינויים במחיר?

**מסקנות אפשריות**:

* נוכל לראות האם **השיח בטוויטר לגבי ביטקוין מקדים או מגיב לשינויי מחיר**.
* להבין **היכן יש יותר התעניינות בביטקוין בעולם**.
* לבדוק **אילו ימים או חודשים מאופיינים בעלייה בציוצים**.
* להשוות בין מקומות בעולם ולהבין **האם יש גישות שונות כלפי ביטקוין במקומות שונים**.

9. תהליך העלאת הדאטה הרלוונטית ל SQLITE :

**התקנה וטעינה של הרחבת SQLite בתוך DuckDB**   
כדי לאפשר ל-DuckDB לעבוד עם ,SQLite יש להתקין ולטעון את ההרחבה המתאימה באמצעות הפקודות:  
התקנת ההרחבה: INSTALL sqlite;  
טעינת ההרחבה: LOAD sqlite;

**יצירת חיבור למסד הנתונים של DuckDB**  
יש לפתוח חיבור למסד נתונים  **DuckDB** בשםdb\_file.duckdb באופן הבא:  
CONNECT 'db\_file.duckdb';

**וידוא טעינת הרחבת SQLite בתוך DuckDB**  
לפני שנוכל לעבוד עם ,SQLite יש לוודא שההרחבה נטענה כראוי שוב באמצעות הפקודות:  
INSTALL sqlite;  
LOAD sqlite;

**חיבור בין DuckDB ל- SQLite-**  
מחברים את מסד הנתונים של **SQLite** (כלומר - (db\_file.sqlite אל תוך ,DuckDB כך שניתן יהיה לקרוא ולכתוב נתונים לתוכו:  
ATTACH 'db\_file.sqlite' AS sqlite\_db (TYPE sqlite);

**העתקת הנתונים מ DuckDB ל- SQLite-**  
עבור כל טבלה ברשימה, יוצרים טבלה חדשה ב SQLite-ומעתיקים אליה את כל הנתונים מהטבלה הקיימת ב-DuckDB.  
התהליך מתבצע על ידי הפקודה:  
CREATE TABLE sqlite\_db.<table\_name> AS SELECT \* FROM <table\_name>;   
כאשר <table\_name> מוחלפת בשם הטבלה המתאימה מתוך הרשימה.

**10.**

קובץ project.py וקובץ sqlite.py

**11א.**

קובץ הדאטהבייס של sqlite בתוך הzip- בשם db\_file.sqlite

**11ב.**

הסבר על כל הטבלאות בקובץ sqlite הקטן :

* 1. **טבלה: bitcoin\_tweets\_small**
* **מספר שורות:** 500
* **מספר עמודות:** 13
* **תיאור:** טבלה זו מכילה מידע על ציוצים בנושא ביטקוין, כולל פרטי המשתמש, תוכן הציוץ ותאריך הפרסום.
* **עמודות בטבלה:**
  + user\_name - שם המשתמש
  + - user\_location מיקום המשתמש
  + - user\_description תיאור המשתמש
  + user\_created - תאריך יצירת החשבון
  + - user\_followers מספר עוקבים
  + user\_friends - מספר עוקבים שהמשתמש עוקב אחריהם
  + user\_favourites - מספר ציוצים שסומנו כמועדפים
  + - user\_verified האם המשתמש מאומת
  + - date תאריך הציוץ
  + text - תוכן הציוץ
  + - hashtags רשימת ההאשטאגים בציוץ
  + source - פלטפורמה ממנה נשלח הציוץ
  + is\_retweet - האם הציוץ הוא ריטוויט
  1. **טבלה :BTC\_PRICE\_PER\_DAY\_SMALL**
* **מספר שורות:** 500
* **מספר עמודות:** 2
* **תיאור**: טבלה זו מכילה מידע יומי על מחיר הביטקוין.
* **עמודות בטבלה:**
  + Date - תאריך
  + - price מחיר הביטקוין באותו יום

---

* 1. **טבלה BTC\_PRICE\_PER\_MONTH:**
* **מספר שורות:** 24
* **מספר עמודות:** 2
* **תיאור** **:**מחיר הביטקוין הממוצע בכל חודש.
* **עמודות בטבלה:**
  + month - חודש ושנה (בפורמט YYYY-MM)
  + avg\_btc\_price - המחיר הממוצע של ביטקוין בחודש

---

* 1. **טבלה : locations\_most\_tweets\_small**
* **מספר שורות: 500**
* **מספר עמודות: 2**
* **תיאור**: טבלה שמציגה כמה ציוצים פורסמו מכל מיקום.
* **עמודות בטבלה:**
  + user\_location - מיקום המשתמש
  + tweet\_count - מספר הציוצים מאותו מיקום

---

* 1. **טבלה: TWEETS\_PER\_DAY**
* **מספר שורות: 222**
* **מספר עמודות: 2**
* **תיאור:** מספר הציוצים שנשלחו על ביטקוין בכל יום.
* **עמודות בטבלה:**
  + tweet\_date - תאריך הציוצים
  + tweet\_count - מספר הציוצים באותו יום

---

* 1. **טבלה: TWEETS\_PER\_MONTH**
* **מספר שורות: 24**
* **מספר עמודות: 2**
* **תיאור:** מספר הציוצים שנשלחו על ביטקוין בכל חודש.
* **עמודות בטבלה:**
  + month - חודש ושנה בפורמט YYYY-MM))
  + tweet\_count - מספר הציוצים באותו חודש

---

* 1. **טבלה: bull\_mention\_by\_location\_small**
* **מספר שורות: 500**
* **מספר עמודות: 2**
* **תיאור:** מציגה את מספר הציוצים בכל מיקום שבהם הופיעה המילה "bull" (שוק שורי)
* **עמודות בטבלה:**
  + user\_location - מיקום המשתמש
  + bull\_mention\_count - מספר הפעמים שבהם הופיעה המילה "bull"

---

* 1. **טבלה : bear\_mention\_by\_location\_small**
* **מספר שורות: 500**
* **מספר עמודות: 2**
* **תיאור :** מציגה את מספר הציוצים בכל מיקום שבהם הופיעה המילה "bear" (שוק דובי)
* **עמודות בטבלה:**
  + user\_location - מיקום המשתמש
  + bear\_mention\_count - מספר הפעמים שבהם הופיעה המילה "bear"

---

* 1. **טבלה: tweets\_per\_month\_compare\_price**
* **מספר שורות: 24**
* **מספר עמודות: 3**
* **תיאור:** השוואה בין מספר הציוצים בכל חודש לבין מחיר הביטקוין הממוצע באותו חודש.
* **עמודות בטבלה:**
  + month - חודש ושנה בפורמט YYYY-MM))
  + tweet\_count - מספר הציוצים באותו חודש
  + avg\_btc\_price - המחיר הממוצע של ביטקוין בחודש

---

* 1. **טבלה : tweets\_per\_day\_compare\_price**
* **מספר שורות: 222**
* **מספר עמודות: 3**
* **תיאור:** השוואה בין מספר הציוצים בכל יום לבין מחיר הביטקוין באותו יום.
* **עמודות בטבלה:**
  + Date - תאריך
  + tweet\_count - מספר הציוצים באותו יום
  + - price מחיר הביטקוין באותו יום

---

* 1. **טבלה: compare\_bull\_bear\_count\_locations\_small**
* **מספר שורות: 500**
* **מספר עמודות: 4**
* **תיאור:** השוואת מספר הציוצים הכולל בכל מיקום, תוך התייחסות לציוצים שמזכירים את "bull" (שוק שורי) ואת "bear" (שוק דובי).
* **עמודות בטבלה:**
  + user\_location - מיקום המשתמש
  + total\_tweets - מספר הציוצים הכולל מאותו מיקום
  + bear\_mention\_count - מספר הפעמים שבהן הופיעה המילה "bear"
  + bull\_mention\_count - מספר הפעמים שבהן הופיעה המילה "bull"

**12 + 13 + 14 .**

הסיפור שאנו רוצים לספר דרך הדאשבורד.

הדאשבורד שלנו מספר את הסיפור של הקשר בין שיח בטוויטר על ביטקוין לבין מחיר הביטקוין .אנחנו רוצים לבדוק כיצד התנהגות השוק והמחירים משפיעים על כמות הציוצים, ואם יש מגמות שיכולות להצביע על קורלציה בין האירועים.

איך נספר את הסיפור?

הסיפור יתפרש על פני מספר גרפים ומפות, כאשר כל חלק יוסיף מידע חדש שיעזור לנו להבין את המגמות:

האם קיים קשר בין מחיר הביטקוין לכמות הציוצים לאורך זמן?

📈 גרף: מספר הציוצים על ביטקוין מול מחיר הביטקוין בחלוקה חודשית

* אנו מציגים שני קווים – אחד שמראה את כמות הציוצים, והשני את מחיר הביטקוין.
* נבדוק מגמות : האם כשהמחיר עולה יש יותר ציוצים? האם יש "גלי התעניינות" סביב תנודות בשוק?
* סיפור אפשרי: ככל שמחיר הביטקוין עולה, אנשים מצייצים יותר (או ההפך, ככל שהמחיר יורד - יש פאניקה ויותר ציוצים).

---

איך נראית התנהגות היומית של המחירים והשיח ברשת?

📉 גרף: מספר הציוצים על ביטקוין מול מחיר הביטקוין לפי ימים

* הסתכלות יותר מפורטת מהגרף הקודם, כדי לזהות תנודות מהירות בשוק.
* נראה האם יש "קפיצות חדות" ביום מסוים, למשל בעקבות חדשות משמעותיות.
* סיפור אפשרי: פתאום רואים קפיצה גדולה בציוצים ביום שבו הביטקוין נפל ב-10% – אולי אנשים דיברו על המכירה ההמונית?

---

באילו מקומות בעולם מדברים הכי הרבה על ביטקוין?

🌍 מפת חום עולמית: המקומות שמדברים הכי הרבה על ביטקוין בטוויטר

* מפה שמראה באילו מדינות ואיזורים בעולם יש הכי הרבה ציוצים על ביטקוין.
* נשאל את עצמנו – האם יש מדינות מסוימות שמובילות את השיח? אולי ארה"ב או הודו?
* סיפור אפשרי: מדינות עם הרבה ציוצים הן מקומות שבהם יש עניין רב במסחר או השקעות בביטקוין.

---

האם יש יותר שיח על שוק חיובי (Bull) או שלילי ?(Bear)

🐂🐻 גרף: מספר ציוצים חיוביים (Bull) לעומת שליליים (Bear) לפי מדינות

* אנחנו מנתחים ציוצים לפי מילות מפתח, למשל אם יש יותר מילים כמו "Bullish" (שוק עולה) או "Bearish" (שוק יורד).
* נבחן איזה מדינות יותר אופטימיות לגבי הביטקוין ואיזה יותר פסימיות.

---

Wordcloudשל ציוצי טוויטר הכי נפוצים לפי התאריך הנבחר בתפריט הנופל

תפריט נופל לבחור בו חודש ושנה וטבלה המציגה בהתאם לכך את כמות הציוצים באותו תאריך ואת מחיר הביטקוין המתאים .

**15.**

קובץ בתוך הzip בשם dashboard.py

**16.**

אין קובץ נוסף

**17.**

קובץ בתוך הzip בשם requirements.txt

**18.**

יש לחלץ הכל מתוך קובץ הזיפ ולפתוח את הפרוייקט , להריץ תחילה את קובץ project.py לאחר מכן את קובץ sqlite.py לאחר מכן להריץ בקונסול את הפקודה streamlit run dashboard.py

**19 + 20.**ייצוג הנתונים שהורדנו בחלק א' על ידי GRAPHDB.

במידה והיינו צריכים לשמור ולמדל את הנתונים כך שיסודרו בצורה שמתאימה לתצוגה גרפית הם היו מסודרים באופן הבא -

* **מבחינת צמתים בגרף:**

יהיה עלינו ליצור שלושה צמתים – עבור **משתמשים**, עבור **ציוצים**, ועבור **האשטגים**.  
בצורה מפורטת יותר, שכוללת גם את הPROPERTIES-:

תהיה לנו צומת **User**, שתייצג חשבון טוויטר מסויים בדאטה שלנו.  
עבורה יהיו כל הפרטים (properties) הבאים: שם, מיקום, תיאור, מתי נוצר החשבון, מס' העוקבים, מס' החברים, מס' המועדפים והאם הוא מאומת (וי כחול).   
(על פי העמודות הבאות בדאטה שלנו: user\_name, user\_location, user\_description, user\_created, user\_followers, user\_friends, user\_favorites, user\_verified)  
  
תהיה לנו צומת **Tweet**, שתייצג ציוץ מסויים בדאטה שלנו.  
עבורה יהיו כל הפרטים (properties) הבאים: תאריך פרסום, מלל, מקור (מהי הגישה לטוויטר) והאם צויץ מחדש.  
(על פי העמודות הבאות בדאטה שלנו: date, text, source, is\_retweeted)  
  
תהיה לנו צומת **Hashtag,** שתייצג האשטאג מסויים בדאטה שלנו.  
עבורה יהיה פרט (property) אחד בלבד, והוא שם ההאשטג (על פי עמודת Hashtags)

* **מבחינת קשרים בגרף:**

יהיה עלינו ליצור שלושה סוגי קשרים – עבור **פרסום**, עבור **הכלת האשטאג** ועבור **ציוץ מחדש**.  
בצורה מפורטת יותר:

יהיה לנו קשר **POSTED**, שייצג פרסום של ציוץ ע"י משתמש. כלומר זהו קשר המקשר בין צומת מסוג User לבין צומת מסוג Tweet.

יהיה לנו קשר ,**CONTAINS\_HASHTAG** שייצג הכלת האשטג מסויים בתוך ציוץ. כלומר זהו קשר המקשר בין צומת מסוג Tweet לבין צומת מסוג Hashtag.

יהיה לנו קשר **RETWEETED,** שייצג ציוץ מחדש של ציוץ מסויים. כלומר זהו קשר המקשר בין צומת מסוג User לבין צומת מסוג Tweet.

ייצוג ויזואלי (סכמה):

תמונה שמכילה עיגול, צילום מסך, ספרה, צבעוני

התיאור נוצר באופן אוטומטי

הסבר על הסכמה:

הסכמה מייצגת שני ציוצים. ניתן לראות שאת שני הציוצים צייצו שני משתמשים ששמותם DeSota Wilson ו-CryptoND. כמו כן, ניתן לראות שמעבר לפוסט שפורסם על ידי המשתמש CryptoND הוא גם צייץ מחדש את הציוץ של המשתמש DeSota Wilson.

לגבי ההשטאגים, ניתן לראות שהציוץ של DeSota Wilson מכיל האשטאג אחד בלבד (#bitcoin) ואילו הציוץ של CryptoND מכיל שלושה האשטאגים (#Thursday, #Btc, #wallet).

הסכמה מייצגת בצורה טובה את הדאטה ואת הקשרים שיש בה, וכמובן שכל צומת מכילה בתוכה את המידע הרלוונטי, כפי שפורט לעיל.

**25.**

תמונה שמכילה טקסט, צילום מסך, גופן, תוכנה

התיאור נוצר באופן אוטומטי

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**26**

**מבנה הקוד והארכיטקטורה**

**שלב ראשון: טעינת הנתונים ושמירה ב- DuckDB**

✅ קראנו שני מקורות נתונים:

* **ציוצים על ביטקוין (Bitcoin\_tweets.csv)**
* **מחירי הביטקוין (BTC\_value\_per\_day.csv)**

✅ הכנסנו אותם למסד הנתונים **DuckDB** וביצענו:

* ניקוי נתונים (הסרת ערכים ריקים).
* יצירת תת-מאגר קטן (500 שורות לדוגמא).
* חישוב ערכים מצטברים כמו **מחיר ממוצע חודשי של ביטקוין** ו**כמות ציוצים ביום/חודש**.

**שלב שני: העברת הנתונים ל-SQLite**

✅ לאחר העיבוד ב-DuckDB, ביצענו **העברה של הנתונים** ל SQLite-ע"י:

* יצירת טבלאות ב-SQLite.
* העתקת את הטבלאות המעובדות.
* שמירת טבלאות נפרדות לניתוחים ספציפיים כמו **ניתוח שיח שורי/דובי (Bull/Bear Sentiment)**.

**שלב שלישי: יצירת הדאשבורד ב-Streamlit**

✅ יצרת **דאשבורד אינטראקטיבי** ב-Streamlit שמציג 5 ויזואליזציות:

1. **📈 גרף קורלציה בין מספר הציוצים למחיר הביטקוין (חודשי).**
   * קו כחול: **כמות הציוצים בכל חודש**.
   * קו אדום: **מחיר הביטקוין הממוצע באותו חודש**.
2. **📉 גרף מגמות יומיות של ציוצים ומחירי הביטקוין.**
   * סידרנו את הנתונים כך שיתאימו לתאריכים ולמנוע חוסר סדר בציר ה-X.
3. **🌍 מפה אינטראקטיבית של כמות הציוצים לפי מיקום בעולם.**
   * שימוש ב Folium-להוספת מעגלים **בגודל משתנה** **לפי מספר הציוצים** לכל מדינה.
4. **🐂🐻 גרף שמראה כמה מהציוצים חיוביים (Bull) וכמה שליליים .(Bear)** 
   * השוואה לפי מדינות.
5. **💬 ענן מילים (Word Cloud) מהמילים הכי נפוצות בציוצים על ביטקוין.**
   * שימוש בספריית wordcloud.

✅ הוספנו **אלמנטים אינטראקטיביים**, כמו:

* **Dropdown לבחירת חודש ולראות נתונים רק ממנו**.
* **אפשרות להציג את הנתונים בטבלה עם הדגשות צבעוניות**.

תמונה שמכילה טקסט, תרשים, קו, צילום מסך

התיאור נוצר באופן אוטומטי

**27.**

תיאור של חלוקת העבודה בינינו –

יחד חיפשנו ומצאנו את המאגר המתאים, והחלטנו מה הסיפור אותו נרצה לספר, מהן השאילתות אותן נשאל ואיך נציג זאת.

עומר שמעוני – חיבור הדאטה ל-duckdb וכתיבת מספר שאילתות, תכנון הדאשבורד וההסברים בו, הכנה של כל ה-GRAPHDB .

אסף חכמון – החיבור ל- SQLITEוכתיבת מספר שאילתות, יצירת הדאשבורד, ניתוח הסיפור לפי תוצאות השאילתות .

**28.**

קובץ readme.txt נמצא בתוך קובץ ה-zip