Delayed Flight Tracker

Liav Steinberg, liavst2@gmail.com, liavst2
Itai Kaufman, itai.k89@gmail.com, itaik
Amit Arie, amit.arie@gmail.com, atstyle

- **תיאור הבעיה:**

בעולם בו הזמן הוא חלק אינטגרלי מחיינו, שאיפתנו היא לשלוט בלוח הזמנים שלנו בכל רגע נתון. לכן בחרנו בפרויקט המבקש להתמקד בהערכת זמני יציאה של טיסות בעיקר עבור אנשי עסקים שלוח הזמנים שלהם צפוף, ונמצאים באופן תכוף בטיסות למקומות שונים, שיוכלו לקבל הערכה גסה לגבי זמן הגעתם ליעד.
הפרויקט שלנו לוקח מסד נתונים של טיסות ומנסה לחזות, לכל טיסה, את זמני היציאה שלה לפי נתונים קודמים שלה כתלות בשדה התעופה, קו התעופה, לקיחה בחשבון איחורים בשל מזג האוויר השורר באזור, בשל בעיות ביטחוניות, בשל איחור הגעת המטוס עצמו, ונתונים נוספים.

- **מקור המידע:**

לפתרון הבעיה השתמשנו במסד נתונים של טיסות בתוך ארה"ב, שנלקח מהאתר הזה:
https://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236&DB_Short_Name=On-Time

המידע שלקחנו מאתר זה מאורגן ב16 קבצים מסוג csv, כאשר 12 מתוכם מכילים נתונים של טיסות יוצאות בשנה האחרונה (2016), אחד לכל חודש בשנה הנ"ל. כל קובץ כזה מחזיק בקירוב 440,000 רשומות עם מספר שדות רלוונטיים כגון קו התעופה, מוצא ויעד הטיסה, האם היא בוטלה או לא, בכמה היא איחרה/הקדימה את זמנה המתוכנן, אם היה איחור - כמה איחור היה בגלל מזג האוויר? בגלל בעיה ביטחונית? בגלל מטוס שהתעכב? ועוד שדות נוספים.

שאר הקבצים מכילים מידע על קווי התעופה ושדות התעופה הקיימים באזור. כל קובץ כזה מחזיק בקירוב 1200 רשומות.

- **<u>גישה לפתרון:</u>**

רצינו לראות איך נתונים סביבתיים משפיעים על איחור הטיסה ולכן יצרנו מספר ויזואליזציות על מנת שנוכל לנתח את נתוני הבסיס באופן ענייני (מוצגים בהמשך). לאחר שניתחנו את הנתונים, הזנו אותם לאלגוריתם למידה בשיטות Logistic Regression, Stochastic Gradient Descent, על מנת לחזות איחור בטיסה נתונה.

- **<u>ניסויים והרצות:</u>**

- **<u>הכנת סביבת העבודה לעיבוד הנתונים</u>**:

ראשית, מכיוון שמקור הנתונים שלנו מאוד כבד, ורצינו לשלוף ממנו מידע ביעילות ובמהירות, אירגנו את כל קבצי הcsv בטבלאות לתוך קובץ db אחד גדול (~540 MB) על מנת שנוכל לשלוף מידע באמצעות שאילתות בשפת sql (שימוש בספריה sqlite3). כפי שניתן לראות בקבצי הקוד המצורפים, את השאילתות כתבנו בקובץ main.py, ושאילתות אלה עובדו בקובץ Parser.py שבו שלפנו את המידע ממסד הנתונים וכתבנו את התוצאה לקבצי json, שעליהם נוכל לנתח ויזואלית את הנתונים. לאחר שניתחנו את הנתונים ע"י מספר שאילתות והצגתם בצורה גרפית, בנינו מסד נתונים חדש עם העמודות הרלוונטיות לפתרון הבעיה. כל טיסה מיוצגת כווקטור שהעמודות בו הן: חודש, יום, גודל שדה"ת מוצא,גודל שדה"ת יעד,גודל חברת התעופה (התייחסנו לגדלים כי הם מהווים יותר משמעות בניתוח הנתונים), זמן המראה, איחור (מינוס מייצג הקדמה), ורשימת ימים מראשון עד שבת המיוצגים כביטים, שבהם 1 באינדקס שמייצג את היום הנתון, ואפס בכל השאר. לדוגמא אם היום הנתון היה שלישי, הרשימה הייתה נראית כך: 0,0,1,0,0,0,0. רשימה זו נועדה להתמודדות עם סוג מידע קטגורי. את כל נתוני הטבלה נירמלנו סביב 0 בגלל אופן מימושו של אלגוריתם הלמידה. כמו כן, הורדנו מהמסד רשומות עם שדות חסרים מכיוון שהאלגוריתם לא היה יודע להתמודד עם רשומות כאלה, וגם מספרם היה זניח ביחס למספר הרשומות הכולל.

- **ויזואליזציה:**

(כל התמונות נמצאות בקובץ ההגשה)

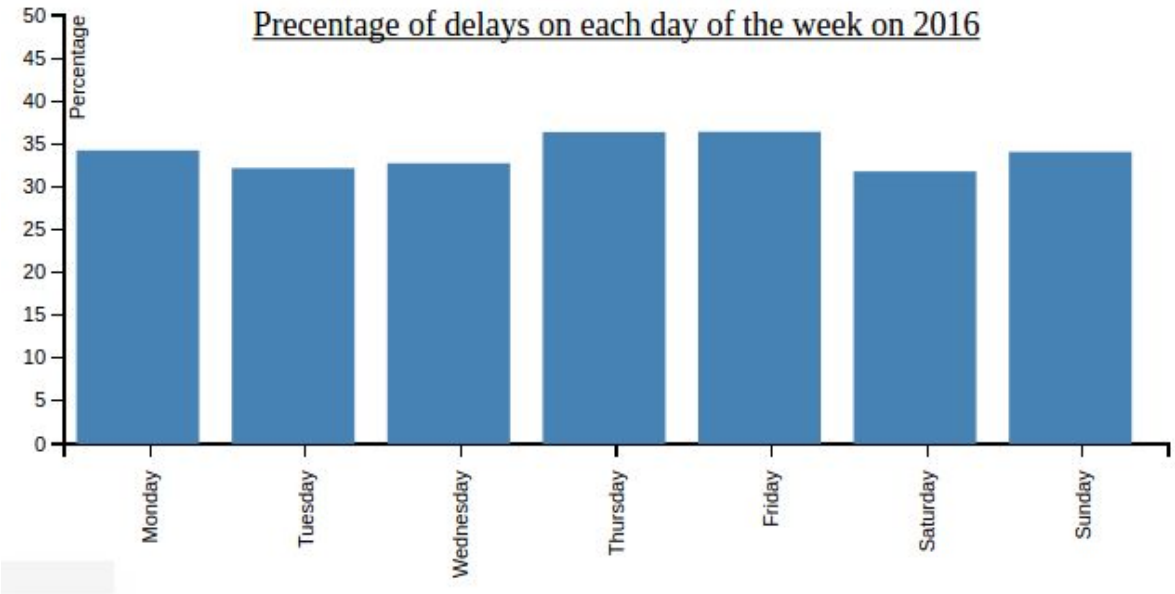
להלן הויזואליזציות בעזרתן רצינו לקבל פרספקטיבה על מסד הנתונים.

גודל שדות התעופה הקיימים באזור הנבחן:

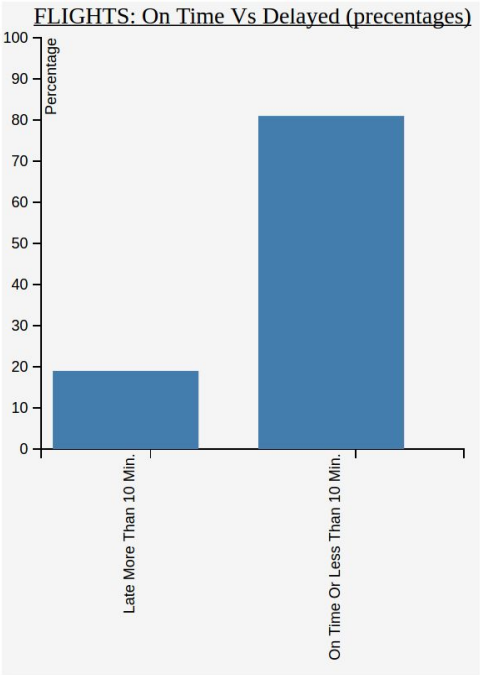
The size of airports around the US according to incoming flights number on 2016:

כמות האיחור לפי שדה תעופה:

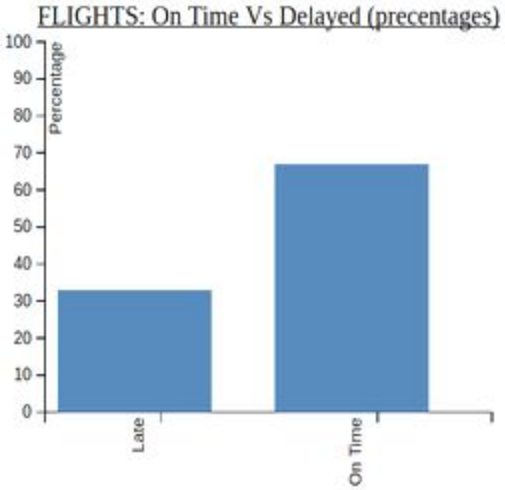
אחוז האיחורים לפי ימים במהלך השנה:



אחוז הטיסות המאחרות יחסית ואלה שהגיעו בזמן:



Threshold = 10



Threshold = 0

- **<u>שלב הלמידה</u>**:
את מסד הנתונים החדש הכנסנו לתוכנה לומדת (בקובץ LogisticLearn.py), בשיטות שצויינו לעיל, כאשר בכל שיטה העברנו פעם אחת סף של 0 ופעם נוספת סף של 10 (איחור טיסה בדקות).

- **<u>הערכת ביצועים:</u>**

לצורך הערכת הביצועים השתמשנו בפונקציה accuracy_score הנמצאת בספריית sklearn של python, ומתפקדת כמו Jaccard similarity. את הפרמטרים לפונקציה זו חילקנו ביחס 20-80 כלומר 80% מהמאגר הופרש להיות הtraining set של הלמידה, ו20% הנותרים הופרשו להיות test set. פונקציה זו סיפקה לנו את רמת הדיוק לכל שיטה של למידה שהרצנו (ראו תוצאות).

- **<u>תוצאות (אחוז רמת דיוק)</u>**:

Threshold	SGD	Logistic Regression
0	65.57%	66.02%
10	79.14%	78.09%

- **<u>אתגרים במהלך הניסויים:</u>**

בהתחלה הייתה בעיה לקרוא את קבצי הcsv בגלל גודלם, וגם להעביר אותם לקובץ db. הבעיה נבעה מכך שאיחדנו מראש את כל הרשומות לקובץ csv אחד שאותו רצינו להעביר לdb, מה שניפח אותו לגודל עצום שהיה קשה לקריאה. לכן הפתרון שלנו לחלק זה היה לקרוא קובץ קובץ ו"להוסיף" על המסד את התוצאה (באמצעות הספריות pandas, sqlite3).
בהכנת הנתונים לאלגוריתם הלמידה נתקלנו במספר בעיות:

1. האלגוריתם מקבל ערכים מספריים ומסד הנתונים שלנו אמנם מכיל ערכים מספריים, אך ללא משמעות מספרית.
2. ניסינו להמיר מספרים אלה לצורתם הבינארית אך דבר זה ניפח משמעותית את מסד הנתונים לגודל שאינו ניתן לתפעול (לפחות במחשבים שלנו). הפתרון שלנו היה להמיר את ימי השבוע לבינארי ואת מספרי החברה המרנו לגודל החברה (כמות הטיסות היוצאות).
3. קריסת האלגוריתם הלומד בגלל ערכי null. אחרי שבדקנו שמספרם זניח ביחס לשאר המסד, השמטנו אותם מהמסד.

בהכנת הויזואליזציה לניתוח, האתגרים המרכזיים היו:

1. שליפת הנתונים ממאגר המידע ושמירתם בפורמט JSON.
2. הצגת הנתונים באמצעות ספריית הגרפים d3 בשפת javascript - מדובר בספריה מורכבת ועשירה בפונקציונליות, אשר מספקת מספר רב של דרכים להצגת המידע בצורה ברורה חזותית, ובכך לאפשר ניתוח ענייני יותר של המידע.

- **<u>עבודה עתידית</u>:**

להמשך עבודה בנושא, ניתן להתרחב לאזורים נוספים מחוץ לארה"ב, לשדות תעופה ולקווי תעופה נוספים המקשרים בין אזורים שונים בעולם, למידה על מסד נתונים יותר רחב הכולל גם נתונים מיותר שנים, אולי גם שילוב גורם מזג האוויר באזור המדובר שעליו אנו מנסים לחזות.

- **<u>סיכום</u>:**

כאמור, מטרתנו בפרויקט זה הייתה לנסות לחזות נתוני טיסה בהינתן מאגר הנתונים של השנה הקודמת. בפרויקט זה ניסינו שתי טכניקות פופולאריות של למידה על מנת להראות שתי דרכים בהם ניתן להגיע להחלטה "האם טיסה X תתאחר?" ברמת דיוק מספיק טובה. בפרויקט זה למדנו רבות על שימוש בספריות מגוונות של python לצורך תפעול מסד נתונים וללמידה, וגם על שימוש בספריות מגוונות של javascript לצורך הצגת נתונים.