עומר ליכטר, ליעד וינברג, גיא סלמה
200857233, 201324332, 302602875

סדנה במדעי המידע

מטרת הפרויקט

חיזוי כמות Likes של דפי Facebook בהתבסס על דפוסי הפעילות בהם

Facebook הינו ממשק מרכזי כיום לשיתוף ותקשורת חברתיים, הן במגזר הפרטי והן במגזר הציבורי והעסקי. פעילות עמוד Facebook פעמים רבות הולכת בקנה אחד עם פעילות האדם, הגוף והעסק אותו העמוד מתעד. עסקים רבים, גופים ציבוריים, עמותות ועוד בוחרים להשתמש בממשק זה ורואים בדף ה-Facebook חלון ראווה לפעילותם, השקפותיהם ומטרותיהם. לצד זה, אנשים פרטיים משקפים דרך עמוד ה-Facebook את מהלך חייהם, חוויותיהם, דעותיהם והשקפותיהם ופרטים אישיים רבים. כל זאת נעשה באמצעות פרסום "פוסטים", תמונות, אירועים, תגובות, "תיוגים" וכו'. מחקרים רבים נערכו בנושא על מנת לבחון כיצד ניתן להעלות את כמות ה-likes עבור עמוד, מתוך הבנה שעמוד Facebook מוצלח ואהוד אשר מגיע להרבה אנשים ביכולתו לקדם לאין שעור את הגוף הציבורי העומד מאחוריו.

בתחילת הדרך, מטרתנו הייתה לחזות הצלחה וכישלון של עסקים מקומיים על בסיס דף ה-Facebook שלהם ומידע מרַשֵּם החברות ומפסקי דין על מצב העסק. לאור קשיים בתיוגם של עסקים שנכשלו, בחרנו לבסוף, בעצת המנחים, להתרכז במרכיב עיקרי של עמוד ה-Facebook – כמות ה-Likes, שכן זוהי מחווה רווחת ביותר ומרכזית ב-Facebook, והיא משקפת בצורה ברורה את הפעילות של הדף, את התמיכה בו ואת העניין של משתמשי Facebook בו. בפרויקט זה ננסה לעמוד על המרכיבים העיקריים המשקפים את פעילות והצלחת עמוד ה-Facebook דרך קישורם והשפעתם על **כמות ה-Likes** לה זוכה הדף. כמו כן, נבחן את ההשערות והמסקנות אותם מציעים מחקרים שונים בנושא.

איסוף המידע

לצורך הבאת המידע על פעילותם של דפי Facebook ציבוריים (כגון עסקים, גופים ציבוריים, עמותות, להקות וכדומה), עשינו שימוש ב-Facebook API [1]. מאחר וקצב שליפת המידע הינו מוגבל, היה עלינו לאסוף את המידע לאורך זמן רב תוך כדי ניתוחו ועריכתו ועיבוי הולך וגדל של שדות המידע הכלולים בו. בשלב ראשון, ביצענו חיפוש של דפים על בסיס כ-32 קטגוריות חיפוש נבחרות (אותן הכנסנו בהמשך גם כ-feature). בהמשך הבאנו את המידע הכללי על הדף (שם הדף, כתובת פיזית, שעות פעילות, פרטי התקשרות וכו'), את המידע על ה-feeds (posts אשר מפורסמים על ה-wall של הדף) ולבסוף את המידע על התמונות.

היות וכמות המידע אותה ניתן לשלוף הינה גם כן מוגבלת, הבאנו לכל דף את 150 ה-posts האחרונים ב-feed ואת 150 התמונות האחרונות, בהנחה כי הם מהווים מדגם מייצג של ה-posts והתמונות הכוללים. השליפה התבצעה באמצעות Facebook Client שהרמנו, בעוד שהקריאות, שמירת המידע ועיבוד ראשוני שלו נעשו ב-apache zeppelin[2].

בסופו של תהליך ארוך שכלל מהמורות רבות היה בידינו מידע על מעל ל-12,000 דפים שונים.

אפיון המידע

עיבוד ראשוני של המידע הגולמי

בשלב ראשון הכרנו ובחנו את השדות השונים שחוזרים מכל אחד מה-end points ועבדנו על טיוב השליפות שלנו. לאחר הגדרת השדות המתאימים עליהם רצינו לעבוד ביצענו את השליפות על כל אחד מהמזהים של הדפים (ראו דוגמא לקריאה ל-API בנספחים, יתר הקריאות מפורטות במחברת zeppelin בתיקיית data_collection). העיבוד הבסיסי של הנתונים שחזרו

---

[1] קישור לתיעוד ה-API: https://developers.facebook.com

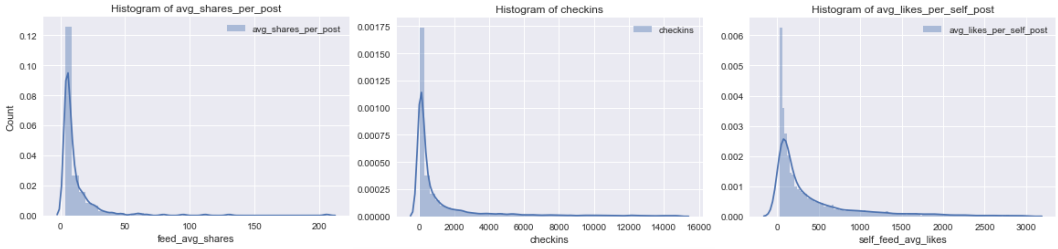
[2] מחברת המאפשרת ניתוח מידע מהיר על גבי sql, scala, spark וכיוב': https://zeppelin.apache.org

מהחיפוש (הכולל מידע בסיסי על הדף) התבצע במחברת Jupyter, בעוד שעיבוד נתוני ה-feeds וה-photos התבצע באמצעות שאילתות SQL ב-zeppelin (השאילתות באמצעותם יצרנו את ה-data sets השונים נמצאות גם הן תחת התיקייה data_collection, במחברת zeppelin ובנפרד). עיבוד זה כלל:

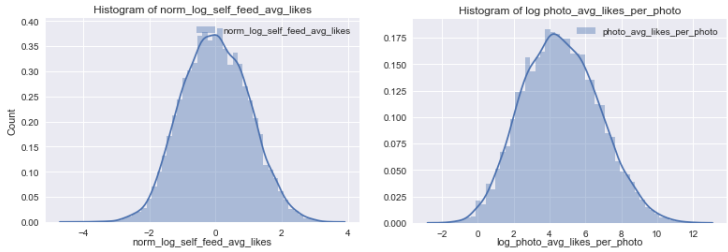
- הפיכת ערכים טקסטואליים לערכים נומריים ומחיקה שלהם מה-data.
- בינאריזציה של משתנים – משתנים אשר מכילים מידע אשר תוכנו לא רלוונטי אלא רק קיומו הוחלפו בערכים בינאריים (לדוגמה, "phone" הומר ל- "has_phone", "description" הומר ל- "has_description" וכו').
כמו כן, בוצעה אגרגציה של תוצאות שונות שחזרו עבור העמוד לכדי תוצאה יחידה (סכימה, ממוצע, תדירות, מקסימום, מינימום, סטיית תקן וכיוב'). יצוין כי לאור השוני בין posts שנכתבו על ידי העמוד ל-posts שאחרים פרסמו על ה-feed של העמוד, בחרנו לעשות הפרדה באגרגרציה לפי כל אחד מהם בנפרד. כמו כן הנתונים תויגו לשמות עם תחילית על פי סוג, לדוגמה "photo" עבור תמונות ו- "self_feed" עבור ה-posts של העמוד.
- יצירת משתנים חדשים בעלי משמעות גבוהה על ידי שילוב של features, דוגמת "photo_upload_frequency" אשר מבטא את סה"כ מספר תמונות/תקופת פעילות כוללת.
- איחוד של שדות עם שמות שונים המכילים תוכן זהה וסינון של רשומות כפולות לפי "page_id".

ניתוח ראשוני של המידע

ננסה ראשית לאפיין את המשתנים השונים המרכיבים את ה- dataset שלנו, על פי חקירת ההתפלגויות, הצגתם באופן ויזואלי והצגת מאפיינים של ההתפלגות באופן מספרי. כבר במבט ראשון, ניתן לראות כי ה- feature אותו אנו מנסים לחזות מתפלג אקספוננציאלית, וכמותו מתפלגים features נוספים חשובים, כמו מספר ה-posts, מספר התמונות, מספר התגובות וכדומה:



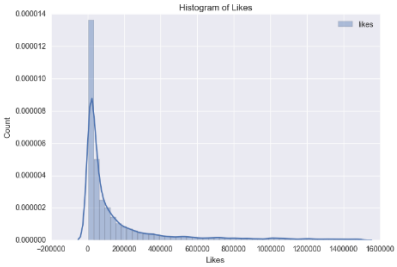
לאור תובנה זו, החלטנו להפעיל על features הללו פעולת log, על מנת לקבל התפלגות מתאימה יותר לניתוח ולעיבוד.
התקבלה התפלגות נורמלית:

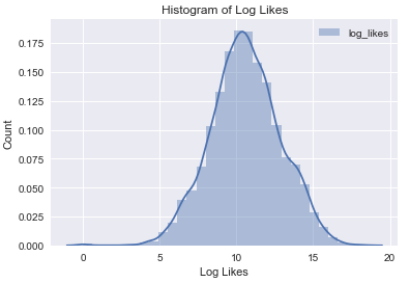


בבחינה ראשונית של features בינאריים נראה כי רובם מוטים באופן חד לאחד הצדדים (לדוגמה, לכולם יש תמונת פרופיל), אולם קיימים גם כאלה המתחלקים בצורה שווה יותר (דוגמת has_hours). משתנים קטגוריים דוגמת מדינות וטווחי מחירים נבחנו והוצגו גם הם. ניתן לראות, למשל, כי חלק נכבד מהדפים עליהם אנו עובדים הינם מארצות הברית.
ככלל, התפלגותם של features נבחרים מוצגת במחברת בחלוקה למידע כללי על הדף, posts של הדף, posts אותם אחרים כתבו על ה-wall של הדף ותמונות של הדף.

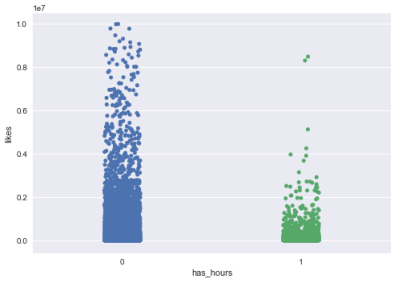
ניתוח Likes

כאמור, המאפיינים השונים לפעילות עמוד Facebook, וביניהם כמות ה-Likes הגלובלי של הדף, הינם משתנים המתפלגים אקספוננציאלית. עבור משתנה כמות ה-Likes, המקסימום עומד על כ- 10^8, והתוחלת על כ- 400,000. לאחר הפעלת פונקציית log על ערכיו התקבלה התפלגות בצורת פעמון גאוסי, בה הערך השכיח ביותר והערך הממוצע מתאחדים.





עוד לפני עיבוד מעמיק של ה-data ניתן לזהות features בולטים באמצעות הצגת קורלציות בסיסיות. נציג לשם הדגמה משתנה בינארי מעט לא צפוי – "has_hours" אשר מסמן האם מוזכרות שעות פתיחה במידע על העמוד. ניתן לראות כי **עמודים אשר אינם מכילים שדה זה זוכים באופן ניכר ליותר likes**.



ניתן להסביר זאת באופן הבא: דפים אשר מכילים שדה זה הינם לרוב עסקים קטנים ומקומיים, דוגמת מקומות בילוי, חנויות מקומיות וכדומה. לעומתם, סביר להניח כי דפים אשר אינם מכילים את השדה הינם בעלי אופי פחות "עסקי" – להקות, עמותות, דפי מעריצים, והרי שדפים מסוג זה מאופיינים בכמות likes גבוהה יותר.

על מנת למפות את הגורמים המשפיעים על החיזוי, בחנו את הקורלציות של המשנים השונים עם ה-feature המרכזי אותו אנו מנסים לחזות, על ידי חישוב מדדי Pearson ו-Spearman: לעומת מדד Pearson, אשר מוגבל להתאמה לינארית, מדד Spearman מאפיין את ההתאמה באופן שאיננו מוגבל להתאמה לינארית, והרי שאפיון זה נכון יותר בשלב זה. ניתן לראות כי ישנו מספר משמעותי של features אשר להם קורלציה > 0.5, ועל כן צפוי כי ערך זה יחסום מלמטה את טיב תוצאות ההרצות של המודלים השונים. ממפת הקרוס-קורלציה בין ה-features המרכזיים ניתן להבחין כי ל- features המשמעותיים קרוס-קורלציה של 0.6~. עובדה זו **הינה משמעותית**, שכן הדבר מעיד כי הם אינם יוצרים יתירות מיותרת ב-data אחד ביחס לשני, ועל כן שילובם בדרכים שונות יכול להביא לבניית חזאי חזק. לאחר הפעלת Log על המשתנים האקספוננציאליים וחישוב הקורלציות מחדש, הקורלציות לפי מדד Pearson עלו באופן משמעותי. ניתן להסיק כי עובדה זו תשפר את תוצאות ריצת המודלים, בעיקר אלה הלינאריים.

```
top correlation to likes using pearson estimator:
=================================================
self_feed_avg_likes            0.52
photo_max_like                 0.51
photo_avg_likes_per_photo      0.50
self_feed_std_likes            0.46
self_feed_max_likes            0.42
```

```
top correlation to likes using spearman estimator:
==================================================
talking_about_count          0.76
self_feed_avg_likes          0.71
photo_max_like               0.69
self_feed_max_likes          0.68
self_feed_std_likes          0.68
```

עיבוד וניקוי המידע, הכנת המידע להרצת המודלים (preprocessing)

מילוי ערכים חסרים (missing data imputation):

- שדות רבים המייצגים את אותו המשתנה הגיעו באופן כפול תחת כותרות מעט שונות סינטקטית (לדוגמה "wereHereCount" ו- "were_here_count"), ולכן אפשר היה להצליב ביניהם במקומות בהם ערך חסר במשתנה אחד קיים במשתנה האחר.
- רבים מהערכים החסרים מייצגים ערך 0, ועל כן השלמנו אותם כך (לדוגמה, כמות likes לתמונות).
- מידע חסר לגבי "location_country" ו- "location_city" הושלמו אחד על ידי השני (למשל ערך חסר עבור מדינה הושלם על ידי ערך קיים של עיר).

סינון features (features removal): על מנת לנקות משתנים אשר ערכיהם קבועים חישבנו את סטיית התקן, וניקינו features בעלי סטיית תקן של 0. כך למשל, סטיית התקן של שדה "has_profile_photo" הינה 0, שכן לכולם יש תמונת פרופיל. בנוסף, ביצענו מעבר על זוגות משתנים בעלי קורלציה גבוהה במיוחד ובחנו האם הם מתיישרים עם ההיגיון, האם חלקם מיותרים, והאם אחד עדיף על פני השני. כך למשל, ישנה קורלציה מושלמת בין מספר ה-posts ש**נכתבו** לבין מספר ה-posts אשר **פורסמו בפועל** ולכן לא היה צורך בשניהם.

נרמול (data normalization): טווחי הערכים ההתחלתיים של ה-features הנומריים שונים זה מזה לעתים מספר סדרי גודל. על מנת להתאים את טווח הערכים כך שיתאימו כקלט למודלים השונים, ביצענו נרמול של הערכים כדלקמן:

1. נרמול משתנה ביחס למשתנה אחר – לאור העובדה שקיימים ערכים מסוימים התלויים בגודלם של משתנים אחרים ביצענו בחלק מהמקרים נרמול של משתנה אחד ביחס לאחר. כך למשל, נרמלנו את מספר התמונות באלבומים השונים במספר התמונות הכולל של הדף אותו הבאנו. היחס הוא זה אשר מהווה המידע המשמעותי, שכן מספר התמונות באלבום מסוים תלוי באופן ישיר במספר הכולל של התמונות שפורסמו. נרמול שכזה הביא לתוצאות בטווח [0,1].
2. z-normalization – על המשתנים האקספוננציאלים הפעלנו log עם מיפוי של אפסים למינוס הערך המקסימאלי ("מינוס אינסוף") ואת תוצאתם נרמלנו באמצעות z-normalization. נרמול זה הביא למרכוז הגאוסיאנים סביב האפס עם תוחלת 0 וסטיית תקן 1. כך, בהיותם ממורכזים אחד ביחס לשני, נכון יותר יהיה להשוות ביניהם ולהכניסם כקלט למודלים. בהתאם לכך גם יתר המשתנים שנורמלו והמשתנים הבינאריים הומרו לערכים מספריים בטווח של {1,-1} במקום {1,0}.
3. min-max – את המשתנים הנומריים שאינם אקספוננציאליים נרמלנו על ידי נרמול min-max, כך שערך המינימום יקבל ערך 1-, וערך המקסימום יקבל ערך 1. גם המשתנים המסונתזים שייצרנו מריצת הרגרסיה הלינארית והרגרסיה הלוגיסטית בהמשך נורמלו בדרך זו.

הצגת משתנים קטגוריים כמשתנים בינאריים (dummies variables/one hot): המרנו את המשתנים הקטגוריים לייצוג {1,0} (ובהמשך ל- {1,-1}), על מנת לתת לקטגוריות משמעות מספרית עבור המודלים. לאור כמות גדולה של קטגוריות שונות (אשר הובילה לכמות גדולה במיוחד של features), איחדנו ידנית קטגוריות דומות זו לזו.

פיצול המידע ל-train-validation-test: חלוקת המידע לשלוש קבוצות שונות, הראשונה (train) ללימוד המכונה (72%), השנייה (validation) לצורכי כיול (18%) והשלישית (test) להשוואה בין מודלים ולבחינת חיזויי ההצלחה (10%). מנקודה זו ועד הבדיקה סופית של המודל הנבחר, לא בוצע כל שינוי ב-test set בכדי למנוע אפשרות להטיה כלשהי בעקבות עיבוד המידע.

סינון ראשוני של נקודות קצה (outlier removal): ביצענו ניקוי ראשוני של ערכים קיצוניים של ההתפלגות, על מנת למנוע הטיות לא רצויות אשר אינן מאפיינות את ה-data, עוד בטרם הרצת מודל כלשהו. ניקוי זה נעשה על בסיס משתנה ה- likes, המתפלג כאמור אקספוננציאלית. כאמור, לאחר הפעלת פונקציית log על ערכיו התקבלה התפלגות בצורת פעמון גאוסי, בה הערך השכיח ביותר והערך הממוצע מתאחדים. אפיינו את חוסר הסימטריה של הפעמון על מנת שלא לשנות את מאפייניו בעת ניקוי outliers, על פי המומנטים מסדר שלישי ורביעי של ההתפלגות:

```
skewness: -0.00136
kurtosis: -0.02257
```

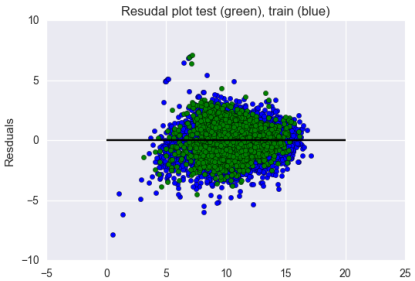
במקרה זה לא נראתה הטיה משמעותית של הפעמון ולכן יכולנו לבחור לנקות ערכים הנמצאים במרחק של למעלה משתי סטיות תקן מהתוחלת. ניקוי זה הביא לצמצום כמות הרשומות בכ- 4.5%.

קוונטיזציה של מספר ה-Likes לערכים דיסקרטיים המתפלגים אחיד: . בעצת המנחים, על מנת לשפר את ביצועי המודלים ולאפשר חיזוי מוצלח, חילקנו את משתנה ה- likes למספר מחלקות (bins) שוות הסתברות. באופן טבעי, הערכים המאפיינים כל bin (כגון ממוצע, אחוזונים) שומרים על ההתפלגות האקספוננציאלית, שכן החלוקה נעשתה באופן שווה הסתברות.

העשרה של features נוספים (feature engineering)

Logistic Regression בינארי – השתמשנו ב-Logistic Regression בינארי על מנת לחזות את המחלקה של כל דף, כאשר כל פעם אימנו את המודל על סף משתנה (בין 0 bin לבין 1 bin בין 1 bin לבין 2 bin ובין 2 bin לבין 3 bin). כל אחת מהריצות הנ"ל יצרה הפרדה טובה יחסית, עם ציוני recall ו-precision הגבוהים מ-80. מעניין לראות כי בהפרדה השלישית הציונים היו טובים במיוחד והתקרבו ל-90. הדבר מצביע על כך שיכולת הפרדה בין דגימות גדולות במיוחד לשאר הדגימות טובה באופן יחסי, יתכן משתחום זה איננו חסום מלמעלה. לאחר שבחנו את תוצאות הריצות השונות ואף ניסינו לבצע fusion שלהם לכדי סיווג יחיד של דגימה למחלקה, החלטנו להשתמש בתוצאותיהם (prediction_probs) כשלושה features נוספים. לא עשינו חיזוי של "נמצא או לא נמצא" במחלקה היות והדבר נעשה במסגרת הריצה הבסיסית של LR multi-class ("one vs all").

Linear Regression – זהו מודל פשוט שלא דורש טיוב פרמטרים, אותו הרצנו בכדי לבחון את יכולות החיזוי שלנו, ללא תלות בחלוקה למחלקות. היות וה-likes מתפלגים אקספוננציאלית, החלטנו לעשות את ההתאמה לפי ה-log של ה-likes. בתחילה ראינו מספר features אשר המקדמים שהותאמו להם ברגרסיה היו גדולים במיוחד ואחרים קטנים במיוחד. לאחר בחינה נוספת נוכחנו לגלות כי קיימת בין אותם משתנים תלות (למשל מדינה ויבשות בהן היו מספק קטן במיוחד של מדינות) ולכן בחרנו להסיר חלק מהם עוד לפני הרצת המודלים כדי למנוע תלויות. בנוסף, ניסנו להבין מיהם ה-outliers של גרף ה-residuals עבור ריצת הרגרסיה הלינארית (כמוצג משמאל). איתרנו את עמודי ה-Facebook הרלוונטים וחקרנו אותם באופן ידני. התגלה כי רובם ככולם הינם דפי Facebook חדשים (חלקם הוקמו ממש באותו היום בו ביצענו את השאילתות) וכי הם מאופיינים במספר תמונות, פרסומים ומספר Likes בודדים. מכאן, התקבלה ההחלטה כי נכון לנקות ערכי קיצון אלה של ההתפלגות. היות והשגיאה הריבועית הממוצעת עומדת על 1.02, ניתן ללמוד כי החיזויים הנומריים באמצעות מודל זה נותנים הערכה טובה לסדר הגודל של ה-Likes של העמוד. לאור קבלת תוצאות טובות באופן יחסי במודל זה החלטנו לעשות שימוש גם ב-Lasso ולהשתמש בתוצאות הנומריות של הרגרסיה הלינארית לאחר נרמול כ-feature נוסף.



בחירת features (feature selection)

בחירת features: לשם בחירת features בשאר המודלים השתמשנו בפונקציה מובנית של sklearn המחזירה features נבחרים לפי סף מוגדר מראש, בצורה זו יצרנו מראש מספר סטים של features בגדלים שונים עבור כל מודל (מדורגים בצורה בסיסית). לאחר שטענו ה-data לפי כל אחת מקבוצות ה-features שנבחרו מראש, ביצענו הסרת outliers נוספת (רק על ה-features שנבחרו, בכדי להימנע מזריקת כמות גדולה מידי של דגימות) באמצעות פונקציה הממשת Grubbs' test – נציין רק ששיטה זו מיועדת למשתנים המתפלגים בצורה נורמאלית, לכן כאשר הפונקציה ניסתה להסיר יותר מידי דגימות בערכי alpha נמוכים מאוד, הנחנו שזה נובע מכך שההתפלגות ב-feature הספציפי רחוקה מידי מלהיות נורמאלית ולא הסרנו את הדגימות.

טיוב פרמטרים (parameter optimization): לאחר שלב בחירת ה-features, טייבנו המשתנים של המודל (היפר פרמטרים) באמצעות gridSearch על כל אחד מקבוצות ה-features, והרצנו בשנית את כלל המודלים לשם דירוגם. לבסוף, בחרנו את הפרמטרים וקבוצת ה-features שהשיגה את הביצועים הטובים ביותר (במונחי f1-score) על ה-validation set.

הרצת המודלים

K-Nearest Neighbors – מודל בסיס ופשוט אותו הרצנו על מנת לקבל benchmark לתהליך כולו. הריצה עצמה עם ערכי k משתנים בטווח שבין 2 ל-70 בקפיצות של 4. בסופו של תהליך הגענו לדיוק של ~0.55 על ה-validation set. בנוסף,

עשינו הרצות של מודל זה גם עם PCA (ללא הורדת ממדים) על מנת להבין במעט את השפעתו על אחוזי הדיוק. לאחר הרצת PCA הגענו לדיוק של ~0.56. מתוך רצון לשמור על משמעות ה-features (בשביל גזירת המשמעויות ישירות מתוך התוצאות), ולאור ההשפעה הזניחה יחסית של ה-PCA במודל זה, החלטנו שלא להרחיב את השימוש בשיטה זו ביתר המודלים.

Logistic Regression (multiclass) – בכדי למצוא את ה-solver ופרמטר ה-multi-class האופטימליים ביצענו הרצות gridSearch על סטים מצומצמים של data, בכדי לשמור על זמני ריצה הגיוניים. מאחר ורק solver אחד תומך ב-penalty מסוג l1 (לו גם היו את הציונים הנמוכים ביותר) החלטנו להתמקד ב-l2 וב-newton-cg שנתן את הביצועים הטובים ביותר. בנוסף, השתמשנו בגישת multinomial ולא one vs rest מאחר ובעצם הוספת ה-features של ההפרדה הבינארית (בשם lr_predict_x) הכנסנו התייחסות של סיווג דומה מאוד להתנהגות one vs rest של המודל (כמובן שגם בדקנו ואכן בצורה הזו גם קיבלנו תוצאות טובות יותר).

SVM (multiclass) – בתחילה, בחנו גם את תוצאות ההרצות של SVM בינארי על ידי מפרידים (כפי שתואר קודם). היות ותוצאות אלו היו פחות טובות מהתוצאות של ריצות דומות של logistic regression החלטנו שלא להשתמש בהם להמשך העבודה. עבור המודל ה-multiclass, בחנו עבור ה-kernel הלינארי ו-rbg kernel את הטיובים של פרמטר המשקולות C ופרמטר γ (רלוונטי ב-rbf בלבד). נוכחנו לגלות כי תוצאותיו של rbf kernel נופלות מאלו של ה-kernel הלינארי ולכן בחרנו להשאיר ב-gridSearch רק את האחרון. הרצנו את gridSearch בטווח לינארי בין 7 ל-15, בו קיבלנו את הביצועים האופטימליים. היות ומודל זה מצפה לקבל רק ערכים בטווח {-1,1} (או {0,1}) ולכן ביצענו נרמול min-max נוסף לפני הרצותיו השונות. יצוין כי זמן הריצה של מודל זה הינו ארוך במיוחד עבור מספר גדול של features. היות והתוצאות עבור מספרים נמוכים יותר של features היו פחות טובות, נאלצנו להמשיך ולהריצו על כלל ה-features, דבר אשר הקשה עלינו רבות בניסיונות השונים לטיוב הפרמטרים.

Lasso Regression – בדומה לרגרסיה הלינארית גם במקרה זה, החלטנו לעשות את ההתאמה לפי הלוג של ה-likes. ביצענו טיוב ראשוני של ההיפר פרמטר α באמצעות LassoCV. שיטה זאת בוחנת ביעילות את ההשפעה של שינויים בפרמטר α על ההתאמה באמצעות cross-validation. כפי שביצענו ברגרסיה הלינארית גם כן בחנו את השגיאה הריבועית הממוצעת ואת גרף ה-residuals שהתקבל. על החיזויים הנומריים שהתקבלו הפעלנו אקספוננט לקבלת חיזוי מספר ה-likes של העמוד ולאחר מכן מפינו אותם ל-bins, על מנת לבחון מודל זה אל מול יתר מסווגי ה-multiclass.

Random Forest – לאחר שמצאנו כי random forest משיג את התוצאות הטובות ביותר במסגרת הבדיקות, החלטנו לכתוב בעצמנו את טיוב הפרמטרים של מודל זה עבור קריטריון gini (אשר השיג תוצאות טובות מאלו של entropy) ועם עם הגבלה ל-250 עצים (n_astimators, אשר הביא לתוצאות טובות יותר מיתר האפשרויות אותן בחנו). לצורך דירוג הפרמטרים, השתמשנו במדד המדרג את ה-features (feature_importance) המובנה במודל הבסיס. מעניין לראות כי במרבית המודלים שהורצו, ה-features שדורגו במקומות הראשונים היו קשורים לLikes שניתנו על ידי משתמשים לפוסטים ותמונות של הדף. בהמשך, טייבנו את הפרמטרים המתאימים ואת הפיטצרים הנבחרים במסגרת מעבר איטרטיבי על אפשרויות שונות של עומק מקסימאלי, מספר מינימאלי של עלים וציון סף במדד feature_importance.

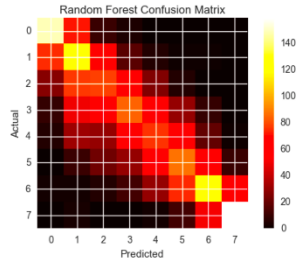
הערכת המודלים (model evaluation)

ניתוח והערכת המודלים

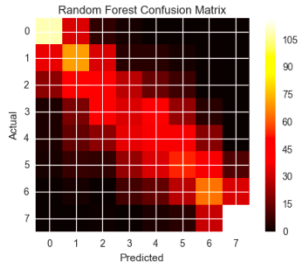
ההשוואה בין התוצאות השונות אותן קיבלנו במהלך הפרויקט נעשתה באמצעות f1 score, המשקלל את ה-recall וה-precision עם התייחסות למספר של true positives, false negatives ו-false positives בכל מחלקה (micro averaging).

הצגת תוצאות ה-multiclass נעשתה באמצעות confusion matrix, מטריצה המדגימה בצורה ברורה ופשוטה את נכונות החיזוי בכל מחלקה (bin). במטריצה זו ציר ה-x מייצג את חיזויי המחלקה וציר ה-y מייצג את המחלקה בפועל. כמו כן, ערכי צבע בהירים מייצגים ערכים גבוהים (כלומר, מספר גדול של דגימות ממחלקה y שנחזו על ידי המודל במחלקה x). נרצה לקבל מרכוז של ערכי צבע בהירים סביב האלכסון הראשי של המטריצה, שכן הדבר יצביע על התאמה גבוהה בין ה-bin הרצוי ל-bin אשר נחזה. בנוסף, על מנת לקבל את הציון המתאים עבור חלוקה לשמונה מחלקות, עם טווח שגיאה של bin יחיד (one-bin-away), מימשנו פונקציה המחשבת את ציון ה-f1 לפי הנוסחה הבסיסית שלו[3].

הצגת תוצאות הרגרסיה נעשו באמצעות residuals plot המציג את תוחלת השגיאה הריבועית של המדגם. לאחר הריצות השונות, ניתחנו את הדפים בעלי תוחלת השגיאה הריבועית הגבוהה ביותר על מנת לשפר את הביצועים שלנו ולזהות נקודות קיצון במידע. כמו כן, עבור lasso לאחר הפעלת אקספוננט והמרת התוצאות חזרה למחלקות, קיבלנו מסווג multiclass אשר תוצאותיו הוצגו כמו יתר מסווגי ה-multiclass. כמו כן, הצגת תוצאות המסווגים הבינאריים נעשתה באמצעות ROC curves ו-precision recall curves פשוטים.




```
Random Forest validation f1-score:0.403189066059
==================
1-Bin-Away scores:
==================
Avg Precision: 0.813694602185
```




```
Random Forest test f1-score:0.414576802508
==================
1-Bin-Away scores:
==================
Avg Precision: 0.807286404038
Avg Recall: 0.80474618688
Avg f1-score: 0.802199648725
```

תוצאות סופיות

כפי שניתן לראות מתוצאות כלל המודלים אותם הרצנו, הגענו לחסמים יחסית אחידים על ה-validation set. עבור ארבעה מחלקות תוצאות ה-f1 score שהתקבלו נעו סביב ה-0.63~ ועבור שמונה מחלקות תוצאות ה-f1 score שהתקבלו נעו סביב ה-0.45~ ו-0.8~ ב-one-bin-away.

מבין שלל המודלים שהורצו במסגרת הפרויקט, המודל שהשיג את הביצועים הטובים ביותר על ה-validation set, לאחר טיוב הפרמטרים וה-features היה Random Forest. מודל זה הורץ גם על ה-test set והשיג ביצועים דומים לאלו שהתקבלו על ה-validation set, עם one-bin-away f1 של 0.81~. יצוין כי היות ומודל זה מכיל אלמנט רנדומי, הרי שהתוצאות המתקבלות בכל ריצה וריצה שונות במעט זו מזו.

לתחושתנו, היכולת להגיע לחיזויים מדויקים יותר מוגבלת ב-data set ההתחלתי אשר היה מצומצם בהיקפו בשל ההקצאות להבאת מידע מ-facebook API, והכיל מגוון גדול מדיי של דפוסי פעילות שונים של דפים, כפי שיוסבר תחת מסקנות.

מסקנות מרכזיות

1. **השונות הגבוהה בדפוסי הפעילות של דפים בעלי likes דומים, פגמה בתוצאות החיזוי -**

הקטגוריות בפרויקט זה נבחרו מתוך שלל הקטגוריות הציבוריות של Facebook. ההנחה הבסיסית הייתה כי כמות ה-"likes" עבור דף הולך בקשר ישר עם כמות ה- "likes" עבור התכנים שבו – ה-posts, התמונות, האירועים וכו'. אולם נוכחנו לגלות שלאו דווקא כך הדבר, וכי שונות גבוהה וסטיות גדולות מאוד מטות את הסיווג באופן משמעותי. נראה זאת על ידי דוגמה: לעמוד של רשת McDonald's[4] ולעמוד של הזמר Bruno Mars[5] יש מספר דומה של לייקים ותדירות דומה של העלאת תמונות ופוסטים חדשים. עם זאת, בעוד שמספר ה-Likes עבור כל פוסט או תמונה של רשת McDonald's הינו נמוך, לכל היותר כמה מאות, מספר הלייקים לפוסט או תמונה של ברונו יכול להגיע למאות אלפים. הבנה זאת ליוותה

[3] על פי הנוסחה; $f1 = 2 * \frac{precision*recall}{precision + recall}$

[4] עמוד פייסבוק של mcdonald's; https://www.facebook.com/McDonalds/?brand_redir=50245567013

[5] עמוד הפייסבוק של הזמר ברונו מארס; https://www.facebook.com/brunomars/

אותנו לקראת סוף הפרויקט ובעקבותיה ניסנו לבצע חלוקה של ה-data לקטגוריות באמצעות קטגוריות החיפוש וקטגוריות מובנות של Facebook, אשר לא הביאו לשיפור בתוצאות לאור מספר קטן של דגימות בכל קטגוריה. בנוסף, ניסנו להגדיר קטגוריות-על המכילות מספר קטגוריות של Facebook או מספר מדינות ואף לבצע חלוקה באמצעות k-means, אולם גם הם לא הועילו ולא הביאו לשיפור בתוצאות.

2. **התייחסות למחקרים בתחום** – מחקרים שונים מציעים פעולות לקידום הלייקים של הדף[6]. נעמוד בקצרה על מספר עצות נבחרות וביטויי שלהם במסגרת הפרויקט שלנו:

   א. תיוג (tag) של אנשים ועמודים אחרים ב-posts, בתמונות, באירועים וכו', מתוך הנחה כי תיוג מביא לפרישה רחבה יותר, שכן הוא מופיע ב-feed של הדפים אשר תויגו. על פי הקורולציות שחושבו באופנים השונים וה-feature-ים אשר נבחרו במודלים השונים, נראה כי נתון זה אינו נמצא בין המשתנים בעלי הקורלציה המשמעותית ביותר לכמות ה-likes. על אף האמור, היות וכמות התמונות והפוסטים אותם הבאנו הינה מוגבלת והרי ומרבית הפוסטים והתמונות לא מכילות תיוגים, יתכן כי לא היה מספיק מידע בכדי להעריך כראוי את השפעת התיוג על כמות ה-likes של העמוד.

   ב. ביצוע פעולות אשר יישמרו על עדכניות ה-post, דוגמת סבבים של תיוג, מחיקת התיוג ותיוג מחדש, שינויים קלים בתוכן הפוסטים וכיוב'. גם במקרה זה, בעקבות מספר נמוך של פוסטים שעודכנו בדאטה (המכילים מידע רלוונטי בשדות self_feed_count_post_updates ו-self_feed_count_post_prc , לא ניתן לאשש או להפיך המלצה זו.

   ג. כתיבת posts קצרים (בין 40-50 תווים ועד 100 תווים לכל היותר). בניסיונות הראשונים להבאת המידע על ה-posts, הבאנו גם את התכנים של ה-posts ואף של התגובות השונות. בעקבות חריגות מכמות ההקצאות נחסמה בפנינו האפשרות להביא מידע פעמים רבות. אי לכך, בסופו של דבר לא הבאנו את התכנים של הפוסטים והתגובות השונות ולכן גם כן אין ביכולתנו לעמוד על האפקטיביות של המלצה זו.

המלצות

1. **מיקוד בקטגוריה יחידה** – כפי שצוין בחלק המסקנות, הדפוסים השונים של דפים בעלי מספר לייקים דומה פגעו בתוצאות החיזוי הסופיות. לעניות דעתנו, במידה והמידע אותו היינו מביאים היה יותר הומוגני (כולל רק דפים של עסקים קטנים, רק דפים של אמנים וכיוב'), דפוסי הפעילות הכלליים של הדפים השונים הייתה דומה הרבה יותר ומאפשרת מתן תחזיות משופרות. לצורך כך, היינו מביאים את כל התוצאות מאחת מקטגוריות החיפוש הרצויות ולא מביאים מספר קטן יחסית של דפים מכל קטוגריה.

2. **הגדלת כמות המידע הנשלף מכל דף** – כפי שתיארנו בחלק המסקנות, גיבוש תובנות קונקרטיות בנוגע לפעילויות ספציפיות שמגדילות את כמות ה-likes של הדף דורש הבאת כמות גדולה יותר של מידע מכל דף. במידה ולא היה לנו מגבלה של כמות זאת, היינו מביאים יותר מ-150 התוצאות הראשונות מכל אחד מהמקורות (תמונות ו-posts). בדרך זו, היה ניתן לעמוד טוב יותר על מועד התחלת פעילות הדף ולהשיג מידע נוסף המתאר פעילות הדף גם בציר הזמן (תדירויות משתנות, seasonality וכו'). בנוסף, לא היינו מגבילים את הסינון לשדות ספציפיים אלא מביאים גם את התוכן של התוצאות, התגובות שפורסמו עליהן וכיוב'. לעניות דעתנו, מידע נוסף זה היה חושף בצורה יותר טובה את דפוסי הפעילות של הדף ומאפשר אבחנה טובה יותר בין הדפים השונים.

---

[6] כתבה המכילה המלצות דומות; https://www.linkedin.com/pulse/50-free-ways-increase-your-facebook-page-likes-gripel-official-?forceNoSplash=true

נספחים

1. לינק Facebook API explorer עם קריאה לדוגמא ל- feed end point, לאחר טיוב השדות המוחזרים:
https://developers.facebook.com/tools/explorer/?method=GET&path=388715684614122%2Ffeed%3Ffields%3Did%2Ccreated_time%2Cfeed_targeting%2Cfrom%2Cis_hidden%2Cis_published%2Clink%2Cmessage_tags%2Cpicture%2Cplace%7Bid%7D%2Cshares%2Csource%2Cstatus_type%2Ctargeting%2Cto%2Cupdated_time%2Cwith_tags%2Ccomments.summary(true)%7Blike_count%7D%2Clikes.summary(true).limit(0)%2Cattachments%7Bdescription_tags%2Cmedia%2Ctarget%7D&version=v2.4

2. קישור ל-git: https://github.com/liadwg/data_mining_workshop.git :
   - מחברת סופית –Like prediction workshop – FINAL:
     https://github.com/liadwg/data_mining_workshop/blob/master/Like%20Prediction%20Workshop%20-%20FINAL.ipynb
   - מחברת zeppelin (איסוף הנתונים) - zeppelin_notebook:
     https://github.com/liadwg/data_mining_workshop/blob/master/data_collection/zeppelin_notebook.json
   - מחברת נספחים - Like Prediction Workshop – Addendum (בדיקות וריצות של מודלים נוספים):
     https://github.com/liadwg/data_mining_workshop/blob/master/Like%20Prediction%20Workshop%20-%20Addendum.ipynb

3. קישור ל-trello: https://trello.com/b/ONPeWkjA/ds-workshop