המכללה האקדמית להנדסה בתל־אביב אפקה
המחלקה להנדסת תעשייה וניהול

כריית מידע – פרויקט סוף

Google Play Store Apps

מרצה:

ד״ר עומר צוק

מגישים:

רון יצחק בורובר

איתי דנינו

המכללה האקדמית להנדסה בתל־אביב אפקה
המחלקה להנדסת תעשייה וניהול

תוכן עניינים

תמצית מנהלים

מטרת הפרויקט הינה לספק למפתחי אפליקציות כלים אשר מטרתם לתת חיזוי שיתבסס על סט נתונים אשר נלקח מאתר "Kaggle", באמצעותו יוכלו לבצע בדיקה של מספר פרמטרים אשר נמדדים באפליקציות מתוך חנות האפליקציות של גוגל, שלבסוף אנו מעוניינים לנסות ולחזור כיצד ליצור אפליקציה "מצליחה", כלומר תזכה לכמות הורדות גבוהה וגם לדירוגים גבוהים.

לאחר ביצוע ניקיון של המידע, ניתחנו את המידע הקיים והשתמשנו בכלים אשר למדנו בקורס, והצלחנו לספק מספר מסקנות שבאמצעותן ניתן להגדיר מהי אפליקציה "מוצלחת".

לצורך קבלת אפליקציה "מוצלחת", הגדרנו כמות הורדות מינימלית של מעל מיליון הורדות, ודירוג מעל 4.0.

באמצעות עמודות נוספות בטבלת הנתונים, חיפשנו קורלציה מסוימת אשר באמצעותה הגענו למסקנה הראשית שלנו, ואכן בשלב ניתוח המידע הצלחנו להראות כי החל ממספר מסוים של ביקורות שמשתמשי אפליקציות ביצעו לאפליקציה מסוימת, אזי ישנו סיכוי גבוה מאוד שאפליקציה תהיה מוצלחת, לכן אנו ממליצים לאותם מפתחי אפליקציות לתת דגש על בקשת מילוי ביקורת מכל משתמש באפליקציה, מעבר לדגשים הנפוצים כמו נוחות, ויזואליות ברורה, חינמיות ועוד.

מבוא

האם ניתן לחזות הצלחה של אפליקציה בחנות של גוגל (אנדרואיד) שתבוא לידי ביטוי בכמות הורדות ודירוג גבוהים?

מטרת שאלה זו הינה להעניק למפתחי אפליקציות פלטפורמה אשר באמצעותה יוכלו לקבל תובנות והכוונה בדבר פוטנציאל פיתוח אפליקציה אשר נחשבת ל"מוצלחת".

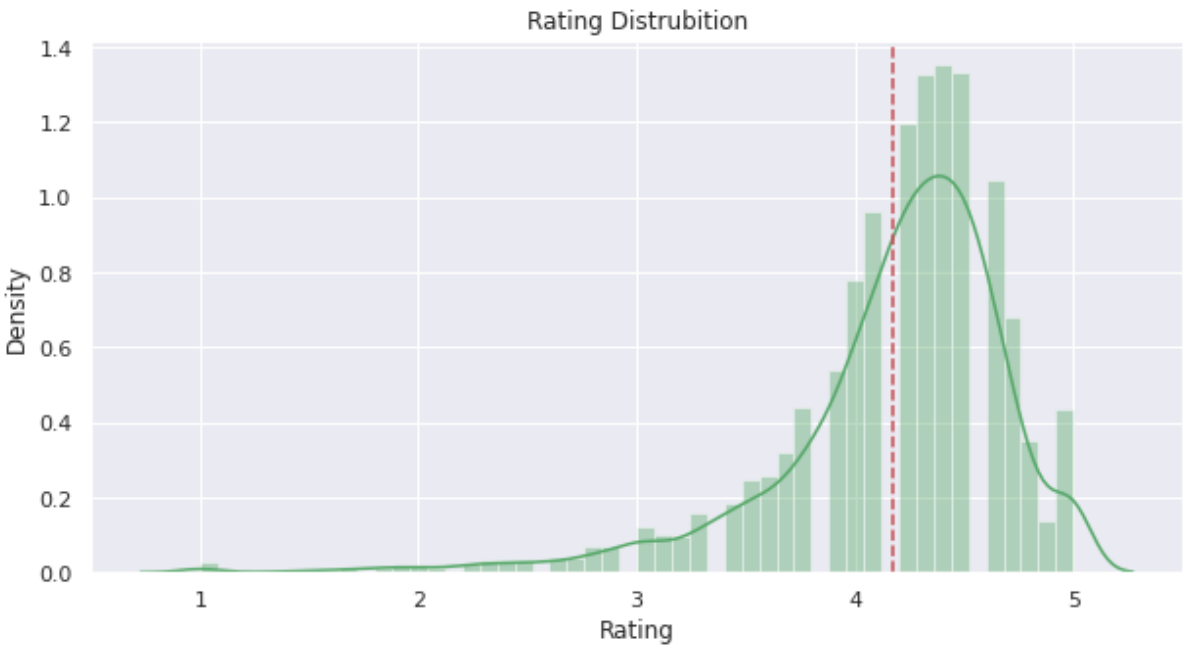
חברה או קבוצת אנשים אשר מעוניינים לפתח אפליקציה יעשו הכל על מנת להגיע לכמות ההורדות הגבוהה ביותר ולאחר מכן לבסס את מעמדה של האפליקציה כאפליקציה "מוצלחת" ע"י קבלת דירוג גבוה מן המשתמשים.

בשנים האחרונות ניתן לראות גידול בכמות האפליקציות בשוק, כאשר לאו דווקא כלל האפליקציות אכן מצליחות "לפגוע" בצורך האמיתי של המשתמשים מבחינת מספר פרמטרים כגון: צורך, עיצוב, נוחות, מחיר ועוד, ולכן טרם פיתוח האפליקציה על המפתחים לבצע חיזוי, על פי מספר פרמטרים אשר רלוונטיים לאותה אפליקציה, לפיהם יוכלו לקבוע האם האפליקציה אשר ברצונם לפתח תצליח.

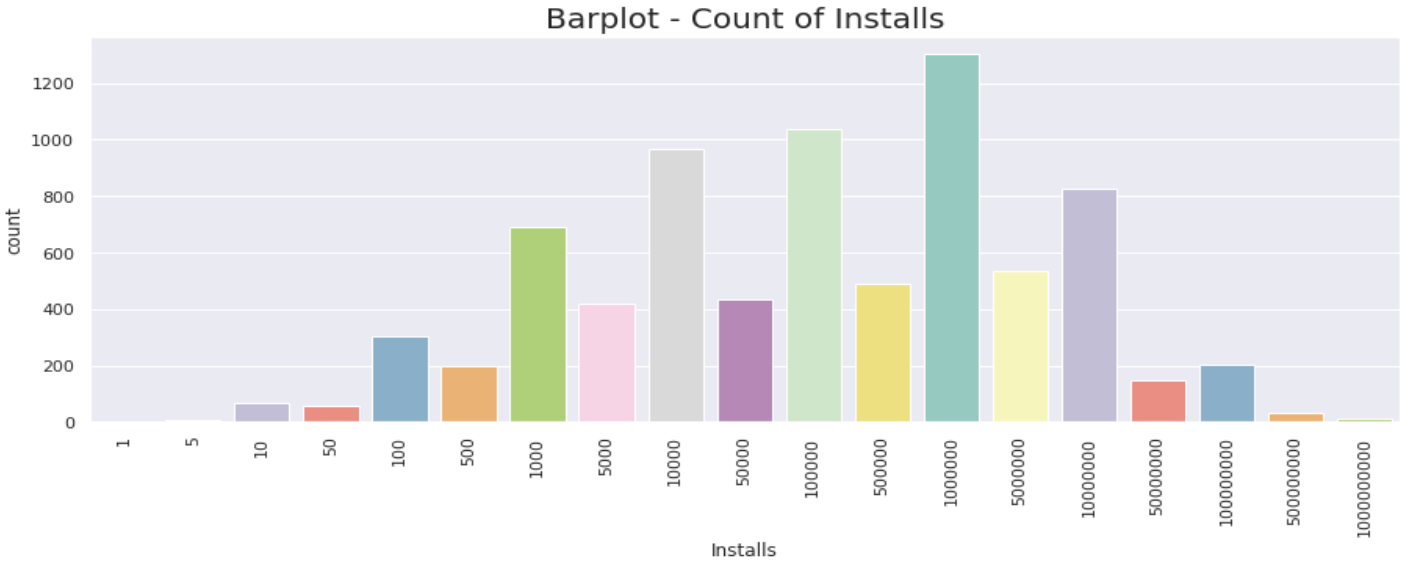
שיטות פעולה

❖ **Data Cleaning**

- תחילה, המרנו את הערכים מעמודת "Size" אשר אינם נומריים לערכים נומריים, ומחקנו את הערכים אשר אינם רלוונטיים.
- מצפייה בטבלה, ישנם 3138 ערכי Null בכל הטבלה, לכן נבצע תחילה מחיקה של ערכים אלו, לא מדובר בכמות נתונים גדולה, ואף אינם מייצגים Time Series כלשהו, לכן ניתן למחוק אותם.
- נגדיר Target Value שיקרא "Success", והינו תלויה בשני פרמטרים: "Rating" ו- "Installs".
- עבור עמודת "Installs" – ביצענו הורדת תווים והפיכת הערכים בעמודה לערכים נומריים.
- עמודת "Rating" הינה בעלת ערכים נומריים בלבד.
- לאחר מכן, קבענו דירוג סף וכמות הורדות מינימלית לצורך הגדרת "הצלחה"- דירוג מעל 4.0 וכמות הורדות מעל מיליון, וכעת אנו מעוניינים לבדוק האם קביעתנו הראשונית נכונה אל מול הנתונים הקיימים בטבלה.
- בגרף זה, ניתן לראות כי הדירוג הממוצע של אפליקציות בחנות של גוגל הינו 4.17 ומכאן אנו למדים כי קביעתנו הראשונית בנושא הדירוג הייתה הגיונית.

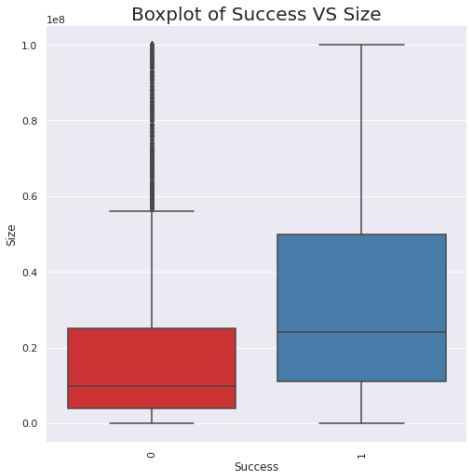


- בגרף הבא, ניתן לראות בבירור גם כאן כי השערתנו הראשונית בנוגע לכמות ההורדות הייתה טובה, כלומר, כמות ההורדות הגבוהה ביותר הינה מעל למיליון הורדות.



- טרם הורדת העמודות "Installs" ו-"Rating" נבדוק קורלציה בין עמודות אלו לבין עמודות נומריות נוספות שיש לנו בטבלה, אשר אנו מעריכים כי עלולה להיות קורלציה מסוימת.
- נבצע המרה לעמודה "Price" מערכים קטגוריאליים לערכים נומריים, לדוגמא: נעבור מ-'18$' לערך 18.
- נפעיל את מבחן Pearson בין העמודות "Rating", "Installs", "Price", "Reviews".
- נבחין תחילה כי בין העמודות "Rating" ל-"Installs" לא קיימת קורלציה (0.053).
- לאחר מכן, נשים לב כי גם עמודת "Price" איננה משפיעה על העמודות "Rating" ו-"Installs", כנ"ל על עמודת "Size".
- ולבסוף נבחין כי בין העמודה "Installs" ל- "Reviews" ישנה קורלציה מסוימת (0.63).

- כעת, נוריד את עמודות "Rating" ו-"Installs" לאחר שהגדרנו עמודה חדשה "Success".
- נחפש קורלציה בין ה-Target Value לבין אחת העמודות. ניתן לראות באמצעות ה-Boxplot הבא כי אפליקציות שהן "מצליחות" אז ככל הנראה ה"Size" יותר גבוה:



- נבצע מבחן חי-בריבוע על מנת לראות קורלציה בין "Success" לשאר העמודות:

	Variable	Chi squared value	p-value
0	Reviews	7639.556250	0.0
1	Size	1082.290598	0.0
2	Genres	981.990479	0.0
3	Price	263.453084	0.0

ניתן לראות שכאשר ה- p-value קטן מ- alpha=0.05, ישנה קורלציה מסוימת בין ה-Target Value לבין הפיצ'רים.

- לבסוף, נוריד את העמודות אשר קבענו שאינן רלוונטיות (בסט הנתונים).

❖ **One Hot**

- לאחר שלא מצאנו קורלציה בין "Success" לבין שאר העמודות שבדקנו, החלטנו לבחור בעמודה "Genres" ובה אולי נמצא קורלציה.
- נמיר את העמודה הקטגוריאלית "Genres" ל-One Hot , לצורך הפעלת אלגוריתמים.

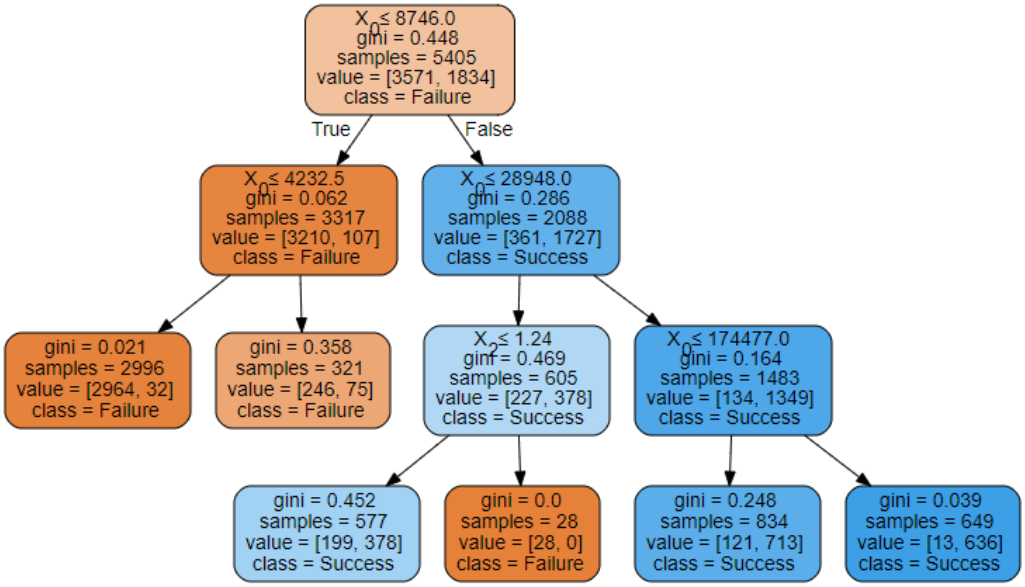
❖ **פעולות טרם הפעלת אלגוריתמים**

- נגדיר משתנה X אשר יאופיין באמצעות Data Frame, ויכיל בתוכו את כלל העמודות למעט עמודת ה-Target Value שהיא עמודת ה- "Success".
- נגדיר משתנה Y הוא ה-Target Value (עמודת ה-"Success"), יאופיין כווקטור שיקבל ערכים של 0 ו-1.
- לקחנו 30% מה-Data לצורך מבחן, ו-70% לאימון.

אלגוריתמים

❖ Decision Tree

➢ כאשר הפעלנו את האלגוריתם, קיבלנו Accuracy של 0.8925 ורצינו לראות אם נוכל לשפר דיוק זה, לשם כך הגדרנו alpha=0.001 ו-max_depth=3, ואכן קיבלנו Accuracy של 0.9158.



➢ לאחר מכן, נסתכל על ה- Decision Tree ונבחין כי ישנה עמודה אחת דומיננטית אשר מיוצגת ע"י X0 – "Reviews".

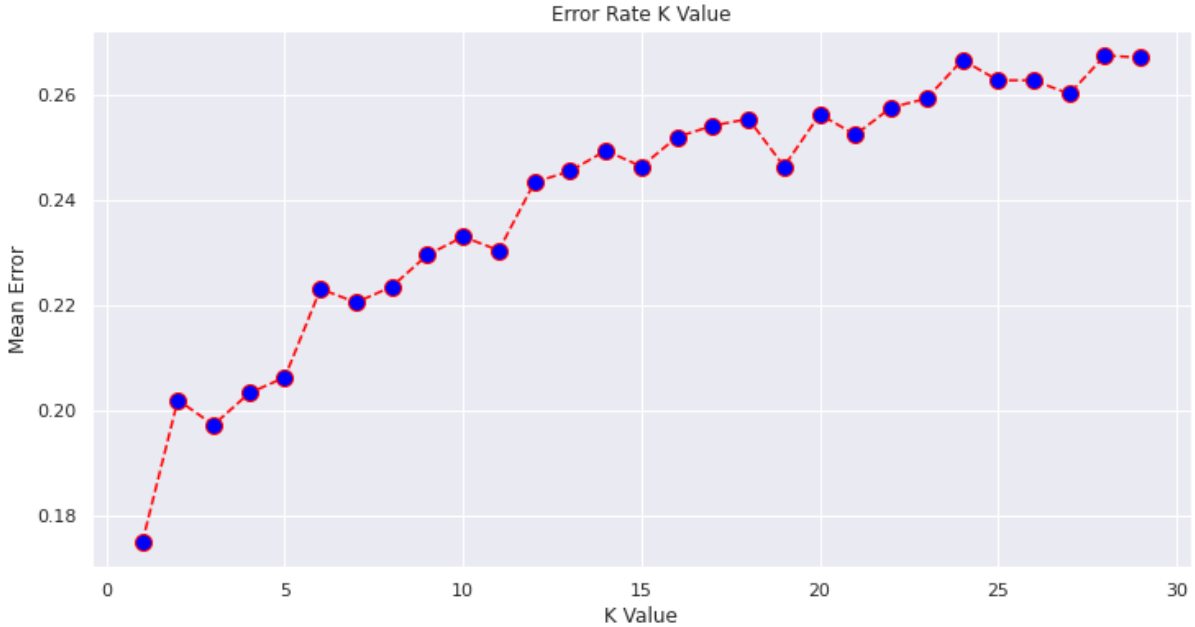
❖ Naive Bayes

➢ לאחר הפעלת אלגוריתם זה, קיבלנו כי ה- Accuracy=0.7962.

➢ אנו מניחים כי הדיוק נמוך יותר מהעץ החלטה כיוון שאין תלות בין הפיצ'רים.

❖ **K-Nearest Neighbors**

- לצורך הפעלת אלגוריתם זה, תחילה נרמלנו את ערכי ה- X_train, ו-X_test (אין צורך בנרמול נתונים משתנה y כיוון שהוא מכיל ערכים בינאריים).
- לאחר הפעלה ראשונית של האלגוריתם, נבחין כי ה- Accuracy=0.7936.
- לצורך שיפור הדיוק, נסתכל על גרף השגיאה ונבין מהו המספר האופטימלי של השכנים.
- כיוון שהסיווג של אלגוריתם זה הינו סיווג בינארי, נבחר מספר אי זוגי של שכנים (K = odd), ובכך נבטיח הכרעה מידית.
- לאחר הרצה, הבחנו כי עבור שכן אחד, נקבל את הAccuracy הטוב ביותר- 0.8252.



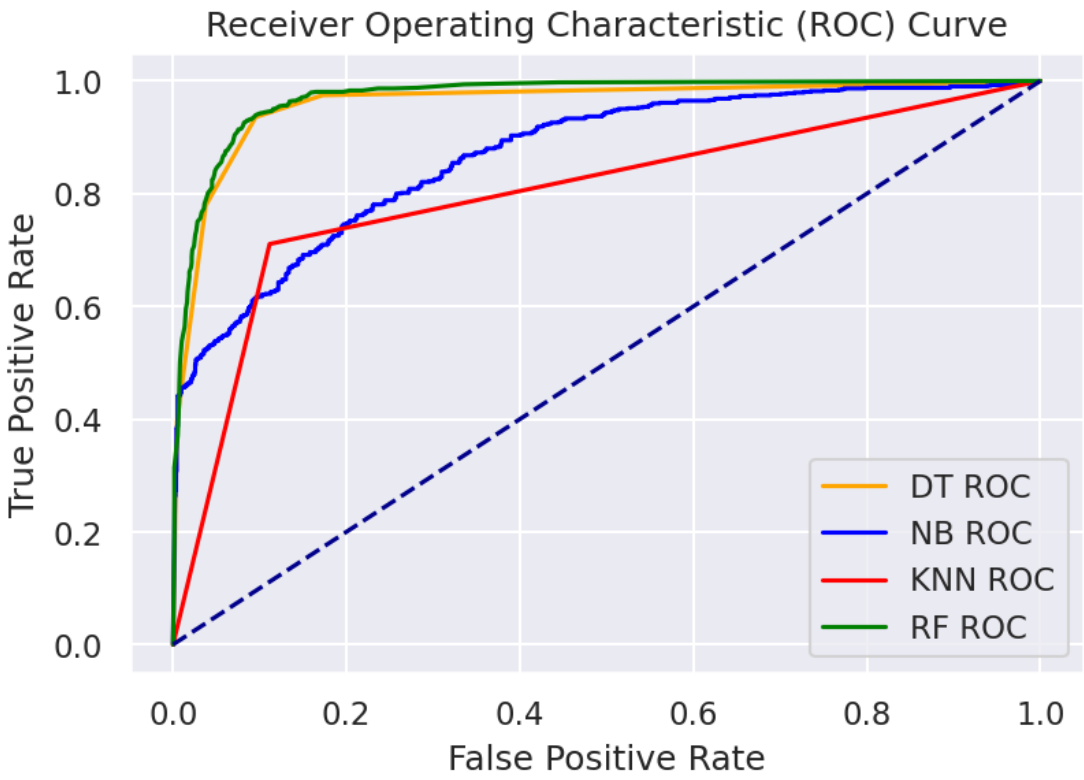
- במקרה שלנו, היתרון בהפעלת אלגוריתם ה-KNN הינו שהוא קל להרצה, לא דורש אימונים ל-Data שלנו, ותמיד נוכל להוסיף מידע בקלות.

❖ **Random Forest**

- לאחר הפעלת אלגוריתם זה, קיבלנו כי ה- Accuracy=0.9188.
- ניתן לראות כי רמת הדיוק באלגוריתם זה הינה גבוהה במעט מרמת הדיוק שקיבלנו באלגוריתם Decision Tree, והדבר הגיוני כיוון שאלגוריתם RF מאחד מספר של עצי החלטה לכדי output סופי.

תוצאות

❖ בשלב זה, ביצענו השוואות בין כלל האלגוריתמים, באמצעות שימוש ב-ROC ו- AUC.



➢ מהסתכלות ראשונית על הגרף, ברור כי האלגוריתמים DT ו-RF הינם העדיפים על שאר האלגוריתמים.

➢ על מנת להחליט איזה מן האלגוריתמים הינו עדיף, ביצענו חישוב לכל אלגוריתם ומצאנו מהו ה-AUC של כל אחד:

	Models	Accuracy	Auc
0	Random Forest	0.918429	0.971917
1	Decision Tree	0.915839	0.960076
2	Naive Bayes	0.796288	0.867510
3	k-Nearest Neighbors	0.825205	0.799660

➢ לכן, ניתן לראות כי האלגוריתם Random Forest הינו הטוב ביותר עם ציון של 97%, כאשר האלגוריתם הבא הינו Decision Tree עם ציון של 96%.

➢ בנוסף, ידוע כי ישנה הנחה שבעת שימוש באלגוריתם Naïve Bayes לא קיימת תלות בין העמודות. במקרה שלנו קיבלנו שהאלגוריתם קיבל ציון נמוך מהעץ החלטה, וכאשר ביצענו מבחן Pearson על העמודות ראינו כי קיימת קורלציה בין חלק מהעמודות, לכן ככל הנראה זוהי הסיבה לכך שהאלגוריתם Naïve Bayes קיבל ציון נמוך יותר.

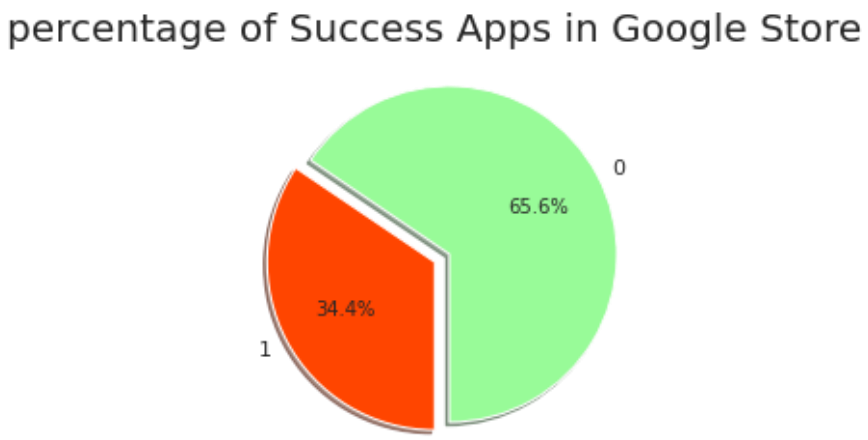
מסקנות

לאחר בחינת כלל האלגוריתמים השונים, והעובדה כי לא קיים הבדל משמעותי בין DT ל- RF, החלטנו להסיק מסקנות מהעץ החלטה אשר הינו ויזואלי. להלן המסקנות:

א. מסקנה ראשית הינה שבעץ החלטה ניתן לראות בבירור כי עמודת "Reviews" הינה משמעותית, כלומר החל מ8,746 ביקורות שאפליקציה מסוימת מקבלת, אזי ככל הנראה היא תהיה "מוצלחת".

ב. מסקנה משנית שניתן להסיק מן העץ החלטה הינה שעמודת "Price" תשפיע על הצלחה החל מהענף השני, ההשפעה באה לידי ביטוי בכך שאם אפליקציה מסוימת תעלה פחות מ-1.24$, אזי ישנה סבירות גבוהה שתהיה "מוצלחת".

בנוסף, לאחר כלל השלבים, קיבלנו מסקנה כוללת כי 34.4% מכלל האפליקציות בחנות של גוגל מוגדרות כיום כ- "מוצלחות".



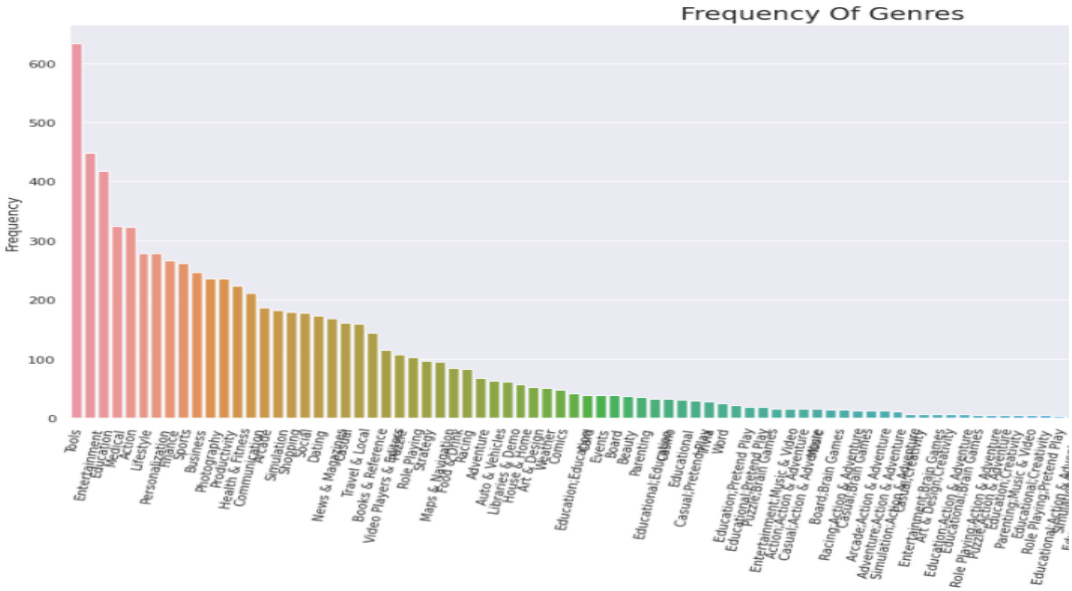
נספחים

❖ נספח א' - מקורות

kaggle

נספח ב' – נתונים כלליים

- בהמשך לממצא הראשוני כי ישנה קורלציה בין "Reviews" לבין "Installs", ניסינו לבדוק האם ישנו קשר גם בין "Reviews" לבין "Rating".
- ביצענו זאת באמצעות בדיקה של האם קיים קשר בין סוג הז'אנר אשר ממנו בוצעו מספר ההורדות הגבוה ביותר לבין האפליקציות שקיבלו הכי הרבה ביקורות.
להלן הז'אנרים השכיחים ביותר:



- בתרשים מעלה, ניתן לראות כי בג'אנר מסוג "Tools" בוצעו הכי הרבה הורדות, אך ניתן לראות בגרף מטה כי דווקא כמות האפליקציות עם כמות הביקורות הגבוהות ביותר הינן מז'אנר "Games", לכן לא הצלחנו למצוא קשר כלשהו אשר יתרום לנו להסיק מסקנות.