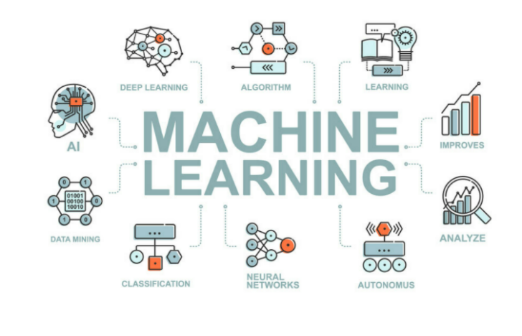
פרויקט גמר – נושאים נבחרים בסטטיסטיקה

חיזוי כמות צפיות של סרטונים ב Youtube על פי מאפייני הסרטון

**מרצה הקורס:** פרופסור ישראל פרמט

**מגישה**

עופרי פוקס– 313528341

**תוכן עניינים**

[תקציר 2](file:///C:\Users\97252\Downloads\פרוייקט%20נושאים%20נבחרים%20בסטטיסטיקה.docx#_Toc61714940)

[מבוא 3](file:///C:\Users\97252\Downloads\פרוייקט%20נושאים%20נבחרים%20בסטטיסטיקה.docx#_Toc61714941)

[מטרת המחקר (השערות + שאלות) 4](file:///C:\Users\97252\Downloads\פרוייקט%20נושאים%20נבחרים%20בסטטיסטיקה.docx#_Toc61714942)

שיטה :  [תיאור והסבר מפורט של המשתנים במחקר הסטטיסטי 4-5](file:///C:\Users\97252\Downloads\פרוייקט%20נושאים%20נבחרים%20בסטטיסטיקה.docx#_Toc61714943)

שיטה : [טיוב הנתונים 5-6](file:///C:\Users\97252\Downloads\פרוייקט%20נושאים%20נבחרים%20בסטטיסטיקה.docx#_Toc61714944)

שיטה : [סטטיסטיקה תיאורית 6-8](file:///C:\Users\97252\Downloads\פרוייקט%20נושאים%20נבחרים%20בסטטיסטיקה.docx#_Toc61714945)

[המודלים בהם השתמשנו : הנחות 9](file:///C:\Users\97252\Downloads\פרוייקט%20נושאים%20נבחרים%20בסטטיסטיקה.docx#_Toc61714946)

המודלים בהם השתמשנו : [לינאריים 10-12](file:///C:\Users\97252\Downloads\פרוייקט%20נושאים%20נבחרים%20בסטטיסטיקה.docx#_Toc61714947)

המודלים בהם השתמשנו : [לא לינאריים](file:///C:\Users\97252\Downloads\פרוייקט%20נושאים%20נבחרים%20בסטטיסטיקה.docx#_Toc61714947) 13-14

המודלים בהם השתמשנו : [תוצאות 15-16](file:///C:\Users\97252\Downloads\פרוייקט%20נושאים%20נבחרים%20בסטטיסטיקה.docx#_Toc61714947)

[דיון ומסקנות 17-18](file:///C:\Users\97252\Downloads\פרוייקט%20נושאים%20נבחרים%20בסטטיסטיקה.docx#_Toc61714948)

[נספחים 19-29](file:///C:\Users\97252\Downloads\פרוייקט%20נושאים%20נבחרים%20בסטטיסטיקה.docx#_Toc61714949)

1. ***תקציר***

בעבודה זו נעשה ניתוח נתונים סטטיסטי לשם בחינת הקשר בין כמות הלייקים, התגובות, הופעת הכתוביות וסיווג הסרטון לכמות הצפיות שסרטון ב Youtube ישיג. המחקר נערך בשנת 2020 ובדק 500 סרטונים העוסקים בעולם ה- Machine Learning. הכלים הסטטיסטיים בהם נעשה שימוש הם: מבחני קורלציה פשוטים, ניתוחי שונות, פקטור אנליסיס ורגרסיה לינארית פשוטה, מרובה ועם משתני dummy. הנתונים נותחו לאור מודל רעיוני המשקף לדעת המחבר את המציאות בצורה הטובה ביותר.

שאלות המחקר נוגעות לציון התופעות המשפיעות ביותר על כמות הצפיות.

מסקנות המחקר מראות כי כמות הצפיות בסרטונים בנושא "לימוד מכונה" מושפעת באופן ישיר מהקטגוריה הספציפית בתחום בו הסרטון עוסק, קיום הכתוביות בסרטון, כמות התגובות, הלייקים והדיסלייקים שיש כבר לסרטון באותו רגע, היום בשבוע בו פורסם הסרטון כמו גם השנה בה הוא פורסם.

1. ***מבוא***

Youtubeהיא חברה אמריקאית המפעילה אתר אינטרנט לאחסון ושיתוף סרטוני וידיאו.

Youtube מהווה חלק מתרבות האינטרנט בחברה המודרנית, אינספור סרטונים נצפים בכל רחבי העולם מדי יום ביומו, וחלקם זוכים לפופולריות רבה ולמספר צפיות רב. החשיפה הרבה לה זוכה האתר חוצה את גבולות האינטרנט, ויוצרת תהודה והשפעה תקשורתיות ענפות. נכון ליוני 2020 Youtube הייתה האתר השני הנצפה ביותר באינטרנט ברחבי העולם על פי דירוג האתרים של אלכסה.[[1]](#footnote-1) סרטונים בעלי פופולריות גבוהה מתוגמלים גם על ידי Youtube וגם על ידי מיליוני המשתמשים שמשתמשים בשירות הסטרימינג ברחבי העולם, ועל כן יש תמריץ ליצור סרטונים טובים ומושכי צופים. ניתן לקבל אינדיקציה למידת הפופולריות הכללית של הסרטון דרך "מדדי מעורבות" ב Youtube כלומר, מדדים של צפיות, לייקים, דיסלייקים ומינויים.

כאמור, אלפי משתמשים מעלים אלפי סרטונים ל Youtube בכל דקה ודקה, ולפלטפורמת ה Youtube אין אפשרות לסווג כל סרטון שעולה בקצב כזה. לכן, פותח אלגוריתם שעושה זאת באופן אוטומטי. האלגוריתם מושפע גם הוא ממאפייני הסרטון, וניתן לומר באופן וודאי שגם מדד הצפיות משפיע על הופעת הסרטון ברשימות החיפוש. [[2]](#footnote-2)

המחקר שלנו עוסק בסרטונים ייחודים של הדרכה/הסבר מעולם התוכן של כל מה שקשור ללמידת המכונה.

למידת מכונה נחשב לתת תחום בתוך נושא הבינה המלאכותית והוא עוסק בפיתוח אלגוריתמים המיועדים לאפשר למחשב ללמוד מתוך דוגמאות שהוא מקבל תוך כדי השימוש בו ובכך לייעל את העבודה ולשפר את חווית השימוש. תחום זה מתפתח מאוד בשנים האחרונות, כיוון שבעולם של היום, למידת מכונה לא שמורה רק לתחמי המחשבים וההייטק, ולמעשה התחום נכנס כמעט לכל דבר בחיים שלנו, כולל בריאות, חינוך ובידור.

בהיותי סטטיסטיקאית לעתיד, בחרתי בנושא הזה דווקא בעקבות ניסיוני הרב עם למידה עצמית דרך סרטונים ב Youtube בעת לקיחת הקורס של "לימוד מכונה". אינספור סרטונים מוצעים לצפייה בחיפוש כל נושא הקשור לתחום, וההיצע רחב מאוד. עם זאת, ישנם סרטונים מפורטים ומקיפים שלא מופיעים בראש רשימת תוצאות החיפוש ויש כאלו שמוצעים בראש הרשימה אך לא מספיק אינפורמטיביים. על כן, נרצה לחקור מה מביא לכמות צפיות רבה יותר, מה שככל הנראה מעלה את הפופולריות הסרטון וקידום הפצתו.

1. **מטרות המחקר:**

**שאלות המחקר:**

כיצד ובאיזה אופן תהיה השפעה למשך הסרטון, כמות לייקים / דיסלייקים, כמות תגובות, אפשרות כתוביות, יום ושנה בו יצא הסרטון - על כמות הצפיות שיהיו לסרטון בתחום Machine Learning.

***השערות המחקר:***

ישפיעו באופן חיובי על כמות הצפיות בסרטון:

* אנחנו מצפים שסרטונים עם כתוביות (התחום הוא תחום נרחב שנלמד בכל רחבי העולם באקדמיה, ולכן כתוביות חיוניות בסרטוני לימוד).
* לא ארוכים מדיי (עד 10 דקות).
* לא קצרים מדיי (מעל 5 דקות).
* כמות לייקים ותגובות רבות – יהיו בעלי נתוני צפיות גבוהים.
* נצפה שבסופ"ש יש מספר רב יותר של צפיות.

ישפיעו באופן שלילי על כמות הצפיות בסרטון:

* נצפה גם שכמות דיסלייקים רבה.
* ארוך מאוד.

1. ***שיטה:***

***נתונים:***

הנתונים מגיעים מאתר KAGGLE. נתונים אלה נאספו ע"י מחקר שנערך בשנת 2020 ובדק 500 סרטונים העוסקים בעולם ה- Machine Learning.

חלק מהמשתנים הינם משתנים הינהרנטיים, אשר נגזרים באופן ישיר מתכונות השיר (כדוגמת אורך השיר, קטגוריה והאיכות ), ואילו חלק אחר של הנתונים הינם מדדים פנימיים של חברת Youtube.

***תיאור והסבר מפורט של המשתנים במחקר הסטטיסטי***

1. viewCount - המשתנה המוסבר במודל. מדד זה מייצג את כמות הצפיות. הערכים במדד זה נעים בין 35 – הסרטון בעל כמות הצפיות הנמוכה ביותר, ל-13,132,984 – הסרטון בעל כמות הצפיות הגבוהה ביותר.
2. Position – זהו ה-ID של הסרטון. נע בין 1 ל500.
3. publishedAt with houer – תאריך ושעה של פרסום הסרטון.
4. publishedAt – תאריך פרסום הסרטון.
5. publishedAt year – השנה בה פורסם הסרטון. נע בין 2008 ל2020.
6. publishedAt month – החודש בו פורסם הסרטון. נע בין 1 ל12.
7. publishedAt day – היום בחודש בו פורסם הסרטון. נע בין 1 ל31.
8. DAY – יום בשבוע בו פורסם הסרטון. נע בין 1 ל7.
9. videoCategoryId – ה-ID של קטוגריית הסרטון. נע בין 1 ל29 אך לא ברצף. ישנם 11 קטגוריות סה"כ.
10. videoCategoryLabel – שם קטוגריית הסרטון. ישנם 11 קטגוריות סה"כ.
11. durationSec – משך זמן הסרטון בשניות. נע בין 24 – הסרטון הקצר ביותר, ל- 3,598 – הסרטון הארוך ביותר.
12. Definition – איכות הסרטון. האיכות היא בHD או בSD.
13. Caption – האם קיימת אופציית כתוביות בסרטון. זהו משתנה בינארי.
14. licensedContent – האם למפרסם רישיון ביוטיוב. 1 כאשר יש רישיון, NULL אחרת.
15. likeCount – כמות הלייקים לסרטון. נע בין 2 – סרטון בעל כמות הלייקים הפחותה ביותר, ל- 337,924 – הסרטון בעל כמות הלייקים הגבוהה ביותר. מאפיין את שביעות רצון הצופים מהסרטון.
16. dislikeCount – כמות הדיס-לייקים לסרטון. נע בין 0 – סרטון בעל כמות הדיס-לייקים המינימלית, ל- 8,457 – הסרטון בעל כמות הדיס-לייקים הגבוהה ביותר. מאפיין את חוסר שביעות רצון הצופים מהסרטון.
17. commentCount – כמות התגובות לסרטון. נע בין 0 – סרטון בעל כמות תגובות מינימלית, ל- 17,771- הסרטון בעל כמות התגובות הגבוהה ביותר. מאפיין את רמת העניין שהסרטון יוצר אצל צופיו.

***טיוב הנתונים:***

* תחילה, על מנת לעבוד עם נתונים רלוונטיים בלבד, בדקנו אם ישנן כפילויות בקובץ הנתונים וניקינו מסט הנתונים שלנו נתונים שאינם רלוונטיים. בדיקה זו העלתה כי אין כפילויות בקובץ הנתונים, והדבר אכן הגיוני לאור קיומו של משתנה ה- position והוא חח"ע.
* הסרנו את משתנה position, אשר אינו רלוונטי ולא מעיד דבר על תוכנו של השיר.
* שמנו לב שבמשתנה licensedContent יש ערך של 1 או NULL ולכן הימרנו את כל ערכי הNULL ל 0 כדי להגדירו כמשתנה דמי.
* בנוסף ראינו שלמשתנים commentCount , dislikeCount , likeCount , viewCount , durationSec ישנם זנבות קיצוניים ועל מנת לנרמל אותם, ביצענו עליהם טרנספורמציה לוגריתמית. לאחר טרנספורמציה זו, המשתנים אכן מתפלגים נורמלית.
* כמו כן הגדרנו את כלל המשתנים הפקטוריאליים למשתנים קטגוריאלים. אלו הם המשתנים: licensedContent, caption, definition, videoCategoryLabel, DAY.

***סטטיסטיקה תיאורית***

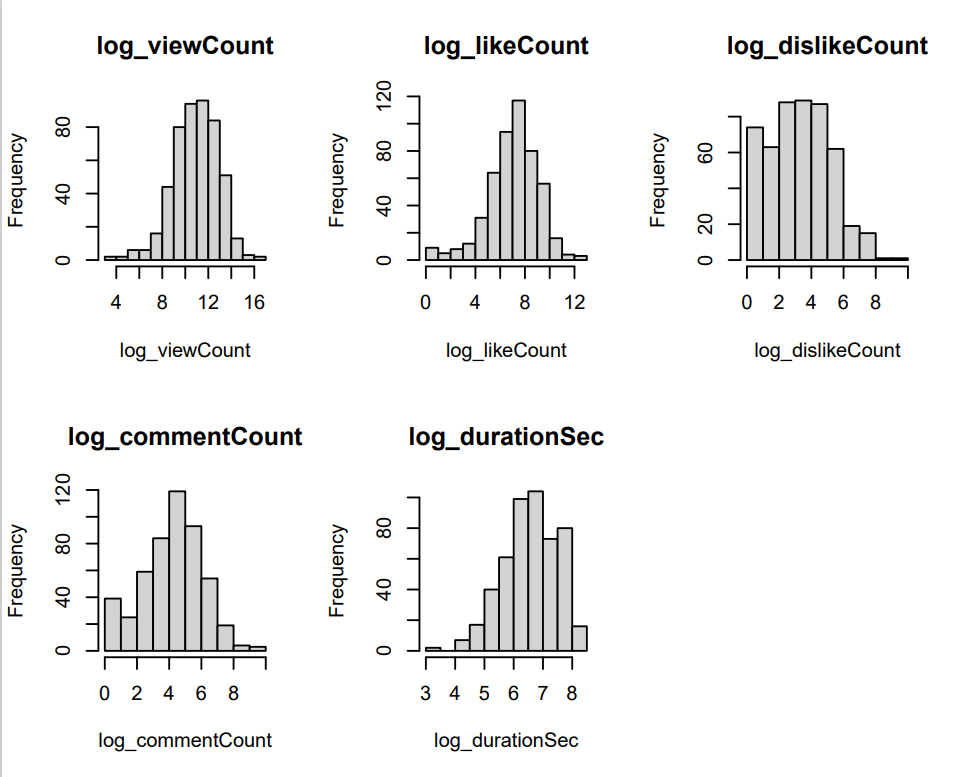
* תחילה, על מנת להבין טוב יותר את הקשרים בין המשתנים, הן בין המסבירים למוסבר והן בין המשתנים המסבירים עצמם, נסתכל על מטריצת המתאמים בין כלל המשתנים הנומריים במודל. כפי שהראינו בחלק הקודם, ישנם מספר משתנים אשר פלטפורמת Youtube בנתה מקומבינציה של משתנים אחרים, ולכן מטריצה זו יכולה לתת לנו עדות למתאם בין משתנים שונים.



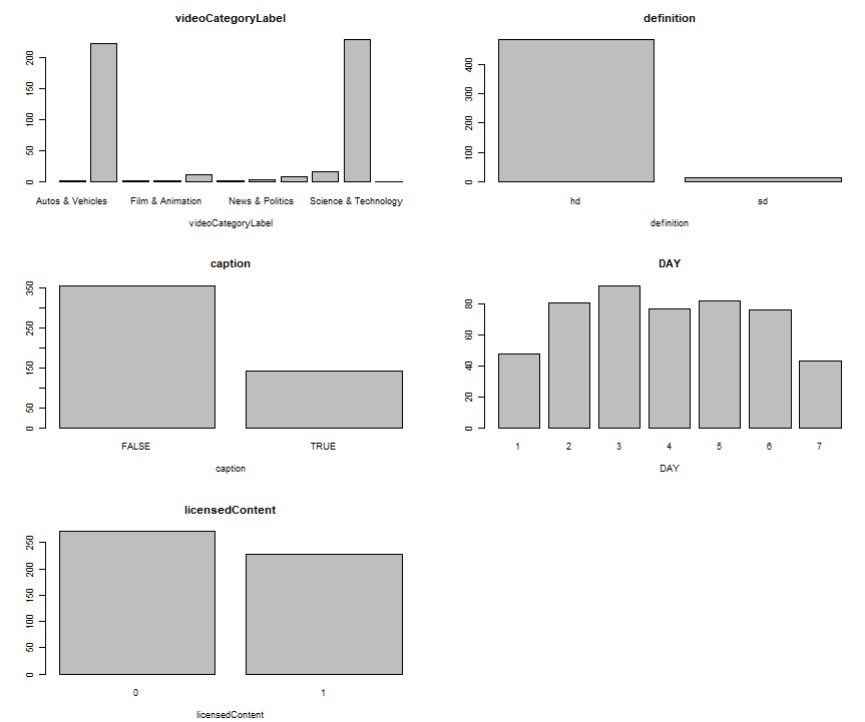
מטבלת הקורלציות המופיעה לעיל ניתן לראות מספר דברים מעניינים המתרחשים בקובץ הנתונים. ראשית, ניתן לראות כי המתאם הגבוה ביותר הוא 0.87 בין המשתנה dislikeCount למשתנה viewCount(המוסבר) ולמשתנה. מתאם זה מאוד הגיוני שהרי כמות הדיס-לייקים מבטאים את רמת חוסר שביעות הרצון של הצופה בסרטון ומשתנה זה מסביר רלוונטי לכמות הצפיות.

בנוסף גם המתאם הגבוה בין dislikeCount ל likeCountשכן שניהם עוסקים ברמת השביעות של השיר.

כמו כן, נראה שיש מתאם מאוד נמוך בין משתנה durationSec לשאר המשתנים.

* לאחר שבחנו את המתאם בין המשתנים השונים, בחנו את ההתפלגות של כל אחד מהמשתנים בנפרד. תחילה בחנו את התפלגות המשתנים הרציפים:

הגרפים של משתנים אלה הוצגו לאחר ניקוי המשתנים הלא רלוונטיים ולאחר ביצוע טרנספורמציות על התפלגויות זנב (על הנתונים הגולמיים, לפני נרמול הנתונים). כעת, ניתן לראות כי מרבית המשתנים אכן מתפלגים נורמלית. (כפי שהסברנו קודם).

* כעת נתאר את התפלגות המשתנים הפקטוריאליים בנתונים:

ניתן לראות שקטגוריות הסרטונים הם בעיקר בנושא פוליטיקה או סוגי רכב.

איכות הסרטונים היא בעיקר בHD.

היום הנפוץ ביותר לפרסום סרטון הוא ביום שלישי.

רוב הסרטונים הם ללא כתוביות.

רוב המפרסמים ללא רישיון Youtube.

***המודלים בהם השתמשנו:***

***הנחות המודל***

במסגרת עבודה זו בחנו מספר מודלים אשר עשויים להתאים לנתונים, שמטרתם היא לחזות בצורה הטובה ביותר את מידת הפופולריות של השיר. מידת הפופולריות הינה משתנה נומרי בדיד שמקבל את הערכים 1-100, ובהתאם לאופי משתנה זה ולאופי המשתנים המסבירים התאמנו את המודלים.

טרם הרצת המודלים, חילקנו את קובץ הנתונים שלנו לסט אימון (train set) וקובץ בדיקה (test set). חלוקה זו התבצעה על פי המקובל בספרות המקצועית, כך ש-80% מהתצפיות הוקצו ל-train ו-20% הוקצו ל-test. בנוסף, עבור כל המודלים שביצענו, בחנו את ה- R-Squared של המודל.

R-Squared גבוה מעיד על מודל מדויק יותר, וקריטריון זה שימש אותנו בסוף העבודה לבחירת המודל המוצלח ביותר.

הנחות המודל, אותן נבדוק ונתקף בהמשך העבודה, הן:

* הנחת הנורמליות.
* הנחת ההומוסקדסטיות.
* אי-תלות בין הטעויות במודל.

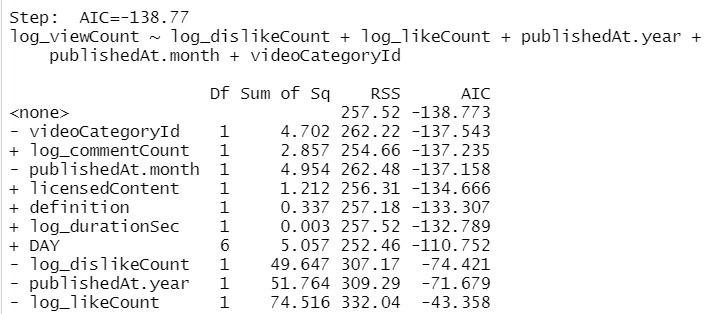
***מודלים לינאריים***

1. **רגרסיה לינארית-** המודל הליניארי הראשון היה מודל רגרסיה ליניארית פשוט. בחרנו במודל זה שכן סביר להניח בתחילת הניתוחים כי ישנו קשר ליניארי בין המשתנים המסבירים שלנו למשתנה המוסבר. על מנת לבחור את המשתנים במודל השתמשנו באלגוריתם ה-stepwise, עם קריטריון AIC כקריטריון החלטה:

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

התקבל המודל הבא:



וכך הסרנו את המשתנים הלא מובהקים- כל מה שמסומן ב +.

והשארנו את המשתנים המובהקים- כל מה שמסומן ב -.

להלן ה- R-Squared שהתקבל מהמודל הסופי:

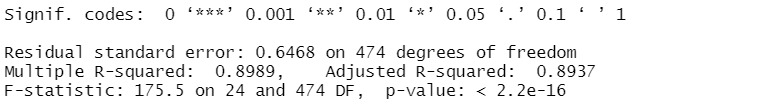


1. **אלגוריתם ה-stepwise** -מודל רגרסיה נוסף שהרצנו בשלב זה הוא בחינה של אינטראקציות מסדר שני בין כל המשתנים (כולל כל משתנה עם עצמו). השתמשנו גם כאן באלגוריתם ה-stepwise על מנת לבחון אילו אינטראקציות אכן רלוונטיות למודל. הסיבה לשימוש במודל זה הייתה חשד שהקשר בין המשתנים המסבירים למשתנה המוסבר אינו ליניארי בהכרח. ניתן לראות בתוצאות כי מדד ה R-Squaredהמתוקנן גדל.

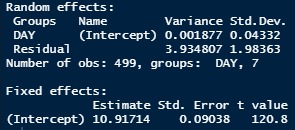
להלן תוצאות המודל:

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

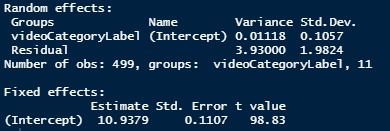
******

******להלן ה- R-Squared שהתקבל מהמודל :

1. **Random Effect****-**העובדה כי חלק מהקטגוריות של המשתנה DAY, המציין את המפתח של הסרטון שבו נכתב השיר, התקבלו כמובהקות, הביאה למסקנה כי יש לבחון גם מודל LMM שבו נכניס Random Effect על משתנה DAY, שכן ייתכן כי נראה התנהגות שונה של הפופולריות כתלות במפתח המוזיקלי של השיר. על כן, הורץ מודל LMM עם חותך בלבד, והתקבלו התוצאות הבאות:

ניתן לראות כי אין הצדקה להכנסת Random Effect על משתנה זה, ולכן פסלנו את השימוש במודל זה.

כנ"ל לגבי משתנה VIDEOCATEGORYLABEL :



1. **Ridge Regression** -כפי שצוין בחלק של הסטטיסטיקה התיאורית, ישנו חשד למולטיקוליניריות חזקה בין המשתנים המסבירים. אנו יודעים כי Ridge Regression מתמודדת בצורה טובה עם נתונים הסובלים ממולטיקוליניריות, ולכן מודל נוסף שבחנו הוא מודל של Ridge Regression. ה-lambda עבור הרגרסיה נבחרה כ-lambda הכי מפוקחת שנמצאת במרחק מינימלי מה-lambda שממזערת את סכום ריבועי הסטיות***.***

גודל ה- lambda שיצא עבור מודל זה הוא:

להלן ה- R-Squared שהתקבל מהמודל:

******

1. **Lasso Regression** -בדקנו גם רגרסיית Lasso אשר דומה במטרתה לרגרסיית ridge בהתמודדות עם נתונים הסובלים מולטיקולינריות אך נותנת קנסות שונים עבור המקדמים של הרגרסיה.

גודל ה- lambda שיצא עבור מודל זה הוא:

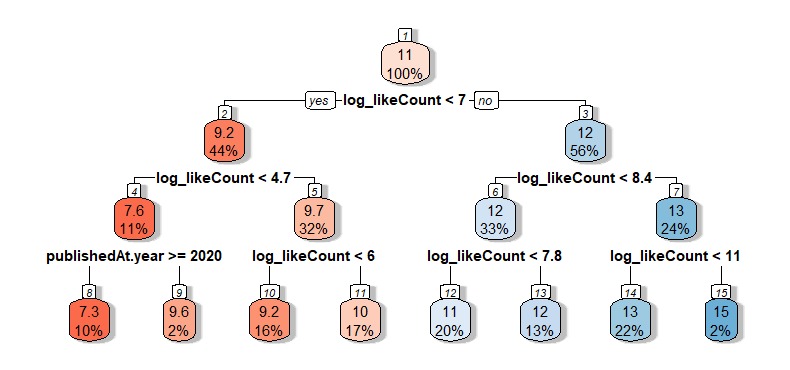
להלן ה- R-Squared שהתקבל מהמודל:



***מודלים לא לינאריים***

1. **Decision Tree –** ההצדקה לשימוש במודל זה נובעת מהדומיין אליו שייכים הנתונים שלנו – סרטוני יוטיוב. סגנונות של סרטונים מסווגים על פי ניואנסים הנגזרים מהמאפיינים של הסרטון, כגון אלה המופיעים בקובץ הנתונים שלנו– כמות הלייקים, כמות הדיסלייקים, שנת הוצאה, חודש הוצאה, כמות תגובות והקטגוריה של הסרטון. סיווג זה נעשה באופן דומה לדרך בה מתנהג האלגוריתם של מודל Decision Tree: האלגוריתם למעשה בוחר בכל צעד איזה משתנה לבחון, וחוזה את מדד הפופולריות על פי ערכים של משתנה זה. לכן, בחיפושינו אחר השיר "האולטימטיבי", ייתכן כי חלוקה זו תאפשר לנו למצוא את שילוב המאפיינים האופטימלי. לאור כך, בחנו מודל כזה על הנתונים שלנו.

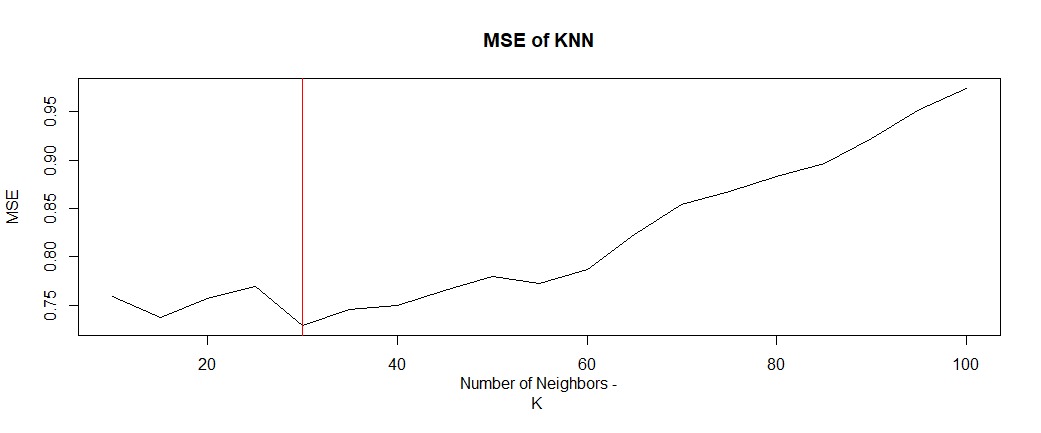
 להלן ה- R-Squared עבור המודל:

******להלן שרטוט העץ שהתקבל:

1. **מודל K-Nearest-Neighbor (KNN) –** מודל זה חוזה את ערך הפופולריות של סרטון מסוים על פי ערך הפופולריות שניתן ל-K הסרטונים שמאפייניהם הם הדומים ביותר לסרטון אותו מנסים לחזות.

הרצנו את מודל ה-KNN על הנתונים, כאשר ביצענו תהליך איטרטיבי למציאת כמות השכנים האופטימלית (פרמטר ה-K), בטווח הערכים מ10 עד 100.

בגרף הבא ניתן לראות שקיבלנו את ה- R-Squared הגבוה ביותר עבור K=30:

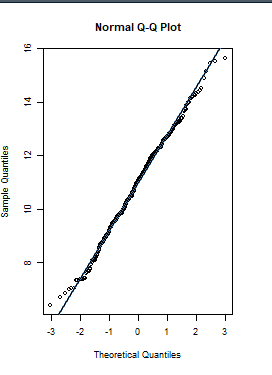


ה- R-Squared של המודל:

******

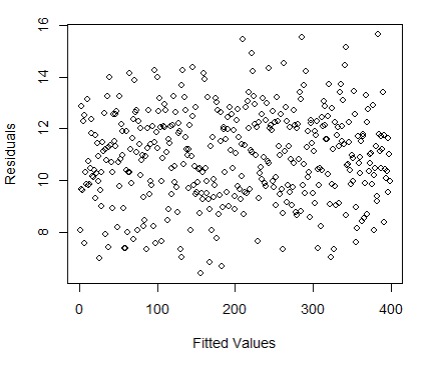
1. ***תוצאות***

נראה כי מבין כל המודלים הלינאריים שבדקנו – מודל ה Ridge Regression הוא בעל ה- R-Squared הגבוה ביותר.

***הנחות המודל הנבחר:***

1. הנחת הנורמליות – הטעות במדגם מתפלגת נורמלית. לשם בדיקת הנחה זו, שרטטנו QQ-Plot, אשר מציג את האחוזונים התיאורטיים של התפלגות נורמלית אל מול האחוזונים בפועל של המדגם:

כפי שניתן לראות בגרף זה, ההתפלגות במדגם מעט מצודדת יותר מההתפלגות הנורמלית, כאשר הדבר בא לידי ביטוי באחוזונים הנמוכים של ההתפלגות ובאחוזונים הגבוהים של ההתפלגות. עם זאת, ניתן לומר כי מלבד כך אין צורה מיוחדת להתפלגות הטעות, ואכן ניתן להניח כי הטעות במדגם מתפלגת נורמלית.



2. הנחת ההומוסקדסטיות – ההנחה לפיה שונות הטעות במודל בלתי תלויה בתצפיות, הנחת שוויון השונויות. על מנת לבדוק הנחה זו, השתמשנו בגרף של הערכים החזויים אל מול השאריות***.***

לא ניתן לראות מגמה ברורה של צורת השונות כתלות בערכים החזויים ולכן לא נניח כי הנחת שוויון השונויות מופרת.

3. אי-תלות בין הטעויות במודל – לאור העובדה כי הנתונים שלנו מורכבים מכ-500 סרטונים אשר נדגמו באופן אקראי ובלתי תלוי מתוך אלפי סרטוני לימוד מכונה הנמצאים ב- Youtube, מתוך כל הקטגוריות השונות של הסרטונים ניתן להניח כי אין תלות בין התצפיות במודל.

1. ***דיון ומסקנות***

קובץ הנתונים הראשוני, הכיל 500 סרטונים אשר עליהם נערך המחקר. ממוצע הצפיות של הסרטונים הינו 291,227.324, עם סטיית תקן של 940,434. על מנת להעריך את טיב המודלים השונים, 80% מהנתונים שימשו לבניית המודלים, ו-20% שימשו לבחינת המודלים.

נבחנו 5 מודלים של לינאריים, מודל של עץ החלטה ומודל KNN. המודל הסופי נבחר על פי קריטריון ה- R-Squared (כפי שמפורט בחלק "שיטות והרצת המודלים"). מודל זה הינו מודל של Ridge Regression.

על פי מדד ה-R Squared, מודל זה מסביר 83.8% מהשונות של צפיות הסרטון.

ערכי המקדמים של המודל הסופי הם:

על פי התוצאות, ניתן לראות כי כל המשתנים אכן מובהקים, אולם המקדמים של חלק גדול מהמשתנים הינם נמוכים ולא משפיעים בצורה משמעותית על הצפיות.

נקיים דיון על המשתנים בעלי ההשפעה הגדולה על הצפיות לפי השערות המחקר שלנו:

1. יום בשבוע – נראה כי השערת המחקר שללנו התבררה כנכונה, ואכן יום שישי (סופ"ש) הוא המשפיע ביותר על כמות הצפיות.
2. זמן סרטון – נראה כי השערת המחקר שלנו התבררה כשגויה ואורך הסרטון פחות משפיע ממה שציפינו.
3. כתוביות – נראה כי השערת המחקר נכונה ואכן קיום כתוביות בסרטון משפיע חיובי על כמות הצפיות בסרטון לעומת אי קיומן.
4. לייקים ותגובות – נראה כי השערת המחקר שלנו התבררה כנכונה ואכן כמות הלייקים והתגובות משפיעות באופן גדול וחיובי על כמות הצפיות
5. דיס-לייקים – נראה כי טעינו בהשערתנו וכמות הדיס לייקים דווקא משפיע באופן חיובי וגדול על כמות הצפיות לעומת הצפיות שכמות גבוהה של דיס-לייקים תשפיע באופן שלילי על כמות הצפיות – כנראה שכמות גבוהה של דילייקים יוצרת עניין וסקרנות אצל הצופים וגומרת להם לצפות בסרטון.

נראה עוד כי:

1. השנה בה פורסם הסרטון משפיע יותר מהחודש בו הוא פורסם. ככל שהשנה יותר מאוחרת הצפיות יותר נמוכות ( הסרטון פחות זמן באוויר).
2. נראה כי קטגוריית הסרטון המשפיעה ביותר באופן חיובי היא כלי רכב.
3. נראה כי איכות הסרטון לא משפיעה גדולה על כמות הצפיות.
4. נראה כי רישיון המפרסם גם הוא לא משפיע באופן גדול על כמות הצפיות.

לסיכום, על אף מגבלות המחקר המתבסס על מדגם קטן וחקר של כוונות ולא התנהגות בפועל, ממצאיו עדיין מעידים על הגורמים לצפייה מרובה בסרטון.

נראה כי כמות הצפיות בסרטונים בנושא "לימוד מכונה" מושפעת באופן ישיר מהקטגוריה הספציפית בתחום בו הסרטון עוסק, קיום הכתוביות בסרטון, כמות התגובות, הלייקים והדיסלייקים שיש כבר לסרטון באותו רגע, היום בשבוע בו פורסם הסרטון כמו גם השנה בה הוא פורסם.

1. **נספחים**

**המודל הנבחר**

**תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי**

**R Script**

**library(magrittr)**

**library(data.table)**

**library(dplyr)**

**library(tidyverse)**

**##install.packages("psych")**

**library(psych)**

**library(ggplot2)**

**library(purrr)**

**library(tibble)**

**library(tidyr)**

**library(ggplot2)**

**library(Hmisc)**

**raw\_data <-read.csv("C:/Users/ofrifox/Documents/לימודים/שנה ג/סמסטר א/נושאים נבחרים/פרוייקט/Machine Learning YouTube Meta Data.csv")**

**#Check for duplicates:**

**duplicated(raw\_data) %>% any()#All the songs ids are unique in each sample.**

**#Removing some variables:**

**clean\_data <- subset(raw\_data, select = -c(position)) ## remove the id**

**## na to 0**

**clean\_data[is.na(clean\_data)] <- 0**

**#Making Duration in log transformation :**

**clean\_data$durationSec <- log(clean\_data$durationSec)**

**clean\_data <- clean\_data %>% rename(log\_durationSec=durationSec)**

**hist(clean\_data[,'log\_durationSec'],xlab='log\_durationSec' , main = 'log\_durationSec')**

**clean\_data$likeCount <- log(clean\_data$likeCount)**

**clean\_data <- clean\_data %>% rename(log\_likeCount=likeCount)**

**hist(clean\_data[,'log\_likeCount'],xlab='log\_likeCount' , main ='log\_likeCount')**

**clean\_data$dislikeCount <- log(clean\_data$dislikeCount)**

**clean\_data <- clean\_data %>% rename(log\_dislikeCount=dislikeCount)**

**hist(clean\_data[,'log\_dislikeCount'],xlab='log\_dislikeCount' , main ='log\_dislikeCount')**

**clean\_data$commentCount <- log(clean\_data$commentCount)**

**clean\_data <- clean\_data %>% rename(log\_commentCount=commentCount)**

**hist(clean\_data[,'log\_commentCount'],xlab='log\_commentCount' , main ='log\_commentCount')**

**clean\_data$viewCount <- log(clean\_data$viewCount)**

**clean\_data <- clean\_data %>% rename(log\_viewCount=viewCount)**

**hist(clean\_data[,'log\_viewCount'],xlab='log\_viewCount' , main ='log\_viewCount')**

**##numeric**

**numerical\_vars<-c('log\_viewCount','log\_likeCount','log\_dislikeCount','log\_commentCount','log\_durationSec')**

**## make "-nf" to 0**

**clean\_data[clean\_data$log\_likeCount<0,"log\_likeCount"]<-0**

**clean\_data[clean\_data$log\_dislikeCount<0,"log\_dislikeCount"]<-0**

**clean\_data[clean\_data$log\_commentCount<0,"log\_commentCount"]<-0**

**## corr in the numeric**

**cors <- function(df) {**

**M <- Hmisc::rcorr(as.matrix(df))**

**Mdf <- map(M, ~data.frame(.x))**

**}**

**formatted\_cors <- function(df){**

**cors(df) %>%**

**map(~rownames\_to\_column(.x, var="measure1")) %>%**

**map(~pivot\_longer(.x, -measure1, "measure2")) %>%**

**bind\_rows(.id = "id") %>%**

**pivot\_wider(names\_from = id, values\_from = value)%>%**

**mutate(sig\_p = T, p\_if\_sig = P, r\_if\_sig = r)**

**}**

**formatted\_cors(clean\_data[,numerical\_vars]) %>%**

**ggplot(aes(measure1, measure2, fill=r, label=round(r\_if\_sig,2))) +**

**geom\_tile() +**

**labs(x = NULL , y = NULL, fill = "Pearson's\nCorrelation", title="Correlations in the data") +**

**scale\_fill\_gradient2(mid="#FBFEF9",low="#0C6291",high="#A63446", limits=c(-1,1)) +**

**geom\_text() + theme(axis.text.x = element\_text(angle = 90)) +**

**scale\_x\_discrete(expand=c(0,0)) +**

**scale\_y\_discrete(expand=c(0,0))**

**#Describing each factorial variable distribution :**

**factorial\_vars <- c("videoCategoryLabel","definition","caption","DAY","licensedContent")**

**par(mfrow = c(3,2))**

**for ( i in factorial\_vars){**

**plot(as.factor(clean\_data[,i]),xlab=i , main = i)**

**# print(class(as.numeric(clean\_data[,i])))**

**}**

**## block**

**clean\_data$DAY <- as.factor(clean\_data$DAY)**

**clean\_data$videoCategoryLabel <- as.factor(clean\_data$videoCategoryLabel)**

**av<-lm(log\_viewCount ~ DAY + videoCategoryLabel +log\_likeCount + log\_dislikeCount + log\_commentCount**

**+ log\_durationSec + publishedAt.year + publishedAt.month+**

**definition + licensedContent,data=clean\_data)**

**summary(av)**

**anova(av)**

**## make factor**

**clean\_data$DAY <- as.factor(clean\_data$DAY)**

**clean\_data$videoCategoryLabel <- as.factor(clean\_data$videoCategoryLabel)**

**clean\_data$definition <- as.factor(clean\_data$definition)**

**clean\_data$licensedContent <- as.factor(clean\_data$licensedContent)**

**clean\_data$caption<- as.factor(clean\_data$caption)**

**# Describing each continues variable distribution :**

**par(mfrow = c(2,3))**

**for ( i in numerical\_vars){**

**hist(clean\_data[,i],xlab=i , main = i) }**

**summary(clean\_data)**

**## make a copy**

**data <- subset(clean\_data, select = -c(position))**

**########################### model########################**

**library(MASS)**

**library(caret)**

**library(tidyverse)**

**library(lme4)**

**library(olsrr)**

**library(plyr)**

**library(readr)**

**library(dplyr)**

**library(ggplot2)**

**library(repr)**

**glimpse(clean\_data)**

**library(glmnet)**

**#install.packages("repr")**

**#install.packages("glmnet")**

**summary(clean\_data)**

**####Train\Test Set**

**set.seed(100)**

**index = sample(1:nrow(clean\_data), 0.8\*nrow(clean\_data))**

**train = clean\_data[index,] # Create the training data**

**test = clean\_data[-index,] # Create the test data**

**dim(train)**

**dim(test)**

**cols\_reg = c('publishedAt.year', 'publishedAt.month', 'videoCategoryLabel'**

**, 'DAY', 'log\_durationSec', 'definition', 'caption',**

**'licensedContent', 'log\_likeCount', 'log\_dislikeCount',**

**'log\_commentCount', 'log\_viewCount')**

**dummies <- dummyVars(log\_viewCount ~ ., data = clean\_data[,cols\_reg])**

**train\_dummies = predict(dummies, newdata = train[,cols\_reg])**

**test\_dummies = predict(dummies, newdata = test[,cols\_reg])**

**print(dim(train\_dummies)); print(dim(test\_dummies))**

**x = as.matrix(train\_dummies)**

**y\_train = train$log\_viewCount**

**x\_test = as.matrix(test\_dummies)**

**y\_test = test$log\_viewCount**

**###EVAL RESULTE**

**# Compute R^2 from true and predicted values**

**eval\_results <- function(true, predicted, df) {**

**SSE <- sum((predicted - true)^2)**

**SST <- sum((true - mean(true))^2)**

**R\_square <- 1 - SSE / SST**

**RMSE = sqrt(SSE/nrow(df))**

**# Model performance metrics**

**data.frame(**

**RMSE = RMSE,**

**Rsquare = R\_square**

**)**

**}**

**#############################################Linear Model###############################**

**model = lm (log\_viewCount ~ log\_likeCount + log\_dislikeCount + log\_commentCount**

**+ log\_durationSec + publishedAt.year + publishedAt.month + DAY +**

**videoCategoryId + definition + licensedContent, data = clean\_data)**

**model2 = lm (log\_viewCount ~ 1, data= train)**

**scope <- list(upper=model , lower=model2)**

**step\_reg <- step(model2 , scope = scope , direction = "both" ,**

**trace=TRUE,k=log(nrow(train)))**

**k <- ols\_step\_all\_possible(model)**

**#library(olsrr)**

**f <- ols\_step\_forward\_p(model, penter = 0.05)**

**b <- ols\_step\_backward\_p(model, prem = 0.05)**

**both <- ols\_step\_both\_p(model, penter = 0.05, prem = 0.05)**

**predictions\_test = predict(step\_reg, newdata = test)**

**eval\_results(test$log\_viewCount, predictions\_test, test)**

**############################################polynomial model**

**model\_pol <- lm (log\_viewCount ~ log\_likeCount +I(log\_likeCount^2)+ I(log\_durationSec^2) + publishedAt.year**

**+ log\_likeCount:log\_dislikeCount + log\_likeCount:log\_commentCount**

**+ log\_likeCount:publishedAt.year + log\_likeCount:publishedAt.month**

**+ log\_likeCount:licensedContent +log\_commentCount:publishedAt.year**

**+ log\_commentCount:videoCategoryId + log\_durationSec:DAY**

**+ DAY:videoCategoryId ,data = train)**

**model2.1 = lm (log\_viewCount ~ 1, data= train)**

**scope\_pol <- list(upper=model\_pol , lower=model2.1)**

**step\_reg\_pol <- step(model2.1 , scope = scope\_pol , direction = "both" ,**

**trace=TRUE,k=log(nrow(train)))**

**step\_predict\_pol <- predict(step\_reg\_pol, newdata=test)**

**summary(model\_pol)**

**# predicting and evaluating the model on test data**

**predictions\_test\_pol = predict(step\_reg\_pol, newdata = test)**

**eval\_results(test$log\_viewCount, predictions\_test\_pol, test)**

**################################ random effect**

**library(glmnet)**

**library(olsrr)**

**model = lm(log\_viewCount ~ log\_likeCount + log\_dislikeCount + log\_commentCount**

**+ publishedAt.year + publishedAt.month +videoCategoryLabel+ DAY**

**, data = data)**

**## day**

**library(lme4)**

**library(glmnet)**

**library(lmerTest)**

**lme.reg.day <-lme4::lmer(log\_viewCount ~ (1|DAY) + (1|videoCategoryLabel), data =clean\_data)**

**summary(lme.reg.day)**

**anova(lme.reg.day,model)**

**##videoCategoryLabel**

**lme.reg.videoCategoryLabel <-lme4::lmer(log\_viewCount ~ (1|videoCategoryLabel),data =clean\_data )**

**summary(lme.reg.videoCategoryLabel)**

**anova(lme.reg.videoCategoryLabel,model)**

**##videoCategoryLabel & day**

**lme.reg.videoCategoryLabel.day <- lmer(log\_viewCount ~ log\_likeCount + log\_dislikeCount + log\_commentCount**

**+ log\_durationSec + publishedAt.year + publishedAt.month+ (1|DAY)+**

**definition + licensedContent+(1|videoCategoryLabel),data = clean\_data )**

**summary(lme.reg.videoCategoryLabel.day)**

**###########################################RIDGE**

**lambdas <- 10^seq(2, -3, by = -.1)**

**# Setting alpha = 1 implements lasso regression**

**ridge\_reg = glmnet(x, y\_train, nlambda = 25, alpha = 0, family = 'gaussian', lambda = lambdas)**

**cv\_ridge <- cv.glmnet(x, y\_train, alpha = 0, lambda = lambdas)**

**optimal\_lambda <- cv\_ridge$lambda.min**

**optimal\_lambda**

**# Prediction and evaluation on train data**

**predictions\_train\_rig <- predict(ridge\_reg, s = optimal\_lambda, newx = x)**

**eval\_results(y\_train, predictions\_train\_rig, train)**

**# Prediction and evaluation on test data**

**predictions\_test\_rig <- predict(ridge\_reg, s = optimal\_lambda, newx = x\_test)**

**eval\_results(y\_test, predictions\_test\_rig, test)**

**print(eval\_results(y\_test, predictions\_test\_rig, test))**

**###########################################LASSO**

**lambdas <- 10^seq(2, -3, by = -.1)**

**# Setting alpha = 1 implements lasso regression**

**lasso\_reg <- cv.glmnet(x, y\_train, alpha = 1, lambda = lambdas, standardize = TRUE, nfolds = 5)**

**# Best**

**lambda\_best\_lasso <- lasso\_reg$lambda.min**

**lambda\_best\_lasso**

**lasso\_model <- glmnet(x, y\_train, alpha = 1, lambda = lambda\_best\_lasso, standardize = TRUE)**

**predictions\_train\_lasso <- predict(lasso\_model, s = lambda\_best\_lasso, newx = x)**

**eval\_results(y\_train, predictions\_train\_lasso, train)**

**predictions\_test\_lasso <- predict(lasso\_model, s = lambda\_best\_lasso, newx = x\_test)**

**eval\_results(y\_test, predictions\_test\_lasso, test)**

**print(eval\_results(y\_test, predictions\_test\_lasso, test))**

**############################################# Not Linear Model###############################**

**#############################################DECISION TREE**

**##install.packages("rpart.plot")**

**##install.packages("tree")**

**library(rpart)**

**library(rpart.plot)**

**library(caret)**

**library(tree)**

**fit <- rpart(log\_viewCount~ log\_commentCount + videoCategoryId + log\_likeCount**

**+ publishedAt.year + publishedAt.month,**

**method="anova", data=train)**

**rpart.plot(fit, box.palette="RdBu", shadow.col="gray", nn=TRUE)**

**# R-SQUARE**

**predictions\_test\_tree = predict(fit, newdata = test)**

**eval\_results(test$log\_viewCount, predictions\_test\_tree, test)**

**###########################################KNN**

**#install.packages('FNN')**

**library(FNN)**

**library(class)**

**Mse\_Vector <- vector ()**

**for ( i in seq(10,100,5) ){**

**knn.1 <- FNN::knn.reg(train=x, test= x\_test , y = train$log\_viewCount**

**, k=i )**

**knn.mse <- MSE(knn.1$pred- (test$log\_viewCount))**

**Mse\_Vector <- append (Mse\_Vector, knn.mse)**

**}**

**{plot(seq(10,100,5),Mse\_Vector, type='l', xlab = "Number of Neighbors -**

**K " , main = "MSE of KNN" ,ylab="MSE")**

**abline(v=30, col = "red")}**

**MSE <- function(x) x^2 %>% mean**

**knn.2 <- FNN::knn.reg(train=x, test= x\_test , y = train$log\_viewCount, k=30)**

**knn.mse <- MSE(knn.2$pred-(test$log\_viewCount))**

**r2.knn <- 1 - knn.mse/var(train$log\_viewCount)**

**r2.knn**

**cat("The R-Square Of the KNN is: ", r2.knn)**

**###################################################best model print**

**library(glmnet)**

**ridge\_reg\_optimal = glmnet(x, y\_train, nlambda = 25, alpha = 0, family = 'gaussian', lambda = 0.3162278)**

**coef(ridge\_reg\_optimal)**

**pval(ridge\_reg\_optimal)**

**##Validate our assumption in the step-wise regression:**

**library(car)**

**par(mfrow=c(1,2))**

**plot(predict(ridge\_reg, s = optimal\_lambda, newx = x),resid(ridge\_reg, s = optimal\_lambda, newx = x),ylab="Residuals",xlab="Fitted Values" )**

**abline(0,0)**

**qqnorm(predict(ridge\_reg, s = optimal\_lambda, newx = x),resid(ridge\_reg, s = optimal\_lambda, newx = x))**

**qqline(predict(ridge\_reg, s = optimal\_lambda, newx = x),col = "steelblue")**

1. <https://he.wikipedia.org/wiki/%D7%99%D7%95%D7%98%D7%99%D7%95%D7%91> [↑](#footnote-ref-1)
2. https://untamedscience.com/blog/does-the-number-of-subscribers-views-likes-under-science-videos-on-youtube-affect-the-behavior-of-the-average-user-these-days/ [↑](#footnote-ref-2)