שיבוץ שלושה שחקנים ראשיים לסרט תוך התחשבות בתקציבו במטרה למקסם רווחים

פרויקט בבינה מלאכותית

מגישות:
חן חבקוק ת"ז 312320997
chenha@campus.technion.ac.il
טל עובד ת"ז 206129710
taloved@campus.technion.ac.il

מנחה:
גיא קושילביץ

הקדמה

רקע

תעשיית הסרטים

קולנוע הוא תחום אמנות העוסק ביצירתם ובהקרנתם של סרטי קולנוע .
הקולנוע הוא מערכת תרבותית, במסגרתה נוצרים מאות ואלפי תוצרי תרבות הנצפים על ידי מיליונים ברחבי העולם. הקולנוע הוא בין השאר כלי תקשורת המונים , ובעל השפעות גדולות על החברה והתרבות.
תעשיית הקולנוע והסרטים היא המצרך האמנותי הפופולרי ביותר.

המילה "קולנוע" היא הלחם של המילים קול ותנועה כאשר הדגש הוא על המילה הקול.
מקורו בהתפתחות הטכנולוגית - הקולנוע בא לאחר הראינוע - הסרט האילם.

בזמן הראינוע ,הבדלי שפה שבין ארץ לאחרת לא היוו מגבלה מכיוון שהסרטים היו אילמים. מדי פעם, כשנוצר צורך, היו מקרינים במקביל לסרט כותרות (במעין שלט נלווה) וכשרצו להקרין סרט בארץ אחרת, החליפו את הכותרות לשפה המדוברת באותה ארץ.

הפקת סרטים היא יקרה מאד לכן ההשקעות הגדולות מתבצעות לרוב על ידי אנשים בעלי ניסיון.
הסרטים מדורגים לרמות שונות, כשרמות גבוהות יותר פירושן השקעה גדולה יותר ויקרה יותר.

הקולנוע ההוליוודי

הקולנוע ההוליוודי החל את דרכו יחד עם הקולנוע עצמו.
הוליווד עצמה הוקמה בתחילת שנות העשרים, כשיוצרי הראינוע נדדו מערבה. שם רכשה קבוצה מהם אדמה מבעליה, ששמם היה הוליווד, והחליטו להשאיר את השם. הוליווד הייתה במקום מושלם למגוון של לוקיישנים שונים. החל מהנוף ההררי מצפון, הים במערב, ההרים והמדבר במזרח, ועמקי קליפורניה הפוריים.

הקולנוע זכה לדחיפה גדולה בשל יצירתו של הקולנוען האמריקני ד.וו.גריפית שהתחיל את הקריירה הקולנועית שלו בתור שחקן, אך מהר מאוד התחיל גם לצלם ותוך זמן קצר גם קיבל הזדמנות לביים. בשנת 1915 ביים את סרטו הראשון "לידתה של אומה", סרט שיצר למעשה את שפת הקולנוע כפי שאנו מכירים אותה היום כמו: עריכה בתוך הסצנה, שימוש בקלוז-אפ וקצב עריכה שונה בהתאם לדרמטיות שבסצנה.
סרטו השני, "אי סובלנות" כלל סצנות המונים ותפאורות בגובה עשרות מטרים, מצלמות שהועלו עם מעליות והמצאות רבות שהפכו את הסרט, שאורכו 163 דקות, לתצוגת איכות קולנועית כבר ב-1916.

קולנוען נוסף וידוע שהביא את הראינוע לפסגות חדשות היה צ'ארלי צ'פלין ,אשר נחשב לאחת מהדמויות החשובות והמשפיעות ביותר בהיסטוריה של תעשיית הקולנוע .סרטיו של צ'פלין הביאו לעולם קולנוע חדש שלא נראה כמוהו, כוריאוגרפיה ייחודית ושימוש מבריק באביזרים ותפאורה.

תור הזהב של הוליווד

הוליווד ידועה כמרכז הפקות הסרטים העולמי. האולפן הראשון בהוליווד הוקם ב1911.
בשנות השלושים ובשנות הארבעים, במהלך התקופה אשר זכתה לכינוי "עידן הזהב של הוליווד", הפיקו אולפני הסרטים בהוליווד בעיקר סרטי מערבונים ,סרטי סלפסטיק (קומדיה משולבת בתנועות גופניות מכאיבות ומרעישות), סרטי "פילם נואר" (סרטי פשע, בעלי אווירה אפלה), סרטים מוזיקליים (מחזמר), סרטי אנימציה מצוירים וסרטים ביוגרפיים. באותה העת אלפי אנשים הועסקו על ידי אולפני הסרטים - שחקנים, מפיקים, במאים, כותבים, פעלולנים, אנשי מלאכה וטכנאים. בבעלותם של אולפני הסרטים היו מאות בתי קולנוע בערים שונות ברחבי ארצות הברית

בהם הוצגו הסרטים שלהם. נקודת השיא של תעשיית הסרטים האמריקאית הייתה במהלך אמצע שנות הארבעים כאשר אולפני הקולנוע הפיקו באותה העת כ-400 סרטים בשנה אשר הוצגו בפני קהל של 90 מיליון צופים אמריקאים מדי שבוע. לאורך השנים, התעשייה התפתחה, צברה ניסיון ושידרגה את היכולות הטכנולוגיות, המקצועיות והאומנותיות.

רקע מהקורס

ראינו בקורס כי תהליך החיזוי הוא תוצאה של תהליכי למידה שונים.
למידה הינה תהליך, המקבל התנסויות כקלט, ומבצע שינויים בבסיס ידע במטרה לשפר, על פי מדד נתון, את היכולת של פותר הבעיות, המשתמש בבסיס הידע לפתור קבוצת בעיות.
ישנם סוגי למידה שונים: כמו למידה מודרכת (supervised learning), למידה לא מודרכת (unsupervised learning), ולמידה מחיזוקים (reinforcement learning).
בקורס התעסקנו בלמידה מודרכת.

<u>שלבי הלמידה המרכזיים:</u>
1. מידול הבעיה
2. איסוף דוגמאות וייצוגן
3. עיבוד מידע
4. שימוש באלגוריתם למידה – מסווג/רגרסור
5. הערכת ביצועים

<u>מושגים</u>:
אובייקט – אלמנט מהסוג בו הבעיה הנדונה עוסקת
תכונה של אובייקט – הינה מידע שמאפיין אותו
מושג – הינו תת קבוצה של מרחב האובייקטים
מושג מטרה – המושג אותו נרצה ללמוד
סיווג של אובייקט – האם הוא שייך למושג המטרה

נפרט על שלבי הלמידה:
בשלב הראשון של הלמידה, מידול הבעיה נרצה להתאים את המושגים לבעייתנו הנתונה, לאחר מכן בשלב השני נרצה לאסוף דוגמאות רבות מהן נרצה ללמוד את מושג המטרה.
נמצא כמה שיותר דוגמאות מתויגות (דוגמא בצירוף הסיווג הנכון שלהן) ונשתדל לצמצם את הדוגמאות הרועשות כלומר דוגמאות המופיעות בצירוף שגוי. כל דוגמה ננסה לפרק לתכונות ולא להסתכל עליה כעל אובייקט אטומי, כדי שנוכל להכליל מהדוגמאות שראינו.
כלומר כל דוגמה תיוצג כווקטור תכונות.
השלב השלישי, עיבוד מקדים של המידע, הינו שלב לא פחות חשוב משלב יצירת המסווג.
בשלב זה נרצה ל"קרצף" ולערוך את הנתונים ככה שיתאימו לפורמט שאותו המסווג מצפה לקבל, וכן נרצה לתקן בעיות איכות הנובעות מחוסר מידע או ממידע לא מעובד מספיק.
שלב זה בדרך כלל דורש הבנה והסקה טובה של הנתונים וכולל למשל: השלמת ערכים חסרים, הסרת דוגמאות רועשות, נרמול נתונים, הורדת מימד ועוד.
בשלב הרביעי, שימוש באלגוריתם למידה, נייצר את המסווג עצמו המקבל מאגר דוגמאות, לומד מהן על מנת לחזות ולפעול על דוגמאות שלא נמצאות במאגר.
השלב האחרון נועד לבדוק את הדיוק של המסווג, כלומר עד כמה ניתן לסמוך על המסווג.
את דיוק המסווג נבדוק ע"י קבוצת מבחן מופרדת לחלוטין, בה נשתמש רק פעם אחת בסיום התהליך.

קיימות בעיות שונות, כמו <u>בעיות בינאריות</u> שמטרתן לקבוע האם אובייקט שייך לקבוצה או לא, ו<u>בעיות רגרסיה</u> שמטרתן לנבא מספר ממשי (מה שאנחנו בחרנו להתעסק עימו בעבודתנו).

הרעיון הכללי של הפרויקט

תיאור הבעיה

בעידן של היום, לאחר התפתחות רבה של תעשיית הסרטים, חשוב לתכנן כל סרט מראש במטרה למקסם את הרווחים שניתן להפיק ממנו.
ולכן, המטרה העיקרית של הפרויקט: בהינתן סרט יחד עם תכונות (מפיקים, במאים, תסריטאים, תקציב הסרט..) המאפיינות אותו, נרצה לייצר מערכת המחזירה את שילוב שלושת השחקנים הראשיים האופטימאלי כך שרווחי הסרט בקופות בתי הקולנוע (box office) יהיו מקסימליים.

דרך הפתרון

את הבעיה שהצגנו החלטנו לפתור בשני שלבים מרכזיים:

1. **רגרסור** -
הרגרסור יקבל כקלט סרט יחד עם התכונות הרלוונטיות המאפיינות את הסרט ויחזיר את הצפי לרווח הסרט בקופות בתי הקולנוע.
2. **אלגוריתם** -
האלגוריתם יקבל סרט עם תכונות המאפיינות אותו, ותחת התקציב הנתון של הסרט, יחזיר את קבוצת השחקנים המתאימה עבורה צפי ההכנסות מהסרט הוא מקסימלי.

כעת נפרט:

<u>שלב א' – בניית הרגרסור</u>

1. <u>השגת הנתונים</u>
מאגר הנתונים הוא המידע עליו מתבסס כל הפרויקט ולכן חשוב מאוד שיכיל נתונים טובים, ובנוסף שיכיל כמה שיותר סרטים – מה שיאפשר לרגרסור ללמוד טוב יותר וכך יגדיל את רמת הדיוק המתבקשת.
ולכן, בשלב הראשון חיפשנו מאגרי נתונים מספיק גדולים שיכילו כמה שיותר תכונות שנראו לנו רלוונטיות לקביעת הרווחים העתידיים של סרט.
לאחר חיפוש נרחב ברשת בחרנו בשני מאגרי נתונים מאתר Kaggle, שהכילו תכונות שונות שנראו לנו חשובות ומשפיעות על רווחי הסרט ואיחדנו אותם למאגר נתונים אחד.
בשלב זה מאגר הנתונים הכיל כ4800 סרטים שונים עם תכונות כגון:
   - כותרת הסרט
   - שחקנים
   - הצוות (במאי, מפיק, תסריטאים...)
   - תקציב הסרט
   - ז'אנרים
   - אתר הסרט
   - מילות מפתח
   - שפת מקור
   - פופולריות הסרט
   - חברת הפקות
   - תאריך יציאה של הסרט
   - רווחי הסרט
   - אורך הסרט (זמן)
   - ממוצע הציון שקיבל הסרט

כעת, עברנו על מאגר הנתונים, וידאנו בצורה רנדומלית חלק מהסרטים לאמת את המידע וניסינו להבין כמה מידע יהיה עלינו להשלים.

בעיות שנתקלנו בהן בחלק זה:

- מציאת מאגר נתונים עם כמות גדולה של סרטים שיכיל כמה שיותר נתונים שלדעתנו יעזרו לחזות את רווחי הסרט – קשה למצוא מאגר יחיד שיכיל את "כל מה שאנחנו צריכים" ולכן בסוף מצאנו שני מאגרים שהיה ניתן לשלב אותם יחסית בקלות וביצענו את השילוב.
- מציאת מאגר נתונים עם כמה שפחות "חוסרי מידע" – במעבר ידני מאוד קשה לראות בכמה חוסרים מדובר, נרחיב על ההתמודדות עם שלב זה בחלק הבא.

2. <u>קרצוף נתונים</u>

מאגר הנתונים שהיה בידינו בשלב זה היה רחוק מלהיות מושלם לעבודה עם הרגרסור. לכן בשלב זה התחלנו לחפש ולכתוב קטעי קוד שיעזרו לנו להשלים את המידע החסר או לתקן מידע קיים (למשל – שמות של סרטים/שחקנים שהכילו תווים בלתי קריאים).
דברים שביצענו:

- השלמת ערכים חסרים של קרוב ל1000 סרטים של תקציב ורווח.
- נרמול ערכי התקציב, הרווח.
- בניית מאגרי שחקנים, במאים, מפיקים ותסריטאים מהמידע הקיים.
- בכל מאגר, נתינת "ציון" מספרי לכל בעל מקצוע. ציון המהווה ממוצע כל הרווחים של הסרטים בהם אותו בעל מקצוע השתתף.
- לבסוף איחדנו את כל המאגרים לכדי DB המכיל עבור כל סרט את:
שמו, "ציון" עבור השחקנים, הבמאים, המפיקים והכותבים, תקציב מנורמל ורווח מנורמל. כאשר לכל סרט יש ציון על פי ממוצע 3 השחקנים הראשיים ולכל שאר בעלי המקצוע יש ציון ממוצע של כלל בעלי אותו מקצוע שהשתתפו בסרט.

3. <u>בחירת אלגוריתם + שלבי הניסויים</u>

במטרה להגיע לרגרסור המיטבי מבחינת תוצאות עבור מאגר הנתונים שאיתו אנו עובדות, נרצה לבחון מספר רגרסורים כאשר על כל רגרסור נרצה לבחון התייחסות שונה לתכונות הקיימות, למשל:

- עבור תכונות כמו שחקן, במאי – האם כדאי לתת ערך לפי הסרט הרווחי ביותר שלקח בו חלק או האם כדאי לעשות ממוצע בין הסרטים בהם השתתף.
- עבור סרטים עם ערכים קיצוניים – האם כדאי להסיר אותם (outliers).
- שימוש בgrid – שתפקידו להריץ את הרגרסור עם פרמטרים שונים ולחפש את שילוב הפרמטרים שיתנו את הערכים האופטימליים.
- עבור תכונות דומות – האם כדאי לוותר על תכונה כלשהי (correlation matrix).
- כחלק מההכנה לשלב בחירת האלגוריתם חילקנו את מאגר הנתונים בתחילה לשתי קבוצת קבוצה אחת מהווה 80% ממאגר הנתונים ו20% הנותרים מהווים את קבוצת הוולידציה. לאחר מכן חילקנו את הקבוצה הראשונה כך ש20% מתוכה מייצג את דוגמאות המבחן ו80% הנותרים את דוגמאות אימון.
סך הכל קיבלנו 3 קבוצות: דוגמאות אימון, דוגמאות ולידציה ודוגמאות מבחן.
לאורך כל שלב זה של בחירת האלגוריתם והניסויים הרגרסורים לא השתמשו ולא נתקלו בדוגמאות המבחן. דוגמאות אלה ישמשו לבחינה הסופית של המודל.

לשם בחינת התוצאות השתמשנו בשתי פונקציות שיתנו לנו אומדן להעריך את התוצאות שהתקבלו:

1. mean_squared_error – מחשבת את ממוצע ריבועי ההפרשים בערכי הרווחים בין הpredicted לבין הactual.
2. avg_absolute_percentage_error – מחשבת את ממוצע אחוז ההפרש בין כל זוג ערכי predicted וactual.

בתחילה השתמשנו בערך הראשון כי רצינו לדעת מהו הממוצע של השגיאות ביחס לכל הדוגמאות. ערך זה שימש כמדד להפרשים בערכם המספרי בין כל דוגמה לחוזי שלה, אך לאחר מספר ניסויים הבנו כי ערך זה לא מספיק לנו להבנת רמת הדיוק של הרגרסור מכיוון שזה יוצר מצב שקול בין דוגמה בעלת ערך רווח גבוה כמו 100 מיליון דולר, עבורה הרגרסור חזה רווח של 101 מיליון (כלומר חריגה של מיליון דולר, שבפועל התבטאה החוזי הוא לא בהפרש גדול ביחס למציאות) לעומת דוגמה עבורה הרווח עמד על מיליון דולר והרגסור חזה עלייה שני מיליון, אמנם מדובר כאן באותה סטייה מבחינת ערך מספרי, אך בפועל יש כאן סטייה של חוזי של פי שניים מהערך הנכון.

לכן החלטנו לנסות לאמוד את התוצאות באמצעות אחוזי שגיאה ולאו דווקא באמצעות הפרש בין הpredicted לactual מתוך ההבנה שאחוז שגיאה עבור הדוגמאות שנתנו יתן לנו באמת מדד עבור החריגה בין החוזי לבין הערך האמיתי.

לאחר מספר ניסויים הבנו שגם ערך האחוזים יכול להיות בעייתי, למשל עבור דוגמה שערך החוזי עליה נמוך יחסית והסטייה ביחס לערך זה היא באלפי דולרים בלבד ("ולא מיליונים") לדוגמה - סרט שהרוויח 10 אלף והרגרסור חוזה 100 אלף, אזי השגיאה היא של 900 אחוז מה שמגדיל מאוד את ממוצע האחוזים כשבפועל הסטייה הייתה של סכומי כסף קטנים באופן יחסי.

בסופו של דבר החלטנו להשתמש בשני הערכים הללו במקביל מכיוון שכל אחד בודק פן שונה, תוך התחשבות במגבלותיהם וחסרונותיהם.

בחרנו לעבוד עם ארבעה רגרסורים במטרה למצוא את הרגרסור בעל התוצאות הטובות ביותר:

1. **RandomForestRegressor** –
אלגוריתם זה מבוסס על שילוב של מס' עצי החלטה (מס' העצים הוא פרמטר לרגרסור).
האלגוריתם דוגם אקראית תתי קבוצות שונות מתוך קב' האימון (עם החזרות/חפיפה בין דוגמאות בקב' השונות) ומאמן את כל אחת מהקבוצות על עץ החלטה. בסוף האלגוריתם משתמש בממוצע התוצאות שהתקבלו כדי לתת חוזי מדויק יותר לערך המתבקש.
יתרונותיו של האלגוריתם הם:
- ✓ הקטנת רעשים (overfitting) – באמצעות כך שדוגמאות רועשות לא משפיעות על כל אחד מעצי ההחלטה ביער, מה שמגדיל את דיוק האלגוריתם.

2. **DecisionTreeRegressor** –
אלגוריתם המבוסס על עץ החלטה. עץ החלטה הוא מבנה דמוי עץ שבו כל צומת פנימי מייצג שאלה ומספר סופי של תשובות, כאשר כל תשובה מיוצגת על ידי בן של הצומת. כל עלה בעץ מייצג החלטה על סיווג .
בהינתן עץ החלטה ודוגמא שנרצה לסווג, ההחלטה על הסיווג תתבצע על ידי שאילת שאלה בשורש העץ, מעבר לבן המתאים לתשובה, וכך הלאה עד שנגיע לעלה, וסיווג הדוגמא יהיה סיווג העלה. סוגים של עצי החלטה הם עצי רגרסיה (כפי שהשתמשנו בעבודתנו) שבהם מותאם ערך רציף לכל תצפית ועצי סיווג שבהם מותאם ערך בדיד או מחלקת סוג לכל תצפית.
עץ החלטה הוא ייצוג פשוט לסיווג דוגמאות. למידה מבוססת עץ החלטה היא אחת הטכניקות המוצלחות ביותר ללמידה מפוקחת באמצעות סיווג. עץ יכול "ללמוד" על ידי פיצול קבוצת המקור לתתי קבוצות, המתבססות על מתן ערך לתכונה. תהליך זה חוזר על עצמו בכל תת-קבוצה באופן רקורסיבי, ונקרא מחיצות רקורסיביות. הרקורסיה מושלמת כאשר כל קבוצות המשנה בצומת בעלות אותו ערך מטרה, או כאשר הפיצול כבר אינו מוסיף ערך לתחזיות.

3. **KNeighborsRegressor** –
"אלגוריתם השכן הקרוב ביותר", הוא אלגוריתם לסיווג ולרגרסיה מקומית.
בשני המקרים הקלט תלוי ב K-התצפיות הקרובות במרחב התכונות.
עבור רגרסור, בהינתן דוגמה חדשה, האלגוריתם מחזיר את ממוצע הערכים של K השכנים הקרובים ביותר. שיטת שקלול נוספת עבור K השכנים הקרובים ביותר היא באמצעות נתינת משקל של d\1 כאשר d הינו המרחק לאותו שכן. בנוסף קיימות מספר שיטות למדידת המרחק בין שני דוגמאות שונות, כגון מרחק אוקלידי, מרחק המינג ועוד .
וכך דוגמאות האימון מיוצגות כווקטור תכונות המרחב רב ממדי, כאשר כל אחד עם סיווג כלומר בעבודתנו בא לידי ביטוי בערך של רווח הסרט. שלב האימון מתבסס על אחסון דוגמאות האימון במבנה נתונים שיאפשר חיפוש מהיר. והקלט הינו וקטור ללא סיווג (סרט ללא פרמטר הרווח). תפקיד הרגרסור לקבוע את רווח הסרט ע"פ ממוצע הרווח של K הסרטים הקרובים ביותר לסרט הנבדק.
חיסרון בשיטה זו היא כאשר התפלגות דוגמאות האימון מוטה למשל במקרה שלנו אם לצור העניין רוב הסרטים בעלי רווח גדול יחסית, אזי רווח דוגמאות המבחן יחושב ע"פ

ממוצע של הקרובים ביותר (כאשר רובם בעלי רווח גדול) וכך רוב הסרטים בדוגמאות האימון יקבלו ערך רווחי גבוה מידי .

4. **LinearRegression** -

רגרסור לינארי משתמש בשיטת הרגרסיה הלינארית. זו שיטה מתמטית למציאת קשר בין משתנה בלתי תלוי X, למשתנה תלוי Y, בהנחה שהקשר ביניהם לינארי.
בשיטה מחושב הקו הישר העובר הנקודות שבמדגם. במצב של קשר ישיר מדויק, כל נקודות המדגם ימצאו על הקו עצמו, אך בפועל גורמים נוספים משפיעים על המדגם והנקודות לרוב מפוזרות מסביב לקן ולא עליו.
במקרה זה הקו מחושב בצורה כזו שסכום ריבועי המרחקים של נקודות המדגם מהקו יהיה הקטן ביותר. כמו כן, קיימת רגרסיה לינארית מרובה המחשבת קשר בין מספר משתנים בלתי תלויים יחד, למשתנה תלוי אחד, כפי שבעבודתנו הרגרסור הלינארי יחפש את הקו עבורו סכום ריבועי המרחקים של דוגמאות האימון ממנו, יהיה הקטן ביותר, וע"י כך ינסה להיות כמה שיותר מדויק ועקבי לדוגמאות האימון כאשר יקבל דוגמת מבחן ויצטרך להחזיר עבורה את הערך מספרי של חיזוי הרווחים מאותו הסרט.

נתאר כעת את התוצאות שהתקבלו לפי סדר הניסויים שביצענו:

ניסוי ראשון

בשלב זה היה בידינו מאגר הנתונים לאחר שלב סידור והשלמת כל הנתונים החסרים, להלן התוצאות לכל אחד מהרגרסורים. נזכיר כי בשלב זה ערכי השחקנים, במאים, מפיקים, ותסריטאים חושבו לפי ממוצע ערכים של הסרטים בהם השתתפו (בהמשך הניסויים נבחן גם את האפשרות לmax בין הסרטים בהם השתתפו).

כפי שניתן לראות בגרפים ה- predict אכן עוקב את ערכי ה- actual, כלומר ניתן לראות שסרט בעל רווח גבוה אכן מתאפיין בחוזי של רווח גבוה **יחסית**, אך גם ניתן לראות שהצפי רחוק מרמת הדיוק שהיינו רוצים לראות בשלב זה, כלומר הסטייה באחוזים אכן גדולה.

אל ערך ה MSE נתייחס בהקשר של האם עלה או ירד בניסוי הבא, מכיוון שהערכים שהרגרסור מקבל הינם הערכים המנורמלים, נרצה להתייחס לשינוי היחסי שלהם בכל ניסוי.

לאחר ניתוח הנתונים שהתקבלו עברנו בחזרה לשלב "קרצוף הנתונים" לבדוק כי לא טעינו במהלך השלב, ולאחר שווידאנו שהנתונים אכן אמינים ע"י מציאת דוגמאות חריגות בודדות ותיקון ערכיהן, הסקנו כי בנוסף יכול להיות שקיימות דוגמאות בעלות ערכים חריגים שמשפיעות על התוצאה משמעותית ואליהן נתייחס בניסויים הבאים.

מקרא כללי עבור הגרפים: ערכי ה-x מייצגים א הסרט עצמו (דוגמת האימון) וערך ה- y מייצג את ערך הרווח המנורמל.

הגרף הכתום מייצג את מה שהרגרסור חוזה ואילו הגרף התכלת מייצג את מה שבפועל קורה, כלומר את התיוג של דוגמת האימון – ערך הרווח של הסרט בפועל.

Random Forest

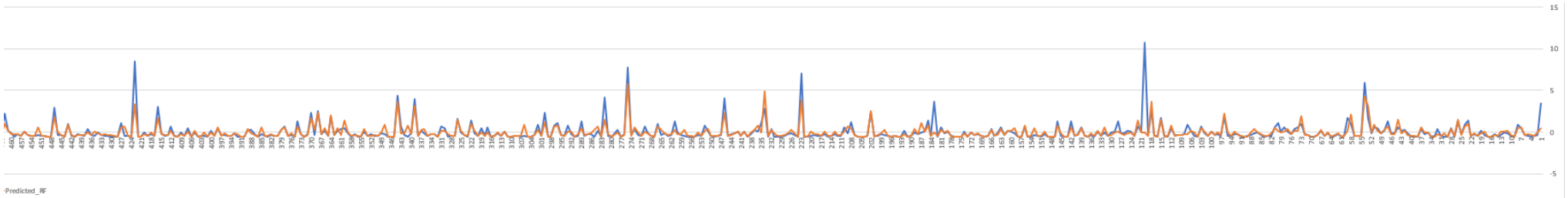
Actual value

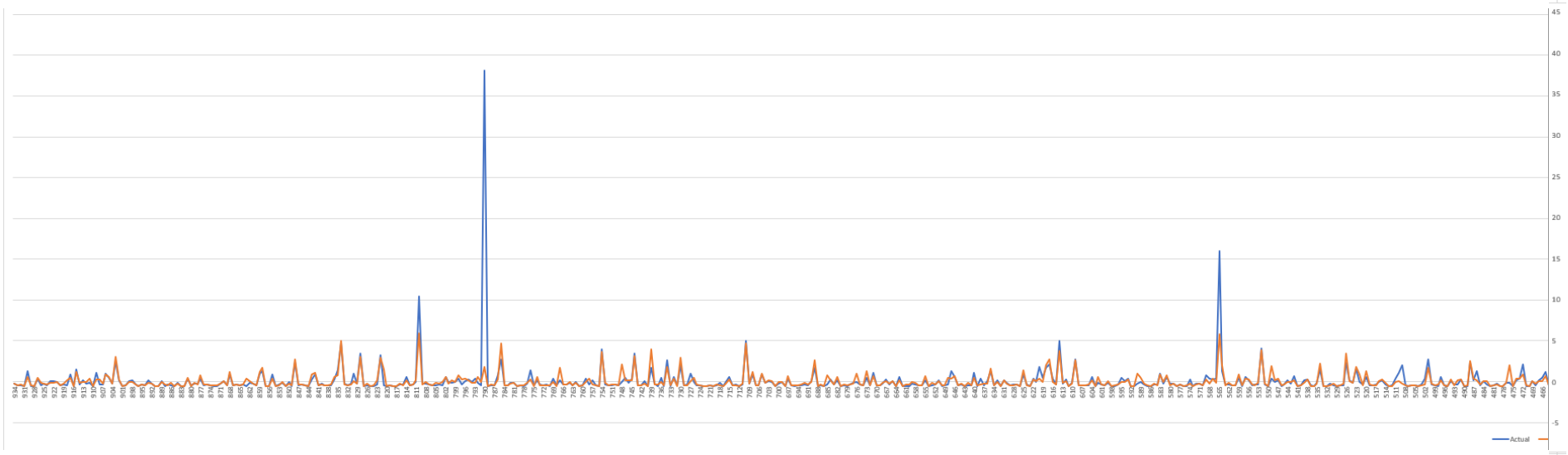
Predicted RF

```
MSE RandomForestRegressor: 1.892672979070471
MAPE RandomForestRegressor: 274.09047710235075
```

ממוצע ריבועי ההפרשים בין התוצאות

ממוצע אחוזי הסטייה





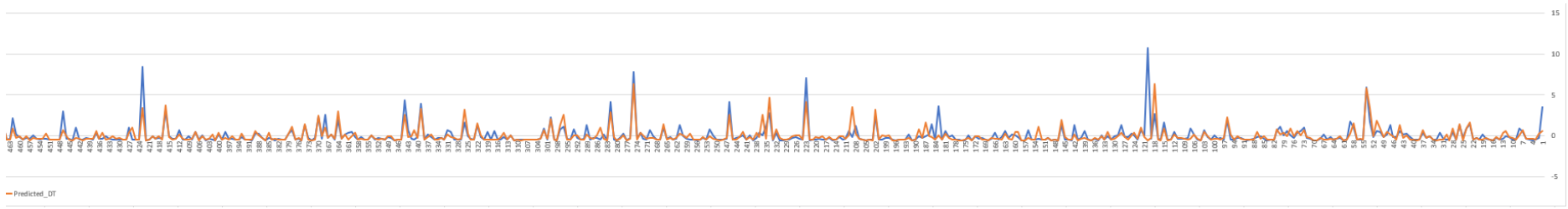
Decision Tree

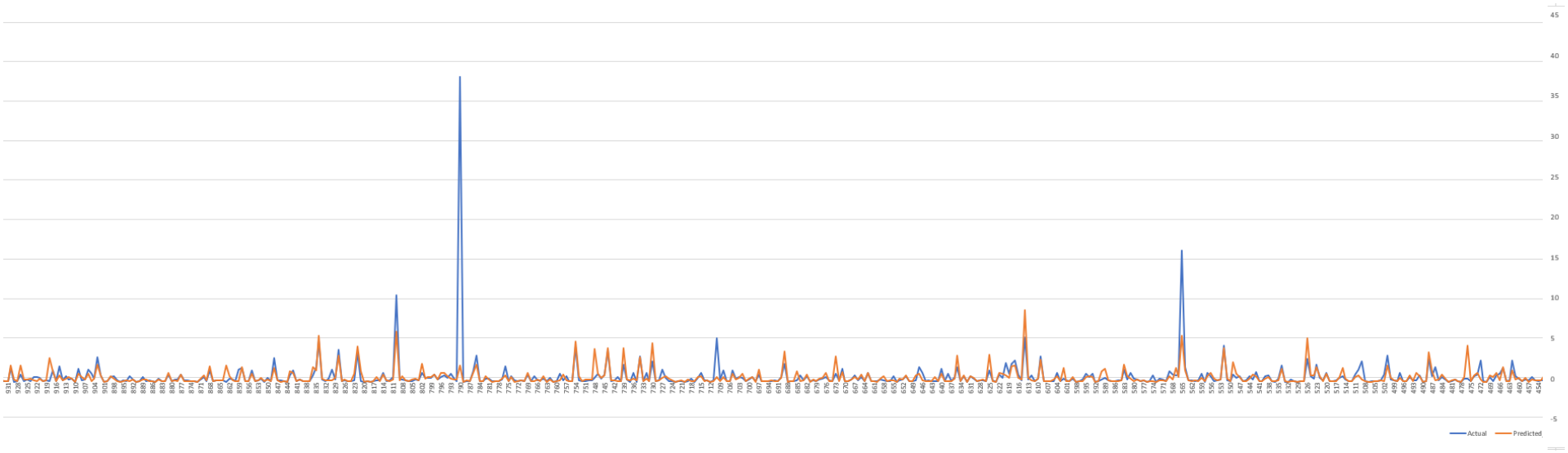
Actual value
Predicted DT

```
MSE DecisionTreeRegressor: 2.111098094211612
MAPE DecisionTreeRegressor: 265.3396291691885
```

ממוצע ריבועי ההפרשים בין התוצאות

ממוצע אחוזי הסטייה





KNN

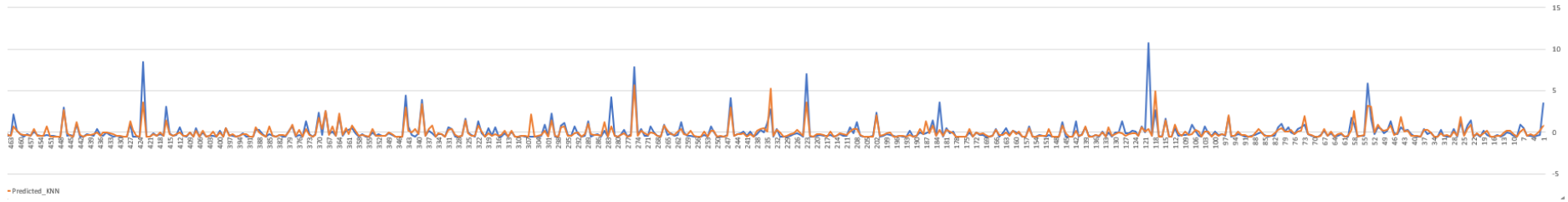
Actual value

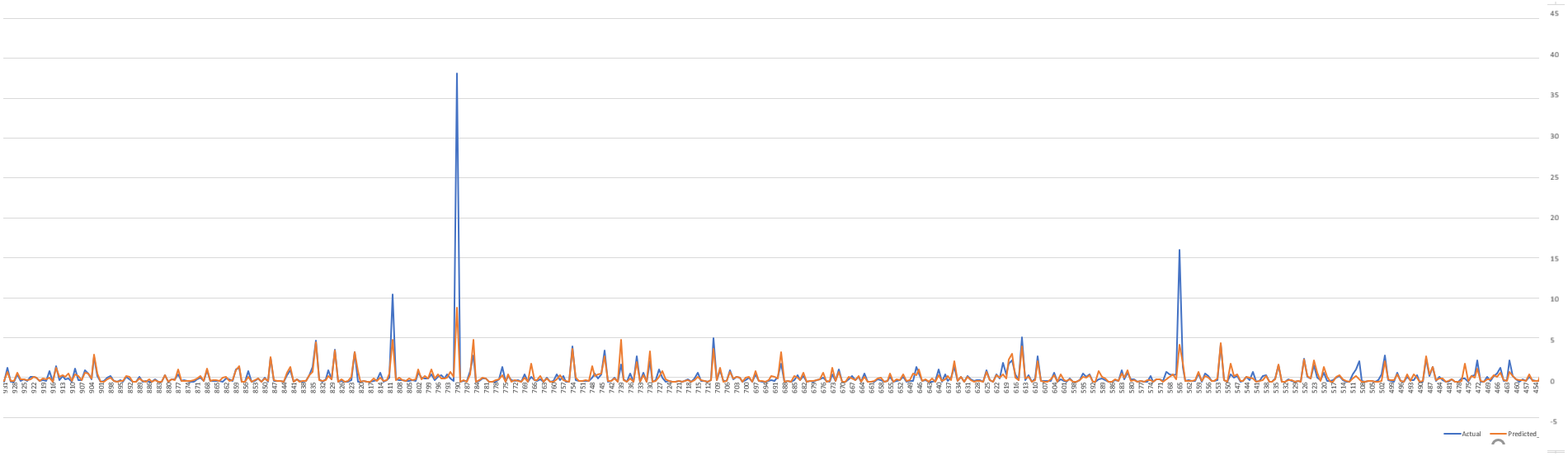
Predicted RF

```
MSE KNeighborsRegressor: 1.4704712846471446
MAPE KNeighborsRegressor: 228.23483064540704
```

ממוצע ריבועי ההפרשים בין התוצאות

ממוצע אחוזי הסטייה





Linear

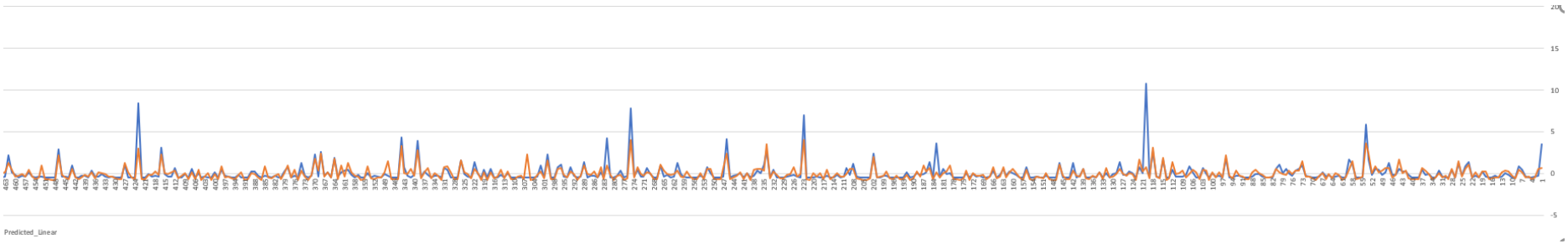
Actual value

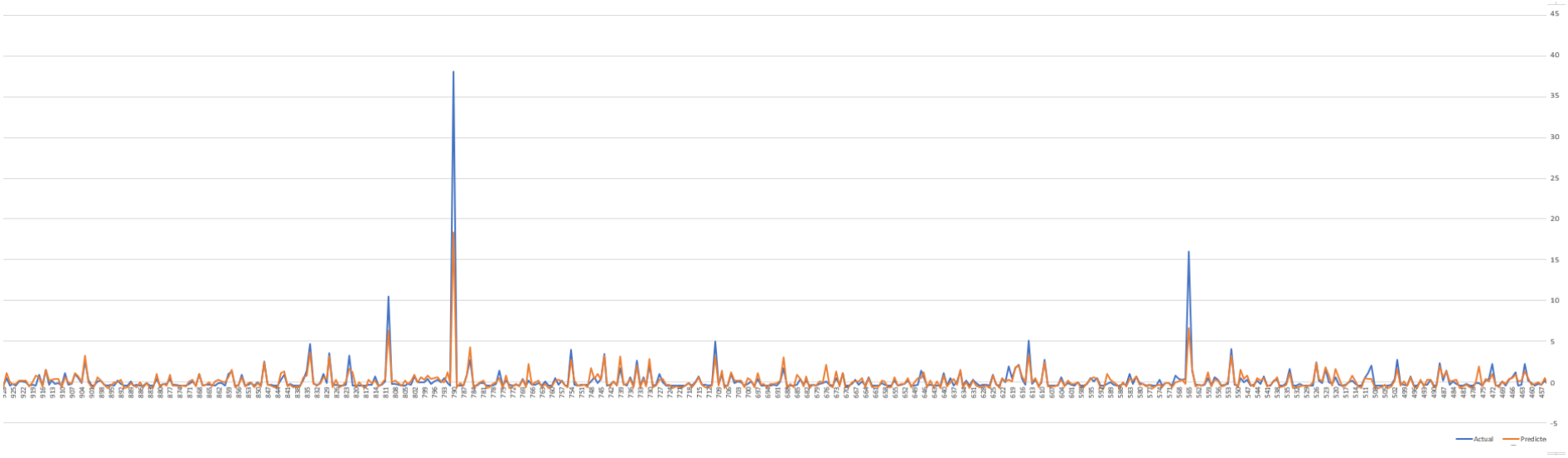
Predicted RF

```
MSE LinearRegression: 0.9150917286003969
MAPE LinearRegression: 492.2365715373164
```

ממוצע ריבועי ההפרשים בין התוצאות

ממוצע אחוזי הסטייה





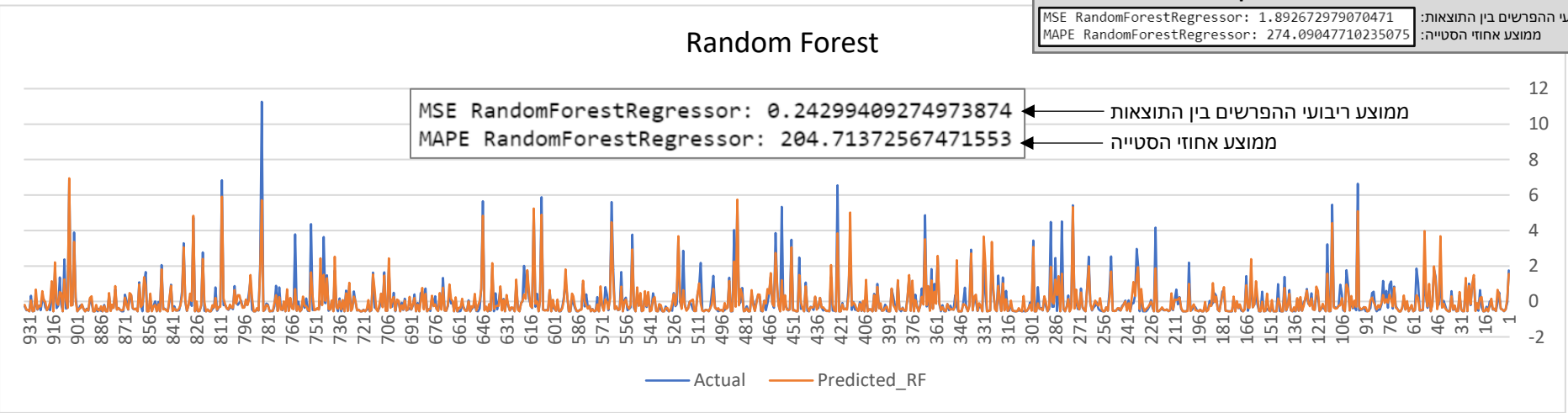
ניסוי שני

בשלב זה החלטנו לנסות להתמודד עם חוסר הדיוק באמצעות outliers, כלומר הסרת דוגמאות בעלות ערך קיצוני ביחס לשאר הדוגמאות.
לכן, הפעלנו את פונקציית הoutliers ובעקבות כך הורדו סרטים שהערכים שלהם היו חריגים (ערכים שהם נכונים אך חריגים ביחס לשאר).
לאחר ביצוע הoutliers צפינו כי ערך הMSE ירד לעומת הניסוי הראשון וזאת מכיוון שכפי שהסברנו לעיל ערכו עלול להיות גבוה ברגע שיש ערכים קיצוניים גבוהים, ולכן בניסוי זה אכן הורדנו ערכים קיצוניים (נמוכים וגבוהים) ואכן כפי שניתן לראות בתוצאות, ערך הMSE ירד באופן משמעותי. בניסוי הקודם הערכים היו מעל 1 וכעת כל הערכים נמוכים מ1.
כמו כן אחוזי הסטייה ירדו ברוב הרגרסורים, ועל כן נסיק שהרגרסורים חזו עבור דוגמאות הoutliers שהורדו, ערכים גבוהים/נמוכים ביחס לערכם האמיתי.
בנוסף ניתן לראות ע"פ הגרפים כי ערכי ה– predict כפי שצפינו עדיין מתנהגים דומה לערכי ה- actual (הרגרסורים חוזים את התוצאות נכון מבחינת המגמות) כלומר הגרפים של הactual ושל הpredict עוקבים אחד אחר השני.

*נשים לב לשינוי שחל בערכי הy של הגרפים לעומת בניסוי הראשון – שכן הורדנו את התוצאות החריגות.

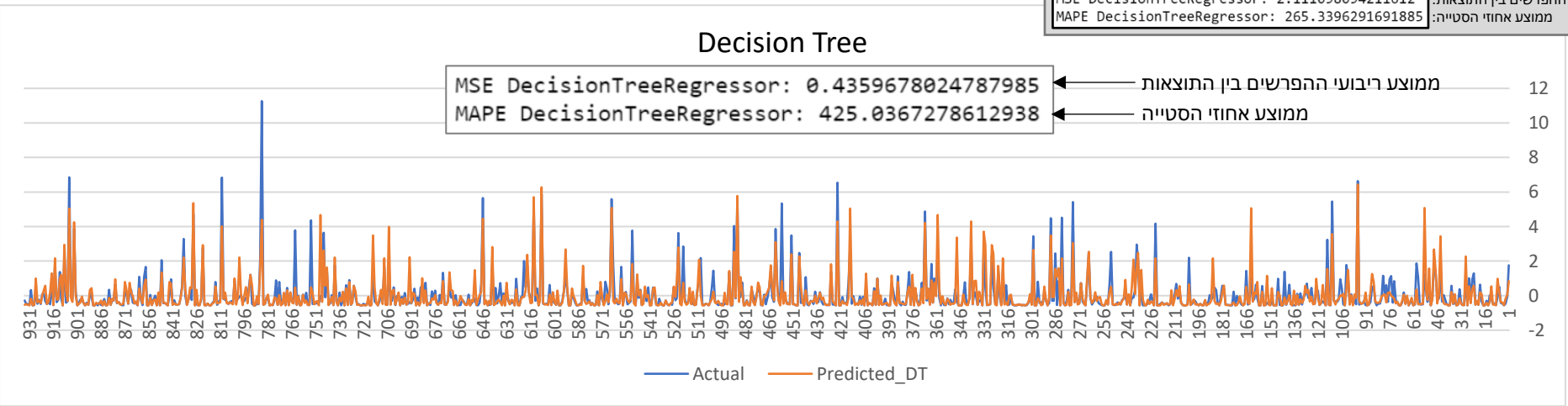
תזכורת לנתונים מהניסוי הקודם:

ממוצע ריבועי ההפרשים בין התוצאות:	MSE RandomForestRegressor: 1.892672979070471
ממוצע אחוזי הסטייה:	MAPE RandomForestRegressor: 274.09047710235075



תזכורת לנתונים מהניסוי הקודם:

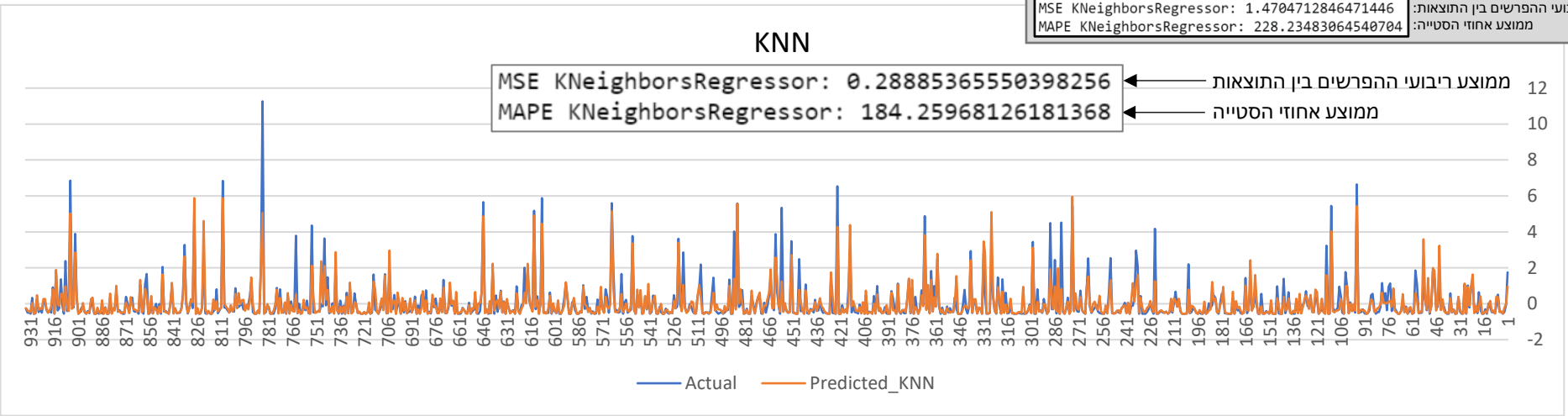
ממוצע ריבועי ההפרשים בין התוצאות:	MSE DecisionTreeRegressor: 2.111098094211612
ממוצע אחוזי הסטייה:	MAPE DecisionTreeRegressor: 265.3396291691885



תזכורת לנתונים מהניסוי הקודם:

ממוצע ריבועי ההפרשים בין התוצאות: `MSE KNeighborsRegressor: 1.4704712846471446`

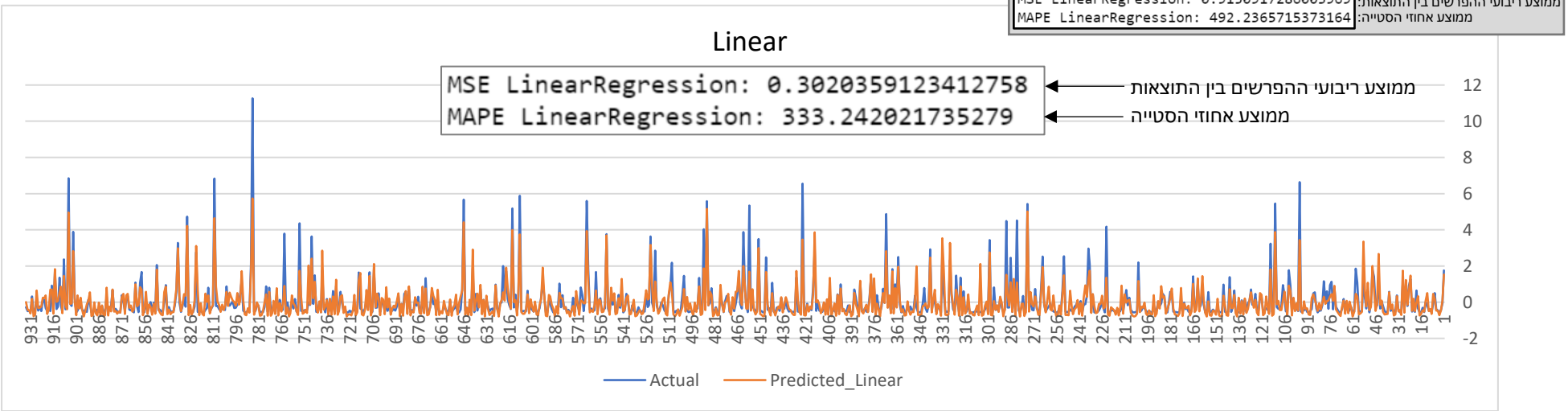
ממוצע אחוזי הסטייה: `MAPE KNeighborsRegressor: 228.23483064540704`



תזכורת לנתונים מהניסוי הקודם:

ממוצע ריבועי ההפרשים בין התוצאות: `MSE LinearRegression: 0.9150917286003969`

ממוצע אחוזי הסטייה: `MAPE LinearRegression: 492.2365715373164`



ניסוי שלישי

בשלב זה ביצענו grid שתפקידו להריץ את הרגרסורים עם פרמטרים שונים ולחפש את שילוב הפרמטרים שיתנו את הערכים האופטימליים, נציג את ההבדלים בתוצאות:

תזכורת לנתונים מהניסוי הקודם(2):

```
MSE RandomForestRegressor: 0.24299409274973874
MAPE RandomForestRegressor: 204.71372567471553
---------------------------------------------
MSE DecisionTreeRegressor: 0.4359678024787985
MAPE DecisionTreeRegressor: 425.0367278612938
---------------------------------------------
MSE KNeighborsRegressor: 0.28885365550398256
MAPE KNeighborsRegressor: 184.25968126181368
---------------------------------------------
MSE LinearRegression: 0.3020359123412758
MAPE LinearRegression: 333.242021735279
```

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

לאחר הgrid:

```
MSE RandomForestRegressor: 0.23246626103541
MAPE RandomForestRegressor: 202.098132134
---------------------------------------------
MSE DecisionTreeRegressor: 0.36892226931022
MAPE DecisionTreeRegressor: 420.349211039048
---------------------------------------------
MSE KNeighborsRegressor: 0.2572097643079988
MAPE KNeighborsRegressor: 176.2991191582476
---------------------------------------------
MSE LinearRegression: 0.2920351203492
MAPE LinearRegression: 331.242021735279
```

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

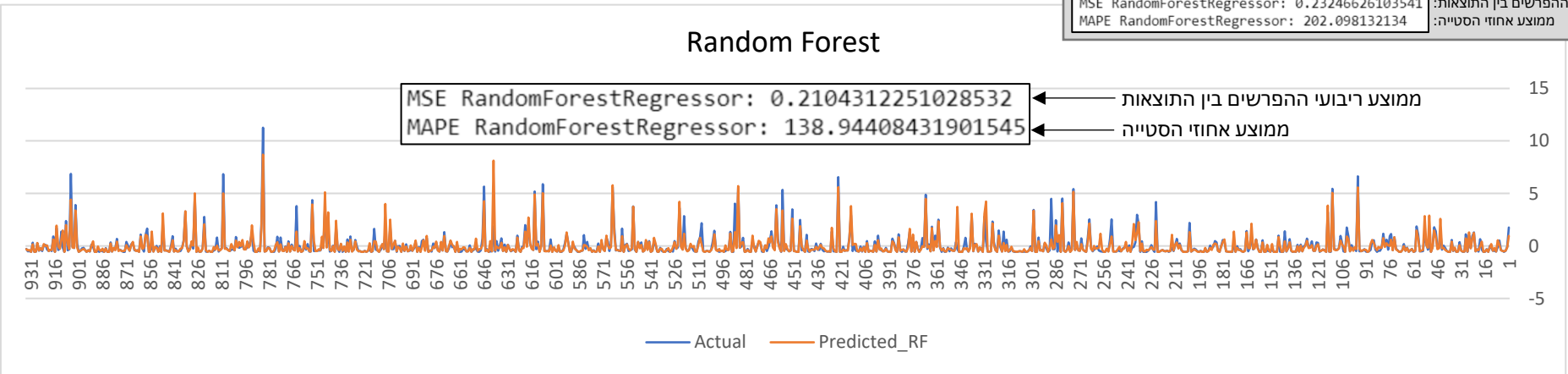
ניתן לראות כי אחוזי השגיאה והMSE אכן ירדו, כלומר הפונקציה אכן מצאה פרמטרים אופטימאליים עבור תוצאות חיזוי טובות יותר, אך לא בהפרש גדול מספיק, ולכן המשכנו לניסוי הבא.

ניסוי רביעי

בשלב זה החלטנו לשנות את אופן חישוב הערכים בDB. כפי שהסברנו, יצרנו מאגר של שחקנים, במאים, מפיקים וכותבים.
בהתחלה כל אחד קיבל ציון לפי ממוצע הרווחים של הסרטים בהם השתתפו. בשלב זה החלטנו לשנות את צורת החישוב הזו ולתת לכל בעל מקצוע (מלבד השחקנים) את הציון שלו לפי הסרט בעל הרווח הגבוה בו השתתף וזאת מכיוון שלאחר מעבר על מאגר הנתונים, שמנו לב כי קיימים במאים, מפיקים וכותבים מפורסמים מצליחים המשתתפים בסרטים רווחים מאוד אך במאגר הנתונים גם מופיעים המון סרטים פחות מצליחים שלהם ולכן לעיתים הציון שקיבלנו קטן לעומת בעלי מקצוע אחרים המופיעים בפחות סרטים במאגר שהיו פחות מוצלחים אך ציונם היה גבוה מאותו ממוצע – לכן חשבנו שממוצע הוא לאו דווקא הערך הנכון להעביר לרגרסור ובחרנו להעביר ערכי מקסימום במקומו.
נצפה כי אחוזי השגיאה ירדו משמעותית וכי גם הMSE ירד, שכן הרגרסור לומד באמצעות דוגמאות אימון נכונות יותר ובעקבות כך הלמידה תהיה טובה יותר.
לאחר ששינינו את הDB קיבלנו את התוצאות שלהלן, ניתן לראות כי כפי שחזינו אכן הגדלנו את הדיוק והערכים ירדו משמעותית.

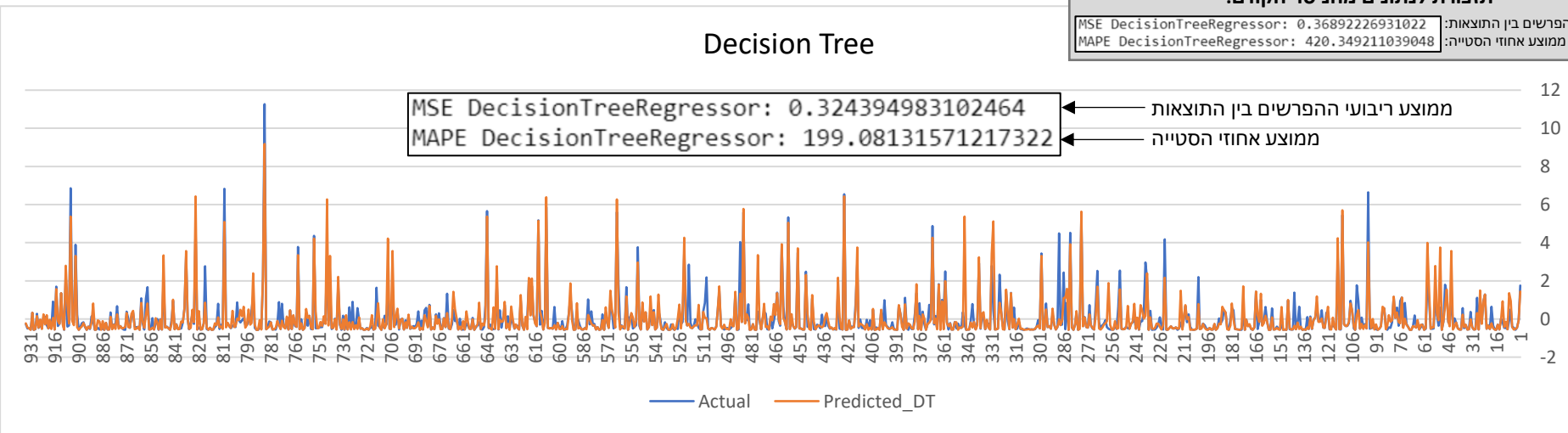
תזכורת לנתונים מהניסוי הקודם:

ממוצע ריבועי ההפרשים בין התוצאות: MSE RandomForestRegressor: 0.23246626103541
ממוצע אחוזי הסטייה: MAPE RandomForestRegressor: 202.098132134



תזכורת לנתונים מהניסוי הקודם:

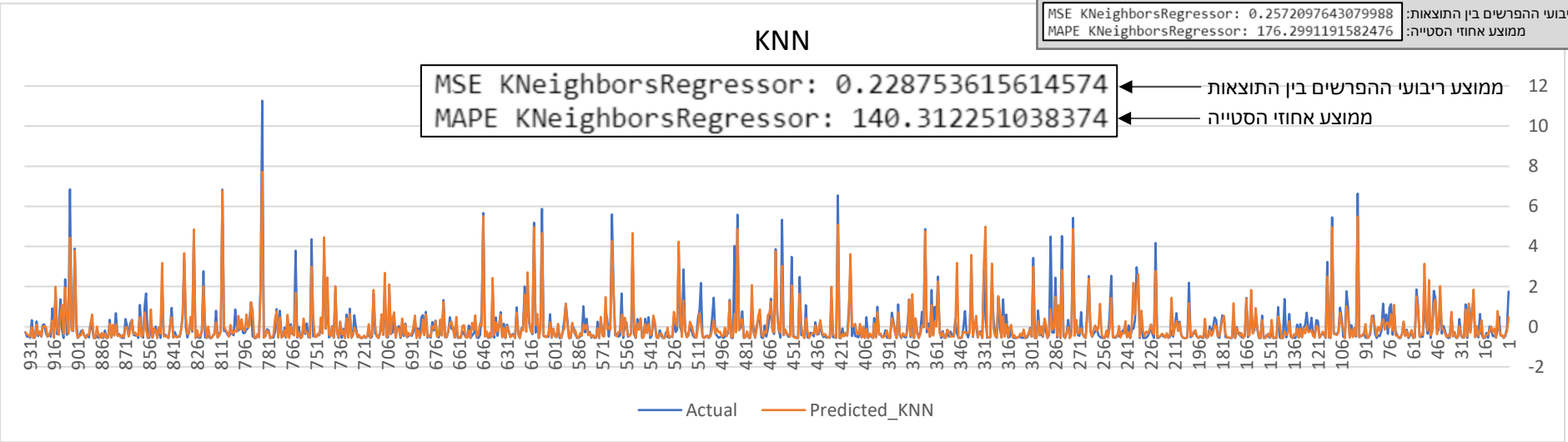
ממוצע ריבועי ההפרשים בין התוצאות: MSE DecisionTreeRegressor: 0.36892226931022
ממוצע אחוזי הסטייה: MAPE DecisionTreeRegressor: 420.349211039048



תזכורת לנתונים מהניסוי הקודם:

ממוצע ריבועי ההפרשים בין התוצאות: `MSE KNeighborsRegressor: 0.2572097643079988`

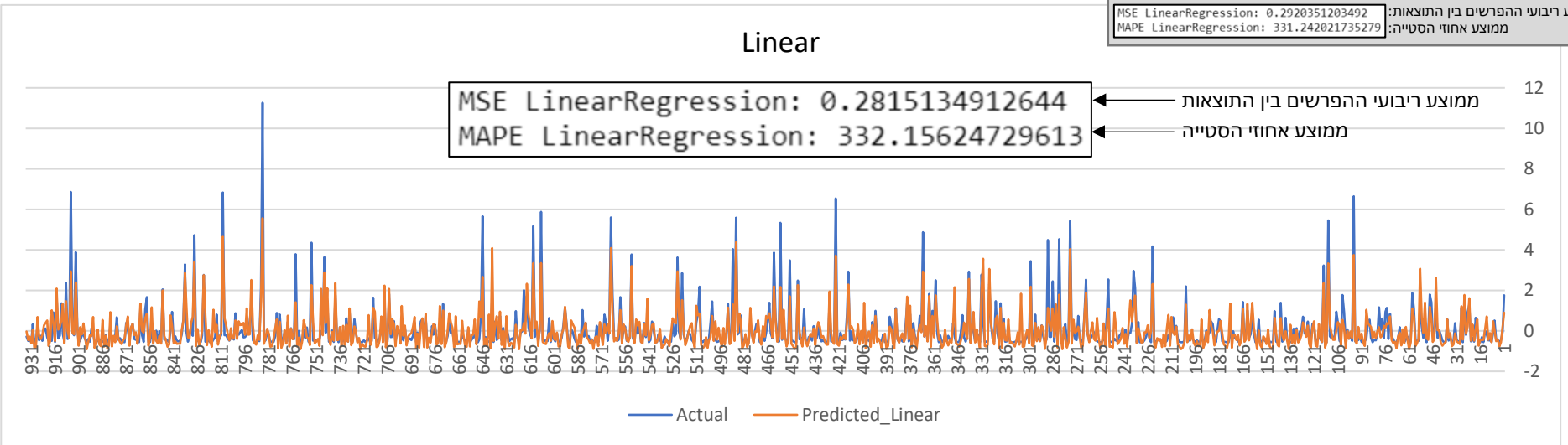
ממוצע אחוזי הסטייה: `MAPE KNeighborsRegressor: 176.2991191582476`



תזכורת לנתונים מהניסוי הקודם:

ממוצע ריבועי ההפרשים בין התוצאות: `MSE LinearRegression: 0.2920351203492`

ממוצע אחוזי הסטייה: `MAPE LinearRegression: 331.242021735279`



ניסוי חמישי

בשלב זה ביצענו grid שתפקידו להריץ את הרגרסורים עם פרמטרים שונים ולחפש את שילוב הפרמטרים שיתנו את הערכים האופטימליים, נציג את ההבדלים בתוצאות:

תזכורת לנתונים מהניסוי הקודם (4):

```
MSE RandomForestRegressor: 0.2104312251028532
MAPE RandomForestRegressor: 138.94408431901545
----------------------------------------------
MSE DecisionTreeRegressor: 0.324394983102464
MAPE DecisionTreeRegressor: 199.08131571217322
----------------------------------------------
MSE KNeighborsRegressor: 0.228753615614574
MAPE KNeighborsRegressor: 140.312251038374
----------------------------------------------
MSE LinearRegression: 0.2815134912644
MAPE LinearRegression: 332.15624729613
```

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

לאחר הgrid:

```
MSE RandomForestRegressor: 0.20901637883097204
MAPE RandomForestRegressor: 136.2154302199941
----------------------------------------------
MSE DecisionTreeRegressor: 0.321562148761056
MAPE DecisionTreeRegressor: 190.6934321982567
----------------------------------------------
MSE KNeighborsRegressor: 0.22428637446122
MAPE KNeighborsRegressor: 140.215613429723
----------------------------------------------
MSE LinearRegression: 0.282153444801
MAPE LinearRegression: 335.5676661038
```

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

ניתן לראות כי אחוזי השגיאה והMSE ירדו במעט או כמעט ולא השתנו.
ייתכן כי הדבר נובע מזה שהערכים האופטימאליים קרובים לערכי ברירת המחדל של הרגרסורים.

ניסוי שישי

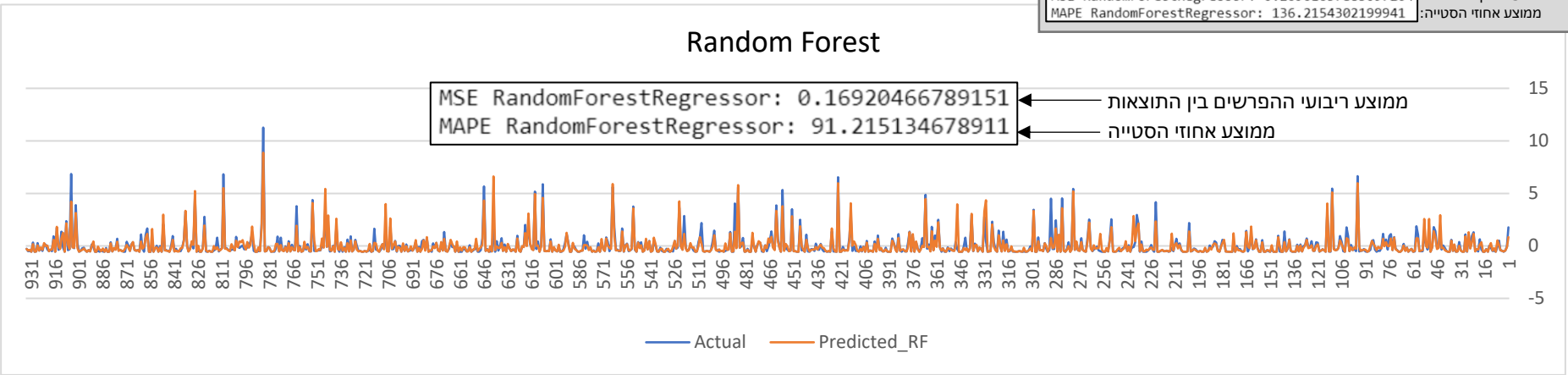
בשלב זה חזרנו שוב לשלב קרצוף הנתונים במטרה לבדוק מה יכול לשפר משמעותית את תוצאות הניסוי. הבחנו כי ערכי הבמאים, מפיקים, וכותבים גבוה משמעותית מערכי השחקנים. (בהתחשב בעובדה כי ערך השחקנים אינו פחות חשוב ואף יותר). ולכן החלטנו לנסות לשנות את ערכי השחקנים שיהיו גם הם ע"פ הסרט הרווחי ביותר ולא ע"פ ממוצע כל הסרטים שאותו שחקן השתתף בהם. ולכן ביצענו באופן דומה לניסוי הרביעי את לקיחת ערך הmax, רק הפעם לשחקנים – כלומר, כל שחקן קיבל את הערך שלו לפי הסרט הרווחי ביותר בו השתתף (בנוסף לשמירת ערכי המקסימום של שאר התכונות מהניסוי הרביעי).

ניתן לראות כי אכן אחוזי השגיאה ירדו משמעותית וכן גם הMSE, מה שעולה בקנה אחד עם העובדה שהשחקנים משפיעים על רווח הסרט וכי הפרש הערכים בין הבמאים, כותבים מפיקים לבין ערך השחקנים פגע בדיוק של הDB וכי דייקנו את הDB, וע"י כך הלמידה הייתה טובה יותר וכך גם התוצאות טובות יותר.

תזכורת לנתונים מהניסוי הקודם:

ממוצע ריבועי ההפרשים בין התוצאות: MSE RandomForestRegressor: 0.20901637883097204

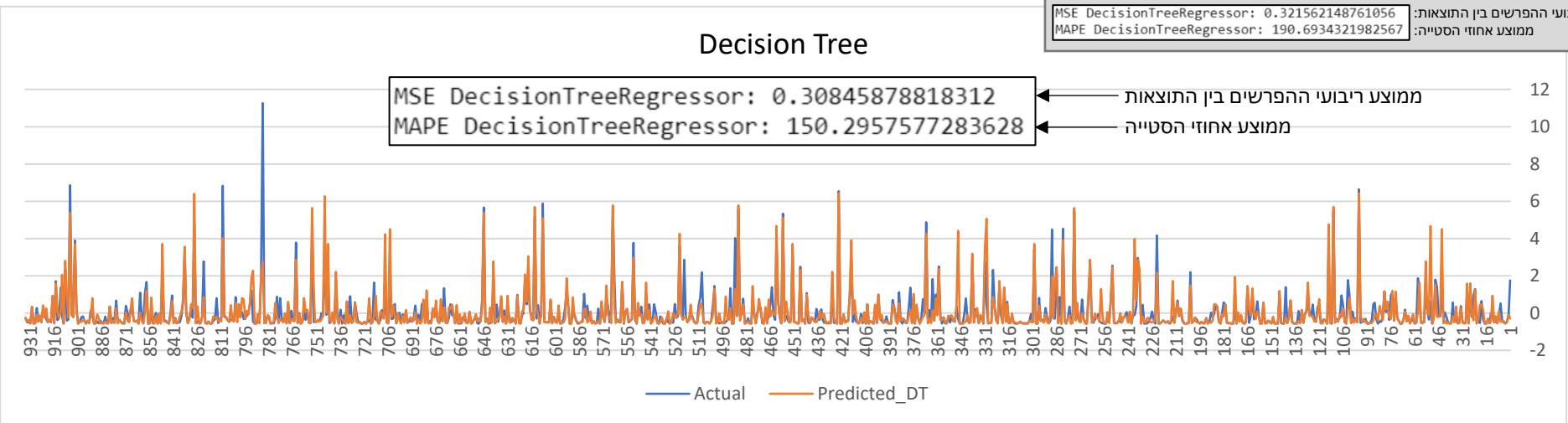
ממוצע אחוזי הסטייה: MAPE RandomForestRegressor: 136.2154302199941



תזכורת לנתונים מהניסוי הקודם:

ממוצע ריבועי ההפרשים בין התוצאות: MSE DecisionTreeRegressor: 0.321562148761056

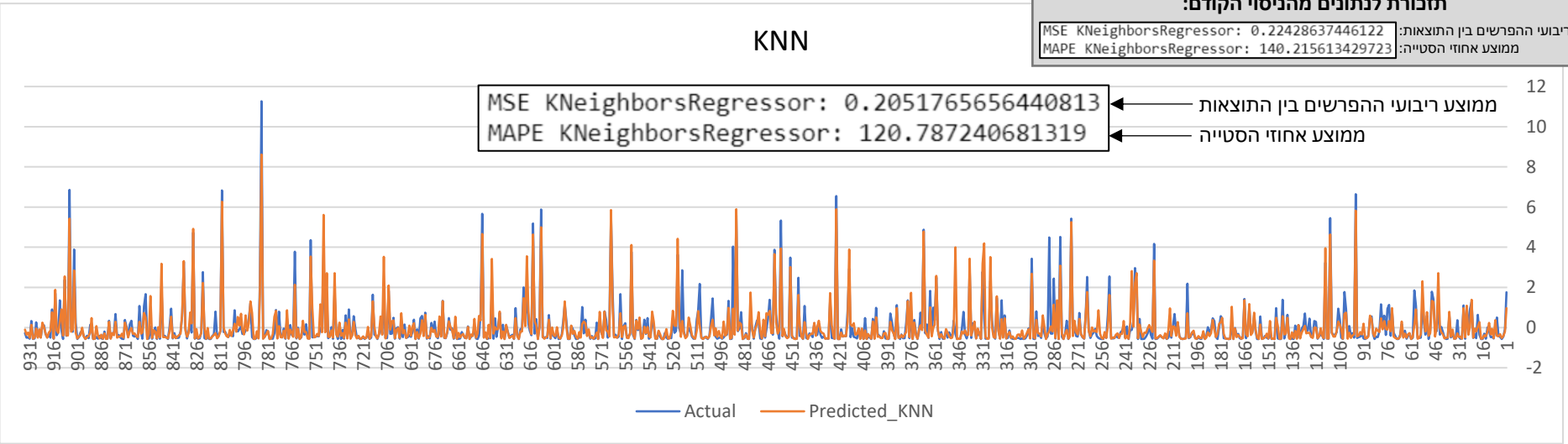
ממוצע אחוזי הסטייה: MAPE DecisionTreeRegressor: 190.6934321982567



תזכורת לנתונים מהניסוי הקודם:

ממוצע ריבועי ההפרשים בין התוצאות: MSE KNeighborsRegressor: 0.22428637446122

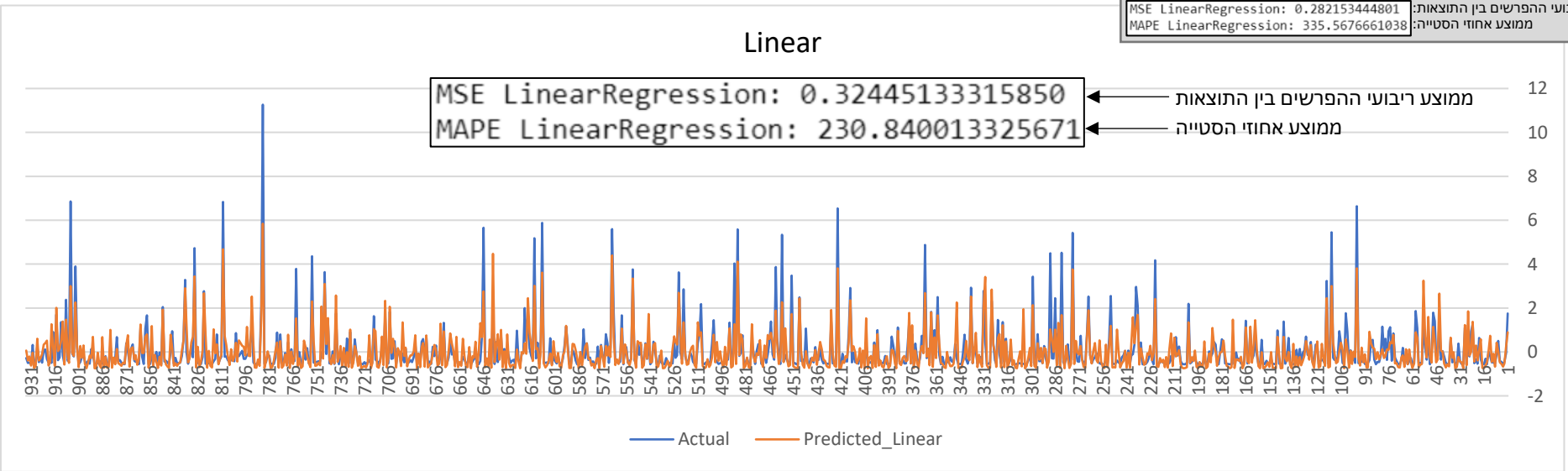
ממוצע אחוזי הסטייה: MAPE KNeighborsRegressor: 140.215613429723



תזכורת לנתונים מהניסוי הקודם:

ממוצע ריבועי ההפרשים בין התוצאות: MSE LinearRegression: 0.282153444801

ממוצע אחוזי הסטייה: MAPE LinearRegression: 335.5676661038



ניסוי שביעי

בשלב זה ביצענו grid שתפקידו להריץ את הרגרסורים עם פרמטרים שונים ולחפש את שילוב הפרמטרים שיתנו את הערכים האופטימליים, נציג את ההבדלים בתוצאות:

תזכורת לנתונים מהניסוי הקודם (6):

```
MSE RandomForestRegressor: 0.16920466789151
MAPE RandomForestRegressor: 91.215134678911
---------------------------------------------
MSE DecisionTreeRegressor: 0.30845878818312
MAPE DecisionTreeRegressor: 150.2957577283628
---------------------------------------------
MSE KNeighborsRegressor: 0.2051765656440813
MAPE KNeighborsRegressor: 120.787240681319
---------------------------------------------
MSE LinearRegression: 0.32445133315850
MAPE LinearRegression: 230.840013325671
```

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

לאחר הgrid:

```
MSE RandomForestRegressor: 0.1653542711213
MAPE RandomForestRegressor: 90.2334511215
---------------------------------------------
MSE DecisionTreeRegressor: 0.3144328188821
MAPE DecisionTreeRegressor: 148.31355702912
---------------------------------------------
MSE KNeighborsRegressor: 0.200037125697
MAPE KNeighborsRegressor: 118.315829054307
---------------------------------------------
MSE LinearRegression: 0.3145668172463
MAPE LinearRegression: 233.21551346768
```

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

ממוצע ריבועי ההפרשים בין התוצאות:
ממוצע אחוזי הסטייה:

ניתן לראות כי גם בשלב הזה כמו בניסוי החמישי, אחוזי השגיאה והMSE ירדו במעט או כמעט ולא השתנו.
ייתכן כי הדבר נובע מזה שהערכים האופטימאליים קרובים לערכי ברירת המחדל של הרגרסורים.

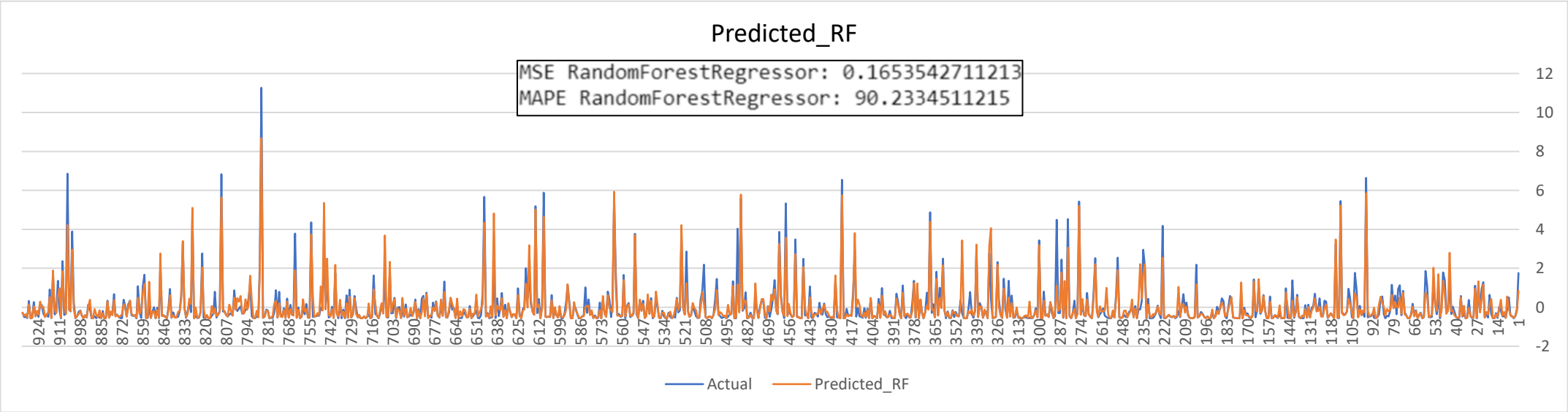
במהלך הניסויים חשבנו לבצע ניסויים נוספים:

ניסוי ראשון - כפי שהסברנו לעיל ערך המפיקים, כותבים ובמאים, מורכב ממוצע/max של כל אותם בלי מקצוע המשתתפים באותו הסרט, ואילו ציון השחקנים לסרט ניתן ע"י ממוצע של שלושת השחקנים הראשיים, לכן חשבנו להגדיל או להקטין את כמות השחקנים איתם נרצה לחשב את ערך השחקן עבור כל סרט, אך נתקלנו בבעיה ששחקנים שהופיעו ברשימה כמספר 4 הורידו משמעותית את ערך השחקנים עבור סרטי קלאסיקות המצליחות בכל הזמנים ולכן החלטנו שזה מוריד את הדיוק וויתרנו על אופציה זו.

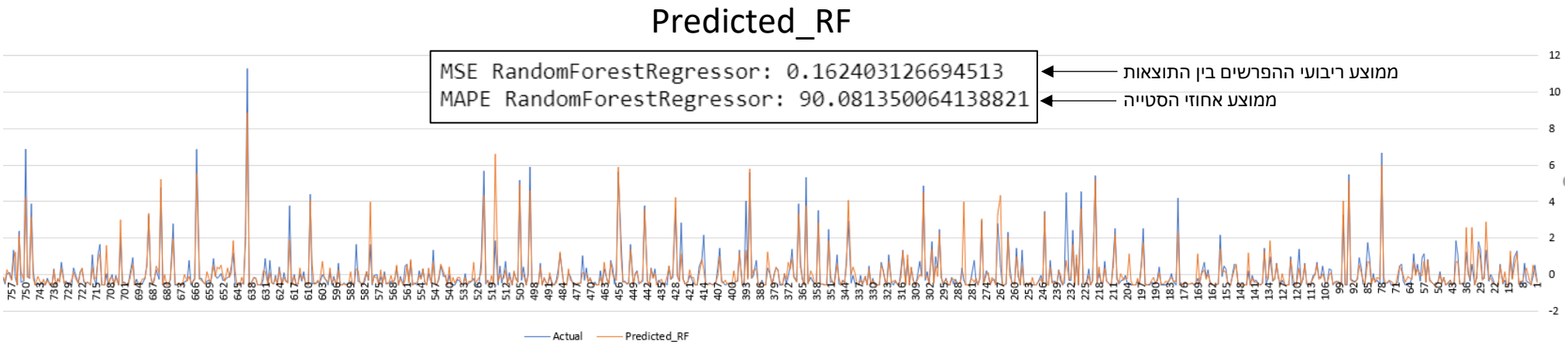
ניסוי שני – במקום לחשב את הערך הממוצע של השחקנים חשבנו לפצל את ערך השחקנים לשלושה ערכים שונים כלומר כל אחד משלושת השחקנים הראשיים יהווה פיצ'ר עבור הרגרסור. אך גם ניסוי זה גרם לסטיות גבוהות באחוזי השגיאה.

ניסוי שלישי - בשלב זה חקרנו שוב את מאגר הנתונים וראינו כי ערך הבמאים והמפיקים מאוד קרוב אחד לשני ולכן רצינו להשתמש ב- correlation matrix על מנת לבדוק האם קיימות תכונות מיותרות (מיותרות בגלל שהן דומות אחת לשנייה ונותנות השפעה כפולה, מה שעלול לגרום לחוסר דיוק).
אך מכיוון שבאמת כמות התכונות (פיצ'רים) אינה רבה מראש, החלטנו לוותר על זה (בפועל הרצנו ולא ראינו הבדלים מאוד משמעותיים) .

ניתן לראות כי לאחר הgrid בניסוי האחרון הגענו לMSE ולאחוזי סטייה מינימליים ברגרסור Random Forest.
בשלב זה חשבנו לבנות אלגוריתם אשר יקבל את ערכי הרגרסורים ויחשב ערך ממוצע לכולם (או לפחות לשניים בעלי אחוז השגיאה הנמוך ביותר)
אך מכיוון שערכי השגיאה של Random Forest היו משמעותית טובים משל שאר הרגרסורים החלטנו לבחור את האלגוריתם Random Forest בלבד.



בשלב זה נשתמש בדוגמאות המבחן שדיברנו עליהן לעיל.
נזכיר כי הרגרסור לא מכיר את דוגמאות המבחן ולא נתקל בהן עד כה.
כעת נשתמש בדוגמאות האימון באמצעותן נאמן את המודל ולאחר מכן נבחן את המודל באמצעות דוגמאות המבחן.
להלן התוצאות:



שלב ב' – אלגוריתם למציאת שילוב שחקנים מיטבי

לאחר שבנינו את הרגרסור המקבל ווקטור עבור כל דוגמת סרט הכולל ערכי תקציב, מפיק, במאי, כותב ושחקנים ומחזיר את הצפי הרווחים מהסרט, נרצה לבנות את האלגוריתם שישתמש ברגרסור שנבנה בשלב א' ויחזיר את שילוב שלושת השחקנים המתאימים לסרט בהתחשב בתקציבו ובמטרה למקסם את רווחיו.

תיאור האלגוריתם

- ✓ **תזכורת** – בשלב קרצוף הנתונים יצרנו מאגר לכל בעל מקצוע כאשר ערכם הינו הרווח המקסימאלי של סרט שבו השתתף לאחר נרמול.
(הנרמול כולל חלוקה ב10^8 ונרמול ע"פ התוחלת והשונות של כל תכונה במאגר).
בנוסף בנינו עשרה מאגרים שמכילים את סך כל השחקנים, מחולקים לערך בין 1 ל10 לפי הצלחת השחקנים (1- הכי נמוך, 10- הכי גבוה) שצפינו שנובעת מהצלחת הסרטים בהם השתתפו.

האלגוריתם מקבל את שמות המפיקים, כותבים ובמאים ואת התקציב (יחד עם מאגר הדוגמאות לשם הלמידה של הרגסור בו הוא משתמש והרגרסור).

בשלב הראשון של האלגוריתם ננרמל את התקציב המתקבל כקלט לאלגוריתם בהתאמה לנרמול שעשינו על דוגמאות המבחן. הערך המתקבל ינורמל לערך שבין 3 ל30, במטרה לחפש אחר שלשות שחקנים מתאימות לתקציב המנורמל.
עבור ערך של תקציב נמוך למשל, יתקבל ערך 3, ונוכל להתאים לו 3 שחקנים שערך כל אחד מהם הוא 1.
עבור ערך תקציב ממוצע למשל, נניח 15, ייבחנו כל השלשות האפשריות (למשל שלשת שחקנים בעלי הערכים: 6,3,6 או 7,4,4 או 8,3,4 במטרה למצוא תחת אחת מהשלשות האלו את ערך הרווח הטוב ביותר).
ועבור ערך מקסימלי, יתקבל ערך 30 כך שנוכל לשבץ לסרט זה 3 שחקנים בעלי ערך 10.
הנרמול הנ"ל הוא רק לצורך בדיקת השלשות המתאימות, אך לא נשלח לרגרסור.

עבור כל אחד מבעלי המקצוע: כותב, במאי ומפיק נחפש את ערכם המספרי במאגר שבנינו בשלב קרצוף הנתונים (לכל אחד יש מאגר משלו), וננרמל אותו בהתאם לנרמול שביצענו לדוגמאות האימון. חיפוש אחר הנתון המספרי הוא במטרה להכין את נתוני הקלט שיתאימו לקלט שהרגרסור קיבל בשלב הלמידה על דוגמאות האימון.
בשלב זה, הכנו את כל הנתונים עבור הרגרסור, מלבד השחקנים.

כעת, ניסינו לחשוב על אפשרויות בהן נעבור על כמה שיותר אופציות על מנת להגיע לתוצאה כמה שיותר מדויקת אך עם התחשבות בזמן הריצה.
כפי שתיארנו עבור תקציב כלשהו, למשל תקציב בעל ערך מנורמל 6, נרצה לעבור על כל שלשות השחקנים שסכומם 6. אמנם כמות השלשות של המספרים שסכומם למשל 6 היא סופית, אך לכל שלשה כזו יש המון אופציות של בחירת שחקנים שיושבים תחת ערכים אלו.
למשל עבור [1,2,3] במצב אופטימלי (מצב בו נקבל באמת את השלשה עם ערך הרווח המקסימלי) נצטרך לעבור על כל שלשות השחקנים האפשריות שערכם 1,2,3, כך שכל שלשה תשלח יחד עם שאר הערכים המנורמלים לרגרסור ונבדוק מבין כל השלשות האפשריות מי נותנת את הרווח המקסימלי. כפי שניתן לתאר, בשל כמות האפשרויות הגדולה של שלשות, זה ייקח הרבה זמן.

<u>לכן חשבנו על מספר אופציות:</u>

- ✓ (תזכורת) **הפתרון הנאיבי** - לעבור על כל שלשות השחקנים האפשריות, לבדוק עבור כל שלשה האם מתאימה לתקציב, לאחר מכן מה הרווח שחוזה הרגרסור, ולהחזיר את השלשה שעומדת בתקציב וגם מחזירה את הרווח המקסימאלי. סיבוכיות הזמן של פתרון זה גבוהה מאוד ולכן ניסינו לחשוב על רעיונות נוספים לחיפוש השלשה.
- ✓ **חיפוש לוקאלי** – לאחר שחקרנו על החיפושים הלוקאלים, ניסינו להשתמש בחיפוש לוקאלי שלא מפתח את כל העוקבים אלא דוגם באופן אקראי שכנים, וכל עוד קיים שיפור ממשיך לנסות לשפר, ברגע שאין שיפור מפסיק את החיפוש.
הדבר באלגוריתם שלנו יתבטא בכך שעבור כל שלשה, למשל עבור השלשה [1,2,3] נדגום באופן אקראי שלושה שחקנים אחד מכל ערך בהתאמה, נשלח לרגרסור את הפרמטרים ונבדוק מהו ערך הרווח הצפוי. לאחר מכן ננסה לשפר את הרווח באמצעות החלפת שחקן בעל ערך 1 בשחקן אחר ובדיקת הרווח הצפוי. אם שיפרנו, נמשיך לדגום אקראית שחקן אחר בעל ערך 1, ברגע שלא שיפרנו, נחזור לשחקן בעל ערך 1 האחרון ששיפר ונעבור לשחקנים בעלי הערך 2. וכך נמשיך עד אשר נגיע לשחקן האחרון ששיפר בעל הערך 3. השימוש בחיפוש הנ"ל גזל המון זמן ריצה, וגם בשלב הניסוי הבנו שזה לאו דווקא הדרך הנכונה ביותר למציאת השילוב המיטבי ולכן החלטנו לוותר על אופציה זו.
- ✓ **חיפוש אלומה** – לבסוף חשבנו על חיפוש אלומה וניסינו להתאים אותו לאלגוריתם שלנו. ע"פ חיפוש אלומה, פיתוח מספר קבוע של מצבים "טובים" – K מצבים, כלומר נבדוק לכל שלשת שחקנים למשל עבור השלשה [1,2,3], K דגימות רנדומליות, ונחזיר את הערך המקסימלי מבניהן.

לאחר מספר ניסויים החלטנו כי חיפוש אלומה בעל K פיתוחים מגיע לדיוק מספק (כלומר מחזיר שלשת שחקנים עם ערך מספיק קרוב לערך הרווח של השלשה הטובה ביותר עבור התקציב הנתון) ואף אינו לוקח זמן ריצה רב.
על מנת להגיע לערך הK הנ"ל החלטנו לבצע ניסויים בדומה לשיטת החיפוש Anytime A*, כלומר הרצנו עבור פרמטרים מסוימים את האלגוריתם ה"בזבזני" בזמן, (המשתמש בחיפוש לוקאלי) ובחנו מהם השלשה ומהו ערך הרווח עבורם.
לאחר מכן התחלנו לבדוק את האלגוריתם המבוסס חיפוש אלומה כאשר ערך הK הנבדק היה ערך 10, בדקנו את הרווח שהתקבל לעומת הרווח שהתקבל באלגוריתם ה"בזבזני", וכך העלנו את ערך הK, כאשר ככל שערך הK עלה כך ערך הרווח התקרב לערך של האלגוריתם השני,
וזמן ההרצה גדל משמעותית.
לאחר מספר ניסויים כאלו (על פרמטרים שונים) החלטנו לקבל את הערך K=30.
כאשר ברוב הריצות התקבלה שלשה שקרובה עד כדי 90% בערך הרווח מערך הרווח המקסימלי המתקבל עבור השלשה המיטבית, ואילו זמן הריצה לא עלה על מספר דקות בודדות.

פרמטר נוסף שהכנסנו לאלגוריתם הוא בדיקה האם זוג מתוך שלשת השחקנים הנבדקים הופיעו יחדיו בסרט מצליח (לשם נתינת בונוס לשילוב השחקנים הנ"ל), כאשר סרט מצליח הוגדר כסרט שרווחו המנורמל מעל 1 (זהו ממוצע רווחי הסרטים המנורמל).
אם זוג שחקנים מופיע ביחד בסרט מצליח, אזי שלשת השחקנים מקבלת תוספת בונוס (שהוגדרה אצלנו תוספת של 3 נקודות לערכי השחקנים הכולל, לפני ממוצע) מכיוון שסביר שזוג שחקנים ששחקנו בעבר בסרט מצליח יעוררו עניין ויגרמו לסרט הנוכחי להצליח גם כן.

לסיכום - פסאודו קוד של האלגוריתם:

קלט: שמות המפיקים, כותבים במאים, תקציב יחד עם המסווג וסט דוגמאות המבחן.
פלט: שמות שלשת השחקנים האופטימאליים עבור התקציב המתאים עבורם הרווח מקסימאלי.

1. חישוב ערך הכותבים, מפיקים, במאים ע"פ מאגרי הנתונים של כל אחד מהם
2. נרמול התקציב עבור הרגרסור
3. נרמול התקציב כפי שהסברנו לעיל (לערכים מ3-30)
4. מעבר על כל שלשות המספרים האפשריות שסכומן שווה לערך התקציב המנורמל
    a. ביצוע K בדיקות רנדומאליות עבור כל שלשה
        i. בדיקה האם זוג שחקנים מהשלושה השתתפו בסרט מצליח
        ii. חישוב ערך ממוצע של שלושת השחקנים (בהתחשב בתוספת בונוס או לא)
        iii. שליחת הנתונים הללו לרגרסור, בחינת הrevenue בהתאם. אם התקבל ערך גבוה יותר של revenue לעדכן את שמות השחקנים החדשים עבור רווח זה.
5. החזרת השלשה בעלת הערך המקסימלי

הערה: ישנו אלגוריתם מעטפת שתפקידו לקבל רשימות של במאים, מפיקים ותסריטאים ולהמיר את ערכם לערך מספרי אותו האלגוריתם מקבל.

כיוונים למחקר עתידי

ישנם כיוונים נוספים לפתרון הבעיה אותם ניתן לבחון, למשל:

1. הוספת מידע על השחקנים (כתכונות נוספות לשיבוצם לסרט) - מציאת והוספת מידע נוסף על כל שחקן כמו גיל, מין, מס' הסרטים שהשתתף בהם, שפות נוספות שהוא דובר וכו' יכול לתת ערך נוסף לבחירת השחקן לסרט המבוקש, תחת דרישות הסרט, דבר שיביא לדיוק התאמת השיבוץ לסרט.

2. לעיתים פחות חשוב אם הערך המתקבל הוא מספרי או אם מחפשים ערך טוב יותר ביחס לערכים אחרים. שימוש במסווג מסוג classifier, עשוי לתת ערכים שנותנים אינטואיציה טובה יותר לפתרון הבעיה (במקום לקבל ערכים מספריים של חוזי של רווח של סרט, לקבל טווח/דירוג מתאים כלשהו) ובנוסף לתת ערכי דיוק טובים יותר עבור המסווג עצמו. במהלך התהליך, כאשר הגענו לבניית האלגוריתם של השלב השני ראינו שהמטרה שלנו לדעת להבדיל בין שיבוץ מסוים של שחקנים לאחר (כלומר מי קיבל רווח גדול יותר) והבחנו שנוכל לקבל זאת באמצעות מעקב אחר המגמה, ויכול להיות שclassifier היה מספיק לשם כך וייתכן כי היינו מקבלות תוצאות מדויקות יותר בשלב הראשון.

3. שימוש באלגוריתם שבנינו לבחירת במאי/מפיק לסרט – לבצע התאמות לאלגוריתם כך שבהינתן שחקנים, תקציב ותסריטאים והפיצ'ר הנוסף (במאי/מפיק) ניתן יהיה לחפש מפיק או במאי בהתאמה לסרט תוך מקסום הרווחים. כך נוכל לפתור בעיות נוספות או להתאים את האלגוריתם ליותר שימושים ממה שצפינו שיפתור בפועל.

4. שימוש במאגרי נתונים נוספים לשם הרחבת המידע הקיים – יתנו לאלגוריתם מאגר שחקנים, במאים, מפיקים ותסריטאים גדול יותר לשבץ ולבחור מתוכו ובנוסף יגדילו את הלמידה שתתבצע וכך את רמת הדיוק שלו.

5. שימוש בתקצירי סרטים ללמידה של תוכן הסרט ובביוגרפיה של שחקנים בעזרת שיטות NLP מתקדמות ומתן פיצ'רים מעניינים נוספים עבורם לשם התאמה מיטבית של שחקנים לתוכן הסרט ולסרטים קודמים מוצלחים שהשתתפו בהם.

סיכום

מטרת הפרוייקט הייתה בניית אלגוריתם לשיבוץ שחקנים עבור סרט בצורה המיטבית.
בתחילת הפרוייקט, חשבנו איך להתמודד עם הבעיה בכל מיני דרכים.
בשלב הראשון חילקנו את העבודה ל2 מקטעים גדולים –

1. <u>למידת הרגרסור ממאגר הנתונים והתכונות שניתנו לו</u>
   שלב זה לקח את עיקר העבודה, בעיקר של הצורך לסדר את מאגרי הנתונים בהם השתמשנו בצורה המיטבית ללמידת המסווג, ובשל הרצון לבחור את התכונות שיתנו לנו את הדיוק הטוב ביותר. שלב זה כלל ניסויים רבים, שאת רובם הצגנו כאן.
2. <u>בניית האלגוריתם בהסתמך על הרגרסור</u>
   בשלב זה ניסינו לחשוב על דרכים שונות על מנת להגיע לתוצאה מיטבית אך עם זאת לשמור על זמן ריצה הגיוני וריאלי.

מכיוון שמטרתנו בחלק הראשון הייתה מציאת רווחים צפויים מסרט, פנינו לעולם חדש של אלגוריתמי למידה שתפקידם לחזות ערכים מספריים ממשיים (בניגוד למסווג שזה מה שראינו עד עכשיו) – רגרסורים.
בהתחלה הופתענו לגלות שממוצע אחוזי הדיוק היה רחוק מאוד ממה שצפינו אך לאחר שחקרנו וביצענו מספר ניסויים הבנו את המשמעות של הסטיות ואת הצורך לאמוד את הדיוק ע"פ ערכים שונים, ובנוסף להסתמך על מגמות הגרפים של הactual אל מול הpredicted.

קשיים שנתקלנו בהם לאורך העבודה

מרכז העבודה הגדול היה בקרצוף הנתונים והכנתם לרגרסור בפרט ולאלגוריתם העיקרי בכלל.
נתקלנו בקשיים מכיוון שהDB שהשתמשנו בו לא היה זהה בכולו, למשל בערך ה-crew עבור הwriter הופיעו שמות רבים כדוגמת "screenplay", "author", "story", "writer" מה שגרם בתחילה להרבה אפסים בערך זה, לכן חיפשנו את השמות השונים למקצוע זה.
כמו כן גם ערך הrevenue והbudget היו חסרים בהרבה סרטים יחסית ולכן בנינו script שיחפש בוויקיפדיה וישלים את הערכים החסרים עבור ערכי הrevenue והbudget.
script זה אכן מילא את החלקים החסרים, אך בחלקם שם ערכים לא הגיוניים ונכונים, מה שגרם בתחילה לשגיאות ברגרסורים וערכי שגיאה גבוהים מאוד.

בנימה אישית,

בתחילת העבודה היה לנו קשה מאוד למצוא נושא שמעניין אותנו ונוגע בנו מספיק כדי שנרצה לפתח משהו סביבו. הכיוונים ההתחלתיים שלנו היו דילמות מעניינות כמו:
"מה הסיכוי, בהינתן אסיר משוחרר עם מידע עליו שהוא יחזור לכלא לאחר שריצה את עונשו הנוכחי?".
הקושי העיקרי שנתקלנו בו בהתחלה הוא שלכל כיוון מקורי כזה, היה חוסר גדול במידע זמין ואיכותי מספיק כדי לבצע תהליך של למידה דבר שלצערנו, חסם אותנו מהתקדמות בכיוונים אלו.
בשלב הבא ניסינו לקחת משהו שהוא אמנם פחות מקורי כמו נושא הסרטים, אך שידוע שקיים לו מידע רב וזמין, ועברנו לחשוב איך אנחנו הופכות את הנושא הלא-מקורי הזה למעניין.
בעזרת גיא, גיבשנו את הרעיון של הפרוייקט שביצענו והתחלנו לרוץ על זה.
במהלך הדרך נתקלנו בהמון קשיים, מאופן סידור המידע ועד לגיבוש הרעיון הכללי של הפרוייקט שהשתנה לאורכו בהתאם לניסויים ולמסקנות שהסקנו.
היו ניסויים שלקח לנו המון זמן להריץ ובסוף גילינו שהכיוון שחשבנו עליו לא באמת עזר או שיפר את הלמידה של הרגרסור כפי שהיינו רוצות, אבל בו בזמן גם הבנו שזה חלק מהתהליך.
התהליך היה מעניין מאוד, במיוחד בחלק הניסויים שבו יכלנו לראות כיצד מה שלמדנו בקורס בינה מלאכותית בא לידי ביטוי באופן מעשי ועוזר בפתרון בעיות מהחיים.
בנוסף, נחשפנו לספריות רבות של עבודה בpython, שהן שימושיות מאוד גם לפרוייקטים שאינם קשורים לבינה ואני בטוחה שנשוב להשתמש בהן כעת לאחר שהכרנו אותן.
הבנו לעומק מתי נעדיף להשתמש במסווג ומתי ברגרסור, גילינו שמאוד קשה לחזות ערכים מדוייקים של מספרים (במקרה שלנו צפי של רווחים) ושגם אם ישנה בעיה שנראה על פני השטח שצריך רגרסור כדי לפתור אותה, כדאי גם לנסות ולחשוב איך אפשר להמיר אותה לבעיית מסווג, והאם זה נוכל בדרך זו גם לפתור את הבעיה.

ביבליוגרפיה

- המושג קולנוע (ויקיפדיה).
- מאגר נתונים הנתון בקובץ csv. של 4800 סרטים ומאפיינים עליהם כולל שחקנים הלקוח מאתר Kaggle - https://www.kaggle.com/tmdb/tmdb-movie-metadata/data
- מאגר נתונים הנתון בקובץ csv. של 2500 סרטים ומאפיינים עליהם כולל ז'אנר, חברה מפיצה, רווחים (Box office) הלקוח מאתר Kaggle - https://www.kaggle.com/yjeong5126/box-office-data-20172019
- השלמת מידע חסר למאגר הנתונים – ויקיפדיה, IMDB.
- מידע על האלגוריתמים:
  - Random forest
  - עצי החלטה
  - KNN, רגרסור לינארי - ויקיפדיה