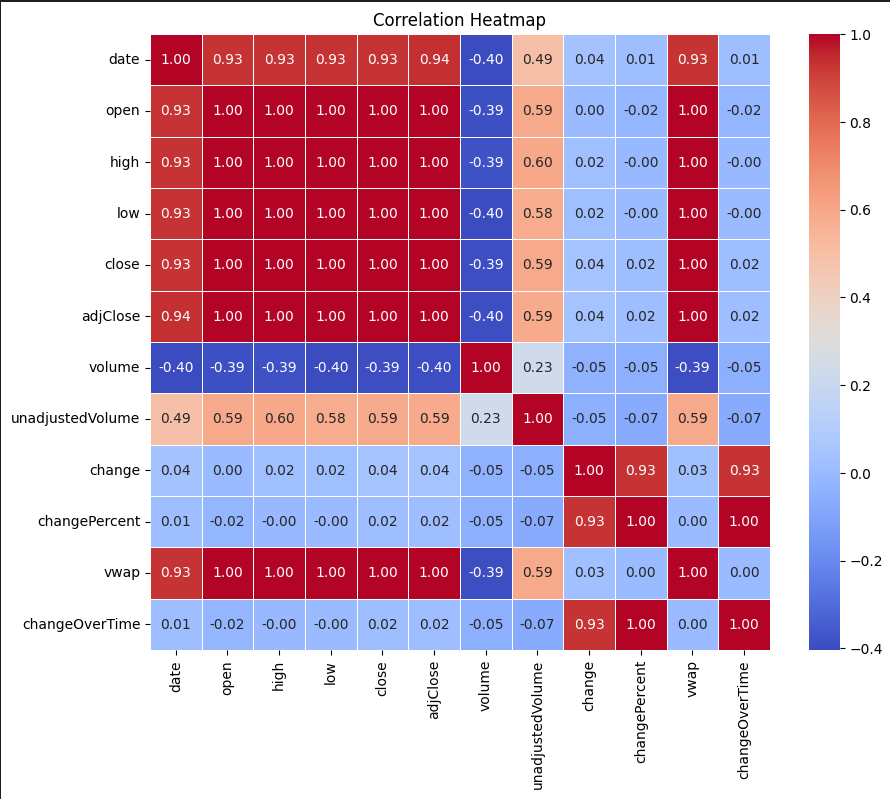
|  |  |  |
| --- | --- | --- |
| מדעים דיגיטליים להיי-טק | | |
| פרויקט מס' \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  פרויקט גמר - **XXX** | | |
| שם הפרויקט: S&P 500 Predictions | | |
| מבצעים: | | |
|  | שמות: גלעד שפיר, רועי ניר | ת"ז: 319092235, 208847459 |
| מקום ביצוע הפרויקט: אוני' תל אביב, עבודה מרחוק | | |
| לשימוש המנחה: דרור יעקובי  הנני מאשר את תכנית העבודה המצורפת והלו״ז המעודכן    חתימה: | | |
| ***תוכן עניינים***   |  |  | | --- | --- | | ***נושא*** | ***עמודים*** | | תקציר | 3 | | רקע על המנייה הנבחרת ומטרתו הסופית | 4 | | יצירת מסגרת נתונים וחיבור ממשקים | 5 | | חקר הנתונים | 5 - 6 | | עיבוד מקדים של הנתונים | 7 - 9 | | הפחתת מימדיות | 10 - 14 | | הגדרת תווית + פונקציות עזר | 15 - 19 | | MLP | 20 - 27 | | SVM | 27 - 28 | | Random Forrest | 29 - 32 | | RNN+ | 32 - 39 | | בחירת מודל מיטבי | 39 | | ביצוע תחזית | 39- 40 | | ניתוח תוצאות ומסקנות + הצעות לפיתוח עתידי של הפרויקט | 41 - 42 |   ***תקציר***  **מטרה:** מטרתו העיקרית של הפרויקט שלנו היא חיזוי מניות בשווקים פיננסיים. בניגוד לניתוח מניות רגיל העוסק בעיקר בניתוח גרפים, קריאת דוחות כספיים או עדכונים חדשותיים, אנו מתמקדים בבסיסי הנתונים המתארים את התנהגות המניות בנושאים כגון מחיר פתיחה, ווליום מסחר וכו'.  **ביצוע:** הבנו שישנם גורמים רבים המשפיעים על המנייה מלבד הנתונים השטחיים שלה. נרצה לבדוק את המאפיינים והנתונים הרלוונטיים שיכולים לעזור לחיזוי המניה. לכן בשלב הראשון, נרצה לאגד מידע הרלוונטי לשווקים של מניות שונות, ולהבין מהו אותו מידע רלוונטי. כדי לבצע חיזוי כזה עלינו להסתכל על המדד שאליו המנייה שייכת. לכן, בשלב המקדים נרצה למצוא אלגוריתם שיסנן ויבחר את הדאטה והמאפיינים הרלוונטים של מניות. נתמקד במדדים וסקטורים אליהם שייכות המניות שבחרנו לחזות. **המדד** משקף את רמת המחירים הממוצעת של קבוצות מסוימות או ענפים מסוימים של מניות הנסחרות בבורסה לניירות ערך. המדד מחושב מתוך מחירי המניות, בדרך כלל על ידי חישוב ערכם המשוקלל. המדד משמש משקיעים הן להערכת רמת המחירים של ענף או קבוצה מסוימים, והן כבסיס להערכת ביצועים למדידת ביצועים של תיקי השקעות להשוואה של ביצועים אלה בטווחי זמן שונים. **הסקטור** ה'וא מגזר מניות מצומצם יותר המתייחס לקבוצת חברות הפועלות באותו ענף או עסק. לעתים קרובות משקיעים מקצים את השקעותיהם על פני מגזרים שונים כדי לגוון את תיקי ההשקעות שלהם ולהפחית סיכונים. בשלב הבא נרצה לבצע **חיזוי** לערך המניה, ונתרכז באי הוודאות של המודל ביחס למניה. בנוסף, נרצה למצוא את המדד ושיטה למדידת אי הוודאות.  **שיטות:** השתמשנו באלגורתמים של ML עדכניים של מודלים סטטיסטיים לניתוח סדרות בזמן (למשל MLP, RNN, Random Forest ועוד).  **דאטה:** בפרויקט שלנו בחרנו להתמקד במדד s&p 500: זהו מדד ניירות ערך הכולל מניות של 500 תאגידים, רובם אמריקאיים. הסיבה שבחרנו במדד זה היא המבנה שלו. אלו הן מניות של חברות ציבוריות גדולות הנסחרות בבורסות אמריקאיות ממגוון רחב של סקטורים, דבר שבעינינו יכול לייצג בצורה טובה מאוד את המצב הכללי במשק ולהיות שימושי בעת חיזויים עתידיים על מדדים או מניות שונים. בפרויקט ביצענו התמקדות נוספת במנייה של חברת אפל (החברה הגדולה ביותר במדד). הסיבה לכך היא שאנו חושבים שעקב משקלה הרב במדד, הניתוח שלה יכול לספק תובנות נוספות לגבי התנהגות של מניות נוספות.  **מסקנות:** המסקנה העיקרית היא שקשה לחזות את התנהגותן של מניות לאורך זמן. זאת הסיבה שעד כה הוקמו מספר גדול של סטארטאפים מלאים באנשי מקצוע מוכשרים ומנוסים בתחום, אך אף אחד עוד לא הצליח לפצח את השוק לחלוטין. בהינתן המידע המצומצם שברשותינו, הצלחנו להגיע לרמת דיוק של 78%, וזאת באמצעות אלגוריתם של למידה עמוקה בשם MLP, וזאת לאחר שהשווינו אותו אל מול מספר לא מבוטל של מודלים ושיטות חיזוי שונות. סך הכל התוצאות שלנו תאמו את הציפיות. ראינו שככל שאנו נשענים על יותר ערכים מהעבר או מגדילים את ערכי הסף אנו מקבלים רמות דיוק גבוהות יותר לחיזויים שלנו על סט המבחן וכמו כן רמת הדיוק יורדת ככל שאנו מנסים להסתכל קדימה יותר אל העתיד. אנו מרוצים מתוצאות הפרויקט וסבורים כי הן יכולות להיות שימושיות למשקיעים שמאמינים בניתוחים גרפיים (יותר מאשר משקיעים שמסתמכים על היכרות עם החברה או ידיעות חדשותיות או פיננסיות). מסקנה נוספת היא היציבות היחסית של מניות גדולות אל מול מניות קטנות, וזו הייתה אחת הסיבות שבחרנו להתמקד במניית אפל מתוך המדד. קישור לגיט של הפרויקט: <https://github.com/GiladShappir/High-Tech-for-science-project/tree/main>  ***רקע על המנייה הנבחרת ומטרתו הסופית***  ***רקע:***  אפל בע"מ (Apple) היא [תאגיד רב-לאומי](https://he.wikipedia.org/wiki/%D7%AA%D7%90%D7%92%D7%99%D7%93_%D7%A8%D7%91-%D7%9C%D7%90%D7%95%D7%9E%D7%99) [אמריקאי](https://he.wikipedia.org/wiki/%D7%90%D7%9E%D7%A8%D7%99%D7%A7%D7%A0%D7%99), המעצב ומפתח מוצרי [אלקטרוניקה](https://he.wikipedia.org/wiki/%D7%90%D7%9C%D7%A7%D7%98%D7%A8%D7%95%D7%A0%D7%99%D7%A7%D7%94), [תוכנות מחשב](https://he.wikipedia.org/wiki/%D7%AA%D7%95%D7%9B%D7%A0%D7%AA_%D7%9E%D7%97%D7%A9%D7%91), [שירותי בידור וצריכה](https://he.wikipedia.org/wiki/%D7%A1%D7%98%D7%A8%D7%99%D7%9E%D7%99%D7%A0%D7%92) ועוד. אפל נסחרת בנאסד"ק: AAPL, והיא הייתה החברה הראשונה שהגיעה לשווי שוק של מעל 3 טריליון דולר.  המוצרים המוכרים ביותר של החברה הם קו המחשבים מקינטוש, סדרת הטלפונים החכמים אייפון, מותגי האוזניות AirPods ו-Beats, שירות המוזיקה אפל מיוזיק, חנות האפליקציות App Store וסדרת מחשבי הלוח אייפד. מוצרים ידועים אחרים הם סדרת נגני המולטימדיה אייפוד, שירות המוזיקה הוותיק אייטונס, שירות הענן iCloud, תוכנת שיחות הווידאו FaceTime, השעון החכם Apple Watch, שירות הצפייה הישירה Apple TV+ והתוכנות המקצועיות Logic Pro ו-GarageBand. מערכות ההפעלה הידועות שנוצרו בפיתוחה של החברה הן מערכת ההפעלה iOS למכשירים ניידים כמו אייפון ואייפד, מערכת ההפעלה macOS (לשעבר OS X) למחשבי המקינטוש, ומערכות ההפעלה watchOS ו-tvOS.  אפל מחזיקה תחתיה רשת [קמעונאות](https://he.wikipedia.org/wiki/%D7%A7%D7%9E%D7%A2%D7%95%D7%A0%D7%90%D7%95%D7%AA) בשם [Apple Store](https://he.wikipedia.org/wiki/Apple_Store) . נכון לפברואר 2021, יש לה 511 חנויות. ב-25 מדינות ברחבי העולם וברבעון האחרון של שנת 2015 ביקרו בחנויות ברחבי העולם כ-120 מיליון אנשים. כמו כן, החברה מפעילה חנות מקוונת באתר האינטרנט שלה. נכון ל-2020, לחברה יש 147,000 עובדים ברחבי העולם.  החברה הגיעה לשווי שוק של טריליון דולר בשנת 2018, ב-19 באוגוסט 2020, הגיעה לשווי שוק של מעל שני טריליון דולר, ובינואר 2022 הגיעה החברה לשווי שוק של מעל שלושה טריליון דולר. החברה הראשונה בהיסטוריה שהגיעה לשווי זה.[2] בתאריך 11 במאי 2022 החברה הודחה מהתואר החברה עם שווי שוק הגדול בעולם על ידי חברת אראמקו הסעודית. אפל היא אחת מחמש חברות הביג טק - חברות טכנולוגיית המידע הגדולות ביותר בארצות הברית.  ***למה בחרנו בה:***   1. **הובלת השוק וחדשנות:** Apple היא שחקן משמעותי בתחומי הטכנולוגיה ואלקטרוניקה. לחברה יש היסטוריה עשירה ומלאה בחדשנות, הכוללת הצגה של מספר מוצרי דגל (כגון אייפון, אייפד, מקינטוש ועוד). למידה על החברה יכולה לספק תובנות לגבי האופן שבו מובילי השוק מניעים חדשנות בתחום, מתאימים עצמם לשינויים בדרישות הלקוחות ומשמרים יתרון תחרותי בשוק. 2. **השפעה כלכלית ושטח גלובלי:** הפעילות של Apple מתרחשת ברחבי העולם ומשפיעה על כלל הכלכלה הגלובלית, כולל שרשראות אספקה, יצור וקמעונאות. ניתוח של מניית אפל יכול לעזור לנו להבין את המורכבות של השוק הגלובלי בתחומים אלו העוזרים להצלחת החברה. זוהי גם הזדמנות גדולה לחקור את ההשפעות הכלכליות והחברתיות של החברה על השווקים בכל רחבי העולם. היות והחברה היא בעלת המשקל/נתח הגדול ביותר במדד שבו אנו מתמקדים במחקר שלנו (למעלה מ-7% מהמדד נכון לרגע כתיבת ספר הפרויקט), אנו סבורים כי הבנת התנהגות המנייה הזו תתרום רבות להבנת התנהגות המדד כולו. 3. **אסטרטגיות שיווק ובניית מותג** Apple :ידועה ביכולות השיווק והמיתוג שלה. המותג של החברה הוא שם נרדף לאיכות, עיצוב וחוויית משתמש. חקר המנייה והחברה עצמה תורמים להבנתנו לגבי אופי החברה והתנהלותה בתחומים אלו.   ***מטרה סופית:***  באמצעות פרויקט זה נרצה להיות מסוגלים לספק תמיכה לקהל משקיעים בשוק ההון על כל צדדיו (גם משקיעים על בסיס יומיומי וגם משקיעים ארוכי טווח). תהליך האופטימיזציה של הפרויקט שלנו מאפשר פנייה לכל קהל המשקיעים בהינתן סט הנתונים הניתן ובחירת ההיפר-פרמטרים במודלים שלנו. אנו סבורים כי המשך עבודה על יצירת ממשק אינטראקטיבי לשימוש בפרויקט שלנו (client-server) יכול לשמש משקיעים נוספים בעתיד בתהליך הדומה לניתוח גרפי של המנייה בה המשקיע מעוניין להיכנס.  ***יצירת מסגרות נתונים***  תחילה יצרנו data frames של המנייה של אפל ע"י בקשת api מאתר המכיל מידע יומיומי עדכני לגבי המניה. הלולאות בתמונה מתארות את יצירת הדאטה עבור כל מנייה ושמירת המידע כמסגרת נתונים באמצעות ספריית pandas. על מסגרות נתונים אלו נעבוד בהמשך הפרויקט, בהתמקדות על מניית אפל.  ***חיבור בין ממשקים***  ***בכדי לעבוד ביחד על הקוד מכמה מקומות שונים, יצרנו סביבת קוד באמצעות הקולאב של גוגל. עדכנו את הקוד דרכו ועבדנו עליו יחד. לצורך הבאת המוצר הסופי, ייבאנו את הקוד מהפלטפורמה הזו חזרה ל-VSCode. כאן מוצג הקוד שבאמצעותו ביצענו את המעבר.***  ***אובזרוציה ראשונית על הנתונים ומשמעות הפיצ'רים הקיימים***    התמונה הראשונה מראה כיצד נראית תצפית התחלתית בדאטה לפני שהתחלנו את העבודה עליו.  התמונה השנייה נותנת הסבר קצר על כל אחד מהפיצ'רים הקיימים בדאטה.  משמעות הפיצ'רים מראה על מולטי קולינאריות הקיימת בין כמה מהם. נטפל בבעיה זו בחלק מאוחר יותר של הפרויקט.    ***ויזואליזציה*** | | |

משום שהניתוח שלנו מתבסס על מחיר המנייה, רצינו לראות את התפלגותו. אמנם משתנה החיזוי שלנו יהיה מחיר הפתיחה של המנייה, אך מובן לנו שניתן גם להפיק מידע רב ממחיר הסגירה של המנייה (באותו יום המסחר) ובפרט מניתוח הקשר בין שני הפיצ'רים הללו.  
בגרפים אלו אנו מציגים כיצד שני אלו התנהגו לאורך השנים (מצד שמאל) ולאורך מספר חודשים (מימין).

­­***קורלציות***

השלב הבא היה להבין את הקשר בין הנתונים השונים בדאטה שלנו. כבר כשראינו את רשימת המשתנים שבידינו, שמנו לב כי ייתכן וניתקל בבעיית מולטי קולינאריות. מולטיקולינאריות היא קורלציה גבוהה של המשתנים הבלתי תלויים (לא משתנה החיזוי). היא עלולה להוביל לתוצאות לא אמינות ולא יציבות של מקדמים במודלים שבהם נשתמש בהמשך. הסיבה לחשש מבעיה זו במקרה שלנו היא אופן יצירת חלק מהמשתנים שלנו. משתנים כמו ChangeOverTime ו- ChangePercent הם למעשה 2 דרכים שונות להצגת אותו המידע/נתון. את הבעיה הזו אנו פותרים בשלב מאוחר יותר של הפרויקט, אך כעת רצינו להכיר בה ולראות עד כמה היא נוכחת בנתונים שלנו.



תמונה שמכילה טקסט, צילום מסך, גופן, תפריט

התיאור נוצר באופן אוטומטי

בצד ימין ניתן לראות את מפת החום המתארת את הקורלציות בין המשתנים השונים שלנו, בעוד בצד שמאל ניתן לראות את הקורלציה של כל אחד מהמשתנים באופן ישיר מול משתנה החיזוי open. בנוסף, ניתן להבחין ברשימת 3 המשתנים הטובים והחשובים ביותר לחיזוי לפי מדד הקורלציה.

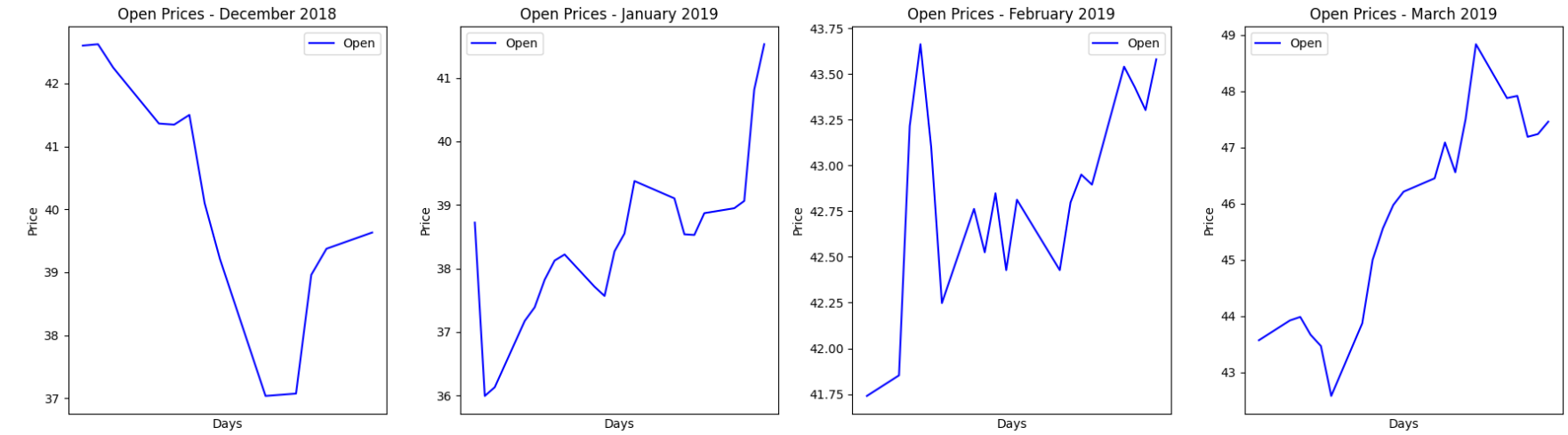
עיבוד מקדים

**עיבוד נתונים מקדים** הוא מונח המתייחס לשלב הקודם לניתוח הנתונים, שמטרתו לשפר את יכולות הניתוח והסקת המידע מהנתונים באמצעות הפעלת כלים המשנים חלקים מסוימים במידע ובאמצעות הסרה של חלקי מידע לא נחוצים. שלב עיבוד הנתונים המקדים הוא שלב חשוב של [הכנת נתונים](https://he.wikipedia.org/wiki/%D7%94%D7%9B%D7%A0%D7%AA_%D7%A0%D7%AA%D7%95%D7%A0%D7%99%D7%9D) בתהליך [כריית מידע](https://he.wikipedia.org/wiki/%D7%9B%D7%A8%D7%99%D7%99%D7%AA_%D7%9E%D7%99%D7%93%D7%A2).

בחלק זה נצלול אל תוך הנתונים וננסה להתמודד בצורה מיטבית עם חריגים/אנומליות, ננרמל את הדאטה לצורך הפעלת אלגוריתמים של למידת מכונה בהמשך, נתמודד עם נתונים חסרים במידה ויש, נטפל במשתנים קטגוריאליים, נוסיף משתנים חשובים חדשים אל תוך הדאטה ולבסוף נקטין את מימדיות הבעיה כדי לקבל פתרון אופטימלי בטרייד-אוף המתרחש בין מורכבות המודל לבין הטיה בתוצאות המודל כתוצאה ממודל פשטני מדי. נעשה זאת תוך כדי ניסיון לשמר את יכולת ההסברה של המודלים שלנו כמה שניתן.

בחלק זה אנו מבצעים חלוקה לסט אימון וסט מבחן כדי שנוכל לאמן את לקראת למידת המכונה הלא מונחית שנעשה בהמשך. משום שאנו רוצים לבצע חיזוי בזמן, לא נעשה את החלוקה בצורה רנדומלית, אלא בצורה כרונולוגית (משום שהתנהגות מנייה אמורה להיות רנדומלית, דרך זו לא תפגע בביצועי המודלים שלנו).

גם לאחר הפיצול, אנו סבורים כי סט האימון שלנו גדול מספיק כך שיהיה מייצג ויוכל לעזור לנו לקבל תחזיות מהימנות לגבי מחיר הפתיחה של המנייה בעתיד (האם יעלה, ירד או יישאר בתוך טווח מסוים שאותו נקבע בשלב מאוחר יותר של העבודה).

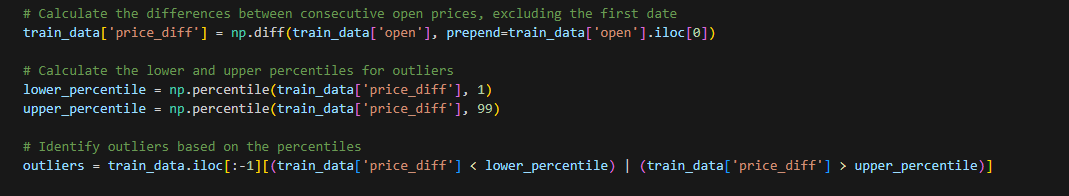


כעת, הפכנו את משתנה התאריך מסוג מחרוזת לסוג DayTime לצורך ניתוח מעמיק והסקת מסקנות בהמשך. כמו כן, רצינו להציג כיצד משתנה החיזוי שלנו מתפלג בצורה חודשית, וזאת כדי להשוות את צורת ההתפלגות המקורית שלו לזו שנקבל לאחר שנתמודד עם ערכים חריגים.

***התמודדות עם חריגים***

בחלקים קודמים של הפרויקט (באמצעות היסטוגרמות) ראינו כי לעיתים יש קפיצות חריגות בהתפלגות משתנה החיזוי שלנו. משום שישנם דברים המשפיעים על החיזוי שלנו שנמצאים מעבר ל-scope של הנתונים שברשותינו (כמו דוחות רבעוניים, משברים פוליטיים וכלכליים, אינפלציה וכו'), נרצה למצוא דרך להתמודד עם תצפיות כאלו כך שלא ישפיעו מעבר לרצוי על תוצאות המודלים שלנו.

דרך ההתמודדות שלנו תהיה שינוי של האחוזון התחתון והעליון של התצפיות. נעשה זאת במספר שלבים:

1. זיהוי האחוזון העליון והתחתון של התצפיות (חריגים) באמצעות np.diff והגדרתם
2. "החלקה" של חריגים באמצעות שיטת Moving Average

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, קו, תרשים, עלילה

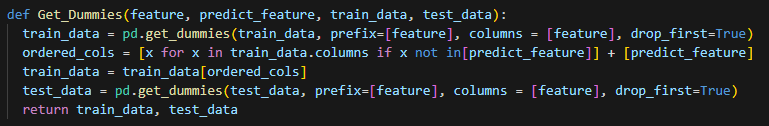
התיאור נוצר באופן אוטומטילמעשה, זיהוי החריגים שלנו היה זיהוי של השינויים החדים ביותר (גם כלפי מעלה וגם כלפי מטה) במחירי הפתיחה של המנייה יום לאחר יום. לאחר שזיהינו את אותם החריגים, בחרנו להחליק אותם באמצעות שימוש בערכי תצפיות מהימים שקדמו לאותו יום חריג ובחלוקת משקל שונה לכל תצפית (ככל שהתצפית יותר קרובה ליום הבעייתי, משקלה והשפעתה על ההחלקה גדל). כאמור, שינינו מספר קטן יחסית של תצפיות (רק 2%) ולכן נצפה לראות כעת שינויים מינוריים בהתפלגות מחירי הפתיחה לאורך זמן:

גם מהסתכלות על חלק מהחודשים שהצגנו, ניתן לראות שההתפלגות מעט שטוחה יותר ושטווח הערכים בגרף הצטמצם מעט. כתוצאה מכך, בגרפים ישנו כעת קו "חלק" יותר.

***התמודדות עם משתנים חופפים***

כפי שציינו קודם, קיימת חפיפה בין חלק מהמשתנים בנתונים שלנו. בחלק זה של הפרויקט ננסה לפתור את הבעיה הזו. דרכי ההתמודדות הללו נועדו להורדת מורכבותם של המודלים שבהם נשתמש בהמשך תוך שמירה על יכולת הסברה מירבית של משתנה החיזוי.

הפונקציה הבאה לוקחת משתנים קטגוריאליים והופכת אותם למשתני דמי (+שימוש על המשתנה date).



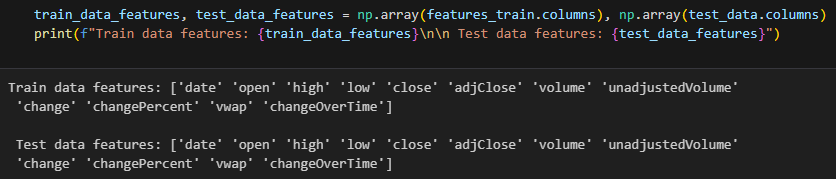
תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

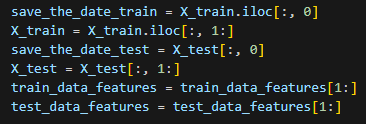
לפני שנוריד את מימדיות הבעיה וננרמל את הנתונים, אנו מחלקים את הנתונים למשתנים ולחיזוי עבור סט האימון וסט המבחן ומוודאים שסט הנתונים שלנו נראה כמו שרצינו:

תמונה שמכילה טקסט, צילום מסך, גופן

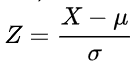
התיאור נוצר באופן אוטומטי



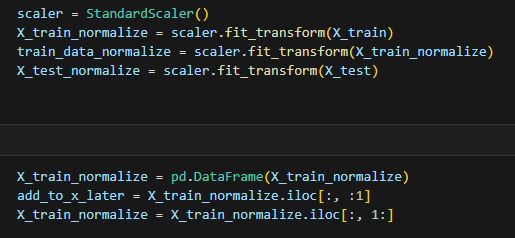
כעת, נוציא מסט הנתונים של האימון שאנו עובדים עליו את עמודת התאריך ונשמור אותה בנפרד (בחרנו בדרך פעולה זו משום שהניתוח שלנו הוא על פני זמן, לכן רצינו להימנע מהורדת משתנה זה בהורדת המימדיות וכמובן שאין צורך לנרמל אותו). אחרי שנסיים את שלב הניתוח המוקדם ולפני השימוש במודלים שלנו, נכניס חזרה את המשתנה date אל תוך סט הנתונים שלנו.



***סטנדרטיזציה של הנתונים***

נורמליזציה של נתונים היא תהליך של הפיכת נתונים לטווח משותף (common range),המאפשר השוואות מדויקות יותר בין נקודות נתונים (data points). סטדנרטיזציה היא תהליך נרמול הנתונים לפי התפלגות נורמלית סטנדרטית (ממוצע שווה ל-0 וסטיית תקן שווה ל-1)

בנתונים שלנו, חלק מנורמלים וחלק לא (לפי הגדרתם). כדי שנוכל להציג את כל הנתונים בהם נשתמש במודלי ה-ML שלנו באופן דומה בשלב מאוחר יותר, נרצה להפעיל תהליך נרמול. תהליך זה יבטיח לנו ביצועים טובים יותר ותוצאות טובות יותר. תהליך זה למעשה "מנקה" נתונים בעייתיים ומשוכפלים כדי להבטיח מבנה לוגי לדאטה סט שלנו.



***תהליך הורדת מימדיות***

ב[סטטיסטיקה](https://he.wikipedia.org/wiki/%D7%A1%D7%98%D7%98%D7%99%D7%A1%D7%98%D7%99%D7%A7%D7%94), [למידת מכונה](https://he.wikipedia.org/wiki/%D7%9C%D7%9E%D7%99%D7%93%D7%AA_%D7%9E%D7%9B%D7%95%D7%A0%D7%94) [ותורת האינפורמציה](https://he.wikipedia.org/wiki/%D7%AA%D7%95%D7%A8%D7%AA_%D7%94%D7%90%D7%99%D7%A0%D7%A4%D7%95%D7%A8%D7%9E%D7%A6%D7%99%D7%94), תהליך **הורדת ממד** (dimensionality reduction) הוא תהליך של צמצום מספר משתנים מתוך כלל המידע, על מנת לשמר כמה שיותר מהמשמעות של המידע עצמו.  
ישנן 2 שיטות לביצוע פעולה זו:

1. **בחירת מאפיינים** (Feature selection)- גישה בה ננסה לבחור תת-קבוצה של משתנים מתוך כלל המשתנים, לתת הקבוצה הזאת נקרא מאפיין (feature). המטרה בגישה זאת היא לבחור את המאפיינים שמייצגים בצורה אופטימלית את המידע.
2. **הטלת מאפיינים** (feature projection) - בגישה זו נמיר את המידע על ידי המרה של מידע המאופיין כמרחב בעל [ממדים](https://he.wikipedia.org/wiki/%D7%9E%D7%9E%D7%93_(%D7%9E%D7%AA%D7%9E%D7%98%D7%99%D7%A7%D7%94)) רבים (high-dimensional space) למידע המיוצג במרחב עם פחות ממדים. ההמרה יכולה להיות המרה ליניארית כמו בשיטת [ניתוח גורמים ראשיים](https://he.wikipedia.org/wiki/%D7%A0%D7%99%D7%AA%D7%95%D7%97_%D7%92%D7%95%D7%A8%D7%9E%D7%99%D7%9D_%D7%A8%D7%90%D7%A9%D7%99%D7%99%D7%9D) (PCA), אך לא בהכרח.

אנחנו נציג אלגוריתם אחד מכל שיטה ונשווה ביניהם באמצעות MSE. אנו עושים זאת משום שבמקרים מסוימים, סוגים בסיסיים של ניתוח נתונים, כמו [סיווג](https://he.wikipedia.org/wiki/%D7%A1%D7%99%D7%95%D7%95%D7%92_(%D7%A1%D7%98%D7%98%D7%99%D7%A1%D7%98%D7%99%D7%A7%D7%94)) או [רגרסיה](https://he.wikipedia.org/wiki/%D7%A8%D7%92%D7%A8%D7%A1%D7%99%D7%94_(%D7%90%D7%A0%D7%9C%D7%99%D7%96%D7%94)) יתבצעו בצורה יותר מדויקת על מידע שעבר הורדת ממדים מאשר על המידע המקורי.

להורדת המימדיות של הבעיה ישנם מספר יתרונות עיקריים:

1. הורדת הנפח הנדרש לאחסון המידע
2. הורדת מידע עודף מובילה לביצועים טובים יותר של אלגוריתמי ML
3. קל יותר להציג ולהבין ויזואלית את המידע כאשר המימדיות נמוכה יותר

עבור שיטת בחירת המאפיינים נבחר באלגוריתם בחירה קדימה, ועבור שיטת הטלת מאפיינים נבחר באלגוריתם PCA. בתקווה, אחד מהם ייתן MSE נמוך יותר וכן גם מספר מימדים נמוך יותר ובו נבחר בסוף התהליך.

**בחירה קדימה**

בחירה קדימה היא שיטה איטרטיבית שבה אנו מתחילים בכך שאין תכונה במודל. בכל איטרציה, אנו ממשיכים להוסיף את התכונה אשר משפרת בצורה הטובה ביותר את המודל שלנו עד שתוספת של משתנה חדש אינה משפרת את ביצועי המודל.

שיטת ה"ענישה" שלנו, שבה נעריך את מספר הפיצ'רים, תהיה C\_p של מלו.

בסטטיסטיקה, Cp של Mallows משמש להערכת ההתאמה של מודל רגרסיה שהוערך באמצעות ריבועים קטנים רגילים. הוא מיושם בהקשר של בחירת מודל, שבו מספר משתני חיזוי זמינים לחיזוי תוצאה כלשהי, והמטרה היא למצוא את המודל הטוב ביותר הכולל תת-קבוצה של מנבאים אלה. ערך קטן של Cp פירושו שהמודל מדויק יחסית.

כדי להעריך את מודל הבחירה קדימה באמצעות הענשה בשימוש Mallow's Cp נשתמש בנוסחאות הבאות:

תמונה שמכילה טקסט, גופן, צילום מסך, שחור

התיאור נוצר באופן אוטומטי

למעשה, כך נראית פעולת האלגוריתם בשימוש ברגרסיה לינארית:

תמונה שמכילה טקסט, תרשים, קו, אוריגמי

התיאור נוצר באופן אוטומטי

לצורך הפעלת האלגוריתם נשתמש בפונקציות הבאות על סט האימון שלנו:

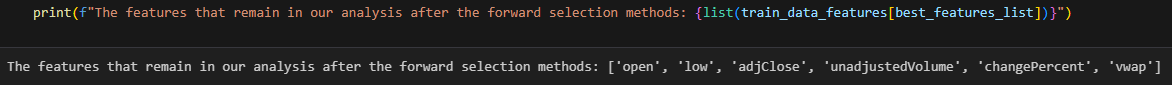
תמונה שמכילה טקסט, צילום מסך

התיאור נוצר באופן אוטומטי

כמו כן, אנו מדפיסים את רשימת הפיצ'רים הטובים ביותר בהתאם למספר הפיצ'רים שנותן את הערך Cp הנמוך ביותר לפי האינדקסים שלהם:

תמונה שמכילה טקסט, צילום מסך, תצוגה, תוכנה

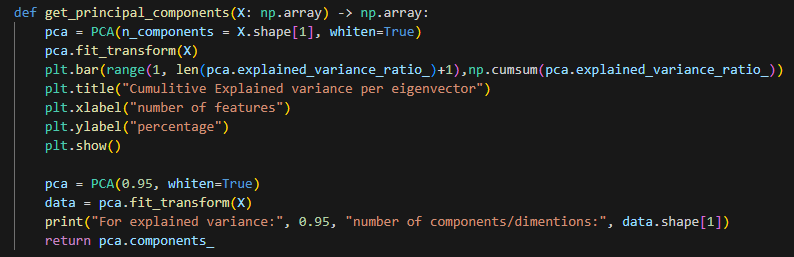
התיאור נוצר באופן אוטומטי



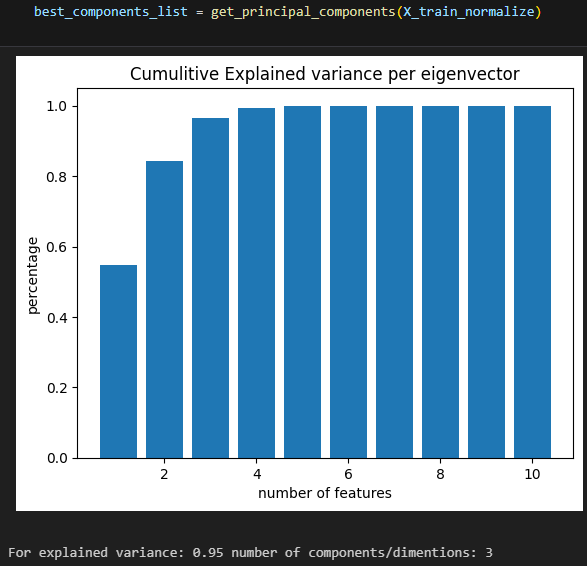
**ניתוח גורמים ראשיים**

ניתוח גורמים ראשיים או PCA [ראשי תיבות](https://he.wikipedia.org/wiki/%D7%A8%D7%90%D7%A9%D7%99_%D7%AA%D7%99%D7%91%D7%95%D7%AA) ב[אנגלית](https://he.wikipedia.org/wiki/%D7%90%D7%A0%D7%92%D7%9C%D7%99%D7%AA) של: (Principal Components Analysis) היא שיטה להתמרה ליניארית של נתונים למערכת קואורדינטות חדשה שבה המידע בקואורדינטות ה[שונות](https://he.wikipedia.org/wiki/%D7%A9%D7%95%D7%A0%D7%95%D7%AA) הוא [אורתוגונלי](https://he.wikipedia.org/wiki/%D7%90%D7%95%D7%A8%D7%AA%D7%95%D7%92%D7%95%D7%A0%D7%9C%D7%99%D7%95%D7%AA) ובעל שונות הולכת וקטנה. שימוש נפוץ בשיטה הוא לצורך מציאת ייצוג מ[ממד](https://he.wikipedia.org/wiki/%D7%9E%D7%9E%D7%93_(%D7%9E%D7%AA%D7%9E%D7%98%D7%99%D7%A7%D7%94)) נמוך למידע מממד גבוה (תהליך הנקרא [הורדת ממד](https://he.wikipedia.org/wiki/%D7%94%D7%95%D7%A8%D7%93%D7%AA_%D7%9E%D7%9E%D7%93)).  
דבר נוסף שחשוב לציין הוא משמעות המימדים הנותרים. בניגוד לתהליך הבחירה קדימה, PCA למעשה יוצר מספר מימדים נמוך יותר ע"י מיזוג המימדים המקוריים. התוצאה היא שהמימדים החדשים איבדו את משמעותם המקורית, משום שהם למעשה צירוף לינארי של כמה מהמשתנים המקוריים (מה המשמעות של מימד שהוא צירוף לינארי של השינוי באחוזים במחיר הסגירה באותו היום והווליום של המנייה באותו היום?).

לאחר התייעצות עם המנחה, קיבלנו החלטה לנסות ולהסביר 95% מהשונות של המודל המקורי באמצעות PCA:



נרצה לראות כיצד נראית השונות המוסברת המצטברת באמצעות מספר הוקטורים העצמיים שמצא המודל:



ניתן לשים לב שלצורך הסברה של 95% מהשונות, יש צורך אך ורק ב-3 מימדים! פחות ממה שראינו בשימוש באלגוריתם בחירה קדימה.

**השוואת המודלים**

ראשית, נשמור בנפרד כל אחד מסטי האימון שלנו לאחר שהפעלנו עליו כל אחד מהאלגוריתמים:

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

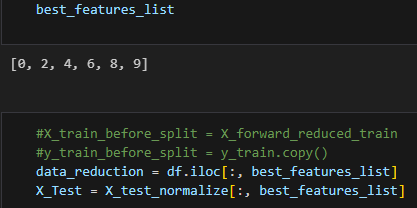
כעת נחשב את התוצאות:

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

**בחירת המודל**

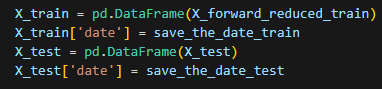
במקרה שלנו, בחירה קדימה השיגה תוצאות טובות יותר (רכבת MSE קטנה יותר), תוך שימוש רק ב-3 תכונות נוספות, בעלות המשמעות המקורית של התכונות (בהשוואה ל-PCA שכנראה איבד מידע רב שגורם לפשרה בין אובדן מידע להפחתת ממדיות נוטה יותר לצד של אובדן מידע מאשר בחירת Forward). לכן נראה שעדיף להשתמש בבחירה קדימה עבור מערך נתונים זה, אם נשתמש ברגרסיה לוגיסטית.

חשוב גם להזכיר שהשתמשנו באלגוריתם בחירה קדימה המבוסס על מודל רגרסיה לוגיסטית עבור כל אחד מארבעת המודלים שלנו שניישם בסעיף הבא. הסיבה לכך היא זמן הריצה הגבוה שלוקח להטמיע בחירת תכונות עבור כל אחד מהדגמים שלנו. אנחנו גם רוצים להזכיר שאנחנו מודעים לכך שאם היינו מיישמים בחירה קדימה לכל דגם בנפרד (ולא מבוסס רק על הרגרסיה הלוגיסטית) ברור שנקבל תוצאות טובות יותר על 3 הדגמים האחרים (מלבד מודל הרגרסיה הלוגיסטית).

כמו כן, נטמיע את אותם הישנויים גם על סט המבחן שלנו כדי שביצועי המודלים עליו יהיו ברי השוואה לאלו שנבצע על סט האימון שלנו בהמשך:

**מניפולציה על משתנים קיימים**

נחזיר חזרה את משתנה התאריך לתוך סט הנתונים שלנו:



תמונה שמכילה טקסט, גופן, צילום מסך, מספר

התיאור נוצר באופן אוטומטיכעת נפצל את עמודת ה-date ל-3 תכונות חדשות (שנה, חודש ויום), נוסיף אותם לדאטה סט ונוריד את עמודת התאריך. נעשה את השינוי גם בסט האימון וגם בסט המבחן:

תמונה שמכילה טקסט, גופן, צילום מסך, מספר

התיאור נוצר באופן אוטומטיכמו כן, משום שהחיזוי שלנו הוא לעתיד, נרצה להיעזר גם בערכים מהעבר של משתנה החיזוי שלנו. לכן אנו מכניסים אותו חזרה אל תוך מערך הנתונים שלנו:



**הגדרת תווית**

בפרויקט שלנו רצינו לנקוט בגישה שונה שעשויה להביא תובנות לקהל גדול יותר. במקום לחזות את מחיר המניה, שיהיה הרבה פחות מדויק, אנחנו רוצים לחזות את ההתנהגות הכוללת של המניה על ידי ניסיון להעריך אם המחיר יעלה, ירד או יישאר בתוך חלון שנוצר על ידי ערכי סף.

בגלל הנתונים שיש לנו על המניה, ובעיקר בגלל המידע שאין לנו (דוחות רווח או רבעון, סיקור תקשורתי על החברה וכו'). לכן, ננסה לנתח את המחיר באמצעות מודלים מתמטיים ובהתבסס על מידע עדכני שיש לנו על המניה.

בחלק זה נקבע חלון ראשוני המורכב מערכי סף תחתונים ועליונים שיחושבו כאשר הממוצע יהיה המחיר של היום והחסם הוא 0.5 סטיית תקן לכל כיוון על ידי ממוצע נע של 30 הימים האחרונים. דרך פעולה זו מסייעת לנו לחזות התנהגויות חריגות של המניה ותקל על סיווגה. בחלק של הפעלה והערכה של מודלי ML, נריץ גם חיפוש ברשת על מספר דרכים ליצירת ערכי סף אלה ונשווה את התוצאות ביניהם כדי לקבל תוצאות אופטימליות.

קטע קוד זה יחזיר לנו את המחיר המקורי, את הממוצע הנע של היום שחישבנו קודם לכן, את ערכי הסף (עליון ותחתון) ואת גודל החלון שלנו:

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

לאחר מכן, אנו מכניסים לסט האימון שלנו עמודה נוספת של תווית. כל תצפית (החל מהיום ה-30 מרגע תחילת איסוף הנתונים משום שאנו משתמשים בחודש האחרון לצורכי החישוב ולכן אין אפשרות לחשב את התצפיות הראשוניות) תקבל עמודה חדשה הנקראת label, שהיא למעשה משתנה החיזוי שלנו בפועל (הנשען על open). אם התצפית נמצאת בתוך החלון בהתאם לתצפית של היום היא תקבל ערך 0, אם מעל לחלון היא תקבל את הערך 1 ואם מתחת אז תקבל את הערך 1- :

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

חשוב לציין שחלק זה מהווה דוגמא בלבד לרעיון שאותו אנו רוצים להציג ובפועל ננקוט בצורת פעולה שונה. כלומר, הכוונה הייתה להציג את הרעיון שמאחורי הפרויקט שלנו במימוש נאיבי ללא אופטימיזציה או שימוש באלגוריתמים מתקדמים של למידת מכונה.

**פונקציות עזר**

הפונקציות הבאות ישמשו אותנו כאשר נשתמש במודלים שלנו לצורך חיזוי הקלסיפיקציה של מחיר המנייה בעתיד.

פונקציה למציאת "טרנד" באמצעות שיטת sliding window. הפונקציה מקבלת סט נתונים, גודל חלון (כמות הימים בעבר שעליה נישען בחיזוי) ואופק (גודל התקופה בעתיד שאליה היינו רוצים לחזות). הפונקציה מחזירה 2 מערכים, הראשון (X) מתאר את רצף הקלט שאותו אנו מכניסים והשני (y) מתאר את רצף הפלט שלנו. הרעיון הוא להעריך באמצעות רצף תצפיות מהעבר את התוויות של רצף מהעתיד:

תמונה שמכילה טקסט, צילום מסך, תוכנה, גופן

התיאור נוצר באופן אוטומטי

הפונקציה הבאה משתמשת בשיטה זהה לפונקציה הקודמת, אך הפעם מטרתה היא לחזות את גודל החלון הנוצר באמצעות ערכי הסף שלנו (ולא אורך התקופה אחורה שעליה אנו מסתכלים בתחזיות שלנו):

תמונה שמכילה טקסט, צילום מסך, תוכנה

התיאור נוצר באופן אוטומטי

בשתי הפונקציות הללו נשתמש בתהליכי האופטימיזציה שנעשה בשלב הרצת והערכת מודלי ה-ML שלנו.

**יצירת מדד accuracy מותאם**

ציון דיוק (accuracy score) בלמידת מכונה הוא מדד הערכה המודד את מספר התחזיות הנכונות שבוצעו על ידי מודל ML ביחס למספר הכולל של תחזיות שנעשו בו בהרצתו. אנו מחשבים זאת על ידי חלוקת מספר התחזיות הנכונות במספר הכולל של התחזיות.

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטילצורכי הבהרה כיצד מחושב המדד והסבר ליצירת המדד המותאם על ידינו, נסביר ראשית מהי confusion matrics. ההמחשה תהיה על קלסיפיקציה בינארית על אף שהקלאסיפיקציה שלנו היא בעלת מספר מחלקות שונות גבוה יותר.

כפי שניתן לראות, קיימות 4 אפשרויות לסיווג התחזית לקלסיפיקציה במודל שלנו: TP,FP,FN,TN.

כל תוצאה תלויה בתחזית המודל שלנו אל מול התוצאה בפועל. אם התחזית זהה למציאות, נסווג אותה כ-True, אחרת כ-False. הסיווג השני הוא חיובי או שלילי, וזאת בהתאם לתחזית שלנו עבור אותה התצפית.

כפי שהסברנו קודם לכן, מדד ה-accuracy למעשה מחושב בצורה הבאה:



במקור, רצינו להשתמש במדד זה במודלים שלנו משום שזהו המדד הסטנדרטי שבו משתמשים במשימות חיזוי בקלסיפיקציה, אך לצערנו נתקלנו בבעיה בשימוש בו. הבעיה טמונה בגודל ערכי הסף שלנו (גודל השרוול שאותו אנו קובעים לצורך הקלסיפיקציה). ככל שערכי הסף גדולים יותר, כך יותר ויותר תחזיות ייכנסו לתוכו ויסווגו תחת המחלקה של אין שינוי. הבעיה נובעת מכך שהסיווג יהיה נכון.

מדוע זו בעיה? משום שסיווג כזה למעשה מחליש את תוצאות המודלים שלנו ומוריד את יכולות ההסברה שלהם והתובנות שניתן להוציא מהן למשתמשים/משקיעים. גודל שרוול גדול מדי יפספס טרנדים של עלייה/ירידה במחיר המנייה היכולים להניב רווח למשקיעים או לכל הפחות ללמד אותם על התנהגות המנייה הצפויה בעתיד.

לכן, בנינו פונקציה מיועדת בדיוק לפתרון בעיה זו. למעשה, אנו מענישים את מדד הדיוק ככל שאחוז גבוה יותר מהתצפיות מסווג כ-0 (אין שינוי משמעותי צפוי במחיר המנייה). משום שאנו מעריכים שנקבל יותר תחזיות כאלו ככל שערך הסף גדול יותר, דרך פעולה זו עונה על דרישותינו. הפונקציה כאמור מחזירה את ערך הדיוק שאותו אנו רוצים לקבל לצורך הערכת המודלים שלנו:

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

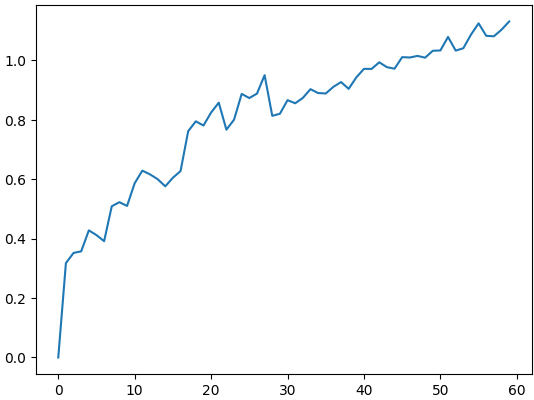
**מציאת שינוי מירבי**

במודלים שלנו מתבצעת אופטימיזציה (GridSearch) על מספר רב של פרמטרים והיפר-פרמטרים. לכן, כדי למנוע זמני ריצה ארוכים מהרצוי, נרצה לדעת האם אפשר לתחום חלק מהפרמטרים בצורה חכמה באמצעות הנתונים שלנו.

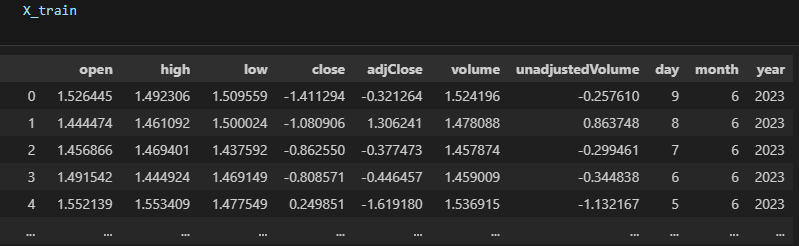
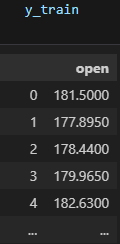
קטע הקוד הבא מראה גרפית את הקפיצות המקסימליות בנתונים בין מחירי המנייה בימים שונים. הוא עושה זאת ע"י חישוב הפער המקסימלי במחיר הפתיחה של המנייה עבור כל הפרש ימים h (עד 60 ימים קדימה) שקיים בנתונים שלנו. הפונקציה המתוארת בקוד תאפשר לנו לבצע הגבלת ערכים חכמה. כאן מוצגת דוגמא עבור ערכים אקראיים:

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי



נרצה לראות כעת איך נראה סט האימון שלנו לפני שאנו מתחילים את שלב התחלת והרצת המודלים. סיווגנו את סט האימון כ-X\_train. הסיווג ל-y\_train ניתן לעמודה שבה מאוכסנים הערכים המקוריים של הפיצ'ר open בדאטה לפני הנרמול (לצורך התחזית), בעוד שב-X\_train יש לנו את המשתנה הנ"ל בצורתו המנורמלת כדי להקל על המודלים שלנו:



כמובן שהאינדקס של התצפיות תואם בין X\_train ל- y\_train.

***הרצת והערכת מודלים של למידת מכונה***

המודלים של למידת מכונה שבחרנו להריץ על הנתונים שלנו לצורך קלסיפיקציה מרובה הם:

1. MLP
2. SVM
3. Random Forrest
4. RNN

בחרנו לשלב גם מודלים של shallow learning וגם מודלים של deep learning. אנו מצפים לקבל ביצועים טובים יותר מלמידה עמוקה, ובמיוחד מ-RNN המתמחה בחיזויים לאורך זמן (או חיזויים של רצפים/השלמות כגון NLP). בחלק זה נריץ את המודולים שלנו, ואם אפשר, ננסה להבין את החשיבות של תכונות בנתונים שלנו לחיזוי התוויות שהתקבלו.

אנו נריץ אותם על מערך הנתונים לאימון שלנו וננסה לשפר אותם ככל האפשר, כמו גם להעריך אותם באמצעות תהליך אימות צולב של סדרות זמן, כמו גם לבדוק את הביצועים שלהם באמצעות ציון דיוק עבור כל מודול שבחרנו. ננסה גם להסביר אם האלגוריתם שלנו מותאם יתר על המידה, ואם כן, כיצד נוכל לפתור (לפחות חלקית) בעיה זו.

בחלק זה, נבחר את המודול שלנו לחיזוי מערך נתוני הבדיקה, בהתבסס על מי מהם ביצע את הטוב ביותר (ציון הדיוק המותאם הגבוה ביותר).

לפני שצללנו פנימה אל המודלים, הורדנו מסט האימון שלנו את העמודות המייצגות את השנה, החודש והיום. עשינו זאת גם כי נתונים אלו אינם מנורמלים (ולכן קיים חשש שהמודלים לא יקבלו אותם כמו שצריך ונקבל תוצאות מעוותות) וגם כי המטרה שלנו היא חיזוי בזמן והסקת מסקנות ממנו, כלומר, נרצה להסתמך על נתונים שאינם התאריך כדי שנוכל לבודד את השפעתם על משתנה החיזוי ללא קשר לתאריך הנבדק (שייתכן שקרו בו דברים נוספים מעבר למה שנמצא אצלנו בדאטה – למשל אסון טבע, קריסת שוק או אפילו דוח רבעוני חיובי מעבר לציפיות). כשנציג את התוצאות בחלק הבא של הפרויקט, נכניס חזרה את הפיצ'רים הללו כדי להציג את הניתוח שלנו לאורך זמן.



***MLP***

פרספטרון רב-שכבתי היא רשת נוירונים שבה המיפוי בין קלט לפלט אינו ליניארי. לפרספטרון רב-שכבתי יש שכבות קלט ופלט, ושכבה נסתרת אחת או יותר עם נוירונים רבים המוערמים יחד.

בעוד שלפרספטרון בתא העצב חייבת להיות פונקציית הפעלה שמטילה סף, כמו ReLU או סיגמואיד, תאי עצב בפרספטרון רב-שכבתי יכולים להשתמש בכל פונקציית הפעלה שרירותית.

MLP היא רשת הנוירונים הבסיסית ביותר.

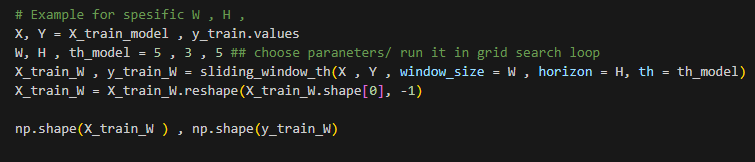
תמונה שמכילה תרשים, טקסט, צילום מסך, עיגול

התיאור נוצר באופן אוטומטי

***הרצת המודל***

לצורך אופטימיזציה של המודל, ביצענו 2 סוגי grid-search (שיטה לביצוע אופטימיזציה של היפר-פרמטרים, כלומר, בהינתן מודל של למידת מכונה וסט נתונים, זו שיטה המוצאת את השילוב הטוב ביותר של ההיפר-פרמטרים של המודל לצורך ביצוע המשימה שאליה נדרש המודל. היפר-פרמטר יכול להיות למשל קצב הלמידה/התקדמות של המודל בכל צעד שלו). ביצענו תהליך זהה עבור כל אחד מהמודלים שבהם השתמשנו בפרויקט.

הסוג הראשון הוא אופטימיזציה של היפר-פרמטרים של המודל עצמו (עבור MLP מדובר למשל במספר השכבות הנסתרות, פונקציית הפעלה, קצב למידה ועוד). כדי שנוכל לעשות זאת, קיבענו את ההיפר-פרמטרים שאותם נרצה לחקור ידנית. דרך זו תאפשר לנו לדעת מהו השילוב הטוב ביותר לכל אחד מהמודלים שלנו בהינתן המודל שאותו אנו מפעילים וסט הנתונים שלנו. חשוב לציין שאת ההיפר-פרמטרים שבדקנו הוצאנו מתוך התיעוד של MLP בספרייה של SickitLearn.

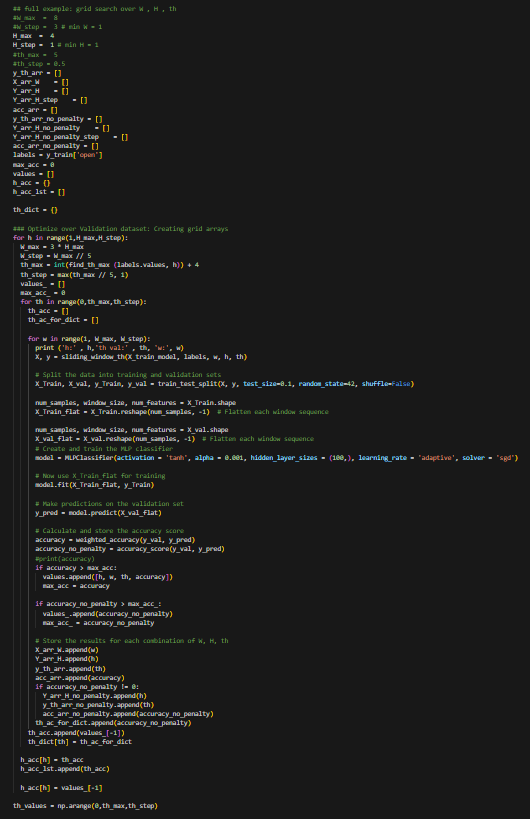


תמונה שמכילה טקסט, צילום מסך, תוכנה, תכונות מולטימדיה

התיאור נוצר באופן אוטומטי

ניתן לראות את השילוב האופטימלי של ההיפר-פרמטרים שמצאנו. נשתמש בשילוב זה ב-GridSearch השני שנבצע. הפעם, אנו מריצים את החיפוש על ההיפר-פרמטרים שיצרנו באופן ידני שאותם נרצה לחקור (W – גודל החלון/הזמן אחורה שאנו מסתכלים, H – גודל התקופה קדימה שאנו מנסים לחזות, TH – גודל ערכי הסף סביב מחיר המנייה שאנו בוחרים לצורך ביצוע הסיווג).   
בקטע הקוד הבא מבוצע תהליך האפטום השני. תהליך זה כולל את הגדרת המשתנים הנחוצים ההתחלתית וכן גם את התאמת המודל ואימונו בהתאם לכל אחד מהפרמטרים המוכנסים אליו. למעשה, שלב זה משתמש בכל פונקציות העזר שהכנו קודם לכן (ביניהן הגדרת המערכים שיקבל המודל באמצעות שיטת החלון הזז, השטחת מימדיות של הקלטים, חישוב מדד דיוק מתוקנן ומקורי ומציאת גודל החלון המקסימלי). כמו כן, אנו בונים מספר מערכים לצורך הצגה ויזואלית מובנת של התוצאות שקיבלנו ושומרים בהם את התוצאות המתקבלות בכל אחת מהאיטרציות של תהליך האופטימיזציה. באמצעות מערכים אלו אנו מסדרים את התוצאות שקיבלנו ומציגים את רמות הדיוק שאליהן הגענו עבור כל ערכי ההיפר-פרמטרים שבדקנו.

בהתאם לרמת הדיוק הגבוהה ביותר שנקבל (max\_acc), נדע מי המודל שביצע את התוצאות הטובות ביותר עבור משימת החיזוי שלנו. את המודל הזה ניקח לצורך ביצוע על פני כל התקופות עבור סט נתוני המבחן שלנו בשלב החיזוי.



את הערכים שסיפקו את הדיוק המתוקנן הגבוה ביותר אנו מציגים כאן:

סדר הערכים משמאל לימין הוא H, W, TH, weighted\_accuracy. חשוב

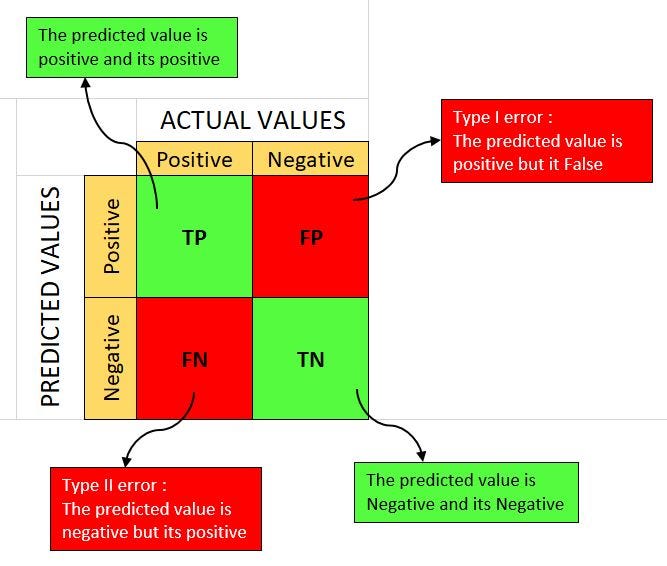
תמונה שמכילה טקסט, גופן, צילום מסך, עיצוב

התיאור נוצר באופן אוטומטיגם לציין שאנו מוסיפים רק את האיברים שמשפרים את רמת הדיוק של המודל

ומציגים בסוף את האיטרציה הטובה ביותר.

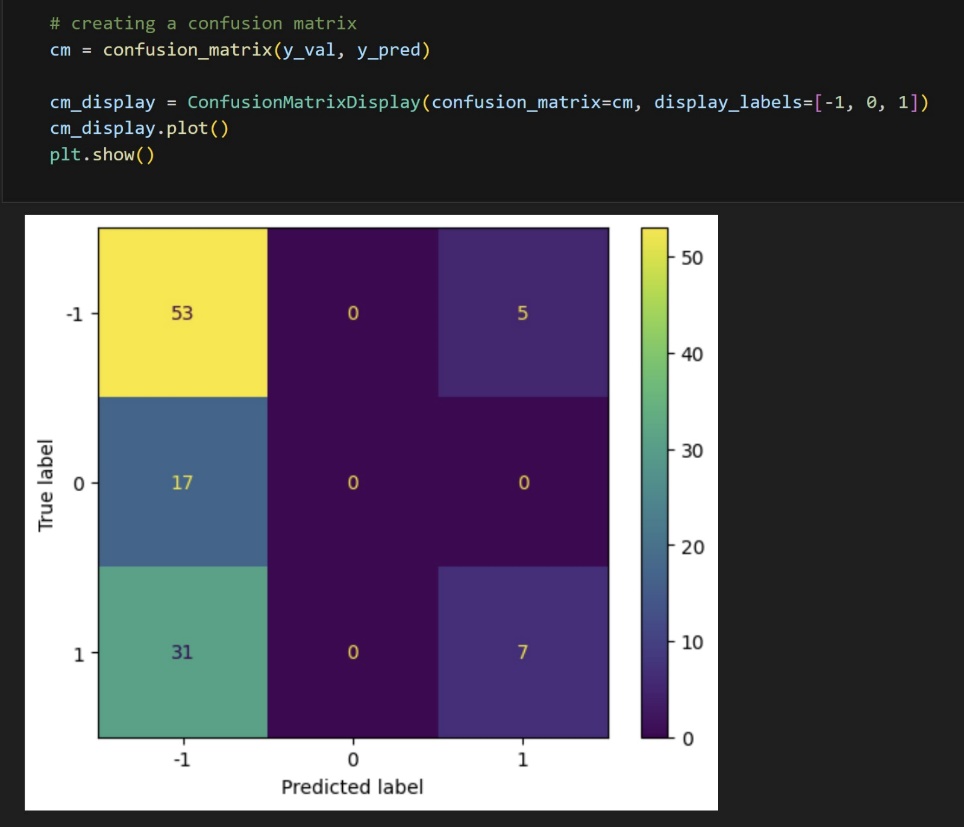
***הצגת Confusion Matrix***

זוהי מטריצה המסכמת את הביצועים של מודל למידת מכונה על קבוצה של נתוני בדיקה. היא משמש לעתים קרובות למדידת הביצועים של מודלי סיווג, שמטרתם לחזות תווית קטגורית עבור כל מופע קלט.

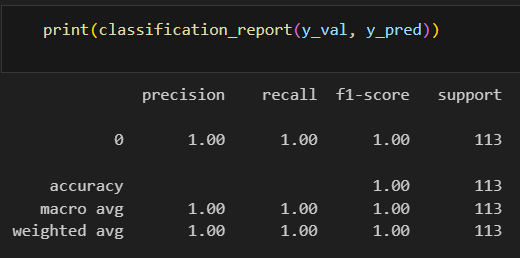


קטע הקוד הבא מציג את המטריצה שהתקבלה כתוצאה משימוש במודל MLP בעזרת השילוב האופטימלי של היפר-פרמטרים שמצאנו על סט הנתונים שלנו. אנו עושים זאת באמצעות השוואת החיזוי/סיווג שלנו על סט הולידציה אל מול התוצאות האמיתיות של סט הולידציה בהתחשב בהיפר-פרמטרים שנבחרו.

מתקבלת המטריצה הבאה (המכילה 3 מחלקות סיווג):



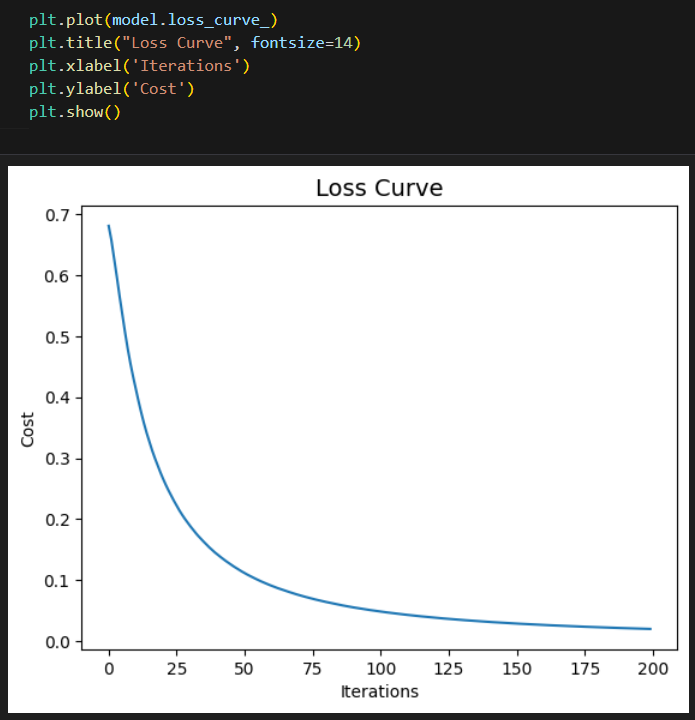
בהתאם לתוצאות הללו, נדפיס את ערכי המדדים השונים המחושבים באמצעות ה-confusion matrix שקיבלנו מהמודל על סט הולדיציה:



***הצגת Loss Curve***

בעולם למידת מכונה, עקומת למידה (נקראת גם עקומת לימוד או עקומת הפסד) היא למעשה שרטוט של ערך פונקציית ההפסד האופטימלי עבור סט האימון כנגד אותו החישוב (עם אותם ההיפר-פרמטרים) עבור סט הוולידציה שיצרו את אותה פונקציית הפסד.   
באופן מופשט יותר, עקומת הלמידה היא עקומה של (מאמץ למידה) - (ביצועים מנבאים), כאשר בדרך כלל מאמץ למידה פירושו מספר דגימות אימון וביצועים מנבאים פירושם דיוק בדגימות בדיקה.  
עקומת למידת המכונה שימושית למטרות רבות, כולל השוואה בין אלגוריתמים שונים, בחירת פרמטרים של מודל במהלך התכנון, התאמת אופטימיזציה לשיפור ההתכנסות וקביעת כמות הנתונים המשמשים לאימון.

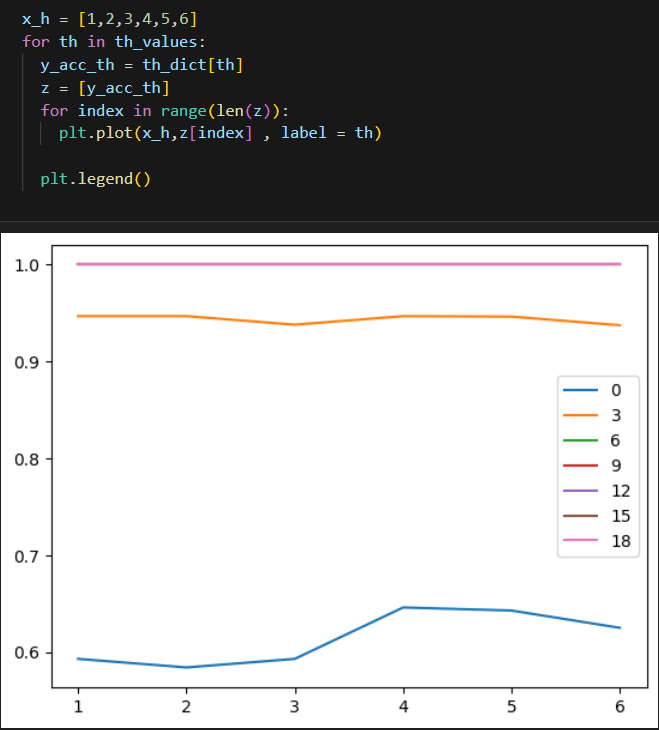
כאן אנו מדפיסים את עקומת ההפסד עבור מודל ה-MLP שלנו:



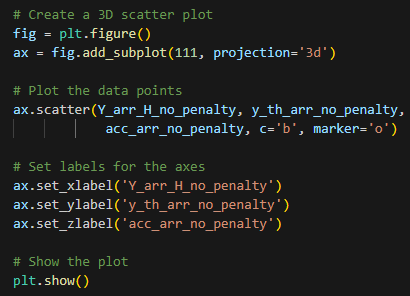
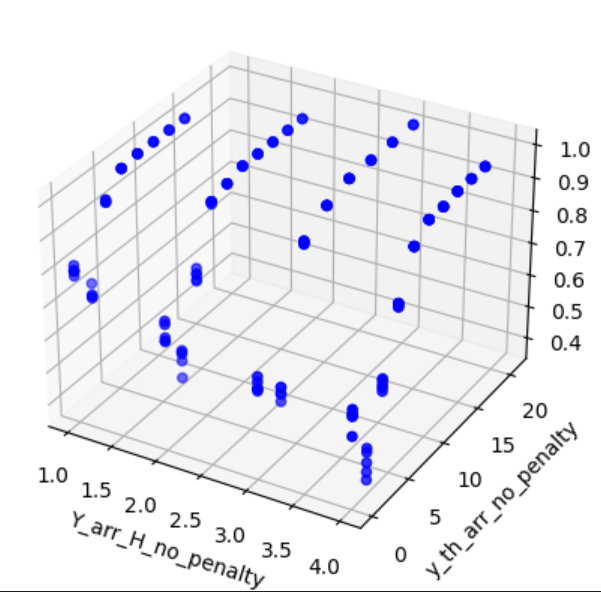
כמובן לא מפתיע אותנו שככל שהמודל עובר יותר איטרציות אימון, כך הלמידה שלו על סט האימון שקיבל טובה יותר, ערכי הפרמטרים שלו מותאמים יותר ולכן גם ערכי החיזוי שלו על סט זה תואמים יותר טוב את ערכם בפועל. הדבר גורר ירידה בערכים המתקבלים בפונקציית ההפסד/הלמידה של המודל שלנו.

***בעיית ה-Thresh Hold***

בשלב מאוד מוקדם בפרויקט הבנו שאנו עומדים בפני בעיה. כאמור, מטרת הפרויקט היא לבצע חיזויים לגבי התנהגות המנייה בעתיד, תוך כך שהחיזויים יהיו אינפורמטיביים למשקיעים. הבעיה היא גודל ערכי הסף שאנו קובעים. סביר להניח שבמרבית הזמן, השינויים בהתנהגות מחיר המנייה יהיו יחסית מינוריים (בהינתן שלא קרו אנומליות לא סבירות כגון אסון טבע או פשיטת רגל פתאומית של החברה – דברים שאינם נכללים בנתונים שלנו). לכן, קביעת ערכי סף גבוהים לגודל השרוול המסווג שלנו יוצרים סיווגים חסרי משמעות. נניח שמחיר המנייה כעת הוא 100$ וערכו נע בין 95$ ל-105$ בשנתיים האחרונות. אם נקבע את ערך סף ההחלטה שלנו ל-15$ מסביב למחיר המנייה, כל הניבויים שלנו יחזו שהתנהגות המנייה לא תשתנה. אמנם כשמסתכלים על מנייה בודדת קשה מעט להבין את הבעיתיות, אך חברת השקעות שרוצה לדעת האם לרכוש מיליוני מניות של החברה הנבדקת תהיה מעוניינת לדעת על שינויים קטנים בהרבה לצורך רווחיות למשקיעים בה. לכן, ערכים כאלו של ערכי סף יהיו חסרי תועלת עבור מספר רב של משקיעים שהיו יכולים להפיק מידע רב מהמודל שלנו.   
סיבה נוספת לכך שניסינו להימנע מבעיה זו היא ציון דיוק מוטה. בהנחה ואנו קובעים ערכי סף גדולים מאוד לצורך הסיווג, המנייה לא תחצה אותם בתקופת החיזוי ולכן כולם יסווגו כ-0 (התנהגות דומה). כאמור, יהיו שינויים (אולי אפילו משמעותיים יחסית) שהמודל לא יכיר בהם. כתוצאה מכך, כל התצפיות העתידיות עבור ערכי פרמטרים אלו יסווגו כ-0, כאשר ביחס לערכי הסף הללו התחזית תהיה תואמת למציאות. התוצאה תהיה רמת דיוק של 100% במודל. זאת תוצאה נפלאה! אך היא די חסרת משמעות ולכן אינה רלוונטית בעינינו.   
במודלים שלנו פתרנו את הבעיה הזו באמצעות פונקציית רמת הדיוק המתוקנת שהצגנו קודם לכן, אך כאן רצינו להציג את הבעיה בצורה גרפית:

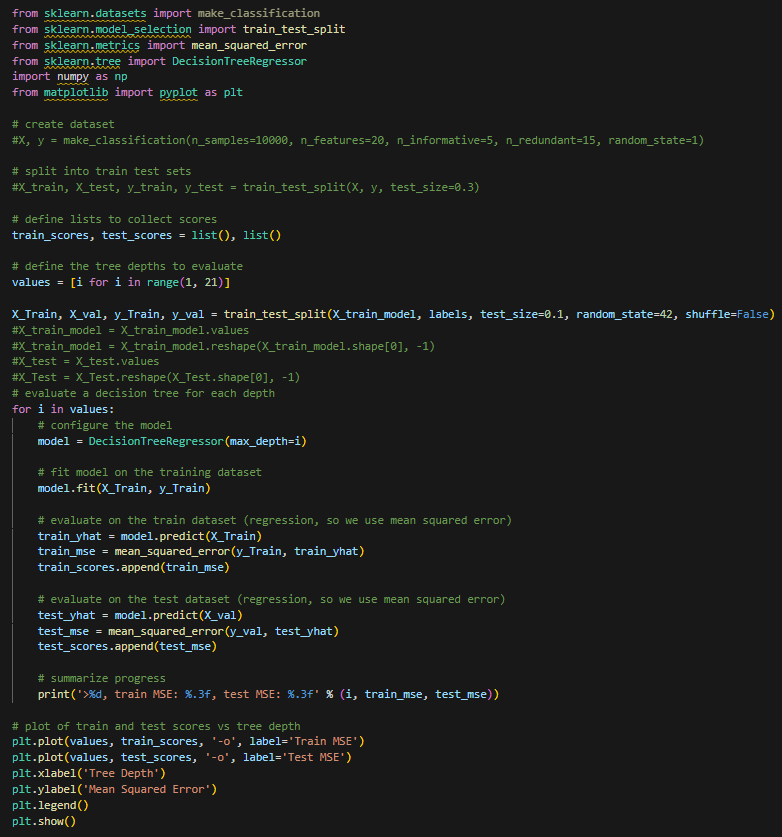
  
גרף זה מציג את רמת הדיוק (הלא מתוקנת – acc\_no\_penalty) של המודל כתלות במספר הימים קדימה (H) שאותם אנו מנסים לחזות. כל אחד מהגרפים מייצג ערך סף חיזוי שונה (TH) שעבורו אנו מפעילים את מודל ה-MLP שלנו. כפי שציפינו שיקרה, החל מערך סף בגודל 6 ומעלה, רמת הדיוק היא 100%. זה כמובן לא מפתיע אותנו, שכן המודל לא יודע לצפות לאנומליות בנתונים ולכן לא חוזה שינוי דרסטי במחיר המנייה בתקופת הזמן הקרובה (פחות משבוע קדימה).

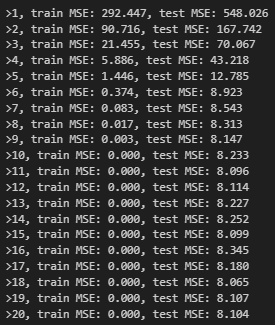
החלטנו להציג את הבעיה גם בצורה תלת מימדית:

­ 

***בדיקת OverFitting***

במידול מתמטי, התאמת יתר היא "הפקה של ניתוח המתאים באופן הדוק מדי או מדויק מדי לקבוצה מסוימת של נתונים, ולכן עלול שלא להתאים לנתונים נוספים או לחזות תצפיות עתידיות באופן אמין". מודל מותאם יתר הוא מודל מתמטי המכיל יותר פרמטרים ממה שניתן להצדיק על ידי הנתונים. במובן המתמטי, פרמטרים אלה מייצגים את דרגת הפולינום. המהות של התאמת יתר היא לחלץ ללא ידיעה חלק מהשונות השיורית (כלומר, הרעש) כאילו וריאציה זו מייצגת את מבנה המודל הבסיסי.  
בחלק הבא, ניסינו לבדוק האם בעיה זו קיימת במודלים שלנו. כדי לעשות זאת, בחרנו להשתמש במודל פשוט יותר מבחינה חישובית מאשר MLP – עצי החלטה! אנו סבורים כי במידה ונקבל התאמה גבוהה מאוד בין סט האימון לולידציה גם עבור מודלים פחות מורכבים, נוכל בוודאות להעריך שאכן בעיה זו קיימת אצלנו במידה מסוימת. חשוב לציין גם שאת הבדיקה עשינו עבור חיזוי מחיר המנייה באופן ישיר, בשונה מבעיית הקלסיפיקציה של שאר המודלים שלנו, ולכן ביססנו את הבדיקה על מדד ה-MSE ולא על רמת דיוק מתוקנת.



את התוצאות ניתן לראות כאן:

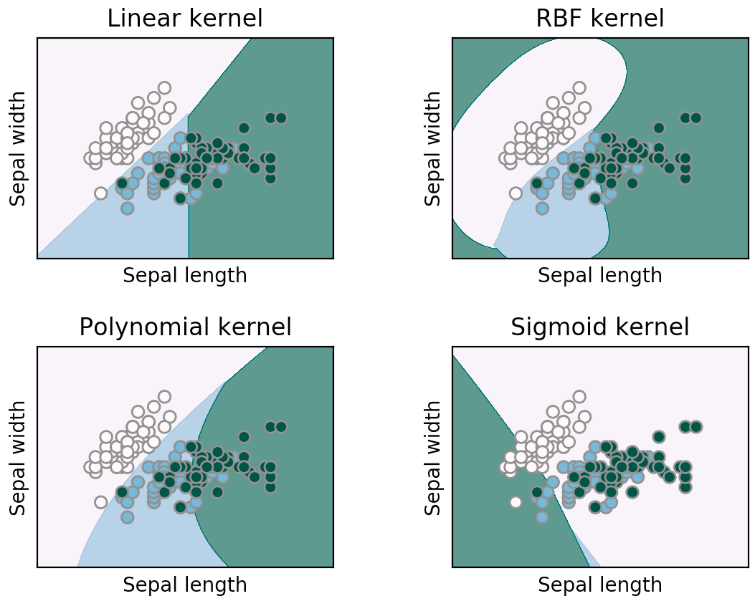
תמונה שמכילה טקסט, צילום מסך, עלילה, קו

התיאור נוצר באופן אוטומטי

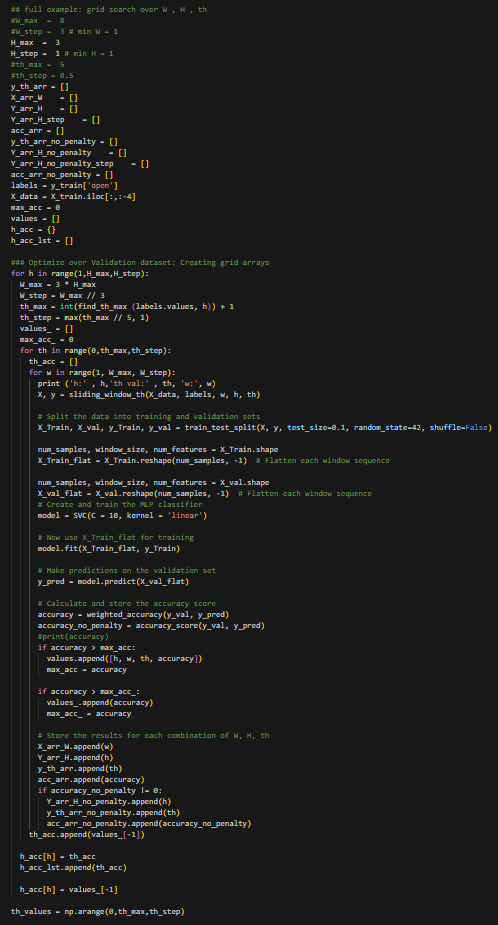
המסקנה הנובעת מהגרף היא שייתכן שבמודלים המורכבים יותר (MLP ו- RNN) ייתכן שהמודל שלנו סובל מהתאמת יתר. זאת משום שעל אף שמחיר המנייה הוא מספר תלת ספרתי עבור מרבית התצפיות, כאשר עצי ההחלטה נהיים עמוקים יחסית, מתקבלת שגיאת אימון 0. יש לקחת עניין זה בחשבון כאשר בוחנים את תוצאות תחזיות המודלים שלנו.

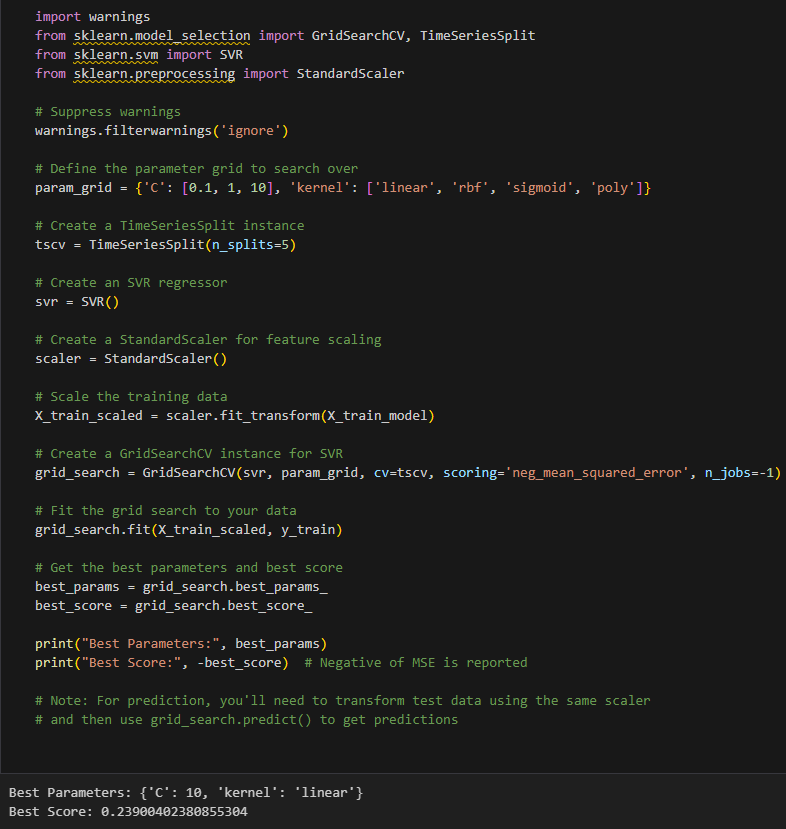
***SVM***

SVM (support vector machines) מייצג מכונות וקטוריות תמיכה. זהו אלגוריתם למידת מכונה מפוקח המשמש הן למשימות סיווג והן למשימות רגרסיה. בהקשר של סיווג, מודל SVM מאומן למצוא על-מישור המפריד בין מחלקות שונות של נקודות נתונים. המטרה היא למקסם את המרווח בין העל-מישור לבין נקודות הנתונים הקרובות ביותר של כל מחלקה. נקודות נתונים קרובות אלה ידועות כווקטורי תמיכה, המשפיעים על המיקום והכיוון של העל-מישור. אלגוריתמים אלו יכולים לטפל בנתונים הניתנים להפרדה ליניארית וכן בנתונים שאינם ניתנים להפרדה ליניארית באמצעות טכניקה הנקראת טריק הליבה. טריק הליבה מאפשר לשרתי אחסון וירטואליים למפות באופן מרומז את נתוני הקלט למרחב תכונות ממדי גבוה יותר, שם הם הופכים לניתנים להפרדה ליניארית. בהקשר של רגרסיה, שרתי אחסון וירטואליים יכולים לשמש גם לחיזוי ערכים רציפים. במקום למצוא היפר-מישור, האלגוריתם שואף להתאים להיפר-מישור שלוכד את מגמת הנתונים. העל-מישור משמש לאחר מכן לביצוע תחזיות על נקודות נתונים חדשות. לשרתי אחסון וירטואליים יש מספר יתרונות, כולל היכולת שלהם להתמודד עם נתונים בממדים גבוהים, החוסן שלהם מפני התאמת יתר, והיעילות שלהם אפילו עם מספר קטן של דגימות.

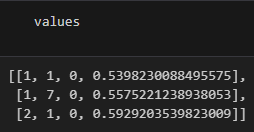
כדי להתמודד עם בעיית סיווג שאינה לינארית, SVM משתמש בטריק ה-kernel כדי לקבל את העל-מישור שנותן את המרחק האופטימלי בין תצפיות ממחלקות סיווג שונות. האלגוריתם המתקבל דומה מבחינה פורמלית, פרט לכך שכל מכפלת נקודות מוחלפת בפונקציית ליבה לא ליניארית. זה מאפשר לאלגוריתם להתאים את ההיפר-מישור בעל השוליים המרביים במרחב תכונות שעבר טרנספורמציה. הטרנספורמציה עשויה להיות לא ליניארית והמרחב שעבר טרנספורמציה הוא רב-ממדי. למרות שהמסווג הוא על-מישור במרחב התכונות שעבר טרנספורמציה, הוא עשוי להיות לא ליניארי במרחב הקלט המקורי.

כאן ניתן לראות דוגמאות לכמה קרנלים אפשריים לצורך סיווג סט הנתונים שלנו למחלקות השונות.

את הרצת האלגוריתם והערכתו נבצע בשיטה זהה לזו שעשינו עבור MLP:

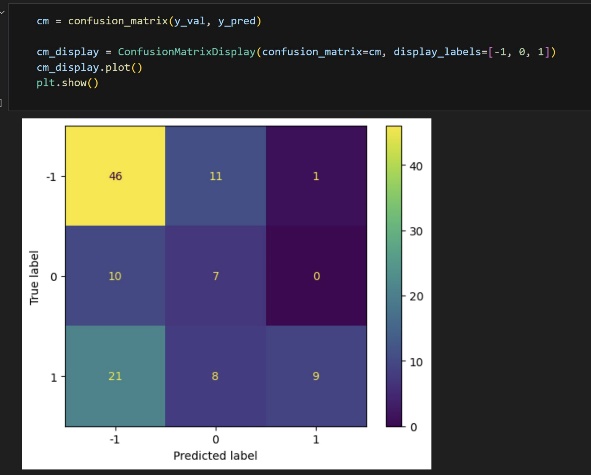


נרצה גם הפעם להדפיס את התוצאה הטובה ביותר שקיבלנו בהינתן הלולאה השנייה:



האיבר האחרון ברשימה הוא הערכים שנותנים לנו את רמת הדיוק המתוקנן הגבוה ביותר. ניתן לראות שהמודל מקבל רמת דיוק של קצת מתחת ל-60% בהינתן הערכים שהכנסנו אליו.

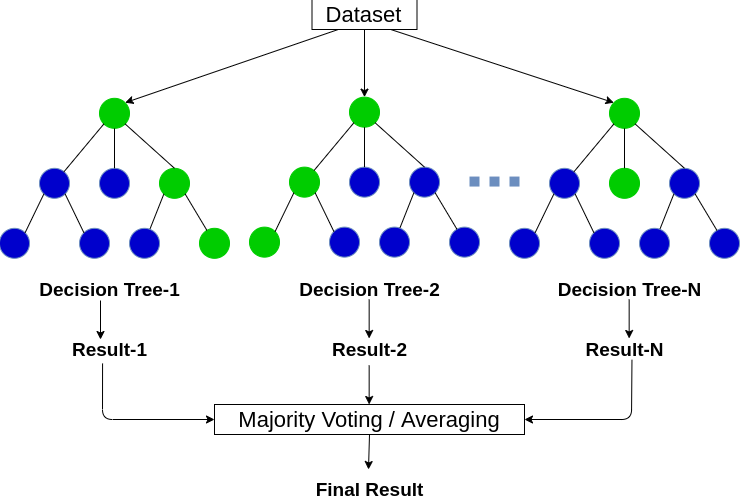
כאן ניתן לראות את ה-confusion matrix של המודל:



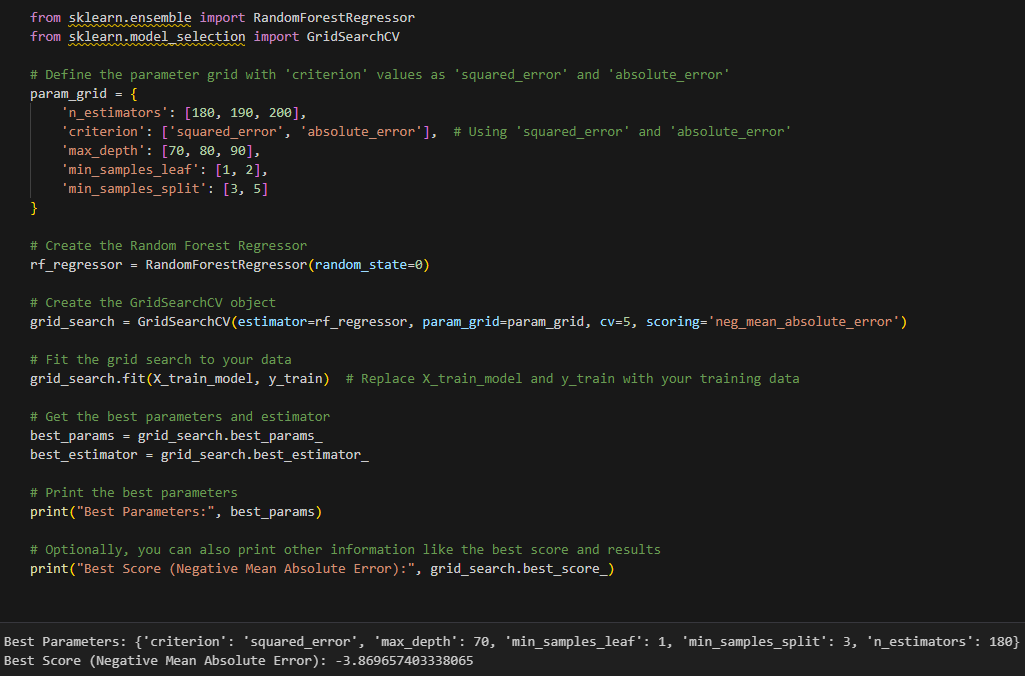
***Random Forrest***

בחלק קודם הצגנו שימוש באלגוריתם של עצי החלטה. הרעיון הבסיסי מאחורי רנדום פורסט הוא ליצור אנסמבל של עצי החלטה, שבו כל עץ מאומן על תת-קבוצה שונה של הנתונים והתכונות. Random Forest מציג אקראיות בשתי דרכים: דגימה אקראית של הנתונים ובחירת תכונות אקראית. במהלך תהליך האימון, כל עץ ביער האקראי מאומן על תת-קבוצה אקראית של נתוני האימון, המכונה דגימות bootstrap. בנוסף, בכל פיצול בעץ, תת-קבוצה אקראית של תכונות נחשבת לקביעת הפיצול הטוב ביותר. כשמדובר בתחזיות, כל עץ בהרכב חוזה באופן עצמאי את משתנה המטרה, והחיזוי הסופי נקבע על ידי צבירת התחזיות הבודדות. עבור משימות רגרסיה, התחזיות מכל עץ הן ממוצעות, ואילו עבור משימות סיווג, החיזוי הסופי נקבע לעתים קרובות על ידי הצבעת הרוב. רנדום פורסט מציע מספר יתרונות. הוא נוטה להיות בעל ביצועי חיזוי טובים, יכול להתמודד עם מערכי נתונים גדולים וגבוהים, והוא נוטה פחות להתאמת יתר בהשוואה לעצי החלטה בודדים. הוא יכול גם לספק מדדים בעלי חשיבות תכונה, המאפשרים בחירה ופרשנות משתנים. בסך הכל, רנדום פורסט הוא אלגוריתם רב-תכליתי ורב עוצמה הנמצא בשימוש נרחב בתחומים שונים בשל חוסנו ודיוקו.

כאן ניתן לראות הצגה גרפית של רנדום פורסט:



כאמור, גם עבור מודל זה נשחזר את 2 שלבי ה-GridSearch:



תמונה שמכילה טקסט, צילום מסך

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, גופן, צילום מסך, עיצוב

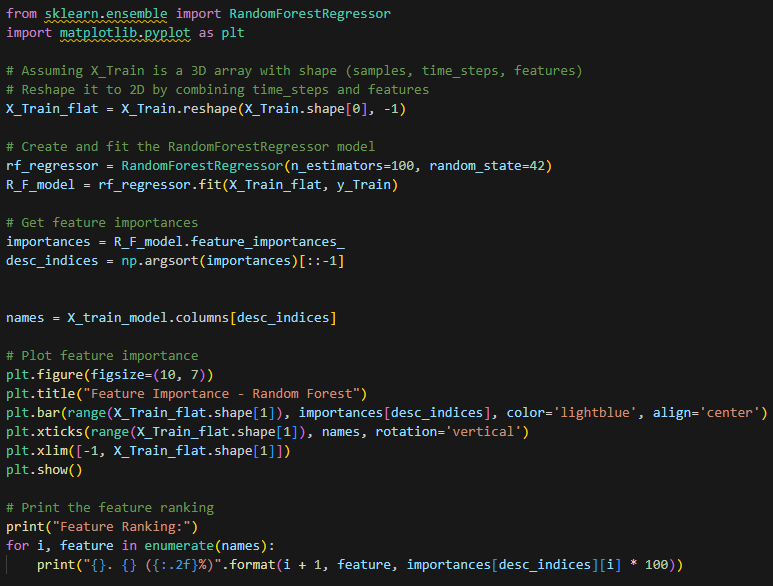
התיאור נוצר באופן אוטומטי

חשוב לנו לציין שלצורך הרצת המודלים בזמן קצר יחסית, בחרנו לבצע את חיפוש שילוב ההיפר-פרמטרים האופטימלי שנבחרים על ידינו בנוגע לבעיה הספציפית של חיזוי בזמן על טווח ערכים מצומצם יחסית. סביר להניח שעבור טווח ערכים גדול יותר היינו יכולים לקבל רמות דיוק גבוהות יותר.

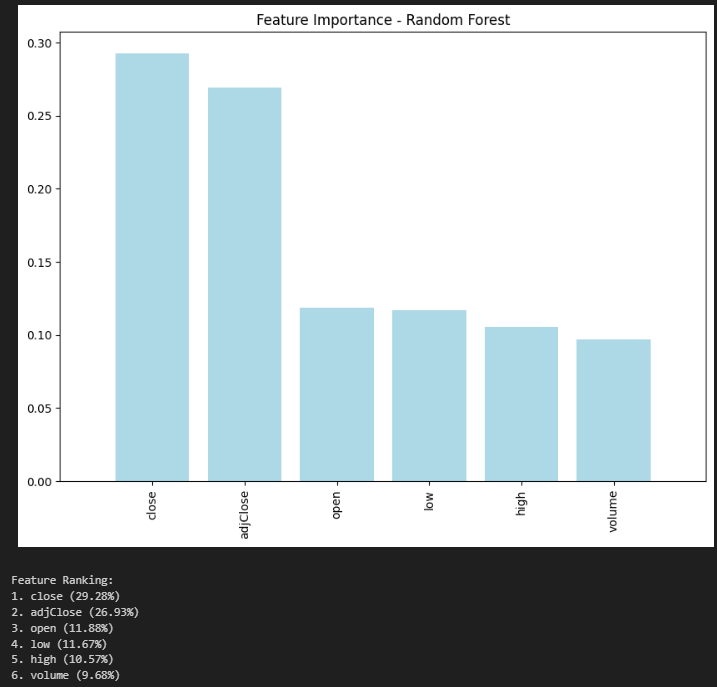
***Feature Importance and RF Visualisation***

בלמידת מכונה, ציוני חשיבות תכונות משמשים כדי לקבוע את החשיבות היחסית של כל תכונה במערך נתונים בעת בניית מודל תחזית. ציונים אלה מחושבים באמצעות מגוון אלגוריתמים שונים, אך אחת הדרכים הפופולריות ביותר היא מסווג היער האקראי. באמצעות אלגוריתם יער אקראי, ניתן למדוד את חשיבות התכונה כאשר ירידת הטומאה הממוצעת מחושבת מכל עצי ההחלטה ביער.

את חשיבות הפיצ'רים השונים למשתנה החיזוי שלנו חישבנו באמצעות קטע הקוד הבא:

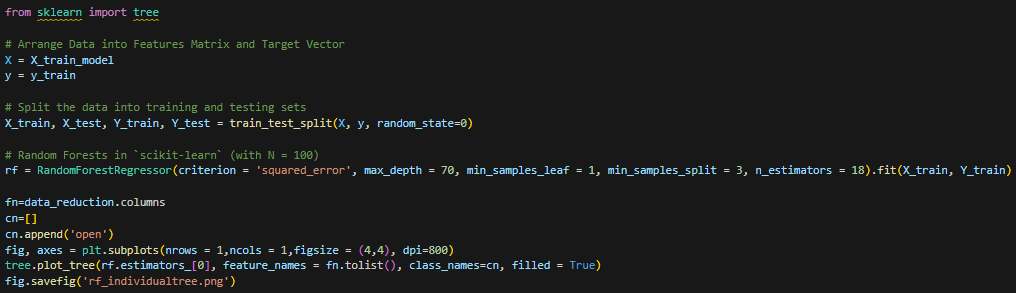


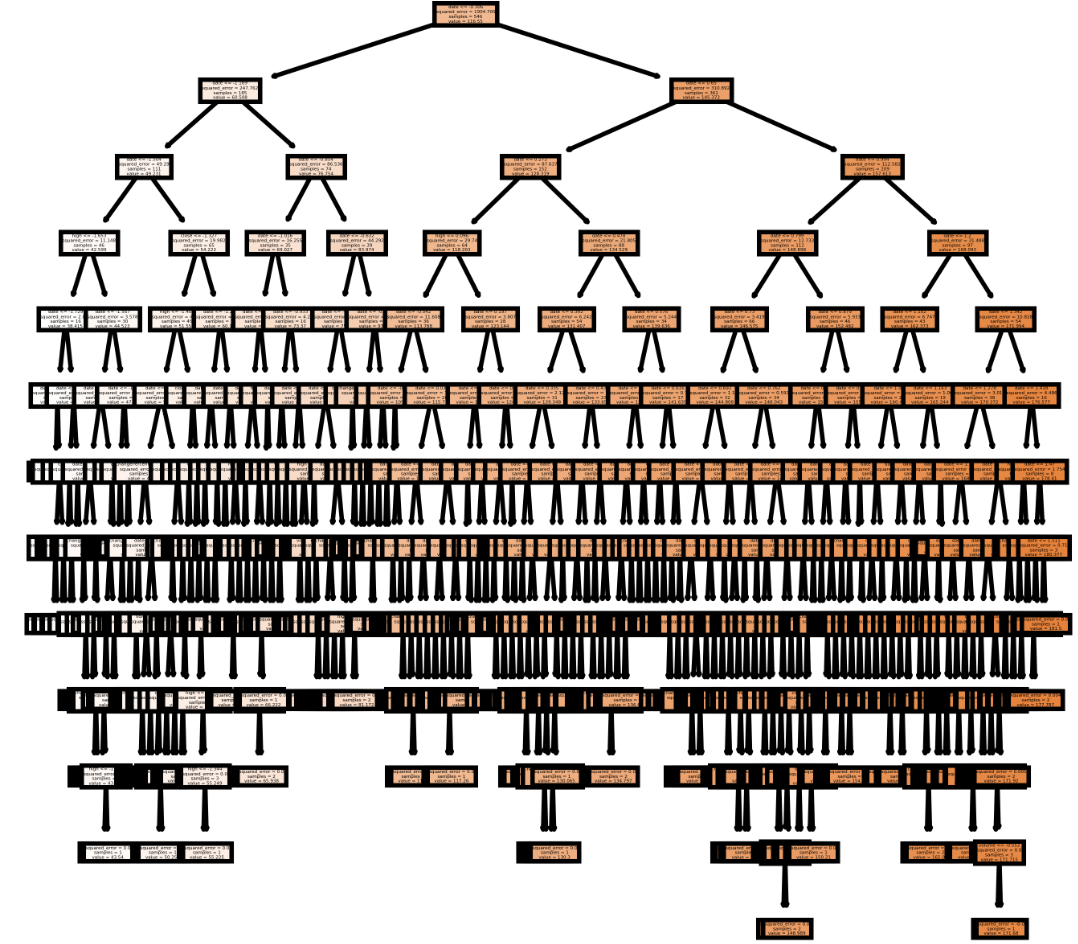
את התוצאות ניתן לראות כאן:



תוצאה מעניינת מחשיבות הפיצ'רים היא שדווקא מחיר הפתיחה אינו הפיצ'ר החשוב ביותר לצורך החיזוי לפי מודל רנדום פורסט!

את העץ שלנו ניתן גם להציג ויזואלית:

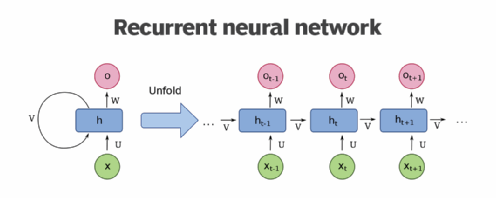




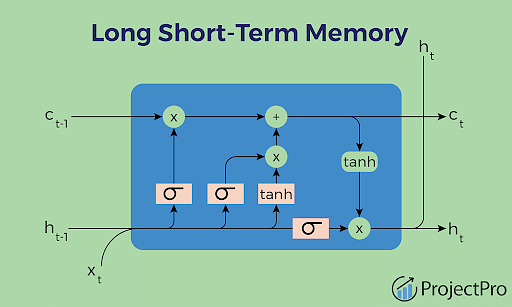
לפי חלקת העץ אנו יכולים לראות כי יער אקראי עובד על מספר עצי החלטה. שרטוט שלהם נותן תחושה בעצם איך המודל חוזה את הערך של משתנה היעד על ידי למידת כללי החלטה פשוטים המוסקים מתכונות הנתונים. כל החלטה בצומת מתקבלת על ידי סיווג באמצעות תכונה אחת מנתוני האימון שלנו.

***RNN and LSTM***

רשת עצבית חוזרת (RNN) היא סוג של רשת עצבית שבה הפלט מהשלב הקודם מוזן כקלט לשלב הנוכחי. ברשתות עצביות מסורתיות, כל הקלטים והיציאות אינם תלויים זה בזה, אך במקרים בהם נדרש לחזות את הערך העתידי, התחזיות הקודמות נדרשות ולכן יש צורך לזכור אותן. כך נוצר RNN, אשר פתר בעיה זו בעזרת שכבה נסתרת. התכונה העיקרית והחשובה ביותר של RNN היא המצב הנסתר שלה, אשר זוכר קצת מידע על רצף. המצב מכונה גם מצב זיכרון מכיוון שהוא זוכר את הקלט הקודם לרשת. הוא משתמש באותם פרמטרים עבור כל קלט כפי שהוא מבצע את אותה משימה על כל הקלט או שכבות מוסתרות כדי לייצר את הפלט. זה מפחית את המורכבות של פרמטרים, בניגוד לרשתות עצביות אחרות.



זיכרון לטווח קצר ארוך (LSTM) הוא סוג של רשת עצבית חוזרת (RNN) שתוכננה במיוחד לטיפול בנתונים רציפים, כגון סדרות זמן, דיבור וטקסט. רשתות LSTM מסוגלות ללמוד תלות ארוכת טווח בנתונים רציפים, מה שהופך אותן למתאימות היטב למשימות כגון תרגום שפות, זיהוי דיבור וחיזוי סדרות זמן.



למעשה, מדובר באלגוריתמים היחידים של למידה עמוקה בהם נשתמש בפרויקט שלנו. בהתאם לכך, ובפרט עקב העובדה שאלגוריתמים אלו נועדו לבעיות של חיזוי בזמן, אנו מצפים לקבל באמצעותם את התוצאות הטובות ביותר שלנו. יש עוד המון מידע לגבי אופן הפעולה שלהם שלא כללנו בספר הפרויקט משום שיש עולם שלם בנושא שאפשר לצלול לתוכו, אך אנחנו יכולים להגיד שמחקר קצר בנושא העלה בנו סקרנות והחלטנו שאנו רוצים לנסות להשתמש בכלים שבהם משתמשים מפתחי למידת מכונה בתעשיית ההייטק.

קטע הקוד הבא מגדיר, מאמן ומעריך מודל Sequential עבור משימת סיווג באמצעות שכבת RNN, עם עצירה מוקדמת למניעת התאמת יתר. זה גם מספק הדמיה של ארכיטקטורת המודל, תוך שימוש בעצירה מוקדמת (עצירה מוקדמת של האלגוריתם כאשר פונ' ההפסד של נתוני הולדיציה לא משתפרת במשך 5 epoches).

תמונה שמכילה טקסט, צילום מסך, גופן

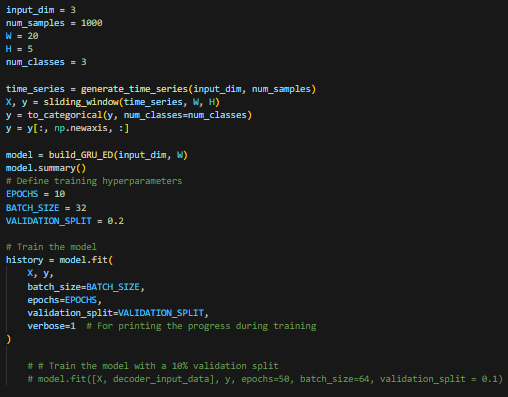
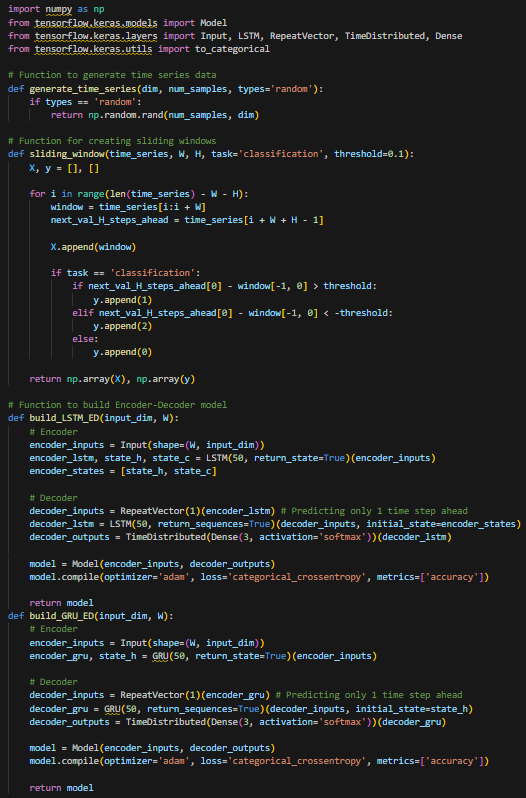
התיאור נוצר באופן אוטומטיתמונה שמכילה טקסט, צילום מסך

התיאור נוצר באופן אוטומטיאת תוצאות המודל על הנתונים שלנו ניתן לראות כאן (נעצר באיפוק ה-7):

תמונה שמכילה קו, טקסט, תרשים, עלילה

התיאור נוצר באופן אוטומטיכפי שצפינו, מודלים מתקדמים אלו מקבלים תוצאות חזקות בהרבה ממודלי ה-shallow learning שבהם השתמשנו קודם. כעת אנו מראים את פונ' ההפסד ורמת הדיוק שהשיג המודל עבור סט האימון אל מול סט הולידציה. כפי שציפינו, ככל שהמודל לומד יותר את סט האימון, רמת הדיוק עבור הולדיציה יורדת, אך במפתיע, למידה רבה יותר מקטינה את פונ' ההפסד של הולידציה.

המטרה הכוללת של קטע הקוד הבא היא ליצור מודל רצף לרצף באמצעות ארכיטקטורת Encoder-Decoder, במיוחד באמצעות שכבות GRU. מודל זה מאומן לחזות את הערך הבא ברצף של סדרת זמן בהתבסס על גישת החלון ההזזה, שבו נעשה שימוש בחלון באורך קבוע של ערכי עבר לביצוע תחזיות לגבי העתיד. הקוד כולל גם שלבי יצירת נתונים ועיבוד מקדים להכנת צמדי הקלט-פלט לאימון ולאחר מכן מאמן את המודל באמצעות הפרמטרים המצוינים. רק לשם ההדגמה, המודל רץ עבור ערכים ספציפיים של טווח חיזוי קדימה (H) וטווח אימון אחורה (W).

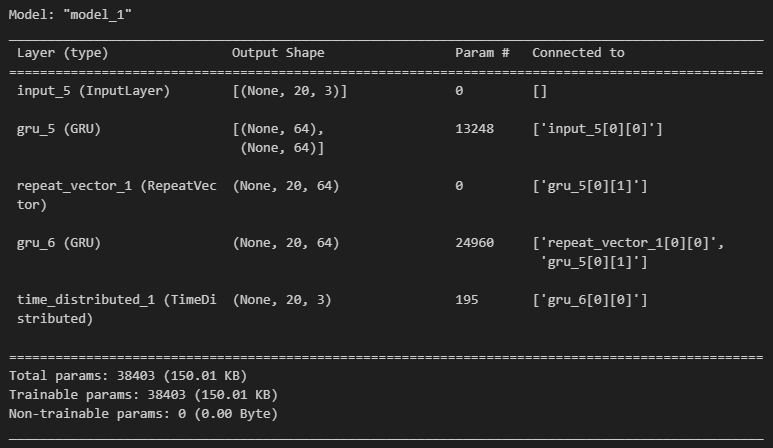


תמונה שמכילה טקסט, צילום מסך

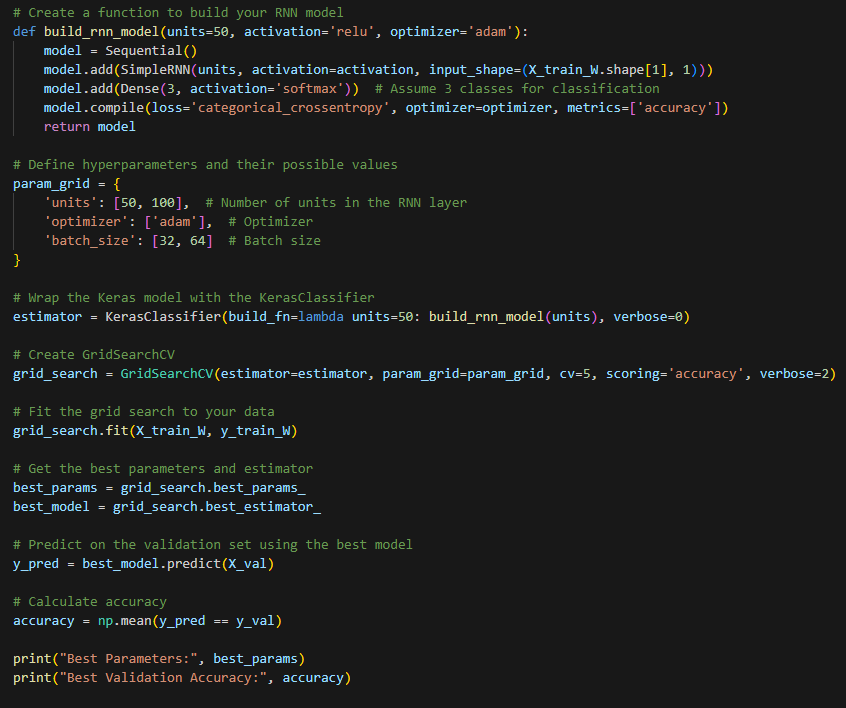
התיאור נוצר באופן אוטומטיחשוב לציין שמתקבלת תוצאה נמוכה יחסית לרמת הדיוק של המודל משום שלא ביצענו GridSearch עבור ערכי הסף, חיזוי קדימה וחיזוי אחורה, אלא רק הראינו את היתכנות המודל ואת הרלוונטיות שלו עבורנו. גם בנסיבות אלו תוצאותיו עולות על אלגוריתמים קודמים:

כשאנו מציגים ויזואלית את התוצאות הללו מתקבלת תמונת מצב מעניינת. נראה שגם ערכי פונ' ההפסד של המודל וגם רמת הדיוק יותר טובים בשלבי הלמידה ההתחלתיים עבור סט הולידציה מאשר עבור סט האימון שלנו. אנו סבורים כי הסיבה לכך היא ביצועי המנייה בתקופת הולידציה. חיתוך הנתונים שלנו נעשה באופן כרונולוגי (תקופת הולידציה היא רציפה ובאה לאחר תקופת האימון). ייתכן שבתקופה זו, מחיר המנייה היה יציב יחסית, ולכן החיזוי ההתחלתי פשוט יותר למודל. מעניין גם לראות שאיפוקים מאוחרים יותר פוגעים מעט בביצועי המודל (הפסד גדל לסט האימון באיפוקים מתקדמים).

כעת ביצענו פעולה זהה עבור מודל נוסף –> ארכיטקטורת encoder-decoder:



לאחר שהצגנו את כל המודלים הללו, נרצה לבצע תהליך דומה של איפטום היפר-פרמטרים דו-שלבי כפי שעשינו לכל המודלים הקודמים, אך הפעם עבור מודל RNN מארכיטקטורת sequential.



כעת נכניס את המודל לשלב השני עם ההיפר-פרמטרים שנבחרו:

תמונה שמכילה טקסט, מכתב, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

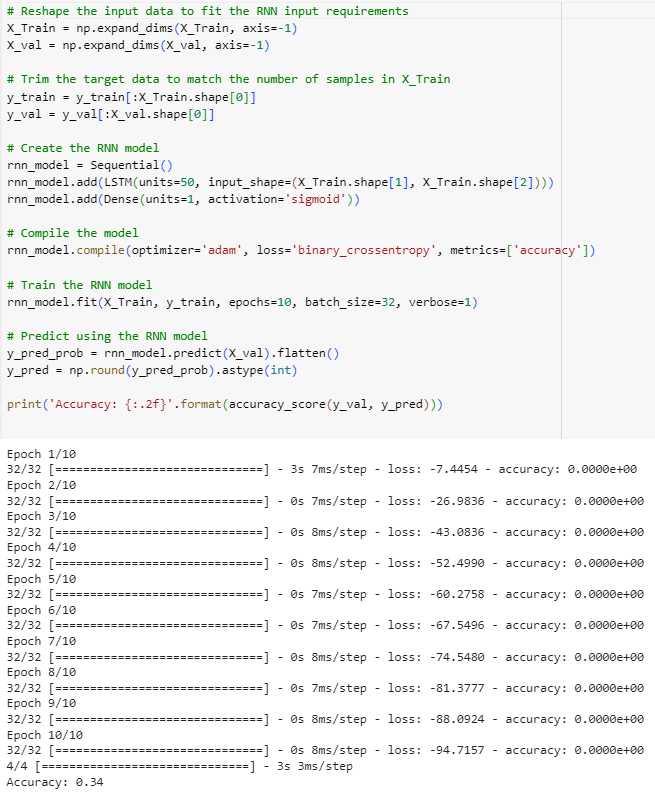
כעת נרצה לראות את התוצאות, נעשה זאת באמצעות הדגמה של המודלים על נתונים מוכנים מראש:

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטיתמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

כעת נרצה לבצע את החיזוי עם המודל שיצרנו:



כך נראית ה-confusion matrix שלו:

תמונה שמכילה טקסט, צילום מסך, צבעוני, תרשים

התיאור נוצר באופן אוטומטי

באופן מפתיע, תוצאות המודל הנ"ל לא היו האופטימליות מבין המודלים שלנו (רמת דיוק של 34% בלבד!). אנו סבורים כי הסיבה לכך היא מורכבות המודלים הללו ביחס לפשטות הנתונים שלנו. ממחקר שביצענו, המסקנה הייתה שמודלים מתקדמים כאלו דורשים כמות גדולה מאוד של נתונים. לצערנו, ככל הנראה כמות המידע שנתנו למודל הייתה יחסית מצומצמת וכתוצאה מכך תהליך הלמידה לא היה מיטבי, לכן החלטנו שלא לבחור בו עבור סט המבחן שלנו.

***בחירת המודל הטוב ביותר***

את המודל הטוב ביותר נבחר לפי רמת הדיוק, כלומר, ייבחר המודל שנתן את רמת הדיוק המתוקנן (weighted accuracy) הגבוה ביותר על סט הולידציה שלנו.

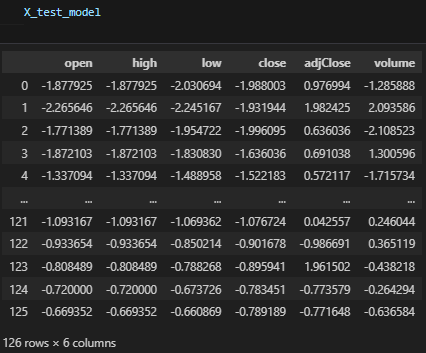
במקרה שלנו, המודל הנבחר הוא MLP. את החיזוי הטוב ביותר (כמעט 7\% רמת דיוק מתוקנן) קיבלנו עבור הפרמטרים הבאים: H = 13, W = 5, TH = 0.

***ביצוע תחזית על סט המבחן***

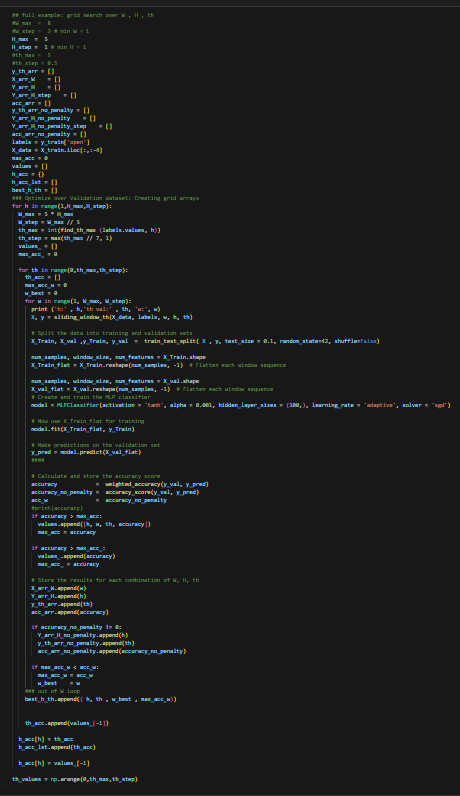
בחלק זה נבצע את התחזיות שלנו על מערך הנתונים של הבדיקה שיצרנו באמצעות המודול שהציג את המיטב מכל המודולים שהרצנו בשלבים מוקדמים יותר של הפרויקט – MLP.

לאחר הבדיקה, נדון בתוצאות שלנו (ועד כמה הן טובות) ובתובנות שקיבלנו מהפרויקט שלנו ביחס למערך הנתונים שבידנו.

קודם כל, נראה איך נראה סט המבחן שלנו שעליו נפעיל את התחזית (ללא משתני התאריך):



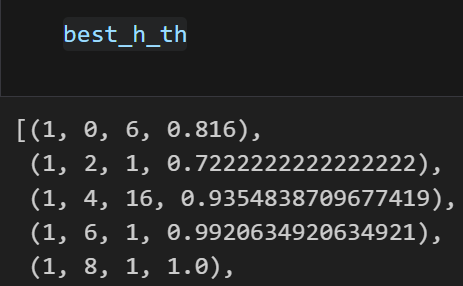
כעת נבצע תהליך זהה עבור כל הנתונים שלנו, כאשר החלוקה תהיה לאימון ומבחן במקום לאימון וולידציה. בסוף נציג גרף לכל W נציג גרף לכל TH, כאשר הצירים יהיו H ו-accuracy. כך נוכל לראות את ביצועי המודל שלנו על כל אחד מערכי ההיפר-פרמטרים ולנתח את מה שקיבלנו. נעשה זאת ע"י שמירה של הערכים בתוך מערך ומתוכם נייצר את כל הגרפים הרלוונטיים.



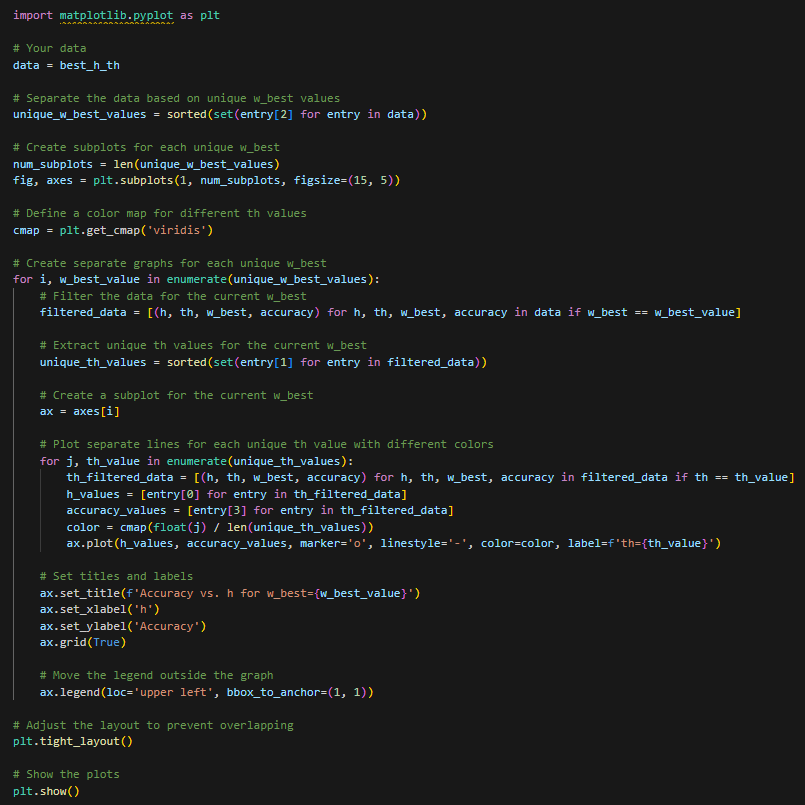
כתוצאה מהלולאה הזו, אנו מקבלים את המערך best\_h\_th. מערך זה מכיל את תוצאות ההרצה המיטביות מהמודל בהתאם ל-W המיטבי שנבחר כהיפר-פרמטר.

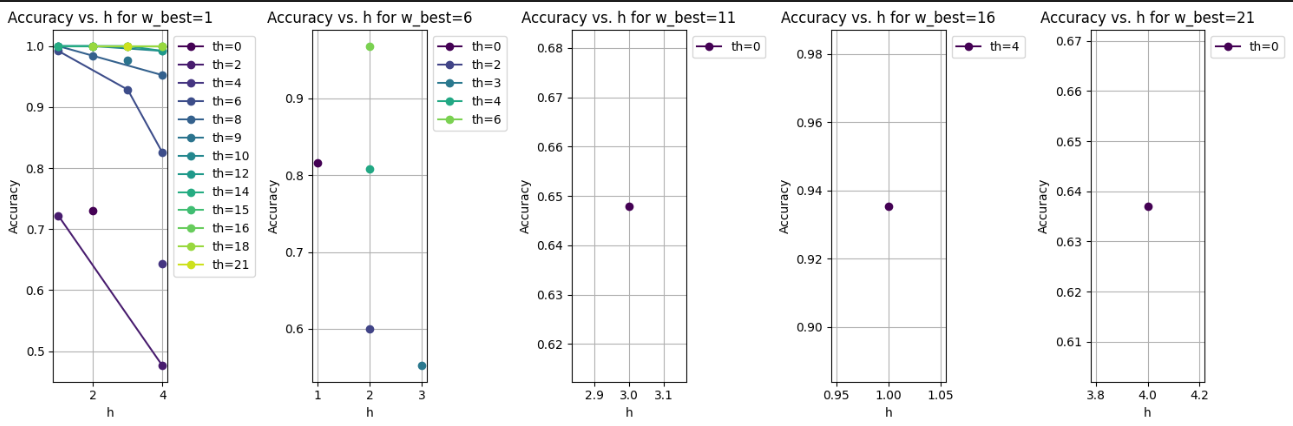
***ניתוח תוצאות ומסקנות***

נדפיס כעת את המערך הרלוונטי:



מערך זה תואם את הציפיות שלנו, כלומר, מציג ערכי רמת דיוק גבוהים יותר ככל שחלון ההסתכלות אחורה או ערכי הסף גדלים וקטן ככל שקפיצת ההסתכלות קדימה גדלה (לא מוצג כל המערך בתמונה).  
כעת נציג גרפית את המערך:





***הצעות לפיתוח עתידי של הפרויקט***

בתחילת העבודה על הפרויקט, חשבנו על מספר דרכים בהן ניתן להמשיך לפתח את הפרויקט:

1. **חיזוי ערכים של מניות נוספות**

בפרויקט התבססנו על המנייה של אפל בלבד. אנו סבורים כי התעמקות וניתוח דומה לזה שביצענו על מניות אחרות יכול להניב מסקנות מעניינות נוספות. ייתכן ומחירי מניות אחרות מושפע אחרת משאר הפיצ'רים הקיימים אצלנו בדאטה (למשל בעת צמצום המימדיות), או שמודלים שונים יוכלו לבצע תחזיות טובות יותר על מניות מסוימות (בשלב האימון של המודלים). לדעתינו, השוואת תוצאות אלו בין מניות שונות יכולה לספק תובנות לגבי אופן חיזוי אופטימלי בזמן של שוק ההון.

1. **חיזוי ערכים של סקטורים**

מדד ה- s&p 500 בנוי מכ-11 סקטורים שונים (ביניהם סקטור אנרגיה, סקטור טכנולוגיה וכו'). כל אחד מהסקטורים מורכב מכמה עשרות מניות בודדות השייכות למדד כולו. חיבור הנתונים של כל המניות השייכות לסקטור מסוים אל תוך סט נתונים יחיד וביצוע חיזוי עליו יכול לנבא תובנות רבות על התנהגות התחום כולו ברמה הגלובלית. לצורך העניין, אם המניות הטכנולוגיות הגדולות ביותר מנבאות עלייה במחירי המניות שלהן בעתיד, ניתן להניח שדבר דומה יקרה גם עבור חברות קטנות יותר. ההערכה הזו יכולה להפיק ערך למשקיעים לגבי הלך הרוח של השוק ומה מתרחש בו ברמה מקיפה הרבה יותר מאשר מנייה בודדת.

1. **חיזוי ערכים של מדדים**

כאן אנחנו מגיעים ל"גביע הקדוש" של ההשקעות. בדומה למדד שבחרנו, קיימים מדדים נוספים ברחבי העולם (גם מדדים יחודיים לישראל כמו מדד תל אביב 125). אמרה מפורסמת בשווקים הפיננסיים היא שלא ניתן לנצח את השוק. הכוונה היא לביזור השקעות, ולכוחו של השוק להמשיך ולהתחזק גם כאשר חלק ממנו קורס. מדובר בחיזוי הרבה יותר מורכב הכולל ניואנסים רבים שאינם קיימים ברשותינו. אך עם כל זאת, אנו חושבים כי זהו הניתוח המשמעותי ביותר והשמיש ביותר לכלים שאותם העמדנו לבדיקה בפרויקט שלנו. חיזוי התנהגות של מדד מרכזי ייתן אינדקציה להתנהגות השווקים בכל תחום שהוא ולכן גם הכי מעשי.