**מבוא**

שנת 2020 נפתחה עם מגפת הקורונה, שיצרה משבר עולמי בכל תחומי החיים, ובפרט בשוק ההון שהתרסק בצורה היסטורית וחסרת תקדים [1].

מאז ומתמיד **משברים בשוק ההון נחשבו להזדמנות** – משקיעים נוהגים לנצל משברים על מנת לקנות ניירות ערך במחירים אטרקטיביים, ולהרוויח את העלייה המגולמת בהתאוששות מהמשבר.

הדרך לניצול הפוטנציאל הנ"ל רצופת מכשולים, ולהלן העיקריים שבהם :

1. קשה מאוד **לחזות** אילו חברות ישרדו את המשבר. כך למשל, לאחר משבר הסאב-פריים בשנת 2008 חברות רבות ירדו מנכסיהן ופשטו רגל [2].
2. קשה מאוד **לתזמן את הכניסה לשוק –** זוהי בעצם נקודת המינימום של נייר הערך המועמד לקנייה, המאפשרת למשקיע את התזמון המושלם להיכנס לשוק במחיר אטרקטיבי.

שיטה נפוצה שהתפתחה במרוצת השנים להתמודדות עם הבעיה הראשונה היא **פיזור**. המשקיע הממוצע יחלק את הונו בין אפשרויות השקעה שונות, כדי להיות חסין מירידות ערך מאסיביות, ולייצר רשת ביטחון הגונה.

אחת הדרכים להשיג את הפיזור הנ"ל היא מכשיר פיננסי בשם **תעודת סל**. זוהי תעודה המונפקת על ידי גוף פיננסי (בנק או בית השקעות), ומאפשרת למשקיע לעקוב אחר נכס בסיס כגון **מדדי מניות** (מדדים המהווים ממוצע משוקלל של מספר רחב של מניות).

ההתמודדות עם **הבעיה השנייה היא שאלת מיליון הדולר של עולם ההשקעות,** ורבים מאמינים שבלתי אפשרי לחזות את התנהגות השוק. במרוצת השנים התפתחו שתי אסכולות מקבילות סביב השאלה הזו : Timing the markets VS Time in the markets [3] (השקעה לטווח הארוך אל מול תזמון השוק).

בפרויקט זה ננסה לאתגר אלגוריתמים שונים של בינה מלאכותית ולמידה עמוקה עם היכולת לתזמן את השוק בזמן משבר, ולמקסם את הרווח של המשקיעים = "קנה בזול ומכור ביוקר".

**תיאור הפתרון המוצע לבעיה**

תמצית הפתרון :

ב-30 השנים האחרונות שוק ההון התפתח במספר מובנים :

* ידע משברים רבים, מהם ניתן ללמוד על אופן ההתאוששות של השוק.
* נוספו אינדיקטורים רבים שנועדו לצוד תופעות שונות בשוק במטרה להבין לאן פניו. האינדיקטורים הללו משמשים משקיעים רבים ככללי אצבע בתכנון ההשקעות שלהם.

ברצוננו, בהינתן נייר ערך מסוים, ויום מסוים בעיצומו של משבר, לתת חיזוי לגבי מידת הכדאיות של כניסה לשוק. יום שפוטנציאל הרווח בו הוא גדול יקבל ציון קרוב ל-1, ויום בעל פוטנציאל רווח מינימלי יקבל ציון קרוב ל-0.

לשם כך, נשתמש במשברים קודמים כדוגמאות אימון לאלגוריתם הלמידה שלנו. כל יום מסחר בעיצומו של משבר יהווה דוגמא.

* **התכונות** של יום המסחר הם אינדיקטורים ממקורות שונים בהם האלגוריתם יתחשב.
* **התיוג** של יום המסחר הוא פוטנציאל הרווח שלו (על פי חישוב שיפורט בהמשך).

תיאור שלבי המערכת :

בשלב זה נתאר בפרוטרוט את החלקים השונים במערכת, וניגע באתגרים המרכזיים בכל אחד מהם.

1. **Data Extractor**

חלק זה אמון על הגדרת "משבר" על סמך patterns של ביצועים. הוא מקבל נתונים של ביצועים היסטוריים של נייר ערך, וקובע תקופות בהיסטוריה בהם נייר ערך זה היה במשבר.

ההגדרה למשבר היא תקופה בזמן שעונה על הקריטריונים הבאים :

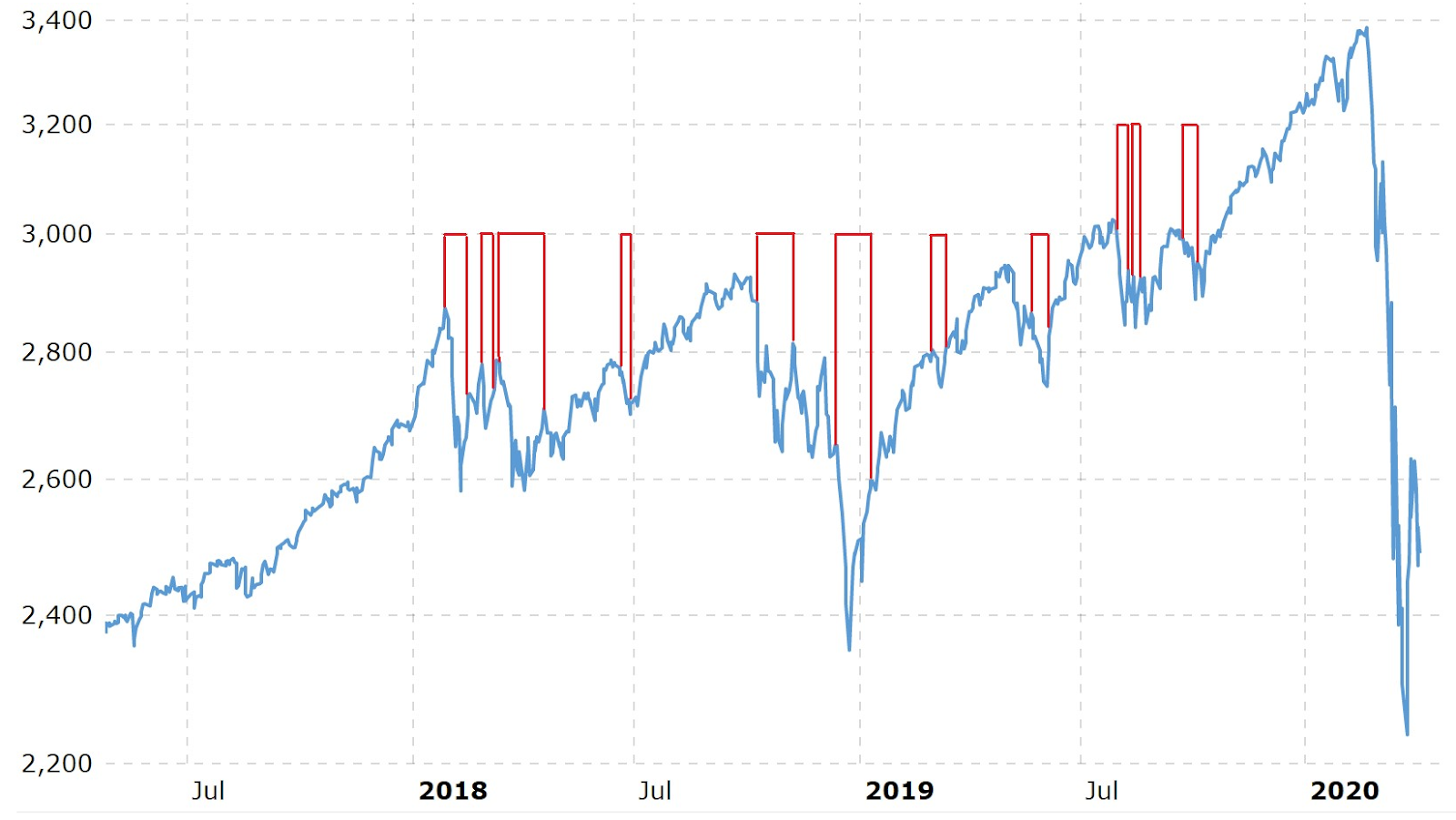
* + אורך מינימלי : לא כל יום של ירידות בשווקים ייחשב כמשבר. נרצה להגדיר סף מינימום (במונחים של ימי מסחר) שמצביע על משבר משמעותי מספיק.
  + פגיעת מינימום : לא כל רצף של ימי ירידה מעיד על משבר. נרצה שהירידה תהיה חדה מספיק (או מתונה וארוכה), ולכן נרצה להגדיר סף למחיר בשיא המשבר (נקודת המינימום ביחס למחיר המקורי).

סיום המשבר ייקבע ליום הראשון בו נייר הערך חזר להיות קרוב למחיר שממנו התחיל המשבר.

הערות :

* + האורך המינימלי, פגיעת המינימום, ורמת הקירבה למחיר המקורי (סוף המשבר) הם היפר-פרמטרים של המודל, וייקבעו בהתאם למדידות על סמך ניסויים שנבצע בהמשך.
  + קיימים משברים בהם נצפו ימי עליות בודדים אשר לא מעידים על סוף המשבר (לדוגמה, כתוצאה מהזרמה ממשלתית של כספים). כדי להתמודד עם הבעיה, נגדיר היפר-פרמטר נוסף שמאפשר ימי תיקון.

להלן ביצועי האלגוריתם על משברים בשלושת השנים האחרונות (בכחול – ביצועי האמת של מדד S&P500 בבורסה האמריקאית, באדום – תקופות אותן האלגוריתם הגדיר כמשבר).



1. **Data Combiner**

חלק זה אמון על איסוף אינדיקטורים ותכונות מימי המסחר שבתוך תקופות המשבר.

הגדרות :

* + Nasdaq, S&P500 : מדדים מובילים בעולם המניות האמריקאי. סחירים בבורסה באמצעות תעודות סל או קרנות מחקות.
  + אינפלציה : נתון המתאר את עליית המחירים במשק. נמדדת באמצעות מדד המחירים לצרכן, הבודק את השינוי במחירו של סל מוצרים קבוע (עלייה של מחירו גוררת עלייה באינפלציה).
  + Earning Ratio : היחס בין רווחי החברה לשווי המניה. זוהי דרך נפוצה לחשב רווח למניה.
  + Dividend Rate : מספר הדולרים שמשקיע מקבל מהחברה בה השקיע לכל מניה שקנה, כתוצאה מרווחי החברה.
  + Dividend Yield : היחס בין ה-Dividend Rate של החברה לבין מחיר המניה שלה.
  + Price-Earnings Ratio : היחס בין שווי השוק של חברה לבין הרווח הנקי השנתי שלה. במדדים, מבצעים ממוצע משוקלל של היחס לפי החברות המשתתפות במדד.
  + CAPE Ratio : מתאר את שווי השוק של מדד ה-S&P500 חלקי הרווח הממוצע, מתואם לאינפלציה, ב-10 השנים האחרונות.
  + מדד הפחד (VIX) : מדד התנודתיות של בורסת האופציות בשיקאגו (CBOE), הנועד להתחקות אחר אלמנטים פסיכולוגיים של משקיעים בשוק.
  + APA : תאגיד נפט אמריקאי. משקיעים רבים רואים בו כמכשיר פיננסי שעוזר במעקב אחר מחיר הנפט בעולם. זה עשוי להיות אינדיקטור טוב לתקופות משבר כי משקיעים רבים עוברים להשקיע בסחורות בתקופות כנ"ל.
  + GLD : קרן סל שמתיימרת לעקוב אחר מחיר הזהב בעולם. זה עשוי להיות אינדיקטור טוב לתקופות משבר כי משקיעים רבים עוברים להשקיע בסחורות בתקופות כנ"ל.
  + TNX : תשואת אג"ח ממשלתי אמריקאי ל-10 שנים. ככלל, אג"חים ממשלתיים נחשבים לאפיק השקעה סולידי ובטוח, ואופן המסחר בהם קשור קשר הדוק לתקופות משבר.

תכונות בתדירות יומיות :

1. מחיר סגירה של S&P500 (Close) : מחיר המניה בתום יום מסחר.
2. מחיר גבוה של S&P500 (High) : העסקה היקרה ביותר (לפי מחיר מניה בודדת) שהתבצעה באותו יום.
3. מחיר נמוך של S&P500 (Low) : העסקה הזולה ביותר (לפי מחיר מניה בודדת) שהתבצעה באותו יום.
4. מחזור מסחר (Volume) : שווי כולל של העסקאות עבור מניה זו שבוצעו באותו יום.
5. מחיר סגירה של nasdaq (Nas-Close).
6. מחיר גבוה של nasdaq (Nas-High).
7. מחיר נמוך של nasdaq (Nas-Low).
8. מחזור מסחר של nasdaq (Nas-Volume).
9. מחיר סגירה של מדד הפחד (Vix-Close).
10. מחיר גבוה של מדד הפחד (Vix-High).
11. מחיר נמוך של מדד הפחד (Vix-Low).
12. מחיר סגירה של APA (Apa-Close).
13. מחיר גבוה של APA (Apa-High).
14. מחיר נמוך של APA (Apa-Low).
15. מחזור מסחר של APA (Apa-Volume).
16. מחיר סגירה של GLD (Gld-Close).
17. מחיר גבוה של GLD (Gld-High).
18. מחיר נמוך של GLD (Gld-Low).
19. מחזור מסחר של GLD (Gld-Volume).
20. מחיר סגירה של TNX (Tnx-Close).
21. מחיר גבוה של TNX (Tnx-High).
22. מחיר נמוך של TNX (Tnx-Low).
23. מחזור מסחר של TNX (Tnx-Volume).

תכונות בתדירות חודשית :

1. אינפלציה.
2. Earning Ratio.
3. Dividend Rate.
4. Dividend Yield.
5. Price-Earning Ratio.
6. CAPE Ratio.

הערות :

1. כדי שנוכל להשתמש במסווג בזמן אמת, נרצה שהתכונות יתבססו על העבר בלבד. לכן בעת סיווג של יום מסחר, המסווג לא ייקח בחשבון תכונות של יום המסחר הנוכחי, אלא של קודמיו.
2. מאותה סיבה, התכונות החודשיות הן במבט לאחור. כך למשל, ביום מסחר של חודש מרץ, נוכל להסתכל על הנתונים החודשיים של ינואר-פברואר.
3. כל תכונה מפוצלת בפועל למספר תכונות, על מנת שנוכל לייצר זיכרון למודל על סמך תקופות זמן משמעותית יותר. כך למשל, נרצה לקחת בחשבון תכונות יומיות של חמישה ימים אחורה, ולא של יום בודד. האורך המדויק הוא היפר פרמטר של המודל.
4. **Features Converter**

כפי שראינו בחלק הקודם, חלק מהתכונות הן בתדירות יומית וחלק בתדירות חודשית. חלק זה אמון על הפיכת התכונות החודשיות ליומיות, כדי שנוכל להשתמש בהן במסווג (שכן הדוגמאות שהמסווג מקבל הן ברמה היומית).

חלק מהתכונות ברמה החודשית מתקבלות ביום הראשון של החודש, וחלק ביום האחרון של החודש. כדי לוודא שבכל יום, כל התכונות עליהן אנחנו מסתמכים זמינות לנו מבעוד מועד :

* + עבור תכונות של סוף חודש – פיצלנו את ערך התכונה לכל יום באותו החודש.
  + עבור תכונות של תחילת חודש – פיצלנו את ערך התכונה לכל יום בחודש הקודם.

כך למשל, תכונה שהתקבלה ב31.3, תיוחס לכל דוגמא מימי חודש מרץ. תכונה שהתקבלה ב1.3, תיוחס לכל דוגמה מימי חודש פברואר.

1. **Model**

חלק זה אמון על תיוג הדוגמאות.

לכל משבר (שכפי שהגדרנו, מאופיין בתאריך התחלה ותאריך סוף), נחשב את מחיר המינימום pricemin ואת מחיר המקסימום pricemax.

נגדיר פוטנציאל רווח למשבר באופן הבא : profitpot = pricemax - pricemin.

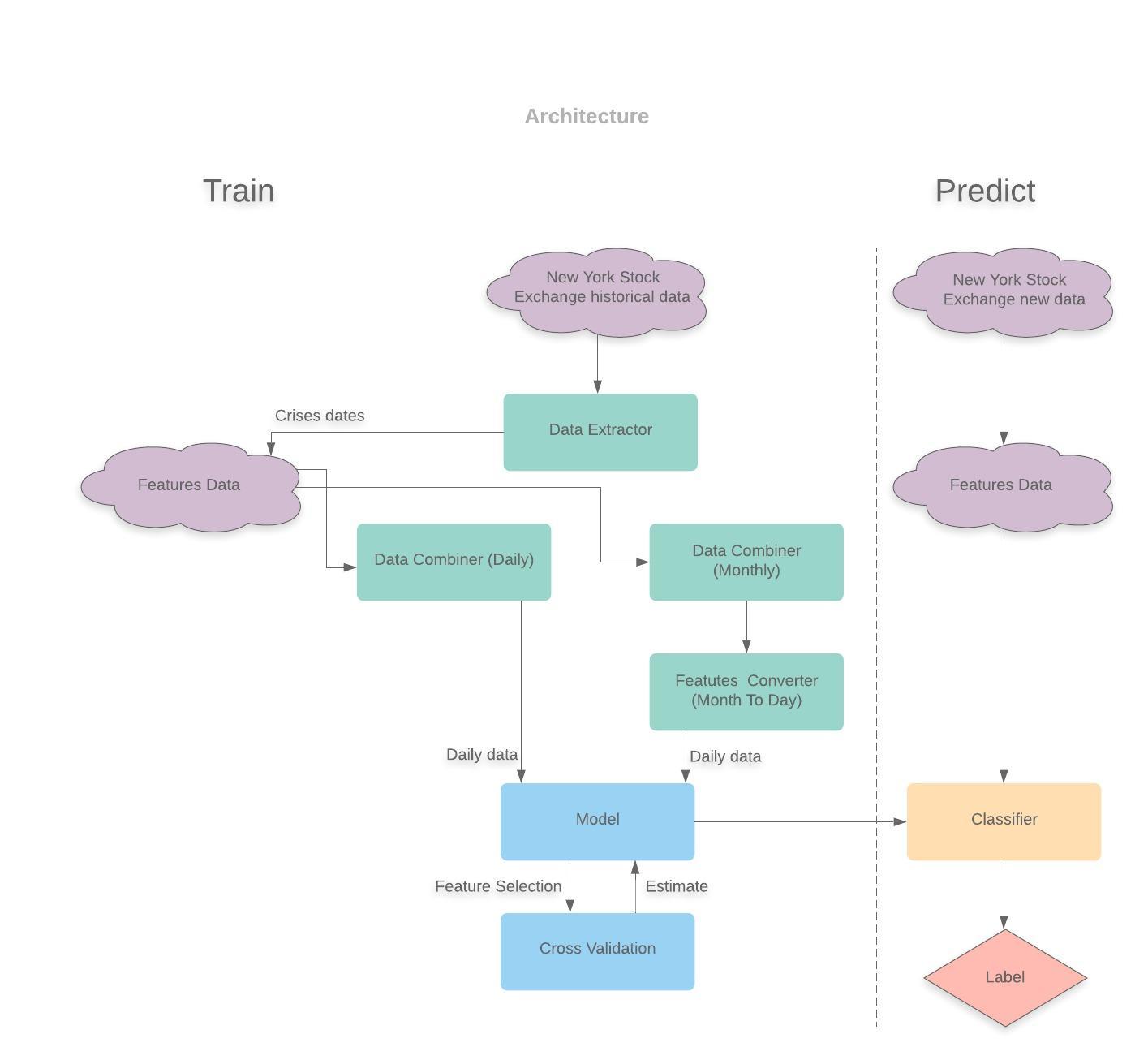
פוטנציאל הרווח של יום מסוים (דוגמה) בתוך המשבר יהיה :

(הערה : הרעיון לחישוב הציון של יום למעשה מחקה את הקונספט של min-max normalization).

1. **Cross Validation**

חלק זה אמון על בחירת תתי-קבוצות שונות של פיצ'רים (בהתאם לאלגוריתמים שיפורטו בהמשך) ואלגוריתם למידה במטרה למצוא אלגוריתם ותת-קבוצה אידיאליים שייבחרו למסווג. הערכת הביצועים תיעשה על ידי K-fold Cross Validation. התוצאות יפורטו בפרק הבא.

מבט-על :



**חקירת הבעיה באמצעות אלגוריתמי בינה מלאכותית**

בשלב זה נתאר ניסויים שעשינו על אלגוריתמי בינה, בהם היינו עדים לקושי להתמודד עם בעיה המבוססת על Time Sequences. תפקידו של חלק זה להדגיש שאלגוריתמי בינה סטנדרטיים מתקשים לענות על הציפיות, וזוהי המוטיבציה למעבר לרשתות נוירונים עם זיכרון.

בחרנו מספר אלגוריתמים לבדיקה, ומספר דרכים לבצע Feature Selection.

לכל אלגוריתם, בדקנו את כל מנגנוני ה-Feature Selection , ועבור תת הקבוצה של תכונות שהתקבלה, בדקנו את רמת הדיוק שלה ע"י Cross-Validation.

חזרנו על הניסוי המתואר עבור מניפולציות שונות שביצענו על ה-Data.

הערה: בשלב זה התיוג, כפי שהוסבר בחלק הקודם, עוגל על מנת לאפשר בחינה גם של מודלים המבוססים על ערכי תיוג דיסקרטיים.

Data Manipulation :

1. Original Data : שימוש ב-Data שתואר בחלק הקודם כפי שהוא.
2. Normalized Data : שימוש ב-Data מנורמל לפי אלגוריתם min-max normalization [4].
3. Standardized Data : הפיכת כל פיצ'ר למפולג נורמלית בתחום שבין 0 ל-1, על פי התוחלת והשונות שלו (שיטת z-score כפי שנלמדת בקורס מבוא למערכות לומדות).

Feature Selection :

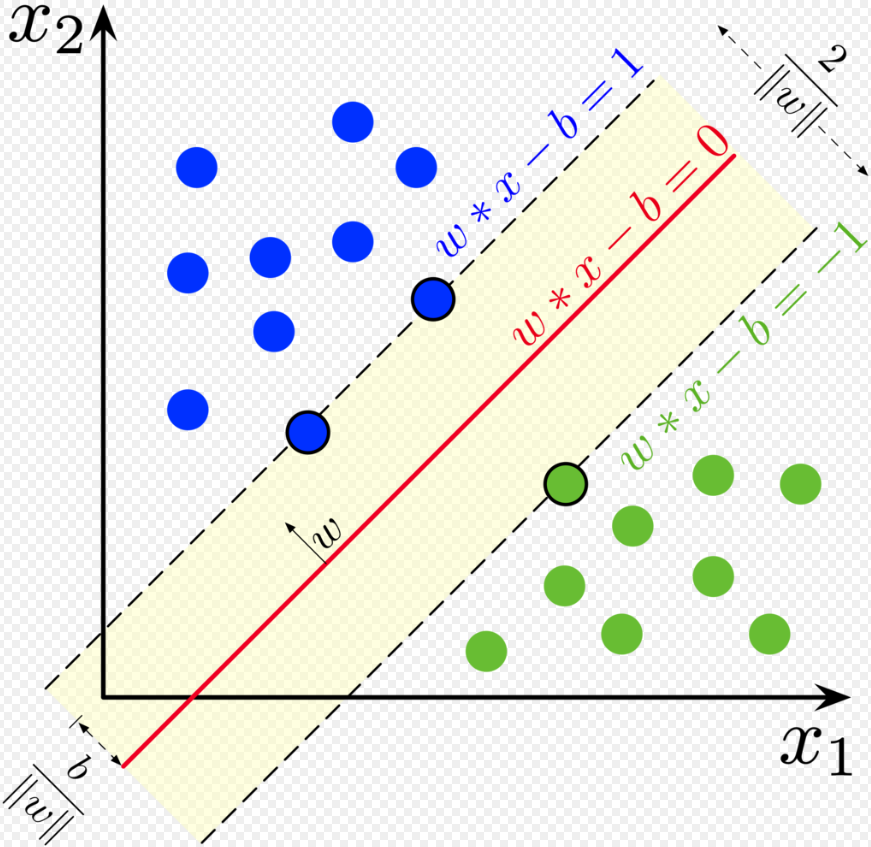
1. selectKBestFeatures : דירוג הפיצ'רים הטובים ביותר על פי פונקציית score, ובחירת k הטובים ביותר תחת הפונקצייה הזו.

פונקציית ה-score שהשתמשנו בה היא mutual\_info\_classif, המנקדת פיצ'רים בהתאם להערכת האנתרופיה שלהם.

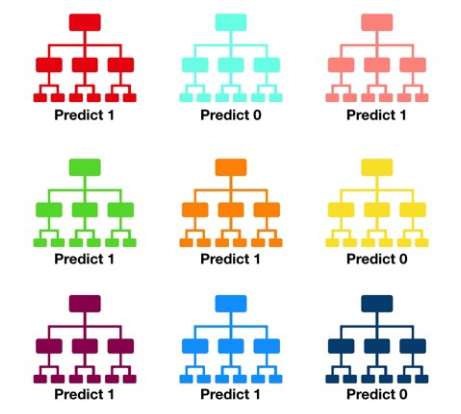
1. selectKExtraTree : מבוסס על ExtraTreeClassifier. אלגוריתם זה בונה יער של עצי החלטה על פי ה-Data, ולאחר מכן דוגם את היער עם הדוגמאות ומפחית את הסיכון ל-overfitting. בדרך זו, הוא נותן ציון בין 0 ל-1 לכלל הפיצ'רים שעומדים לרשותו. תחת ההנחה זו, בחרנו את k הטובים ביותר.

אלגוריתמים :

1. SVM : אלגוריתם Support Vector Machine המבוסס על הפרדה לינארית של דוגמאות מתויגות. מטרתו למקסם ככל הניתן את המרווח (Gap) בין המפריד הלינארי לבין הדוגמאות הקרובות לו (מכלל הקטגוריות) [5].



1. Random Forest : יער של עצי החלטה. בקבלת דוגמה חדשה כלל עצי ההחלטה מנסים לסווג את הדוגמה, וההחלטה מתקבלת על פי majority principal [6].



1. KNN : תיוג לפי K השכנים הקרובים ביותר על פי תכונות מוגדרות ומרחק מוגדר.



ניסוי – מטא אלגוריתם :

1. לכל data מקבוצת Data Manipulation :
   1. לכל feature selector מקבוצת Feature Selection :
      1. לכל algorithm מקבוצת האלגוריתמים :
         1. לכל k (מספר הפיצ'רים) מ-1 עד כמות הפיצ'רים המקסימלית :
            1. הפעל את feature selector K על ה-data וקבל תת קבוצה של פיצ'רים.
            2. אמן את ה-algorithm על פי ה-data המצומצם לפי תת הקבוצה שנבחרה.
            3. בדוק את הדיוק על פי 5-Fold Cross Validation.

בתוצאות הניסוי נציע לכל data, feature selector ו-algorithm את ה-k האידיאלי בלבד.

תוצאות הניסוי :

לכל הרצה נתאר את רמת הדיוק של האלגוריתם שהתקבל ואת מספר הפיצ'רים שהאלגוריתם בחר, כאשר יש סך הכל 122 פיצ'רים זמינים.

* Original Data :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **KNN** | **Random Forest** | **SVM** |  | |
| 16.5% | 18.7% | 13.4% | *Accuracy* | **selectKBestFeatures** |
| 22 | 120 | 2 | *Features* |
| 18.3% | 18.5% | 16.8% | *Accuracy* | **selectKExtraTree** |
| 10 | 102 | 1 | *Features* |

* Normalized Data :

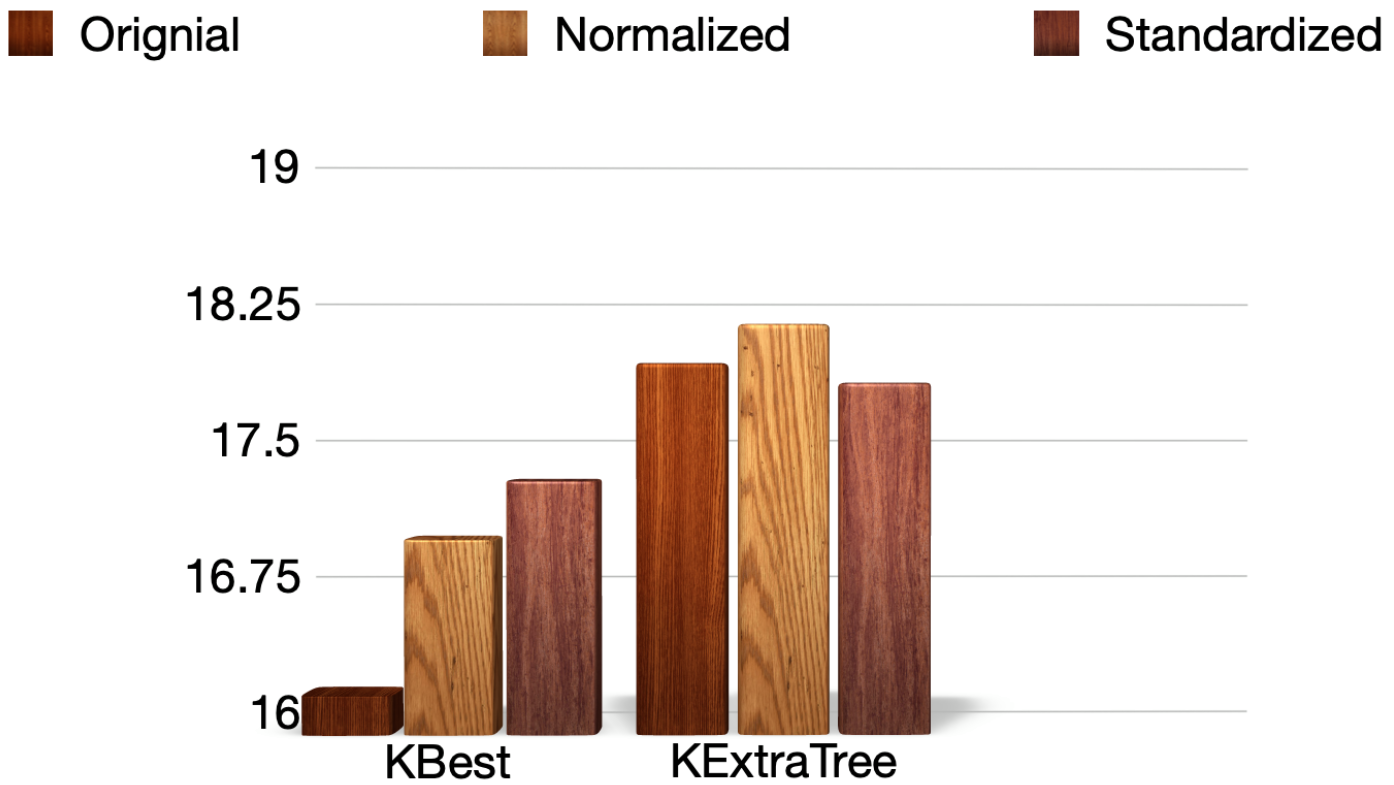
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **KNN** | **Random Forest** | **SVM** |  | |
| 16.5% | 18.6% | 15.9% | *Accuracy* | **selectKBestFeatures** |
| 7 | 120 | 1 | *Features* |
| 17.6% | 18.3% | 18.4% | *Accuracy* | **selectKExtraTree** |
| 8 | 6 | 2 | *Features* |

* Standardized Data :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **KNN** | **Random Forest** | **SVM** |  | |
| 18% | 18.7% | 15.3% | *Accuracy* | **selectKBestFeatures** |
| 7 | 120 | 109 | *Features* |
| 17.6% | 18.2% | 17.7% | *Accuracy* | **selectKExtraTree** |
| 4 | 49 | 17 | *Features* |

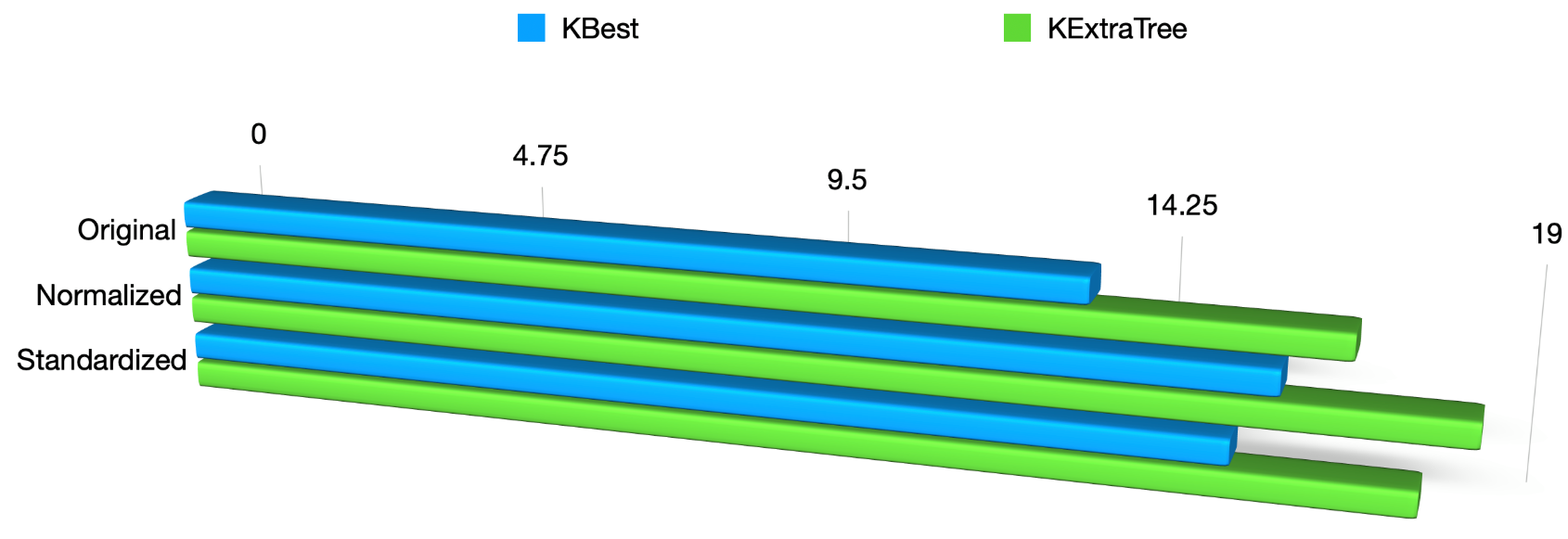
מסקנות :

1. **selectKExtraTree** מביא לתוצאות טובות יותר מ-selectKBestFeature.



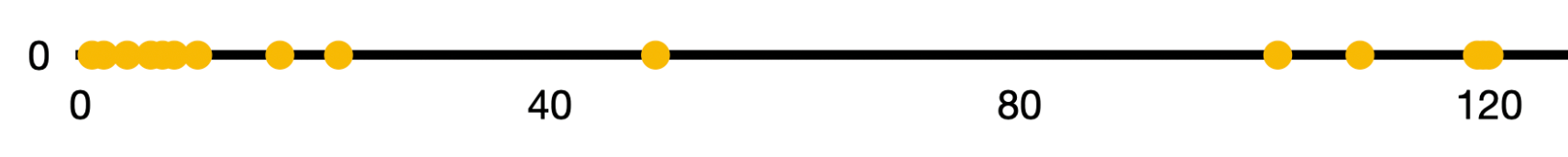
1. **נרמול** מאחד הסוגים מביא לתוצאות טובות יותר מאשר ה-Data הגולמי. זאת לאור שהתכונות במסווג בעלות טווח ערכים משתנה מאוד, ובהתאם גם לכל פיצ'ר יש תוחלת ושונות ייחודיים.

בגרף : תוצאות של Data מנורמל לעומת תוצאות של Data גולמי לפי אלגוריתם SVM.



1. **האלגוריתמים לא מתאימים לבעיות הקשורות באופן הדוק ל-Time Sequences.** ניתן לראות זאת הן **ברמת הדיוק הנמוכה** – מה שמעיד על יכולת דלה של הכללה, והן **ברמת השונות של מספר הפיצ'רים** שנלקחים בחשבון – מה שמעיד על אקראיות בסיווג.

בגרף : פיזור של מספר הפיצ'רים שנבחרו בניסויים השונים שתוארו.



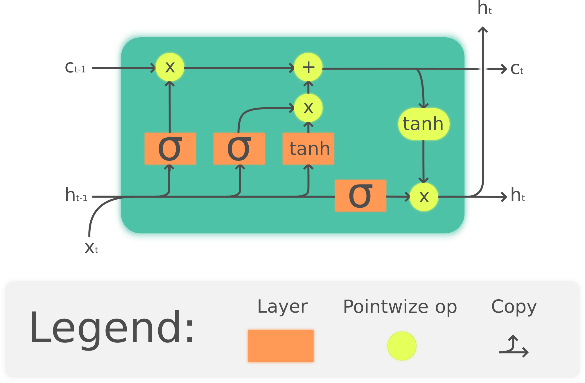
**חקירת הבעיה באמצעות Deep Learning Algorithms ובפרט LSTM**

רקע :

רשת LSTM הינה ארכיטקטורת RNN (Recurrent Neural Network). בניגוד ל-FeedForward Neural Network, ל-LSTM יש Feedback connections שמאפשרים לרשת להתייחס לאוסף הדוגמאות כאוסף סדרתי.

דוגמה טובה לכך היא ילד שרואה סרט ובו סצנה של שני אנשים רצים עם כדור. אם על סמך סצנות קודמות הוא יודע שמדובר בסרט על כדורסל, הוא יסיק שבסצנה ישחקו כדורסל בסבירות גבוהה. ההסקה הזו נובעת מההתייחסות לסצנה (הדוגמה) כמידע עוקב לסצנות קודמות, ולא כ-Single data point.

הייחוד העיקרי של LSTM הם האופרטורים בתוך תא ה-LSTM, שמאפשרים לו לזכור חלק מהמידע ולשכוח חלק אחר.



עקב כך, מספר מניפולציות קודמות עשויות להשפיע, לטובה או לרעה, על רמת הדיוק של המסווג :

* **שימוש ב-data לא רציף** : כזכור, פיצלנו את ה-data על פי משברים. כתוצאה מכך LSTM מקבל דוגמאות רק מתקופות משבר והמידע איננו רציף.
* **שימוש ב"פוטנציאל הרווח"** : כזכור, הגדרנו "פוטנציאל רווח" כפונקציה של מחירי מינימום ומקסימום בתוך משבר מסוים. כלומר, פונקציית התיוג שלנו קשורה קשר הדוק בהיות ה-data מחולק למשברים. ויתור על החלוקה למשברים – משמעותו ויתור על נוסחת "פוטנציאל הרווח".
* **שימוש ב-lookback** : לרשת LSTM, כפי שהוסבר, יש מנגנוני זיכרון מובנים, וניסיון "להעמיס" עליה זיכרון באופן ידני עשוי להביא למצב של over-information.
* **שימוש בנתונים חודשיים** : הנתונים החודשיים "נמרחים" בצורה זהה על פני ימים באותו חודש ועשויים להשפיע על התוצאות , כתכונות שאינן מפרידות היטב בין הדוגמאות.

ניסויים :

על מנת לבחון את ההשפעות של הנקודות שהועלו לעיל, ולבחור אסטרטגיה להמשך, נבצע מספר ניסויים ובהם נבדוק את טיב התוצאות על קבוצת מבחן :

1. שימוש ב-data רציף מול שימוש ב-data תלוי-משברים. על מנת להשוות בין השניים, ננסה בכל יום לחזות את מחיר הסגירה של יום המסחר הבא (שכן פונקציית התיוג המקורית איננה מוגדרת היטב עבור data רציף). את מדידת הדיוק נבצע על סמך המרחק הממוצע וסטיית התקן של החיזוי מהתיוג האמיתי. כמו כן, נציג את התנהגות ה-Loss function בכל אחד מהמקרים.

*תוצאה* :

|  |  |  |
| --- | --- | --- |
|  | **לא רציף** | **רציף** |
| Predict VS Ground Truth | מרחק ממוצע : 140.9$  סטיית תקן : 66$ | מרחק ממוצע : 89.4$  סטיית תקן : 76.3$ |
| Loss Function |  |  |

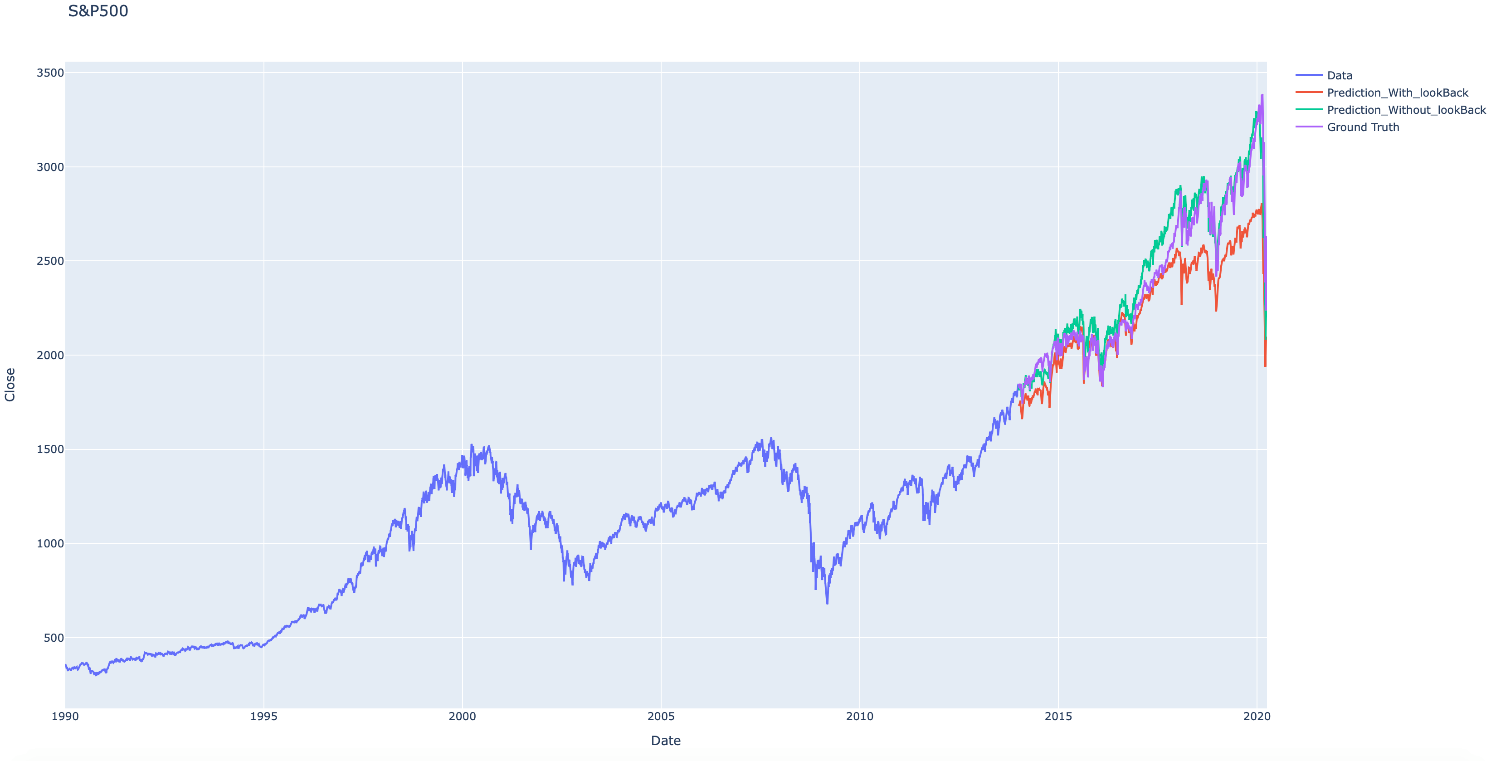
*מסקנה* :

הרשת מתמודדת טוב יותר עם Data רציף. הסיבה לכך עשויה להיות הרציפות עצמה, או העלייה הדרמטית במספר הדוגמאות כתוצאה מכך שלא עושים filter out לימים שאינם במשבר (בערך פי 3 דוגמאות).

1. שימוש ב-lookback אל מול אי שימוש. את מדידת הדיוק נבצע ע"י השוואת שני החיזויים לתיוג האמיתי לאורך זמן.

*תוצאה :*

|  |  |  |
| --- | --- | --- |
|  | **With Lookback** | **Without Lookback** |
| Predict VS Ground Truth | מרחק ממוצע : 161$  סטיית תקן : 138.3$ | מרחק ממוצע : 71.5$  סטיית תקן : 57.7$ |
| Loss Function |  |  |



*מסקנה* :

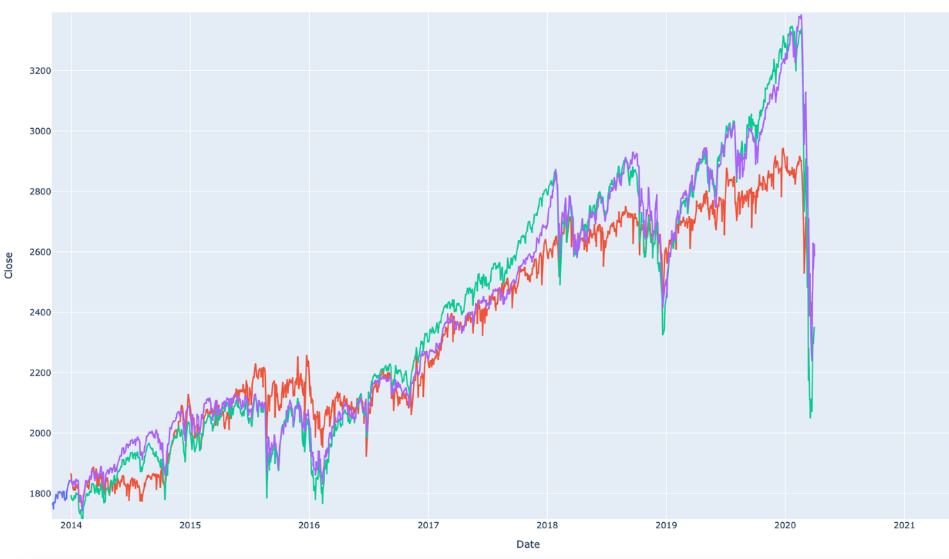
החיזויים ללא Lookback טובים יותר, ואת ההשוואות הבאות נעשה על בסיס נתונים ללא Lookback.

הערה – שימו לב ל-Scaling של פונקציות ה-Loss. אף על פי שלמראית עין With Lookback Loss Function נראית טובה יותר, בפועל הביצועים של Without Lookback Loss Function טובים משמעותית.

1. שימוש בתכונות יומיות בלבד אל מול שימוש בתכונות יומיות וחודשיות יחד. את מדידת הדיוק נבצע ע"י השוואת שני החיזויים לתיוג האמיתי לאורך זמן.

*תוצאה :*

|  |  |  |
| --- | --- | --- |
|  | **With Month Attributes** | **Without Month Attributes** |
| Predict VS Ground Truth | מרחק ממוצע : 48.9$  סטיית תקן : 41.2$ | מרחק ממוצע : 92.9$  סטיית תקן : 88.6$ |
| Loss Function |  |  |



*מסקנה :*

על אף פי שהתכונות החודשיות נמרחות על פני הימים השונים, הן מסייעות משמעותית בסיווג, וכדאי להשאיר אותן.

המסקנה הכוללת מהניסויים היא שאסטרטגיית הלמידה היא הבאה :

* נשתמש ב-Data רציף ולא תלוי משברים.
* נחזה את מחיר ה-Close ולא את פוטנציאל הרווח. בהמשך נסביר איך חוזרים להתחשב בפוטנציאל הרווח שהגדרנו.
* נחזה על סמך התכונות ללא Lookback.
* ניקח בחשבון את התכונות החודשיות.

Hyper Parameters Tuning :

כעת, כשבחרנו אסטרטגיה, יש מספר היפר-פרמטרים באלגוריתמים שניתן לכוונן על סמך ביצועים. על מנת להגיע לרמת הסיווג המיטבית, בדקנו את כל הפרמוטציות שבאות לידי ביטוי בטווח הערכים הבא :

|  |  |  |
| --- | --- | --- |
| **Description** | **Possible Values** | **Parameter** |
| מספר מחזורי האימון של הרשת. | 18,22,26,30 | Epochs |
| מספר דוגמאות האימון בכל סיבוב. | 32,48,64,72,96 | Batch Size |
| מספר היחידות בתוך תא LSTM. | 30,40,50,60,70,80 | Units |

מדובר בבעיית אופטימיזציה ארבע-מימדית – רמת הדיוק של המסווג כתלות בשלושה פרמטרים. לכן, במקום לנתח את התוצאות על פני טבלה, נשתמש ב"מפת חום". ככל שהתוצאה מרשימה יותר – היא תהיה ירוקה יותר במפה. תוצאות גרועות במיוחד יסומנו באדום. המוטיבציה לשימוש בשיטה זו נעוצה בכך שהרשת מחזירה תוצאות באופן לא דטרמינסטי – ולכן נרצה לראות "אזורים" מוצלחים של תחומי ערכים לפרמטרים, ולאו דווקא ריצה מושלמת אחת.

פונקציית הדיוק תלויה בשני פרמטרים :

* Mean : המרחק הממוצע בין ה-label המקורי לבין prediction label.
* Std : סטיית התקן של המרחק ביניהם.

והרי התוצאות :

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **(Epochs, batch size, units)** | **Mean** | **std** | **(Epochs, batch size, units)** | **Mean** | **std** | **(Epochs, batch size, units)** | **Mean** | **std** | **(Epochs, batch size, units)** | **Mean** | **std** |
| (18,32,30) | 94.3 | 78.9 | (22,32,30) | 230.8 | 194.5 | (26,32,30) | 70.1 | 81.4 | (30,32,30) | 213.2 | 195.4 |
| (18,32,40) | 73.1 | 57.4 | (22,32,40) | 64.7 | 75.6 | (26,32,40) | 50.8 | 61.4 | (30,32,40) | 170.5 | 158.6 |
| (18,32,50) | 58.7 | 65.3 | (22,32,50) | 64.2 | 64.4 | (26,32,50) | 70.8 | 87.9 | (30,32,50) | 235.7 | 187.3 |
| (18,32,60) | 176.2 | 114.8 | (22,32,60) | 115.4 | 73.1 | (26,32,60) | 60 | 57.7 | (30,32,60) | 91.6 | 105.2 |
| (18,32,70) | 57.1 | 38.5 | (22,32,70) | 106.8 | 71 | (26,32,70) | 56 | 66.9 | (30,32,70) | 116.1 | 83.2 |
| (18,32,80) | 114.5 | 124.7 | (22,32,80) | 175.5 | 114.5 | (26,32,80) | 121 | 115.4 | (30,32,80) | 242.3 | 203.1 |
| (18,48,30) | 288.6 | 158.7 | (22,48,30) | 302.7 | 154.8 | (26,48,30) | 383.8 | 180.8 | (30,48,30) | 73.1 | 69 |
| (18,48,40) | 287 | 202.6 | (22,48,40) | 147.3 | 135.3 | (26,48,40) | 167.3 | 112.1 | (30,48,40) | 112.2 | 91.5 |
| (18,48,50) | 88.3 | 51.5 | (22,48,50) | 278.8 | 186.4 | (26,48,50) | 214.9 | 132.6 | (30,48,50) | 296.1 | 142 |
| (18,48,60) | 285.9 | 140.2 | (22,48,60) | 240.7 | 150.6 | (26,48,60) | 152.1 | 123.9 | (30,48,60) | 140.1 | 99.5 |
| (18,48,70) | 164 | 81.5 | (22,48,70) | 107.6 | 73 | (26,48,70) | 86 | 62.5 | (30,48,70) | 343 | 174.3 |
| (18,48,80) | 75.3 | 41.4 | (22,48,80) | 54.7 | 57.2 | (26,48,80) | 245 | 156.2 | (30,48,80) | 171.3 | 139.5 |
| (18,64,30) | 435.3 | 178.4 | (22,64,30) | 65.8 | 58.7 | (26,64,30) | 94.9 | 68.1 | (30,64,30) | 108.8 | 67 |
| (18,64,40) | 181.3 | 84.1 | (22,64,40) | 248.2 | 103.2 | (26,64,40) | 111.5 | 75 | (30,64,40) | 111.2 | 56.1 |
| (18,64,50) | 396.4 | 152.8 | (22,64,50) | 120.4 | 67 | (26,64,50) | 121.5 | 75.5 | (30,64,50) | 125.7 | 64.8 |
| (18,64,60) | 190.9 | 145 | (22,64,60) | 50.7 | 45.9 | (26,64,60) | 184.3 | 106.5 | (30,64,60) | 84.5 | 53.2 |
| (18,64,70) | 343.8 | 130.9 | (22,64,70) | 107.6 | 71.4 | (26,64,70) | 99.5 | 62 | (30,64,70) | 112.1 | 54.6 |
| (18,64,80) | 320.6 | 140.7 | (22,64,80) | 52.6 | 42 | (26,64,80) | 181.1 | 98.3 | (30,64,80) | 185.3 | 87.9 |
| (18,72,30) | 149.2 | 83 | (22,72,30) | 54.5 | 42.1 | (26,72,30) | 182.9 | 96.5 | (30,72,30) | 91.2 | 97.6 |
| (18,72,40) | 79.1 | 75.4 | (22,72,40) | 64.3 | 42.9 | (26,72,40) | 88.1 | 53 | (30,72,40) | 177.8 | 106 |
| (18,72,50) | 160.6 | 52.7 | (22,72,50) | 182.7 | 110.6 | (26,72,50) | 233.6 | 119.2 | (30,72,50) | 95.8 | 108 |
| (18,72,60) | 176.1 | 74.4 | (22,72,60) | 74.1 | 56.1 | (26,72,60) | 135.2 | 73.6 | (30,72,60) | 258.4 | 130.6 |
| (18,72,70) | 229.7 | 83.5 | (22,72,70) | 122.1 | 60.8 | (26,72,70) | 132.2 | 69.5 | (30,72,70) | 59 | 49.1 |
| (18,72,80) | 251.1 | 112.2 | (22,72,80) | 52.5 | 44.5 | (26,72,80) | 178.8 | 96.2 | (30,72,80) | 130.6 | 75.4 |
| (18,96,30) | 87.5 | 58.1 | (22,96,30) | 148.2 | 155.2 | (26,96,30) | 343.5 | 222.5 | (30,96,30) | 327.6 | 190 |
| (18,96,40) | 84.8 | 52.2 | (22,96,40) | 75.2 | 46.8 | (26,96,40) | 108.2 | 118.5 | (30,96,40) | 356 | 223.5 |
| (18,96,50) | 46.1 | 41.4 | (22,96,50) | 126.2 | 74.5 | (26,96,50) | 147.2 | 69.7 | (30,96,50) | 429.2 | 250.9 |
| (18,96,60) | 143.6 | 85.4 | (22,96,60) | 183.9 | 116.5 | (26,96,60) | 185.3 | 68.5 | (30,96,60) | 163.6 | 173.5 |
| (18,96,70) | 57.5 | 49.4 | (22,96,70) | 142.6 | 90.5 | (26,96,70) | 118 | 83.6 | (30,96,70) | 131.1 | 65.8 |
| (18,96,80) | 183.7 | 104.3 | (22,96,80) | 178.5 | 114.1 | (26,96,80) | 258.4 | 218.3 | (30,96,80) | 122.5 | 92.8 |

*מסקנות*

* קבוצת הפרמטרים (Epochs=18,Batch Size=96, Units=50) מביאה לתוצאות האידיאליות ביותר בריצה הנ"ל.
* עבור תוצאות סיווג אידיאליות, קיים יחס הפוך בין Epochs ל-Batch Size. כך למשל, התוצאות הטובות ביותר התקבלו עבור (Epochs=22, Batch Size=72) ועבור (Epochs=26, Batch Size=32).
* נבחר את המסווג עם הפרמטרים (Epochs=26, Batch Size=32, Units=70).
* על מנת להמחיש את ההבדל בביצועים, נשווה בין זוג מסווגים – המסווג הנבחר, ומסווג שנתן תוצאות גרועות :



**יישום אסטרטגיית מסחר על סמך תוצאות המסווג**

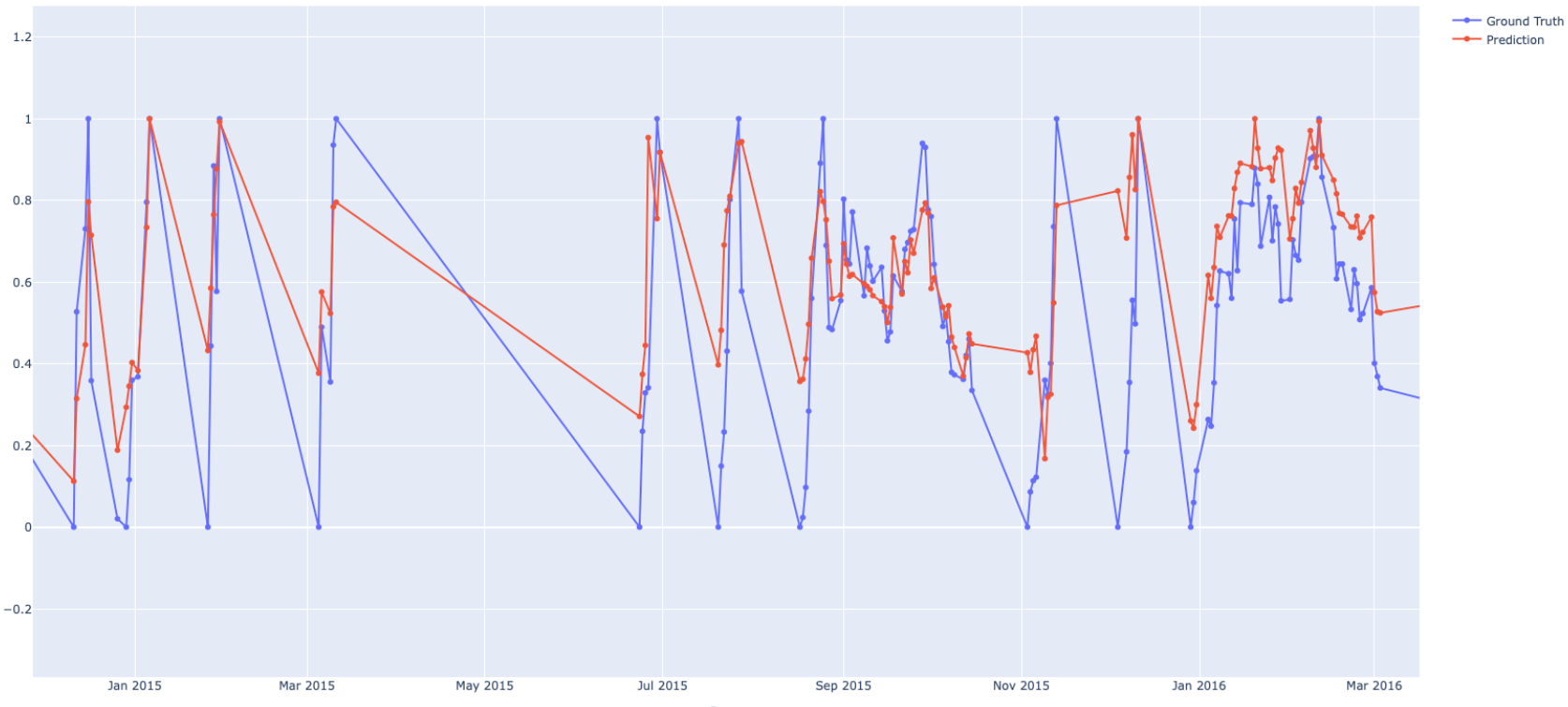
רקע

בחלק זה נחזור להשתמש בפונקציית פוטנציאל הרווח שהגדרנו. המשמעות היא שהמסווג ייתן ציונים בין 0 ל-1 לימים במהלך משבר, וככל שהציון גבוה יותר, כך לכאורה פוטנציאל הרווח עולה.

שאלה שעשויה להישאל היא איך מחלצים את פוטנציאל הרווח ממחיר הסגירה, שכן כפי שראינו בחלק הקודם, הרשת חזתה את מחיר הסגירה (Close). התשובה לכך, אם כן, היא שחיזוי פוטנציאל הרווח יחושב מהחיזוי של Close, כשם שפוטנציאל הרווח האמיתי חושב מה-Close.

הערה : להבדיל מפוטנציאל הרווח האמיתי, חיזוי פוטנציאל הרווח לאו דווקא יהיה בין 0 ל-1 (היות שהוא מבוסס על מחירי סגירה שחזינו ואינם מדויקים). עקב כך, תוצאות המסווג עבור פוטנציאל הרווח ינורמלו.

את התוצאות הנ"ל נרצה לתרגם לאסטרטגיית מסחר. בחלק זה נציע מספר אסטרטגיות שמהוות נוסחה סגורה, ומסתמכות על ניבויי המסווג בלבד. נבחן את ביצועי האסטרטגיות האלה בהינתן סכום כסף מסוים ותחת משברים מקבוצת ה-test שלנו. כלומר, נבדוק, בהינתן סכום התחלתי של 100K ואסטרטגית מסחר, כמה כסף הצלחנו להרוויח במהלך משבר מסוים.



אסטרטגיה #1 – All In Strategy

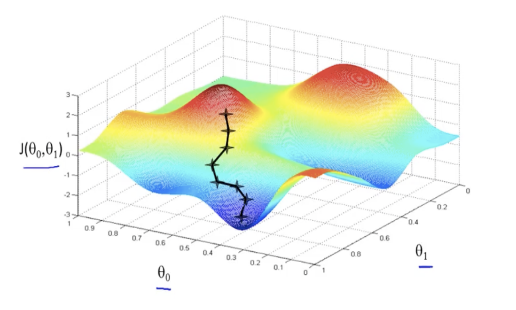
רעיון : התאמה לאנשים שלא רוצים להתעסק במסחר יום-יומי, אלא להשקיע את מיטב כספם בנקודה יחסית בטוחה, ולממש בנקודה משתלמת.

אלגוריתם :

1. בכל יום, חשב את פוטנציאל הרווח profit\_day.
2. קנה בכל כספך את נייר הערך ביום הראשון בו profit\_day גדול מ-0.8.
3. ממש בכל כספך את נייר הערך ביום הראשון בו profit\_day קטן מ-0.5.

אסטרטגיה #2 – Gradient Strategy

רעיון : מעט הסתמכות ע vל ביצועי המסווג ביחס למציאות, ויותר הסתמכות על ביצועי המסווג ביחס לעצמו (ביום האתמול)

1. בכל יום, חשב את פוטנציאל הרווח profit\_day, ואת פוטנציאל הרווח של אתמול profit\_yesterday.
2. נסמן : profit\_gradient – הפרש הפוטנציאלים. invested\_money – סכום הכסף המושקע ברגע נתון. uninvested\_money – סכום הכסף שנותר לא מושקע ברגע נתון.
3. בכל יום בו profit\_day גדול מ- profit\_yesterday :
   1. סכום הכסף להשקעה = uninvested\_money \* abs(profit\_gradient).
4. בכל יום בו profit\_day קטן מ-profit\_yesterday :
   1. סכום הכסף למימוש = invested\_money \* abs(profit\_gradient).

אסטרטגיה #3 – Calculated Risk Strategy

רעיון : הסתמכות נרחבת על ביצועי המסווג.

1. בכל יום, חשב את פוטנציאל הרווח profit\_day.
2. נסמן : invested\_money – סכום הכסף המושקע ברגע נתון. uninvested\_money – סכום הכסף שנותר לא מושקע ברגע נתון.
3. בכל יום בו profit\_day גדול מ-0.6 :
   1. סכום הכסף להשקעה = uninvested\_money \* profit\_day \* 0.5.
4. בכל יום בו profit\_day קטן מ-0.4 :
   1. סכום הכסף למימוש = invested\_money \* (1- profit\_day) \* 0.5.

**ביבילוגרפיה**

|  |  |
| --- | --- |
| <https://www.maariv.co.il/business/world/Article-760391> | [1] |
| <https://he.wikipedia.org/wiki/%D7%94%D7%9E%D7%A9%D7%91%D7%A8_%D7%94%D7%9B%D7%9C%D7%9B%D7%9C%D7%99_%D7%94%D7%A2%D7%95%D7%9C%D7%9E%D7%99_(2008)> | [2] |
| <https://zoefin.com/learn/market-timing-vs-time-in-the-market/> | [3] |
| <https://www.codecademy.com/articles/normalization> | [4] |
| <https://he.wikipedia.org/wiki/%D7%9E%D7%9B%D7%95%D7%A0%D7%AA_%D7%95%D7%A7%D7%98%D7%95%D7%A8%D7%99%D7%9D_%D7%AA%D7%95%D7%9E%D7%9B%D7%99%D7%9D> | [5] |
| <https://towardsdatascience.com/understanding-random-forest-58381e0602d2> | [6] |
|  |  |