פרויקט סוף קורס- Machine Learning

שם: מיכאל בן עמוס

ת.ז: 203862263

מייל: michaelba8@gmail.com

קישור ל- GIT: https://github.com/michaelba8/Bitcoin-prediction-ML-project

תיאור הבעיה:

מבוא:

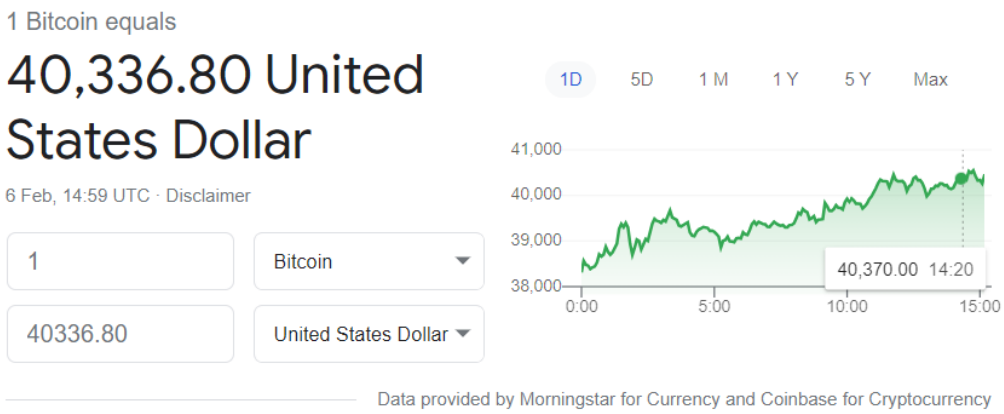
בשנים האחרונות פרצו לתודעה המטבעות הדיגיטליים ובראשם כמובן המטבע הראשון והמפורסם ביותר Bitcoin (ביטקוין).

הרעיון של המטבע הדיגיטלי הוא להחליף בסופו של דבר את המטבעות הרשמיים של המדינות בעולם, ובעצם בגדול להעלים את הצורך בבנקים.

ל bitcoin כמו לכל מטבע יש ערך, בניגוד למטבעות מוחשיים, הייצור של מטבע זה הוא מוגבל וכיום, תהליך הייצור של המטבעות נמוך מאוד, ועקב הביקוש הרב שיש למטבע זה, הערך שלו עלה וממשיך לעלות בצורה מסחררת.

לצורך העניין בתאריך ה- 10.06.2016 ערכו של מטבע bitcoin אחד היה $610, היום כעבור פחות מ-5 שנים ערכו הנוכחי (לתאריך 06.02.2021) הוא $40,336, **עליה** של מעל 6500%.

עם זאת, בהסתכלות קרובה יותר על גרף ה- Bitcoin, ערכו לא רק עולה, אלא עולה ויורד לסירוגין בצורה מאוד לא יציבה ועם תדירות שינוי מאוד גבוהה, לפניכם תמונה של גרף ביטקוין ביום רגיל הפרוסה על 15 שעות, ניתן לראות כמה הגרף "קופצני" ולא יציב.



המטרה:

אחרי הרקע הקל למטבע ה-Bitcoin נעבור למטרת הפרויקט, בהתחשב באופי הגרף וחוסר היציבות של המטבע, נרצה לפתח בוט שסוחר ב- Bitcoin למטרות רווח כספי, הבוט יודע לבצע 2 פעולות פשוטות:

1. קנייה ומכירה (לונג):
פעולה זאת היא פעולת קנייה של כמות מסוימת של ביטקוין (פוזיציה) בהנחה שהוא יעלה בערכו ואז מכירה כאשר הוא הגיע לערך הרצוי, ובכך להרוויח את ההפרש (מינוס עמלה).

2. פעולת מכירה וקנייה (שורט):
זאת אפשרות שניתנת בכל זירות המסחר של המטבעות הדיגיטליים, וזה כאשר חוזים כי הערך הנוכחי הוא גבוה והוא הולך לרדת, אז 'מוכרים' בזירת המסחר (לא חייב להחזיק בביטקוין לפני) כמות מסוימת של ביטקוין, ואז כאשר הוא יורד במחיר 'קונים' אותו חזרה וכך מרוויחים את ההפרש של הירידה, כמובן שאם הערך שלו עולה אז מפסידים.

עמלה- כל פעולה כזאת מחייבת בעמלה מטעם זירת המסחר, של 0.1% מהערך בו סחרת, לכן הבוט יעבוד בצורה הבאה, עבור ערך קבוע הניתן לשינוי בקוד, שהגדרנו כרגע 0.2% :

לונג-

שורט-

בשני המקרים, נחכה לראות עליה/ירידה של הערך המצופה. אם הערך יעלה/ירד לכיוון הנגדי ממה שצפינו ב-0.1%, נודה בהפסד, ונצא מהפוזיציה בהפסד של 0.2%

איפה נכנס ה- Machine Learning?

על מנת שהבוט הזה אכן ירוויח, נרצה לדעת לחזות את העליות/ירידות בערך על פי נתונים הניתנים להשגה, ולכן נבנה פונקציה המבוססת על מודל של ML, החוזה אחת לדקה האם היא צופה שבדקות הקרובות הערך יעלה/ירד ב0.2% או יישאר בטווח בו הוא נמצא.

ה- DATA

לאחר חיפוש מעמיק באינטרנט, השגתי דאטה המכיל כל דקה אפשרית של 21 חודשים, מחודש מרץ 2019 עד אוקטובר 2020. זהו דאטה שפורסם באופן חינמי, על ידי זירת מסחר הביטקוין הגדולה בעולם binance.com . כל שורה בדאטה מייצגת דקה אחת ומכילה את 12 הפרמטרים הבאים:

OpenTime- זמן התחלה open- ערך בדולרים בתחילת הדקה high- ערך גבוה ביותר

Low- ערך נמוך ביותר close- ערך בסוף הדקה volume-פרמטר סחר מסוים

closeTime- זמן סיום Trades- מספר עסקאות Ingored- ללא משמעות

ו- takerBaseAssetVolume, takerQuoteAssetVolume , quoteAssetVolume נפח של מסחר מסוים

וקטור ה- Y:

על מנת להגדיר וקטור Y שעליו ילמד המודל, בניתי פונקציה שרצה על הדאטה, ועבור כל דקה נבחר את ערך המטבע בסוף הדקה (close) נקרא לו val, נרוץ קדימה על הדאטה, ונחפש ערך (ב high או ב low) גדול או קטן מ

val ב0.2% או יותר. אם הוא גדול נסמן את ערך ה-Y להיות 1, אם הוא קטן נסמן 1- ואם עברו כמות דקות מסוימת (פרמטר של הפונקציה אני עבדתי עם 3-10 דקות), ולא היה שינוי מהטווח של 0.2%+- נסמן 0.

זאת הגדרה השקולה למטרה שלנו, כשנקרא בלייב את נתוני הדקה האחרונה ניקח את ערך הסיום של הדקה (הערך הנוכחי), וננסה לחזות האם הערך צפוי לעלות או לרדת בטווח של 0.2%

עבודה על ה- DATA :

פרמטרים שהוסרו עוד לפני תחילת העבודה:

- Ignored- פרמטר חסר משמעות שערכו תמיד 0
- CloseTime, OpenTime -מייצגים את זמן ההתחלה והסיום של הדקה, לא רלוונטים לחיזוי הערך

פרמטרים ששונו:

ערך הביטקוין תמיד במגמת עליה לכן ערך ארבעת התכונות שמייצגות את הערך בדאטה שלנו (high low, open, close) יעלו באופן כללי בהתאם לזמן בו נבדקו.

אלגוריתמי ה- ML הם לינארים ולכן יבנו מודל שגוי עבור ה-DATA הזה. לכן מה שעשיתי זה במקום להשתמש בערך האבסולוטי שלהם, השתמשתי ביחס בין כל ה-4 וכך עליית הביטקוין לא תשפיע על תהליך הלמידה, ובנוסף מגמת העלייה/ירידה תקבל ביטוי מספרי בדאטה. לקחתי את הערכים high, low, close וחילקתי אותם ב-open.לאחר מכן את הפרמטר open הסרתי מהדאטה.

כעת יש ברשותנו 3 תכונות חדשות עבור כל דקה:

- היחס בין הערך המקסימלי לערך ההתחלתי (high)
- היחס בין הערך המינימלי לערך ההתחלתי (low)
- היחס בין הערך הסופי לערך ההתחלתי (close)

ככל שהתקדמתי בפרויקט הבנתי ש-3 התכונות הללו הן התכונות העיקריות שמשפיעות על המודל, בנוסף לתכונות האלה הוספתי גם את התכונה trades (מספר עסקאות בדקה), ואת כל שאר התכונות הסרתי מהדאטה, כיוון שהן גורמות ל-overfitting.

בדיקת דיוק מותאמת:

לפני שנמשיך לתצוגת הגרפים והסברים על מימוש האלגוריתמים, ארצה להגדיר בדיקת דיוק קצת שונה:

עבור בדיקת דיוק רגילה עם המטריצה המוכרת נקבל מצבים שאינם רלוונטים למטרת הפרויקט:

חיזוי של מחלקה 0:

מצב שאינו מעניין, אם המודל חוזה שהערך לא ירד ולא יעלה בדקות הקרובות, אז פחות חשוב לי לדעת אם הוא צדק או לא צדק, כיוון שהבוט לא יעשה שום פעולה בכל מקרה ולכן לא ירוויח ולא יפסיד.

לכן בניתי טסט קטן שבודק את הדיוק רק כאשר המודל חוזה 1 או 1-

בנוסף בטסט הנ"ל שילבתי פונקציה שבניתי שחוזה בעזרת המודל את המחלקה, אך פונקציה זאת תחזה את המחלקה עבור הסתברויות גדולות יותר אשר מתקבלות כ-input (בחרתי 62% ,בניגוד לערך הדיפולטיבי שהוא 34% עבור 3 מחלקות). זאת על מנת לצמצם עוד את הסיכון להפסד.

ולכן הייתי צריך גם ליצור משתנה שמחזיק את היחס בו הפונקציה מחליטה על פעולה (שורט/לונג) ביחס לפעמים שנקראה (אחת לדקה). משתנה זה קראתי לו attack_ratio, וצריך למצוא איזון בינו לבין אחוז הדיוק כדי להגיע לרווח מקסימלי.

הרצת אלגוריתמי הלמידה

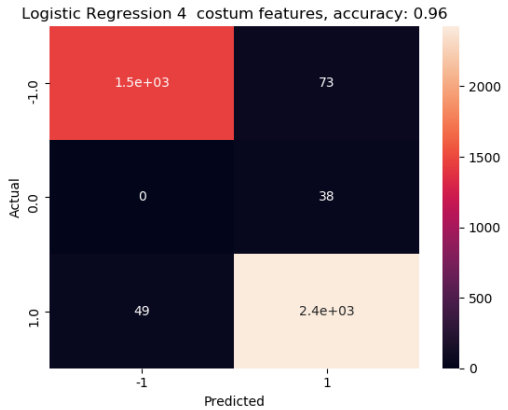
1. (חשוב לציין שכל התחזיות בגרפים הבאים הן תחזיות מותאמות עם הסתברות של 62% ומלעה על פי המודל)

-האלגוריתם הראשון בו השתמשתי הוא Logistic Regression של הספריה sklearn.

גרף דיוק מותאם:

Accuracy: 96%

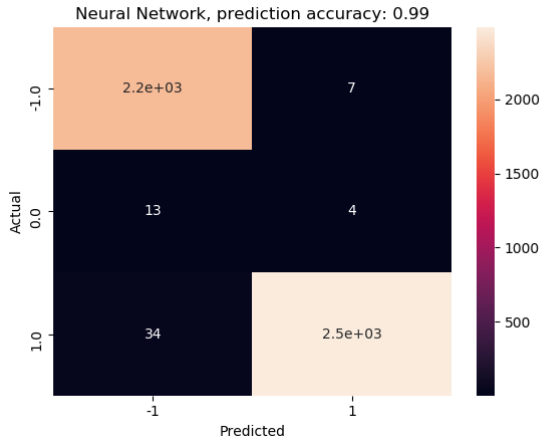
Attack ratio: 5%



-האלגוריתם השני שבו השתמשתי הוא רשת נוירונים גם של הספריה sklearn

Accuracy: 99%

Attack ratio: 5.8%



2.פרמטרים וכוונון אלגוריתמים:

Logistic regression:

C- השתמשתי בפונקציה מאחד מתרגילי הבית על מנת למצוא את ה-C הטוב ביותר, שורה זאת נמצאת בקוד תחת הערה, עקב זמן ריצה גדול (דאטה של 810,000 דוגמאות). ה-C הטוב ביותר הוא 10^(-4), אך בחרתי ב-c=10, אסביר בהמשך מה הסיבה.

רשת נוירונים:

Alpha- האלפא המומלצת על ידי הספריה היתה 1e-05, פרמטר אשר נתן דיוק גבוה מאוד מבחינת צרכי המודל, אבל ההתכנסות הייתה יחסית מאוד איטית. ככל שהגדלתי את האלפא (פי 10) קיבלתי התכנסות מהירה יותר אך תוצאות פחות טובות. לבסוף השארתי אותו 1e-05.

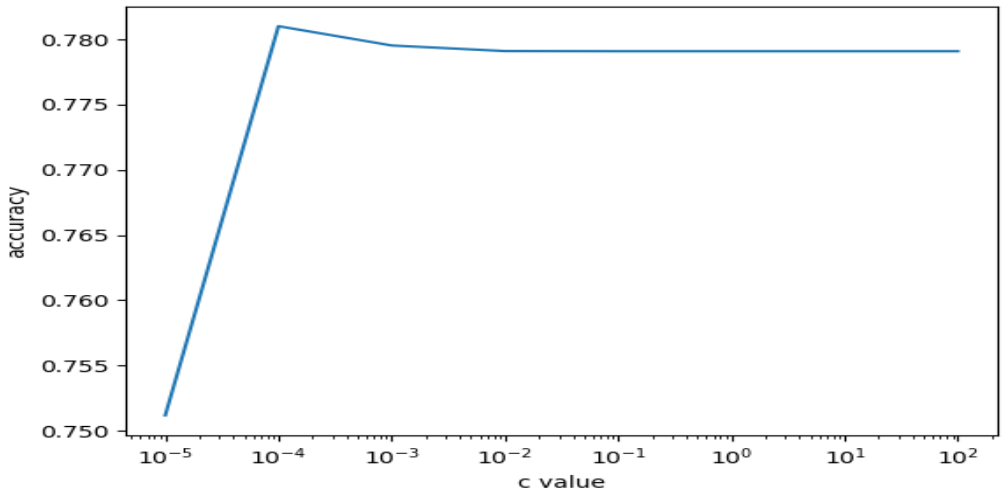
מספר שכבות- (14,3), שוב ככל שהגדלתי את המספר השכבות (14) כך קיבלתי תוצאה גדולה יותר אך זמן הריצה עבור בניית המודל היה גדול מאוד.

.3

Logistic regression:

בגרף הבא ניתן לראות כי ה-C הטוב ביותר מבחינת דיוק כללי היה 0.0001, אך לצרכים שלשמם נעשה הפרויקט הזה, ה-C הזה אינו מספיק טוב, וזאת כיוון שהוא מדייק מעט יותר אבל רק בגלל שהוא חוזה יותר פעמים את המחלקה 0 (לא צופה שינוי מהותי בדקות הקרובות), ולכן הוא אולי מדייק יותר, אבל זהו חיזוי שלא משרת את המטרה של המודל.

ולכן ה-C שנבחר כך שנותן את הדיוק הטוב ביותר הוא C=10. אמנם הדיוק באחוז או שניים נמוך יותר, אך זה בגלל שהוא "מסתכן" יותר.



רשת נוירונים:

עקב זמני ריצה מאוד גבוהים ובעיקר מאמץ רב מאוד של המחשב, לא יכולתי לבנות גרפים המשווים בין כמה רשתות נוירונים עם פרמטרים שונים, ולאחר מספר הרצות עם פרמטרים שונים הגעתי למסקנה שהפרמטרים האופטימליים עבור רשת הנוירונים הם אלו שציינתי למלעה, 14 שכבות + alpha=1e-5

4.תכונות חשובות ביותר:

רציתי להשתמש באלגוריתם mutual information בו השתמשתי באחד התרגילים, אך מסתבר שהוא אינו מתאים כל כך עבור הדאטה הזה, והתוצאות שקיבלתי ממנו סתרו את התוצאות בשטח כשהרצתי את המודל. לכן בדקתי כל אחת מן התכונות בנפרד וראיתי מה הדיוק שמתקבל עבור כל תכונה בנפרד, את התוצאה הכנסתי לרשימה ממוינת (חשובה ביותר משמאל לימין) וזאת הרשימה:

```
['num_trades', 'low_ratio', 'high_ratio', 'qav', 'volume', 'close_ratio', 'taker_base_vol', 'taker_quote_vol', 'open_time', 'open', 'high', 'low', 'close', 'close_time', 'ignore']
```

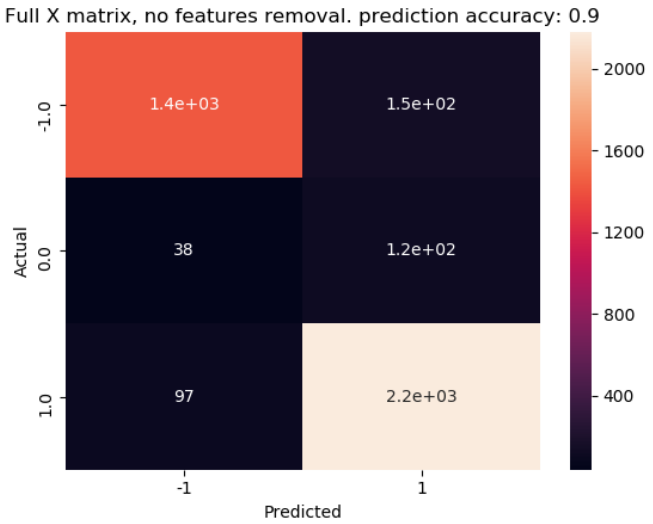
(התכונות low_ratio, high_ratio, close_ratio , אלו התכונות המותאמות שיצרתי מתוך הדאטה)

כמובן שנכונות הרשימה לא מדויקת במאה אחוז מול התוצאות בשטח (לאחר אינספור הרצות שונות ובדיקות), אך התוצאה קרובה מאוד.

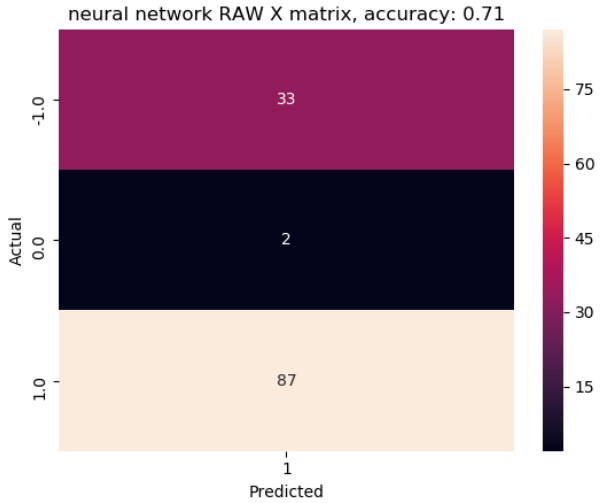
5. על מנת להתמודד עם Overfitting, כמו שצוין למלעה זרקתי מספר תכונות שההשפעה שלהן על החיזוי היא קטנה והם פוגעים בדיוק המודל.

זוהי תוצאה של רגרסיה לוגיסטית אשר לא נעשה שום עיבוד על מטריצת ה-X שהוכנסה בצורה נקייה (נרמול בלבד) שוב עם הסתברות 62% ומלעה. כאשר ערך המשתנה attack_ratio=3.5%

לכן אפשר להסיק שלא רק שנפגע הדיוק ב-5% אלא גם יחס החיזויים של 1/1- מול המחלקה 0 נפגע בצורה משמעותית, ולכן הפגיעה ביעילות המודל היא כפולה, בנוסף האלגוריתם לא התכנס לאחר 300 איטרציות.



תוצאה של רשת נוירונים, ללא עיבוד מקדים על הדאטה (רק נרמול) של מחיקה ושינוי תכונות:



כמו שניתן לראות תוצאה חלשה מאוד, במיוחד כשהמשתנה attack_ratio=0.1%. אפשר להגיד בצורה חד משמעית שזהו מודל כושל (בנוסף הוא חוזה רק עליות ולא ירידות), הנובע בעיקר מ-overfitting

וכבונוס אני אראה לסיום את התוצאה של רשת נוירונים עם הסתברות גבוהה מ-95%

Attack_ratio=5.3%. תוצאה מרשימה מאוד, ולכן אני חושב שזה יהיה המודל שירוץ על הבוט שאני אבנה. (נעשתה סימולציה על 25 שעות אחרונות 11.2.2021 ויצא רווח של 10%-12%)