**היבטים מעשיים בכריית תוכן אינטרנטי לאפליקציות עסקיות – פרויקט סיום**

**נושא הפרויקט:** Stock Market Prediction

לקליטת טבלת הנתונים עליה יבוצע העיבוד והמחקר, תחילה יש להזין נתיב המוביל למיקומה:

**שאלה 1 –Data Exploration**

* השלב הראשון שהיינו צריכים לבצע לצורך data exploration הוא לנקות ולנרמל את ה-data:

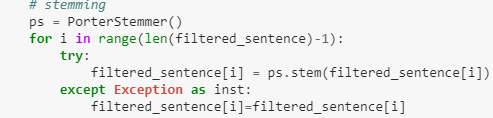
1. ביצוע Lowercase עבור כל מילה בטקסט:



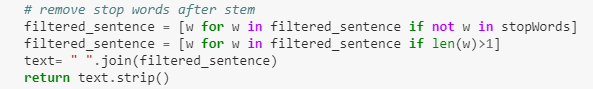
1. ביצוע Tokenizing עבור הטקסט (הפרדה למילים בצורת טוקן):



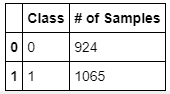
1. ביצוע Stemming עבור כל טוקן, תוך שימוש באלגוריתם porter (לא השתמשנו באלגוריתם Lemmatization מכיוון שלא רצינו להביא כמה מילים עם אותה משמעות, לאותה צורה. כלומר, רצינו לשמר את המילה המקורית שנכתבה בטקסט ורק לנרמל אותה לצורה הבסיסית שלה, ע"י "חיתוך" הסוף שלה).



1. הורדת Stop Words מן הטקסט, בכדי להתעלם ממילים נפוצות מידי בשפה האנגלית, אשר עלולות "להשתלט" על הטקסט וכתוצאה מכך להשפיע על התפלגויות המילים האחרות בידיעות (אותן אנו רוצים לחקור).



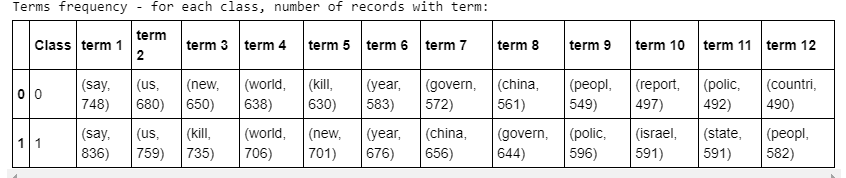
* לאחר ניקוי הנתונים, בדקנו אם הסיווג הבינארי הנתון הינו מאוזן/לא מאוזן (כלומר, אם רוב מוחץ של הרשומות שייכות לסיווג מסוים – 0 או 1), לכן בדקנו כמה רשומות יש עבור כל סיווג:



ניתן לראות כי התפלגות הרשומות עבור כל Class הינה תקינה ומאוזנת (כלומר, אין רוב רשומות תחת class מסוים).

* בדיקת תדירויות **12** ה-terms הכי נפוצים בכל אחת משתי המחלקות, **בשני האופנים:**

1. עבור כל class, בדקנו **בכמה רשומות** מופיע כל term (כלומר, אם הופיע באותה רשומה כמה פעמים אז ספרנו זאת כפעם אחת בלבד לאותה רשומה).



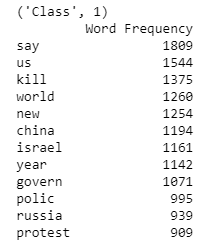
ניתן לראות כי בשתי המחלקות יש מילים נפוצות (כלומר, ברשומות רבות) משותפות (לאחר שעברו תהליך stemming): say, us, new, world, kill, year, govern, china, people, polic (10 מילים סה"כ, מתוך 12).

יתרה מכך, מספר המופעים (התפלגויות) של כל אחת ממילים משותפות אלה דומה. לדוגמה:

* Say מופיעה במחלקה אפס 748 פעמים ובמחלקה אחת 836 פעמים.
* Us מופיעה במחלקה אפס 680 פעמים ובמחלקה אחת 759 פעמים.
* New מופיעה במחלקה אפס 650 פעמים ובמחלקה אחת 701 פעמים.

כלומר, לא ניתן להסיק באופן מובהק איזה מילים נפוצות **ייחודיות** **למחלקה** עלולות להצביע על ירידת/עליית מדד המנייה.

1. עבור כל class, בדקנו **כמה פעמים** מופיע ה- .term



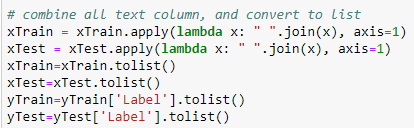
כאמור, **גם** בבדיקת התפלגות המילים **בדרך ב'** ניתן לראות כי רוב המילים הנפוצות משותפות לשתי המחלקות, ובעלות התפלגות דומה. בנוסף, במחלקה 1 ניתן לראות כי המילה kill מופיעה ברשימת 12 המילים הנפוצות למחלקה – זו היא מילה אשר היינו מצפים כי תופיע **רק** במחלקה 0 (לאור אופייה השלילי ובכך שמחלקה 0 מזוהה עם ירידת מדד המנייה). כמו כן, מילה בעלת אופן שלילי נוסף אשר מופיעה במחלקה 1 היא protest.

**שאלה 2 – Building Machine Learning classifiers**

בכדי שנוכל לאמן ולבחון כל אחד מן המודלים, חילקנו את ה- data לשתי קבוצות:

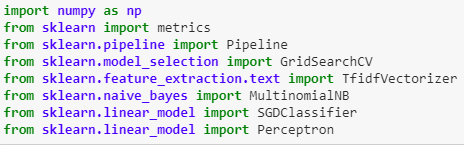
* קבוצת אימון (Train) – 70 אחוז מרשומות ה-data.
* קבוצת מבחן (Test) – 30 אחוז מרשומות ה-data. 

(xTrain, yTrain עבור קבוצת האימון ו- xTest, yTest עבור קבוצת המבחן)

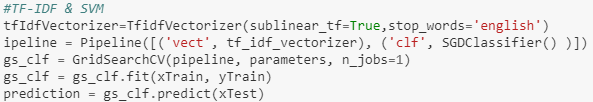
לאחר מכן, המרנו את הקבוצות הנ"ל לצורת רשימה, בכדי שנוכל לאמן ולבחון את המודל:

המודל הראשון בו השתמשנו הוא מסוג Non-Keras:

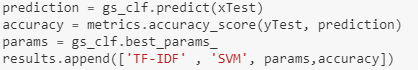
* השתמשנו במודל מסוג SVM (SGD classifier).
* קריאה לספריות הרלוונטיות:



* הסיווג נעשה באמצעות שלושה פרמטרים:
* Feature extraction של הנתונים.
* שיטת Machine Learning באמצעותה יתבצע האימון והסיווג.
* פרמטרים של למידת המכונה.



* כפי שניתן לראות, יש שימוש ב – pipeline וב- gridSearch, אשר מחזירים את הפרמטרים והתוצאות הטובים ביותר, עם הפרמטרים אשר קיבל המסווג. אימון המודל התבצע על קובץ האימון (train), ודיוק המודל נבדק על קובץ המבחן (test).
* השתמשנו ב- feature extraction בצורת tf-idf, מכיוון שרצינו להמיר את טקסט הקלט לצורה שאיתה תוכל לעבוד למידת המכונה (צורת וקטור) – כפי שנלמד בכיתה. בצורה זו, החילוץ של טקסט הפוסטים הינו יעיל יותר.
* בחירת שילובים שונים של פרמטרים ללמידת המודל, כך שנוכל לבחון קומבינציות שונות ולקבל את הפרמטרים הרלוונטיים ביותר לנו:
* תוצאות הסיווג (הפלט של אימון המסווג), אוחסנו בתוך רשימה בשם results:



* בבואנו לבדוק את טיב המודל, החלטנו לבדוק גם את דיוק המודל וגם את מדד ה- AUC area) under the curve):



* התוצאות שהתקבלו:

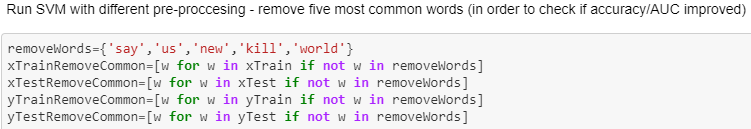
דיוק המודל

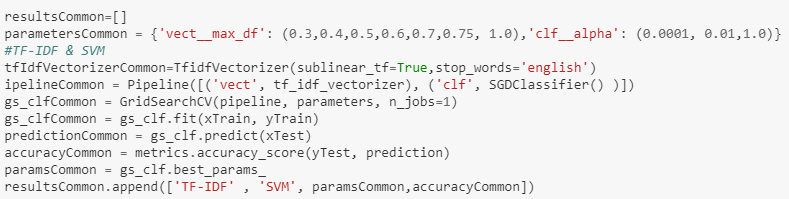
הפרמטרים האופטימליים



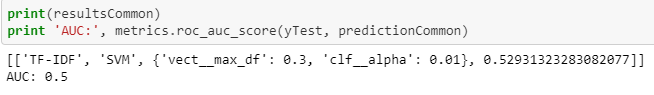
מדד ה-AUC

גם מדד הדיוק וגם מדד ה-AUC הינם ממוצעים, אך לא נמוכים.

* לאחר מכן, החלטנו לבצע tuning למודל. כלומר – לבצע שינוי בתהליך ה – Pre processing, בכדי שנוכל לראות האם נוכל להביא לידי שיפור תוצאות המודל. השינוי שביצענו הינו הורדת חמשת המילים הנפוצות ביותר (בשתי המחלקות) מקבצי האימון והמבחן. הסיבה להורדת מילים אלה היא על-מנת למנוע "השתלטות" של מילים נפוצות המשותפות לשתי המחלקות, וע"י כך להראות בצורה טובה יותר את ייחודיות כל מחלקה בפני עצמה.
* אימון, מבחן המודל, לאחר שעבר שינוי ב – pre processing :



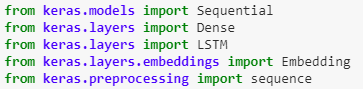
* התוצאות שהתקבלו:



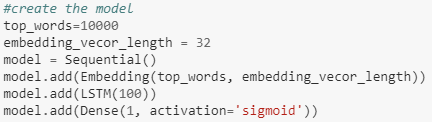
כפי שניתן לראות, הסרת 5 המילים הנפוצות ביותר לשתי המחלקות, לא הביאה לשינוי בפלט המסווג.

המודל השני שבנינו הוא מסוג Keras:

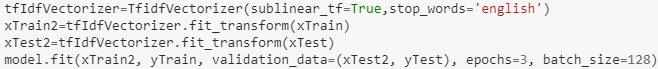
* השתמשנו במודל מסוג LSTM-RNN (רשת נוירונים(.
* קריאה לספריות הרלוונטיות:

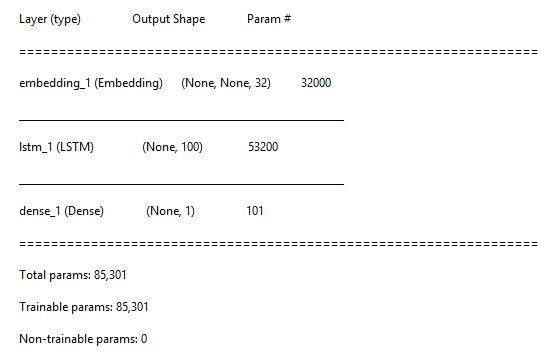


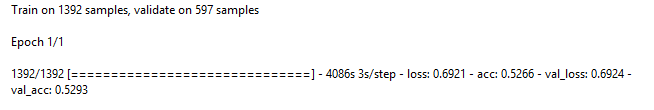
* כפי שנלמד בכיתה, השכבה הראשונה ברשת הנוירונים היא שכבת Embedded:
* מספר המילים הראשונות מתוך המילון שנוצר אשר בחרנו (top words) הוא 1,000.
* וקטור ה-embedded הינו בגודל 32 נוירונים.
* השכבה השנייה ברשת הנוירונים היא שכבת Hidden:
* מספר נוירוני ה- output הינו 1.
* בעלת פונקציית אקטיבציה מסוג sigmoid.



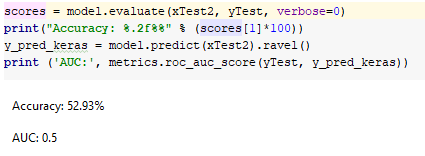
* אימון ומבחן המודל:
* Epochs בעל ערך 1.
* Batch בגודל 128 רשומות.



* הערכת המודל:
* הפלט שהניב המודל:



* דיוק המודל:



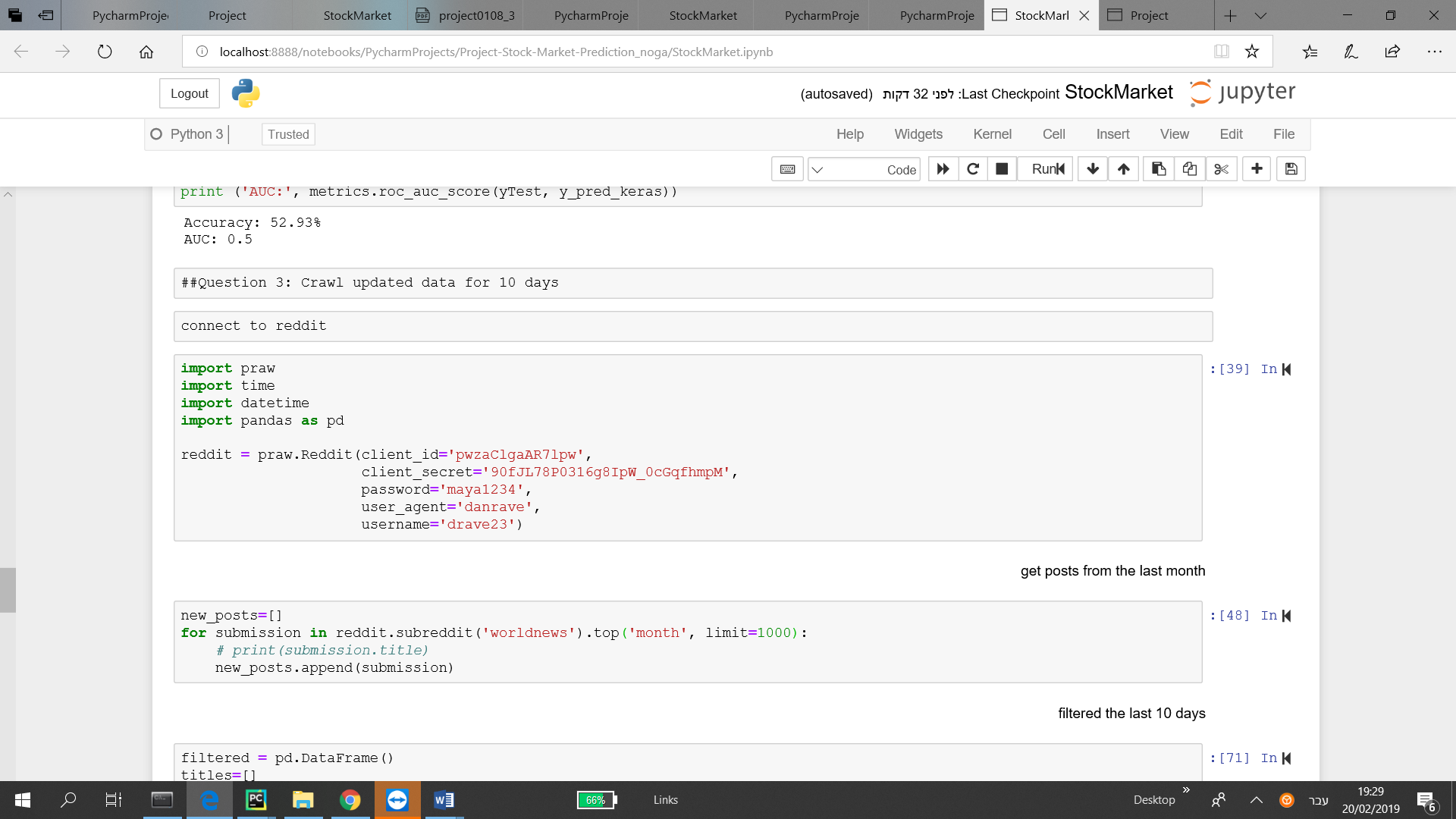
כפי שניתן לראות, הדיוק ומדד ה – AUC הינם בעלי אותם הערכים שהניב מודל ה -SVM , אותו אימנו קודם לכן.

**שאלה 3 – predict the DJIA direction per day**

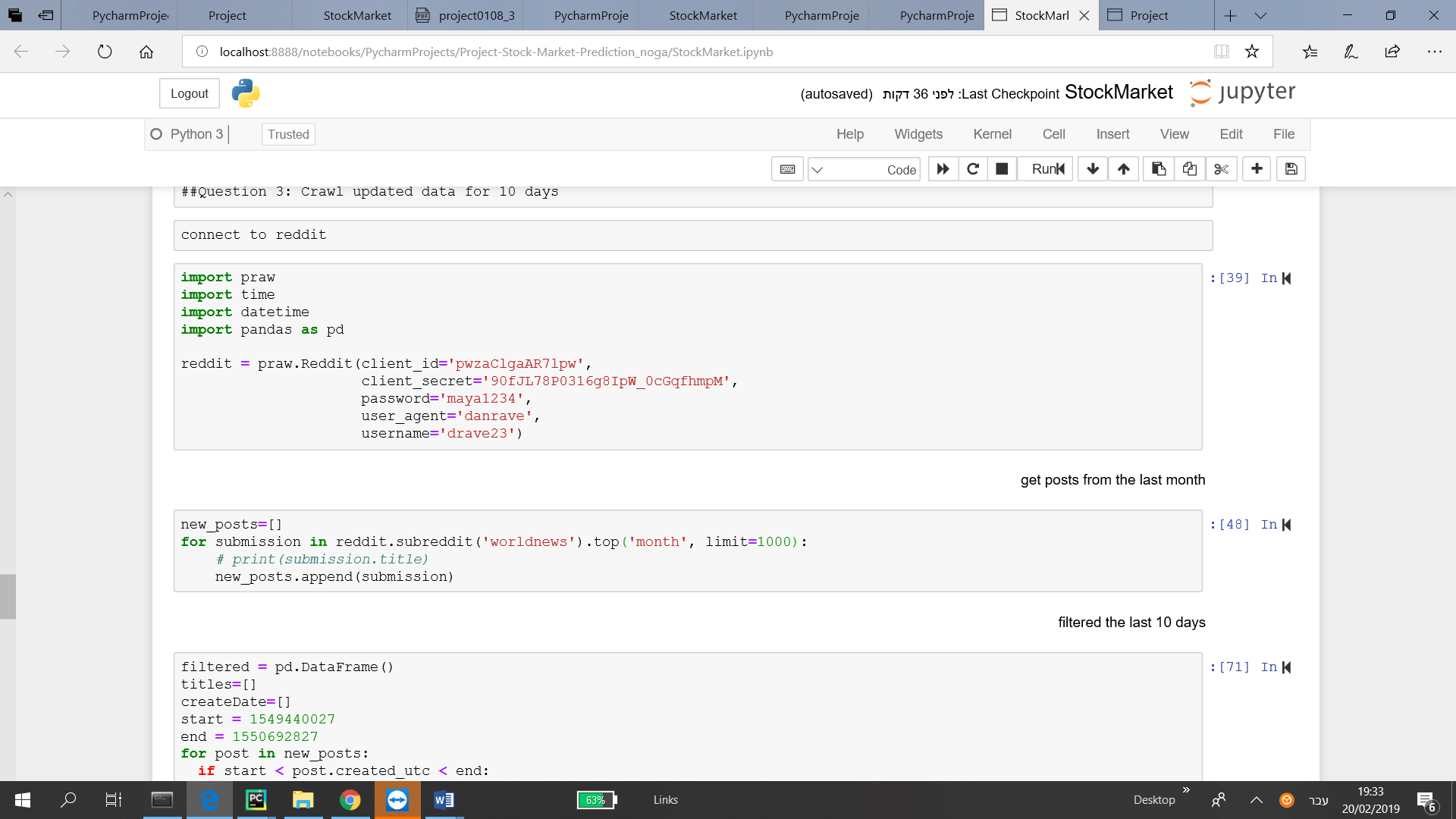
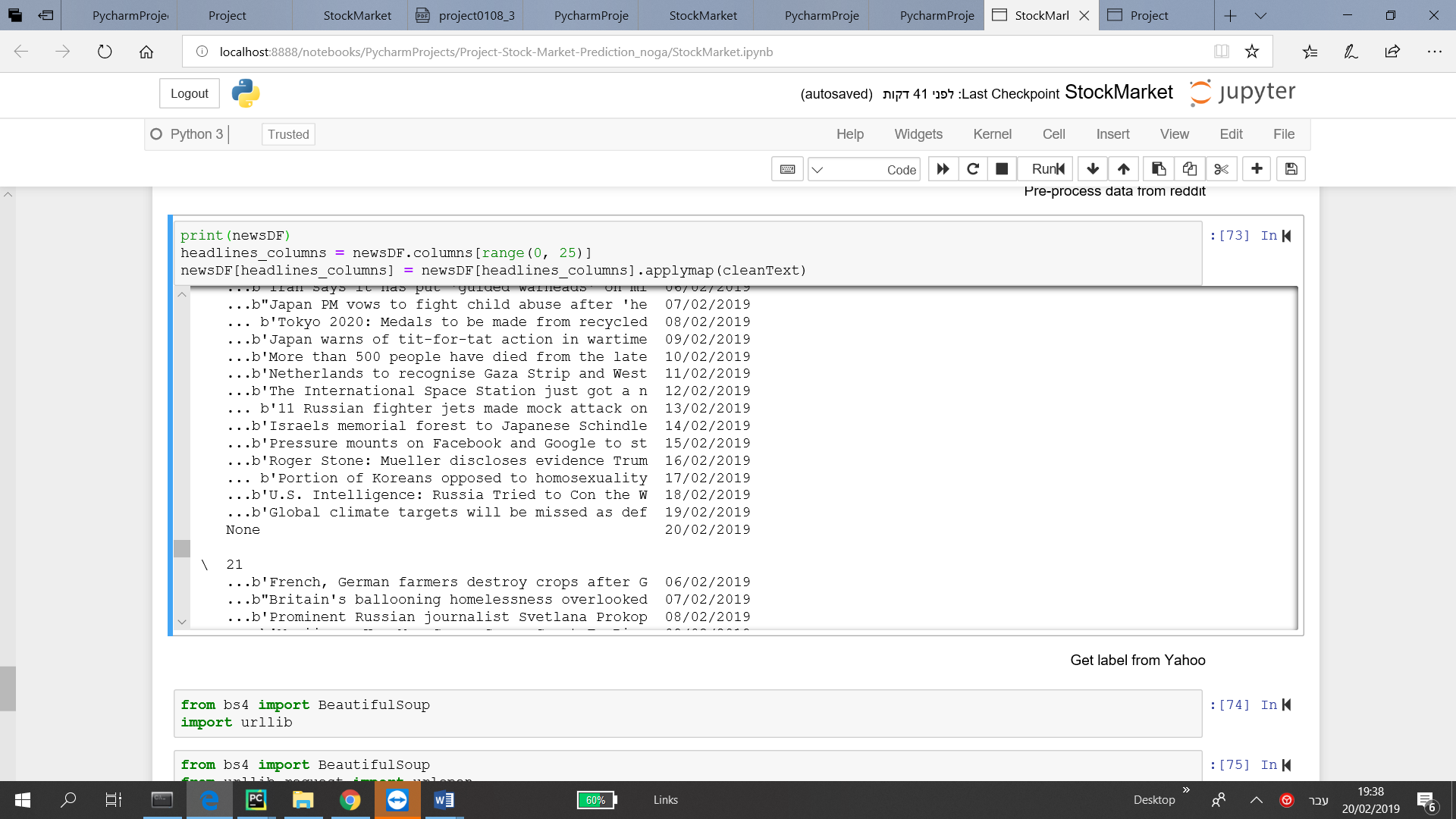
לאחר שאימנו את המודל, נחלץ מהאתר של reddit 1000 ידיעות מה-14 ימים האחרונים כך שעבור 10 ימי מסחר האחרונים בדאוג ונס, ננסה לחזות האם המדד עולה או יורד.

בכדי לחלץ את המידע הרלוונטי מהאתר של reddit השתמשנו בספריית -praw (התחברות באמצעות OAuth)

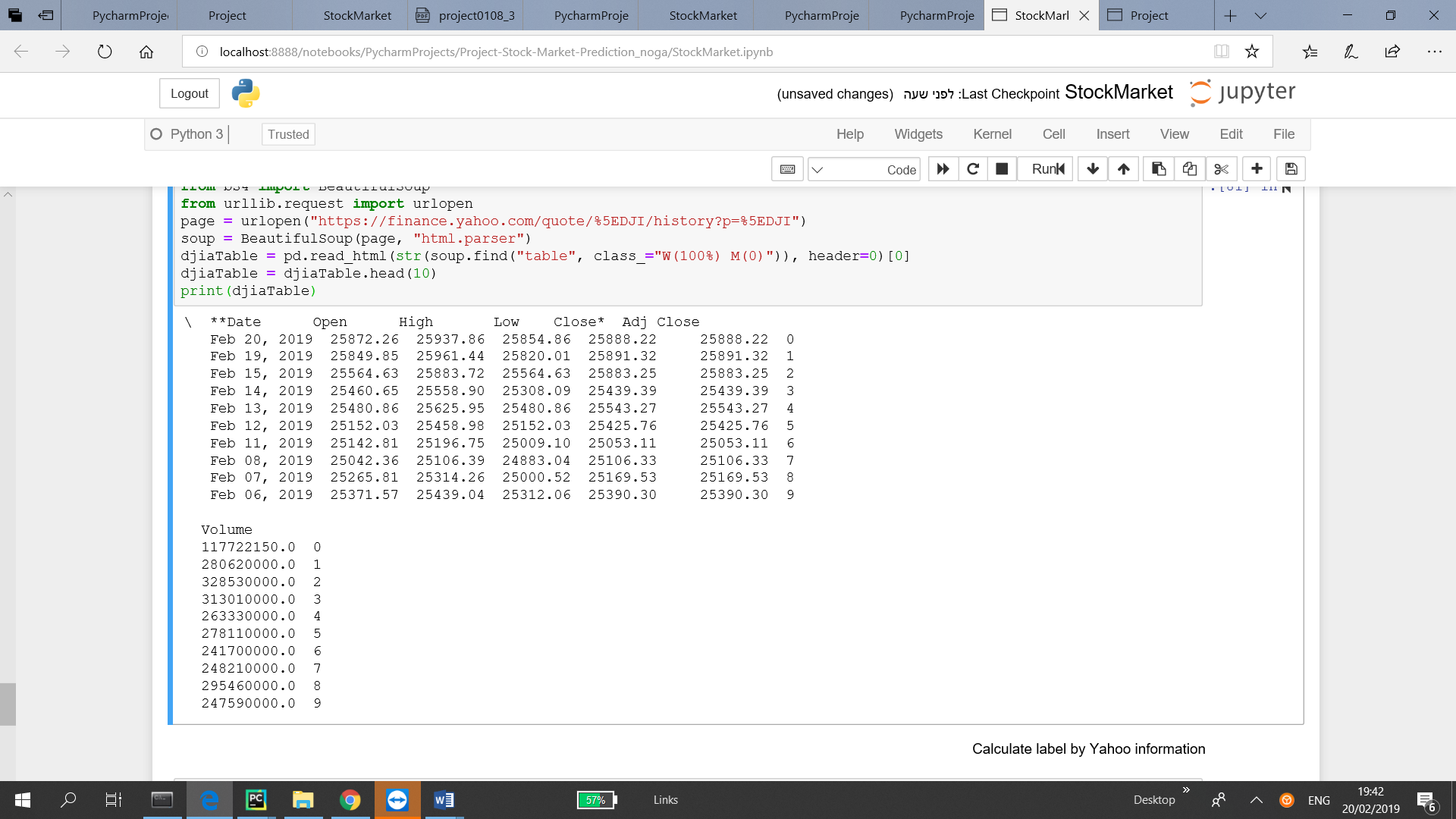
פרטי ההתחברות:



בכל בקשה ניתן לחלץ עד כ- 1000 ידיעות לכל היותר. לאחר החילוץ, סיננו את הידיעות לתאריכים המתאימים (6-20.2.19) מיינו את הידיעות לפי תאריך וביצעו ניקיון לטקסט..



בנוסף, חילצנו מהאתר של Yahoo נתונים עבור מדד הדאו ג'ונס של ה-10 ימי מסחר האחרונים כאשר סיווגנו את הרשומה כ-1 אם המנייה עלתה או לא השתנתה ו-0 אם המנייה ירדה.

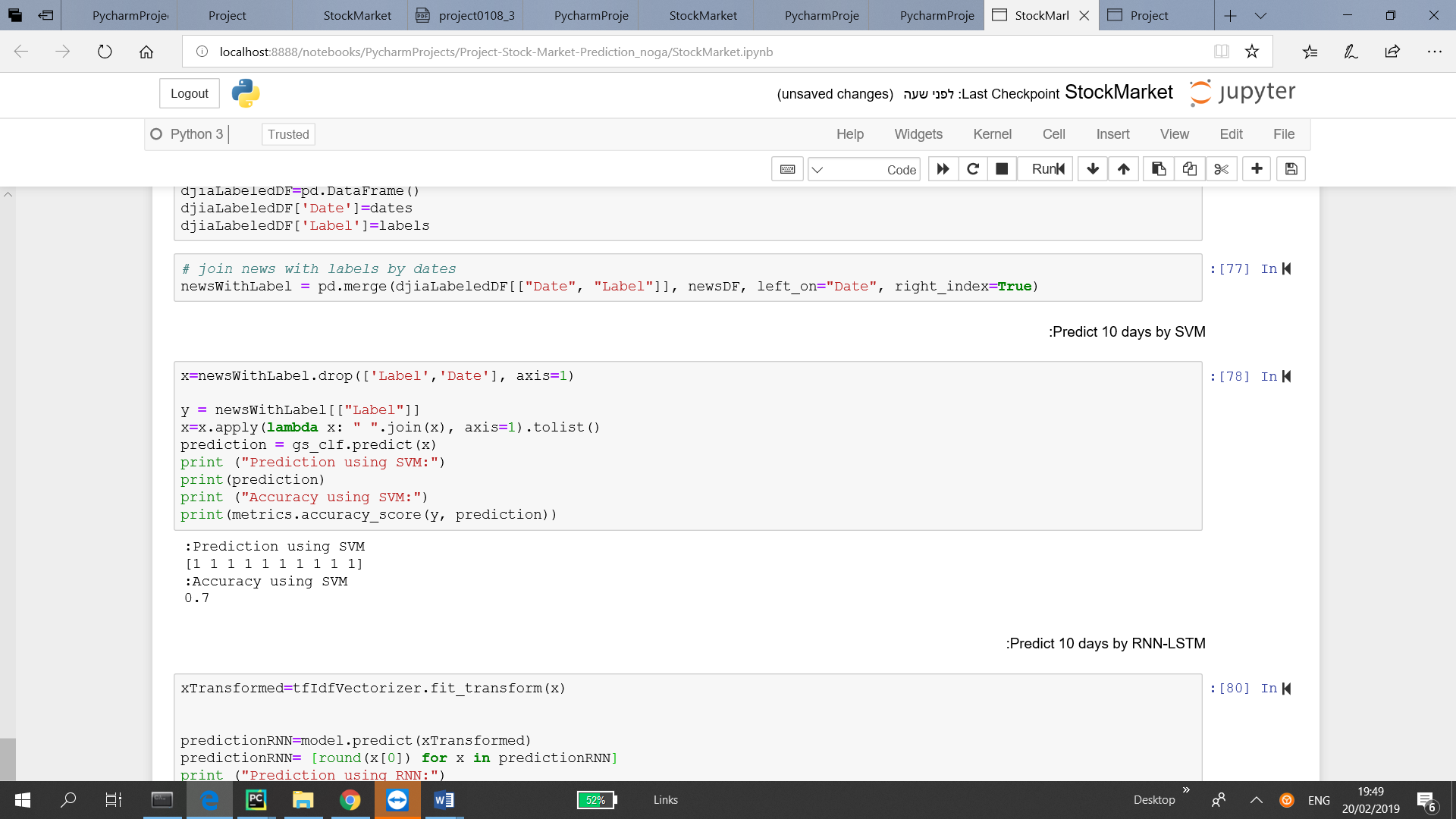


לאחר שיש בידינו את הסיווג ואת הידיעות לכל יום נתון, חיברנו את הנתונים לטבלה בכדי שנוכל להשוות את חיזוי המודלים למה שקרה בפועל.

הרצנו שני מודלים:

1. מודל svm :

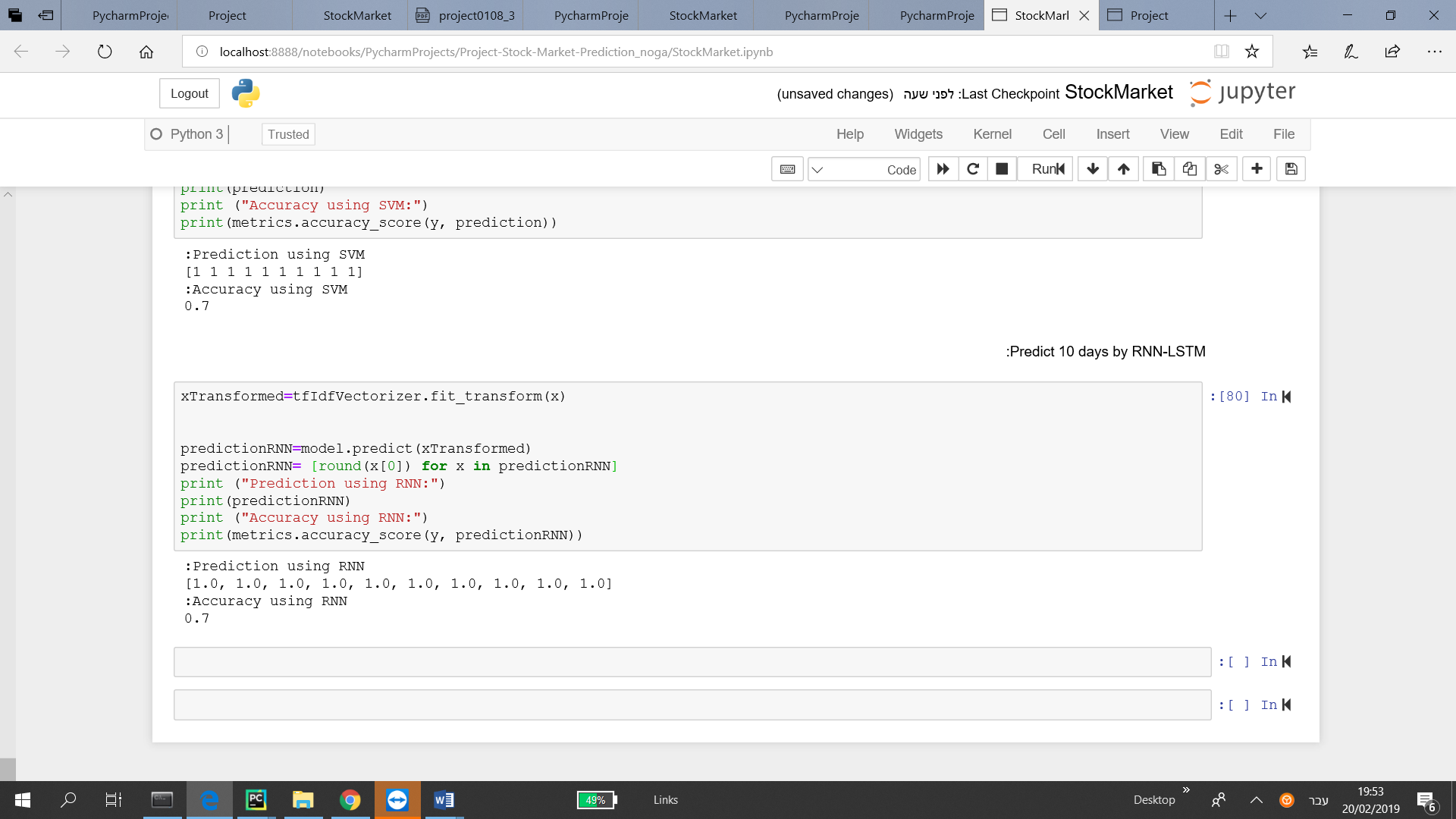
חזינו כי בכל יום המדד יעלה וקיבלנו דיוק של 0.7.



1. מודל keras באמצעות LSTM-RNN:

גם כן חזינו שבכל יום המדד יעלה וקיבלנו דיוק של 0.7.

עבור כל יום קיבלנו חיזוי בין 0 ל-1,כך שעבור חיזוי גבוה מ- 0.5 עיגלנו ל-1.



אתגרים וקשיים:

* במהלך העבודה השתמשו בכלים רבים אשר למדנו בקורס וחיבורם לכדי עבודה אחת הייתה לא פשוטה. חילוץ הנתונים ועבודה עם dataset שונים כך שנוכל לנקות טקסט לא רלוונטי על מנת לקבל דיוק גבוה יותר
* הרצת והתקנת keras דרשו מאיתנו זמן רב. הרצת המודל עם epoch = 3 ו-3000 top words נערכה בערך כ-8 שעות. ולכן נאלצנו להוריד את ה- epoch בכדי לקבל תוצאות בזמן סביר.
* חילוץ ידיעות מהאתר מהאתר של reddit דרשה מאיתנו יצירתיות שכן לא מצאו api שמחלץ ידיעות לפי תאריכים, אלא רק ידיעות יומיות, שבועיות או חודשיות..

מסקנות: