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

**נושא הפרויקט:** Stock Market Prediction

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

**שאלה 1 – Data Exploration**

* השלב הראשון שהיינו צריכים לבצע על-מנת data exploration הוא לנקות את ה-data, בכדי שיהיה מנורמל:

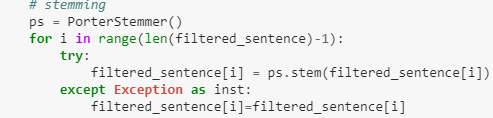
1. ביצוע Lowercase עבור כל מילה בטקסט:



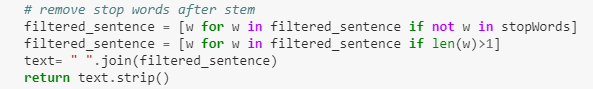
1. ביצוע Tokenizing עבור הטקסט (הפרדה למילים בצורת טוקן):



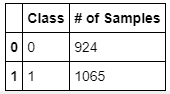
1. ביצוע Stemming עבור כל טוקן, תוך שימוש באלגוריתם porter (לא השתמשנו באלגוריתם Lemmatization מכיוון שלא רצינו להביא כמה מילים עם אותה משמעות, לאותה צורה. כלומר, רצינו לשמר את המילה המקורית שנכתבה בטקסט ורק לנרמל אותה לצורה הבסיסית שלה, ע"י "חיתוך" הסוף שלה).



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).



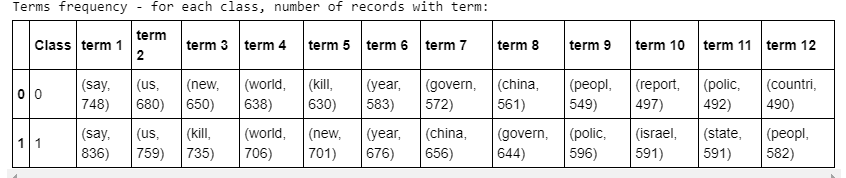
* לאחר ניקוי הנתונים, בדקנו אם הסיווג הבינארי הנתון הינו מאוזן/לא מאוזן (כלומר, אם רוב מוחץ של הרשומות שייכות לסיווג מסוים – 0 או 1). לכן, בדקנו כמה רשומות יש עבור כל סיווג:



ניתן לראות כי התפלגות הרשומות עבור כל Class הינה תקינה ומאוזנת (כלומר, אין מספר רב מידי של דגימות שמדד DJIA Adj עלה/ירד עבורן).

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

1. עבור כל class, בדקנו עבור כל term **בכמה רשומות** הוא מופיע (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).

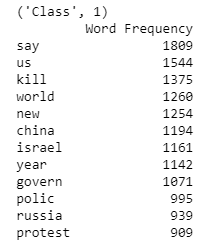
יתרה מכך, מספר המופעים (התפלגויות) של כל אחת ממילים משותפות אלה דומה. לדוגמה:

* 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.

כמו כן, מחלקה אפס לא מכילה מילים נפוצות בעלות אופי שלילי (כדוגמת מילות שלילה), אשר היו יכולות להצביע על תהליך ירידת מניות השערים - דבר אשר כן ציפינו שיקרה בתהליך חקר הנתונים.

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



כאמור, **גם** בבדיקת התפלגות המילים **באופן 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.

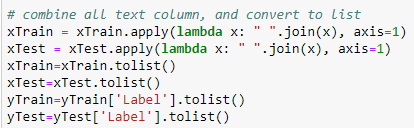
**שאלה 2 – Building Machine Learning classifiers**

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.

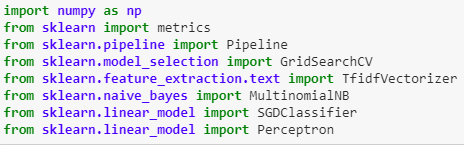


(xTrain, xTest עבור קבוצת האימון ו- yTrain, yTest עבור קבוצת המבחן)

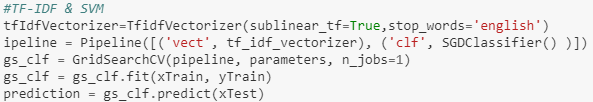
לאחר מכן, הימרנו את הקבוצות הנ"ל לצורת רשימה, בכדי שנוכל לאמן ולבחון את המודל:

המודל הראשון שבנינו הוא מסוג Non-Keras:

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



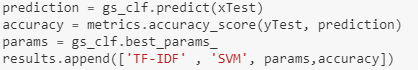
* הסיווג נעשה באמצעות שלושה פרמטרים:
* Feature extraction של הנתונים.
* שיטת Machine Learning באמצעותה יתבצע האימון והסיווג.
* פרמטרים של למידת המכונה.



* 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:



* התוצאות של 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):



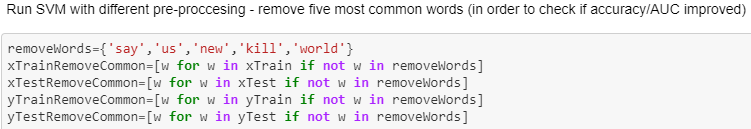
* התוצאות שהתקבלו:

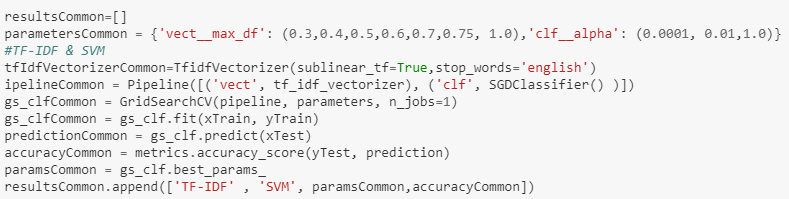
הפרמטרים האופטימליים

מדד ה-AUC

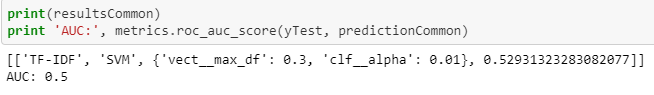
דיוק המודל

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 5 המילים הכי נפוצות (בשתי המחלקות) מקבצי האימון והמבחן:
* אימון, מבחן המודל, לאחר שעבר שינוי ב – pre processing :



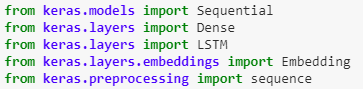
* התוצאות אשר התקבלו:



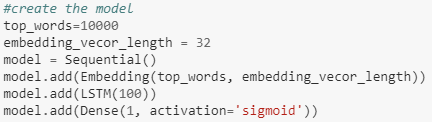
כפי שניתן לראות, הסרת 5 המילים הנפוצות ביותר לשתי המחלקות, לא הביאה לשינוי בפלט המסווג.

המודל השני שבנינו הוא מסוג Keras:

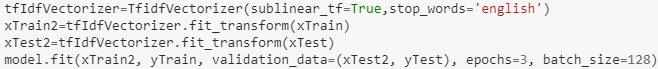
* השתמשנו במודל מסוג LSTM-RNN (רשת נוירונים(.
* קריאה לספריות הרלוונטיות:



* כפי שנלמד בכיתה, השכבה הראשונה ברשת הנוירונים היא שכבת Embedded:
* מספר המילים הראשונות מתוך המילון שנוצר אשר בחרנו (top words) הוא 2,000.
* 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 3.
* גודל batchבגודל 128 רשומות.



* הערכת המודל:
* הפלט שהניב המודל:
* דיוק המודל: