פרויקט בניתוח טקסט פיננסי – קורס במבוא ללמידה עמוקה

רקע לפרויקט

בפרויקט נלקחו החברות אשר נמצאות במדד ה S&P 500- מדד אשר מהווה ברומטר לשוק האמריקאי, כמעט כל החברות הגדולות והחשובות נמצאות שם . הדאטה שאספנו על החברות מורכב מהטקסט המתומלל של Earnings Calls שמקיימות החברות בכל סוף רבעון.

Earning calls הוא מפגש שמנהלים השדרה הניהולית של החברה עם המשקיעים שלה ואנליסטים ומטרת הפגישה היא תחילה לספר על הרבעון שעבר על החברה, מה התוצאות הפיננסיות שלה, מהם האתגרים שעומדים בפניה ובהמשך לתת למשקיעים ולאנליסטים לשאול את המנהלים שאלות על החברה.

בחלק הראשון של כל Earning Call, מנהלי החברה מציגים את מה שנעשה בחברה ובחלקו השני נפתח דיון לשאלות תשובות של האנליסטים והמשקיעים אל מנהלי החברה.

מטרת הפרויקט הינה למצוא האם קיימת קורלציה בין הנאמר ב Earning Call לבין תשואת המניה של החברה באותו הרבעון עד ליום לפני פרסום הדוח, עבור הרבעון הרביעי (Q4) נסתכל על התשואה השנתית מכיוון שזהו הדוח השנתי של החברה ובו החברה מציגה נתונים שנתיים.

תיאור הפרויקט

שלב 1 - איסוף נתונים:

הדאטה מורכב מנתונים שנאספו בין השנים 2010 ועד 2022.

תחילה חילצנו את החברות הנמצאות בקבוצת S&P 500 ולאחר מכן את תוכן הEarning Calls של כל חברה לכל רבעון, זאת באמצעות ספריית request שדרכה ניגשנו לAPI שממנו שלפנו את הנתונים. את מחירי המניות ומדד ה S&P 500 חילצנו ברמה יומית לכל חברה באמצעות ספריית yfinance .

שלב 2 - Pre Processing:

יצירת הלייבלים

מכיוון שמדובר בדאטה שאיננו מתויג החלטנו ליצור לייבלים בעצמנו ולראות האם יש קורלציה בין הטקסט לבין הלייבלים שיצרנו, את הלייבלים יצרנו בצורה הבאה:

תחילה חישבנו לכל חברה תשואה רבעונית עבור רבעונים 1,2,3 ותשואה שנתית עבור רבעון 4.

התשואה חושבה כך שלכל חברה לקחנו את מחיר המניה יום לפני פרסום הדוח ומחיר המניה לפני פרסום הדוח שקדם – עבור דוחות רבעוניים ועבור הדוח השנתי אותו דבר רק עבור תקופה של שנה.

בחישוב התשואה לא לקחנו את המחיר שביום הפרסום מכיוון שביום זה יש תזוזה חזקה במחיר המניה שנובע עקב נפח מסחר גבוה ומניפולציות של משקיעים, לכן הנחנו שזה יפגע בקורלציה מכיוון שאנחנו מנתחים טקסט שרוב המידע בו כבר ידוע למשקיעים, למשל אם חברה מסוימת נכנסת לעסקה גדולה זה מידע שיתפרסם בפומבי עוד לפני פרסום הדוח.
בנוסף רצינו להתחשב בהשפעת השוק לכן לכל תשואה שחישבנו לתקופה השוונו לתשואה שעשה מדד ה S&P 500 באותה תקופה. בחרנו בטווח ביטחון ש 3%-+ וייצרנו את הלייבלים בצורה הבאה –

$$Label = \begin{cases} 1, & Company\ yield - sp500\ yield > 3\% \\ 2, & Company\ yield - sp500\ yield < 3\% \\ 0, & otherwise \end{cases}$$

כלומר עבור תשואות שגבוהות ב3% מתשואת השוק נתייחס בכך שלחברה היו ביצועי יתר, 3% מתחת ביצועי חסר ואחרת ביצועים נטרלים.

נציין כי בימים בהם לא מתנהל מסחר לא קיים דאטה על מחיר המניה בשוק, כמו למשל ימי חג או שבת. על מנת להתמודד עם בעיה זו חישבנו ימים לאחור עד שהגענו ליום עם מסחר.

עיבוד הטקסט

תמלולי הEarning Calls מכילים בין 10 ל20 עמודים של מלל. כמות מאוד גדולה של מלל שקשה למודל להתמודד איתו עם הכוח חישוב שזמין לנו, במיוחד אם מדובר במודל מורכב. תחילה הורדנו

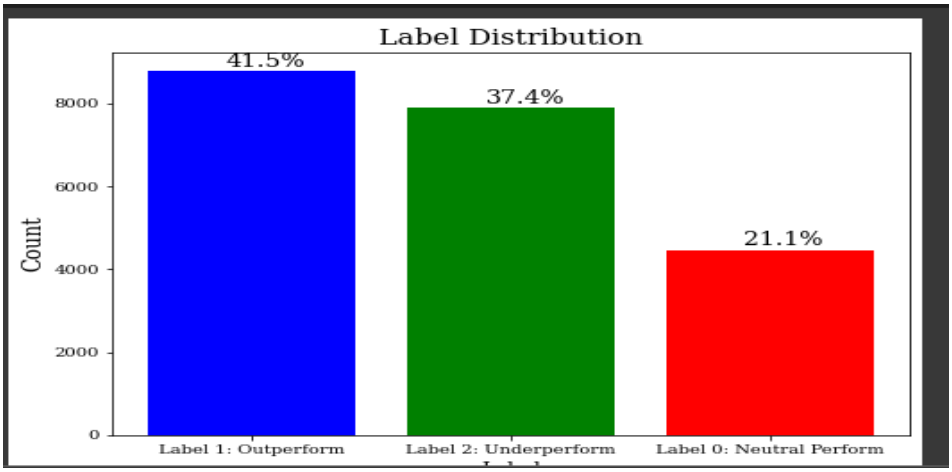
לכל sample את 2 הפסקאות הראשונות. פסקאות אלה הן פסקאות פתיחה האומרות שלום לכולם ומציגות את הנוכחים בחדר. הן פחות רלוונטיות לפעילות החברה והגורמים המשפיעים על תשואת המניה. הורדנו את ה stop words בדומה לטכניקה שראינו בתרגול כמו כן הורדנו סימנים מיוחדים למעט סימני ('.','\n'). בנוסף, על מנת להפוך את הדאטה ליעיל יותר השתמשנו בקורפוס בשם **Loughran-Macdonald** קורפוס זה מכיל את כל המילים שהופיעו בדוחות במסמכים פיננסיים כמו למשל דוחות כספיים, כתבות כלכליות מאמרים ועוד.. כמו כן בקורפוס יש מידע על כמה פעמים המילים הופיעו בסך הכל בכל המסמכים. ראינו ש50% מהמילים מופיעות עד 450 פעם וכאשר מספר הממוצע של המילים הוא 350 אלף מילה. בעבור כל משפט בטקסט בדקנו שהוא מכיל מילים שנמצאות במאגר זה וגם האם קיימת בו מילה שנמצאת ב50% העליונים של תדירות המילים. במידה ולא התקיימו תנאים אלה הסרנו את המשפט.

בנוסף, השתמשנו בטכניקות של איחוד פסקאות מתחת לאורך מסויים ופיצול פסקאות מעל אורך מסויים, כך שנוכל להכניס למודל דטא מאוזן.

<u>שלב 3 Models</u>

בסופו של דבר ייתכן ולא קיימת קורלציה בין תשואת המניה לבין הEarning Calls . מחירי המניות מושפעים ממספר מאוד גדול של גורמים מלבד פעולות החברה ולמרות הניסיון שלנו לסווג את הלייבלים באופן מנטרל את השפעות השוק, לא ניתן לנטרל דברים אחרים כמו מאפייניים מיוחדים של כל חברה שנלקחים בחשבון רק בצורה ספציפית ואירועים גאו פולייטים שכולים להשפיע על חברה ספציפית יותר מאשר חברה אחרת, כמו כן גם שווי השוק של החברה והמגמה שבה היא נמצאת לא נלקחו בחשבון במודל. אנו רצינו נטו לבדוק האם ניתן למצוא בעזרת שיטות מתקדמות לניתוח טסט קורלציה שחשבנו שהיא קיימת. הרצנו מספר מודלים שונים בדרכים שונות אך כולם קיבלנו accuracy סביב ה41-44 אחוזים.

הלייבלים מתפלגים בצורה הבאה:



לכן המודל הנאיבי שיחזה הכל "1" יספק ביצועים של 41.5%.

המודלים:

א. מודלים פשוטים של רשתות נוירונים - את המודלים הללו יצרנו שיהיה לנו מאין בסיס שאותו נרצה לעבור ולשפר.

1. מודל ראשון, מודל עם שכבות Fully connected, למודל זה יצא דיוק של 41.3% דומה לדיוק הנאיבי.
2. מודל שני, מודל נוסף של שכבות Fully connected, פה יצא דיוק של 40%
3. מודל שלישי, מודל LSTM עם שכבות Fully connected, יצא דיוק של 37%
4. מודל רביעי, מודל CNN עם שכבות נוספות של Fully connected , יצא דיוק של 42.5% אחוז מעל המודל הנאיבי.
5. מודל חמישי, שילוב של RNN ו CNN, יצא דיוק של 36.7%.

לסיכום, לא הצלחנו להשיג תוצאות משמעותיות באף אחד מהמודלים.

ב. מודל - FinBert

מודל זה הוא גרסא של מודל BERT המפורסם של גוגל אך שאומן על דטא פיננסי ביניהם דטא של דוחות כספיים.
לקחנו את מודל זה ועשינו עליו Fine Tuning , כלומר אימנו את המודל הזה על הדטא שלנו והוספנו שכבות קונבולוציה וFully Conected. מכיוון שמודל זה יודע להתמודד רק עם משפטים באורך לכל היותר של 510 תווים (התו הראשון והתו האחרון שומרים לסימון של תחילת המשפט וסופו) היה לנו קשיים רבים איך נוכל לאמן את המודל בצורה נכונה מכיוון שכל טקסט שלנו מורכב מהרבה יותר מ510 תווים.
לכן לכל סאמפל חילקנו את הטקסט לחלקים באורך של 450 תווים (לגדלים גדולים יותר המודל קרס) תוך שמירה על מאפייני הטקסט כמו פסקה ומשפטים, מכיוון שיש לנו המון דטא לאמן לא לקחנו עבור כל חברה את כל הsub texts שלה מפאת מגבלות כוח חישוב, לכן עבור כל חברה בחרנו בצורה רנדומלית 64 קטעים באורך של 450 תווים. כך שעבור כל transcript הצלחנו לכסות בערך 70% ממנו, הtranscript הממוצע מורכב מ92 חלקים באורך 450.

במודל עצמו הגדרנו כך שכל batch יהיה בגודל של 64 דוגמאות כך שכל batch בעצם ייצג sample.
הגדרנו את פונקציית ההפסד שלנו בכך שעבור כל batch היא תחשב את ההפסד וכך המשקולות יתעדכנו בהתאם. כלומר מכיוון שכל batch מייצג 64 קטעים מסאמפל אחד נוכל לתת להם את הלייבל של הטקסט בכללותו וכך לחשב את הפונקציית הפסד וכך נשמור על קהורנטיות המודל.
אנו נתנו לייבלים לטקסט בכללותו ולא לקטעים ממנו וגישה זו פותרת את הבעיה.

במודל זה עשינו מספר רב של נסיונות עם שכבות שונות והגענו לדיוק גם פה של **42.5%.**
זהו דיוק שאינו מספק ולא שווה את הזמן ריצה הרבה שהמודל רץ בו מעל ל-10 שעות וכוח החישוב הרב שהוא דורש, 38 GPU RAM.

למרות שהדיוק לא גבוה בחלק זה למדנו הכי הרבה, למדנו על מודל BERT וסוגיו, איך הוא פועל, איך השכבות שלו פועלות, איך פועל הEmbadding שלו וכמובן החלק הכי מסובך איך לעשות לו Fine Tunening. כמו כן למדנו איך לעבוד על GPU ולמדנו על שיקולי זיכרון ואיך ניתן לצמצם זיכון ולמנוע memory leak, למדנו איך להעביר מידע וחישובים בין CPU ל GPUוגם איך לצמצם שימוש בזיכרון בשיטות כמו half precision. בנוסף, הצלחנו ליישם על המודל טכניקה שבה נוכל לעבד טקסט גדול בכללותו יותר ממה שהמודל יכול לעבד.

ג. מודל XGBoost

במודל זה רצינו לראות האם אינדיקטורים נוספים מלבד הטקסט יוכלו להגדיל את הסבר הקורלציה ולכן בחרנו להשתמש במודל שעובד על דאטה טבלאי, לכל חברה יש מספר פיצ'רים נוספים שהחלטנו ליצור.

- עמודות dummies - יצרנו תחילה עמודה לכל רבעון, ז"א הוספנו 4 עמודות המייצגות רבעון ומציינות באיזה רבעון הייתה הדגימה הנוכחית שלנו. בנוסף, יצרנו dummies למגזרי פעילות. קיימים 11 מגזרי פעילות שונים במדד הS&P 500 . מידע זה גם חולץ באמצעות ספריית yfinance .
- הוספנו את אחוז המשקל של כל חברה מהמדד, גם כאן באמצעות ספרית yfinance לכל שנה. נזכיר כי S&P 500 הינו מדד משוקלל של 500 חברות, בכל שנה ביצענו חישוב למשקל כל חברה מכלל המדד.
- השתמשנו בספרייה בשם "textblob" ובה קיימות פונקציות אשר מחלצות את ההקשר של המשפט ונותנות ציון אם בין מינוס אחד לאחד. כאשר אחד זה חיובי ומינוס אחד זה שלילי. עבור הטקסט יצרנו שתי מטריצות שונות שבאמצעותן נבדוק את המודל, אחד בשיטת count vectorizer (BOW)ופעם שניה בשיטת TF-IDF.
- הוספנו שתי עמודות נוספות שהן אחוז המילים החיוביות בתוך הטקסט ואחוז המילים עם ההקשר השלילי במשפט. את המילים הפכנו לטקונים. תחילה, נבדוק את מודל ה- xgboost על הדאטה של הטקס בלבד, ולאחר מכן נוסיף בכל פעם עמודות שונות. באחד המודלים הוספנו את הסקטורים של החברות, ובמודל אחר רק את העמודות של ההקשר של המילים והמשקלים של החברות.

במהלך העבודה, ניסינו ליישם את שיטת ה GridSearch. לצערנו זה לקח המון שעות והמודל קרס בשל תעדוף משימות ולוח זמנים צפוף החלטנו שלא להריץ אותו.

הרצנו מספר פעמים את המודל כאשר הדאטה הכיל בכל פעם פיצ'רים שונים בניסיון לראות האם נצליח לתפוס דברים נוסיף שישפיעו על הקורלציה מלבד תמלול ב Earning Call. בין הפיצ'רים שהוספנו הין, משקנה דאמי של איזה רבעון, מה הסנטימנט של הטקסט, ומה המשקל של החברה במדד.

כלל התוצאות היו בטווח של 42-39 בדומה לכלל המודלים שהרצנו ללא שינוי משמעותי.

אנחנו מניחים שכנראה חלק מעמודות הסקטור לא תורמות ללמידה של המודל. באמידת המודל עם העמודות של אחוז המילים החיוביות והשליליות, הרבעונים, המשקל וסמטימנט המילה, קיבלנו accuracy של 42.5% גם.

הקשיים שעמדו בפנינו באמידת המודל היו הוספת פיצ'רים רלוונטים לטקסט שאינם כספיים, וכך גם להבין אילו עמודות מכילות משתנים רלוונטיים. בנוסף, זמני הריצה של חלק מהפונקציות כמו stemmer והמודלים הגבילו את כמות ההרצות שיכולנו לבצע, והכמות המוגבלת של computational units.

<u>תוצאות</u>

לסיכום בכל המודלים קיבלנו תוצאות דומות ללא קשר לרמת סיבוך המודל וזה סביב ה42.5% מאוד דומה למודל נאיבי, לכן ניתן להגיד שקשה למצוא קורלציה בין הטקסט לתשואתה של המניה.

<u>מסקנות</u>

העולם הפיננסי הוא עולם המורכב ומושפע מנושאים שונים ומגוונים. תשואת מניה יכולה להשתנות בכל רגע ובשל שלל רחב של סיבות. ייתכן וזו אחת הסיבות לקורלציה הנמוכה.

במודל XGBOOST נלקחו פרמטרים נוספים מעבר לתמלול הדוחות, למרות זאת לא עלה הaccuracy. העובדה שלא לקחנו פיצ'רים טמפוררים אולי גרמו לכך שלא היה שיפור, אך רצינו לשאר נאמנים למבנה העבודה שקשור לטקסט גרידא ולא לדברים שקשורים למחירי המניות.

עבודה עם טקסט- במהלך תהליך העבודה למדנו כי למודלים קשה להתמודד עם טקסט בעל מסה גדולה. הלמידה של המודלים, ממגוון סוגים ולאו דווקא מודל ספציפי, יעילה יותר ומדויקת יותר בעבור טקסטים קצרים ופשוטים.

רעיונות להמשך

עלו בפנינו מספר רעיונות שרצינו לבחון בעקבות מסקנות שונות שהגענו אליהן לקראת סיום תהליך העבודה.

רעיון 1 – להשתמש בembedding של מודל ה Bert. שכבת embedding זו היא ממימד 3 , כאשר כל וקטור מיוצג בצורה הבאה (1,x,768) כאשר x זה מספר המילים בקטע ו768 זה הייצוג שלהם בוקטור embedding. רצינו לכל פסקה בטקסט לקחת את חלק הembedding שלה וכך למשל עבור טקסט המורכב מ50 פסקאות נקבל 50 וקטורים שכל וקטור הוא הembedding של הפסקה במודל ה BERT, כמובן רק החלק של embedding בוקטור ושכל פסקה תהיה לכל היותר 510 תווים. לכל דוגמה נרצה להכניס את כל הווקטורים שלה למודלים של רשתות נוירונים כמו למשל RNN עם LSTM ולבדוק את התוצאה מאשר embedding נאיבי כמו שעשינו בפרויקט.

רעיון 2 – הרעיון השני שלנו הוא לחלק את כל הטקסט לפסקאות וכל פסקה להכניס למודל ה Finbert מכיוון שמודל זה הוא מודל SequanceClassfication נוכל לקחת את שכבת הsoftmax האחרונה ובעצם לקבל את 3 ההסתברויות לכל לייבל. מודל זה הוא מודל שמזהה Sentiment כאשר הם מחולקים ל3 סוגים חיובי, שלילי ונטרלי בדומה לייבלים שלנו. נגדיר מספר פסקאות שניקח עבור כל דוגמה, למשל 50 לכן עבורם נוכל לבנות מאין "תמונה" בגודל של 768X50X3 כאשר 768 זה ה embedding מהBERT דוגמאות נקבל מטריצה שדומה בעיקרון לייצוג RGB אנחנו מודעים שזה לא תמונה אך נוכל להשתמש עליה ברשתות קונבולציות הבנויות למודלים של ניתוחי תמונה ואולי נצליח להגיע לתוצאות. במקרה זה המימד יהיה (10000,50,3) כך בעצם אנחנו לוקחים דטא חד מימדי של טקסט ומעבירים אותו למרחב דו מימדי וכאשר כל channel מהווה ייצוג סנטימנט של הפסקה לפי מודל Finbert, channel זה מורכב מהעוצמה של כל סנטימנט כמו בRGB על צבעים.

<u>קישורים:</u>

הסבר על מילון Loughran-Macdonald -

https://sraf.nd.edu/loughranmcdonald-master-dictionary

קישור לגיט על המודל FinBert שהשתמשנו בו –

https://github.com/yya518/FinBERT