הבחנה בין ידיעות כוזבות לידיעות אמינות תוך מציאת הפיצ׳רים המשמעותיים לסיווג

דוח פרויקט בבינה מלאכותית 236502

מגישות:

שני אמסלם 322791724
מיכל עוזרי 325719052

Shaniamsalem@campus.technion.ac.il
michal.ozeri@campus.technion.ac.il

רקע על הפרויקט:

עם התקדמות הטכנולוגיה בשנים האחרונות, ישנו ריבוי במקורות המידע שמהם אנשים שואבים את הידע שלהם בנוגע לאירועים המתרחשים מסביב לעולם. מקורות מידע אלו, שלא כמו בתקופות קדומות יותר, מאפשרים הפצה הרבה יותר נרחבת של מידע וחדשות, בעלות נמוכה על ידי כל אדם במגוון פלטפורמות.
בעידן שבו זמן רב יותר מושקע ברשתות החברתיות, אנשים נוטים יותר לצרוך חדשות דרך פלטפורמות אלו מאשר דרך אמצעי תקשורת מסורתיים, כמו עיתונים או טלוויזיה.
היתרונות המרכזיים של מדיה חברתית הם הנגישות המהירה והעלות הנמוכה, לצד האפשרות לקיים אינטראקציות חברתיות סביב החדשות, כמו שיתוף, תגובות ודיווחים בזמן אמת. [1]
במחקר שנערך בארצות הברית, 62 אחוזים מהמבוגרים דיווחו על קבלת חדשות במדיה החברתית ב-2016, בעוד בשנת 2012 רק 49 אחוזים דיווחו על קבלת חדשות מאתרי חדשות חברתיים. [1]
ניתן לראות את הקפיצה המשמעותית שיש בשימוש ברשתות חברתיות ככלי תקשורת במהלך השנים. עם זאת, יחד עם היתרונות הללו, כל אלו מהוות קרקע פורייה להפצת מידע כוזב ("Fake News") במסווה של ידיעות אמינות שקל מאוד למשתמשים להאמין לו גם ללא אישור של גורם רשמי. [5] למידע כוזב פוטנציאל להשפעות שליליות גלובליות במגוון תחומים כמו פוליטיקה, בריאות, ביטחון, דיפלומטיה ועוד. דוגמה ראשונה היא על קמפיין הבחירות לנשיאות ארה"ב בשנת 2016, אשר הושפע ממידע כוזב. [5]
לאור ההתפתחויות האחרונות במדינה שלנו, אנו עדים להשפעות ההרסניות של המידע הכוזב המופץ ברשתות החברתיות על דעת הקהל העולמית כלפי המלחמה וכלפי מדינת ישראל. נושא זה מאוד רלוונטי ונוגע לנו באופן אישי ולכן החלטנו לחקור כיצד בינה מלאכותית יכולה לבוא לעזרה בזיהוי של מידע כוזב, ולתכנן מודל קלסיפיקציה, שיקבל כקלט ידיעות טקסט ויפלוט האם הן כוזבות או נכונות.

עבודות בנושא:

מצאנו מספר עבודות בנושא ב- GOOGLE SCHOLAR, שמפורטות כמאמרים.
במאמר הראשון בשם *Fake News Detection Using Machine Learning Approaches* [3]
מתואר שימוש בלמידת מכונה על מנת לסווג נכונה בין ידיעות כזב לידיעות אמינות.
במאמר נעשה שימוש בכלים של NLP ו- Scikit-learn, על מנת באמצעותם לחלץ פיצ'רים מתאימים שיעזרו לבניית המודל הסופי ולבעיית הסיווג שלנו. גם בעבודה שלנו המפורטת מטה השתמשנו גם בכלים אלה ובעוד כלים נוספים על מנת להצליח ליצור מודל סופי שיסווג נכונה.
במאמר השני בשם *Fake News Detection on Social* [1] מפורטת שורש הבעיה תוך התקדמות הרשתות החברתיות והמדיה החברתית. מפורטים במאמר האתגרים בזיהוי ידיעות כזב, שכן הן מתפשטות במהירות רבה ולעיתים נראות עבור צופה תמים כאמיתיות. זאת כמובן על מנת להטעות ולגרום לתפיסת מציאות שגויה שקשה לתקנה. מפורט כי על מנת לשפר את היכולת לזהות ידיעות כוזבות יש לשלב בנוסף לניתוח התוכן הכולל כותרת, שפה ותמונות את האינטראקציות סביב הידיעה: תגובות, שיתופים. בדר"כ בידיעות כוזבות נבחין בשימוש בבוטים.
במאמר השלישי בשם *A Survey of Fake News* [2] מתבצעת סקירה על נושא ידיעות הכזב, אשר מפרט על הקושי בזיהוי הידיעות הכוזבות, ההשפעה שלהם על הדעה הציבורית תוך פירוט על ההיבטים הפסיכולוגיים והחברתיים שלהן. המאמר מדגיש את הצורך בשימוש בלמידת מכונה כדי להבדיל בין ידיעות כזב לידיעות אמינות.
תוך סקירת העבודות, המאמרים והפרויקטים הקיימים, החלטנו שנרצה בפרויקט שלנו להתמקד מעבר לבעיית החיזוי בפיצ'רים שיעזרו לנו לסווג נכון. זה מאוד סיקרן אותנו לגלות אילו מאפיינים עוזרים למסווג שלנו לסווג טקסט למידע כוזב ולמידע אמיתי וללמוד מכך. נוכל ללמוד מכך המון ואף לעזור למשתמשים שאין בידיהם כלי של בינה מלאכותית להצליח לפתח חשיבה ביקורתית ולהצליח לזהות בסיכוי גבוה בעצמם חדשות כוזבות.

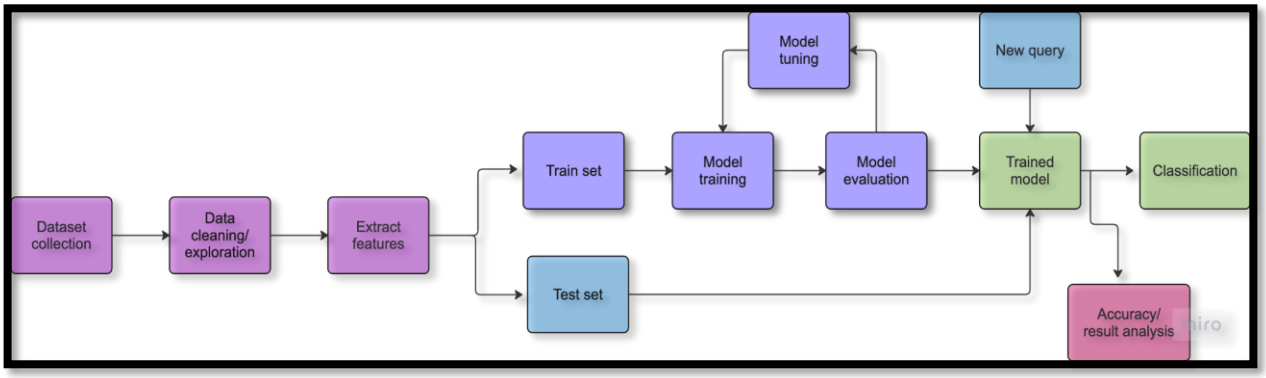
תיאור הפתרון המוצע לבעיה:

מטרת הפרויקט שלנו היא לבנות מודל קלסיפיקציה, שיקבל כקלט ידיעות טקסט ויפלוט האם הן כוזבות או נכונות. בנוסף, תוך בניית העבודה סיקרן אותנו לגלות מה הם הפיצ'רים המשמעותיים שבעזרתם המסווג סיווג, לנסות לראות האם הם יוכלו לבדם לסווג נכונה. על כן, במהלך הפרויקט נציג אותם ואת המסקנות שגילינו.

כדי לבצע זאת, ביצענו מספר שלבים:

1. איסוף נתונים.
2. עיבוד הטקסטים וחילוץ פיצ'רים.
3. ויזואליזציה של ה -data.
4. למידה: אימון המודלים עם ה- train ומציאת אחוזי הצלחה על ה- test.
5. ביצוע feature selection והסקת מסקנות על ובעזרת הפיצ'רים המשמעותיים.
6. אימון מחדש על המודלים הקיימים ועל מודלים חדשים.
7. יצירת מודל סופי / השוואה בין מודלים שונים, הקצאה של פיצ'רים חשובים. טבלה של יתרונות וחסרונות לכל מידול.

תרשים זרימה המתאר את אופן הפעולה שלנו המפורט מעלה:



בעמודים הבאים נפרט על כל השלבים.

שלב 1: איסוף הנתונים

התחלנו את התהליך של הלמידה באמצעות datasets המכילים טקסט וסיווג הטקסט המתאים.
השאיפה שלנו היא לבסוף לבחון את המודל שלנו על תמלילים של דובר צה״ל, אשר הינן ידיעות אמיתיות ונרצה כי המודל הסופי יצליח לסווגן כאמת.
בנוסף, נרצה להשתמש בידיעות שידועות ככוזבות, ולראות האם אכן המודל הסופי שלנו מצליח לסווגן כשקר.
איסוף הנתונים חשוב ביותר שכן עליו יתבסס המודל הסופי שנבנה, ועל כן עליו להיות מגוון, מייצג ומורכב מספיק, כדי שהפתרון שלנו והמודל הסופי יהיה מוצלח.

קישור ל- dataset הבאים בהם השתמשנו בתור התחלה:

https://www.kaggle.com/datasets/saurabhshahane/fake-news-classification

החלטנו להתחיל מהdata מ Kaggle מכיוון שלפי דעתנו הוא נראה אמין (ניתן לראות לפי הציון שנתן Kaggle ולפי דירוג של אנשים מצד ימין למעלה), הוא מכסה 4 datasets שונים על מנת למנוע overfitting, והוא מכיל דגימות רבות כאשר רובן המוחלט אינן דגימות לקויות. לדוגמה, בdataset של הטקסטים הצגנו את כמות הדגימות לאחר הסרת שכפולים ודגימות בהן אין טקסט וראינו כי נותרו 62,719 מתוך 72,134 דגימות.
אספנו עוד datasets אותם נוכל לשלב בהמשך במידת הצורך:

https://github.com/KaiDMML/FakeNewsNet/tree/master/dataset
https://github.com/kapilsinghnegi/Fake-News-Detection/tree/main

כאמור, בנינו ידנית dataset המורכב מידיעות אמת של דובר צה"ל עליו נרצה לבדוק את שיעור ההצלחה והכישלון שלנו על ידיעות אמת.

שלב 2: עיבוד הנתונים וחילוץ הפיצ'רים

בעבודה שלנו השתמשנו ב- dataset המכיל בכל שורה טקסט, כותרת וסיווג מתאים. מכיוון שלדגימות רבות לא הייתה כותרת, החלטנו להפריד את הdata ל-2 datasets: אחד המכיל רק טקסטים, ושני המכיל רק כותרות, מתוך מחשבה שלכותרות וטקסטים מאפיינים שונים בהקשר של המשימה שלנו. התחלנו בעיבוד ולמידת ה- data של הטקסטים. בסוף כל תהליך הלמידה שנתאר מטה, נרצה להתכנס למסווג סופי לטקסט שיצליח לסווג כל טקסט שיקבל בצורה טובה ואמינה כידיעת אמת או כזב. בתהליך עיבוד המידע נעשה שימוש בספריית nltk שסיפקה לנו מאגרי מילים כגון: stopwords ופונקציות עיבוד טקסט. רעיונות לפיצ'רים נלקחו מהמאמר [5] מלבד פיצ'ר מספר 6.

תהליך העיבוד:

1. הסרת שכפולי דגימות ודגימות ללא טקסט.
2. חישוב פיצ'ר תדירות stopwords (מילים שלא מוסיפות ערך למשפט) לכל טקסט והסרת את ה stopwords מכל טקסט (שמרנו את הטקסט ללא ה stopwords בעמודה נפרדת). מעתה עובדים על הטקסט לאחר הסרת ה stopwords.
3. שימוש ב SentimentIntensityAnalyzer של nltk.sentiment.vader על מנת לבצע Sentiment Analysis לכל טקסט וקבלת 4 פיצ'רים: neg - ההסתברות שהסנטימנט של הטקסט יהיה שלילי, pos - ההסתברות שהסנטימנט של הטקסט יהיה חיובי, neu - ההסתברות שהסנטימנט של הטקסט יהיה ניטרלי, compound - ציון משוקלל של הסנטימנט הכולל בטקסט, מנורמל.
4. חישוב פיצ'ר תדירות מילים הכתובות באותיות גדולות בלבד.
5. חישוב פיצ'ר תדירות סימני הפיסוק בטקסט.
6. חישוב פיצ'רים של תדירויות מילים מתוך קטגוריות במילון Moral Foundation Dictionary. המילון פותח ע"י צוות חוקרים בראשות ג'ונתן היידט וג'סי גרהם כחלק מעבודתם על פסיכולוגיה מוסרית ושונות תפיסת המוסר בין תרבויות [6]. המילון מחולק לקטגוריות ובכל קטגוריה רשימת מילים המתאימה לה. לכל קטגוריה חישבנו פיצ'ר המתאר את תדירות המילים בטקסט הלקוחות מאותה קטגוריה. תחת כל קטגוריה יש מילים התומכות בה (virtue) ומילים המנוגדות לה (vice). הקטגוריות הן: care, fairness, loyalty, authority, sanctity סה"כ 2 * 5 קטגוריות = 10 פיצ'רים.
7. ביצוע Lemmatization על הטקסט (החלפת כל מילה בשורש שלה) וחישוב TF IDF על הטקסט. כתוצאה מכך התווספו מאות פיצ'רים כאשר כל פיצ'ר מייצג ציון TF IDF של מילה מסוימת. השתמשנו ב- TfidfVectorizer של sklearn.

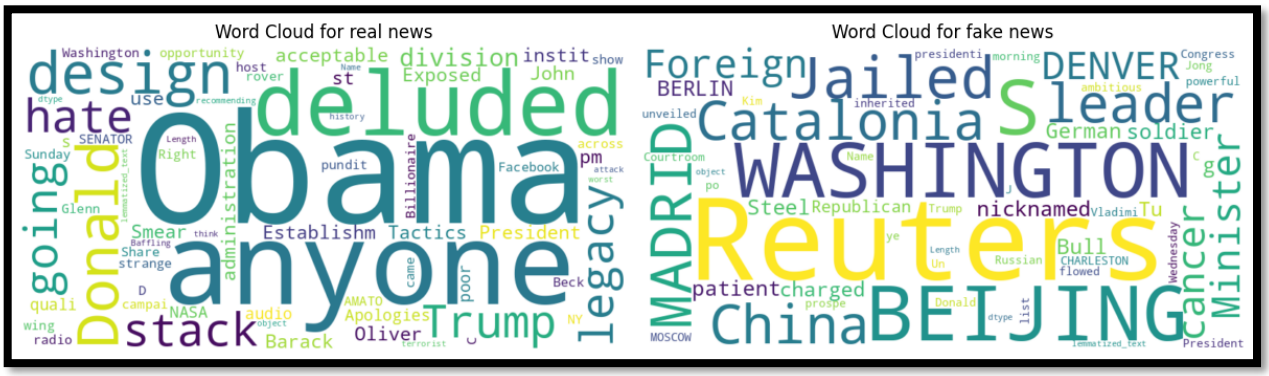
את כל השלבים הנ"ל ביצענו בנוסף על סט המבחן כאשר ה tfidf vectorizer בו השתמשנו לסט המבחן הוא זה שאומן על סט האימון בלבד. את סט האימון והמבחן שמרנו לקבצים לשימוש עתידי. בנוסף הפרדנו את הטקסטים המעובדים (ללא stopwords ולאחר Lemmatization) ושמרנו אותם גם לקובץ.

בשלב זה עצרנו את תהליך חיפוש הפיצ'רים כי רצינו לראות את הביצועים של המודלים על הדאטה שקיבלנו בהמשך נוכל לחזור לשלב זה ולחלץ יותר פיצ'רים.

שלב 3: ויזואליזציה של ה data

1. Word Cloud

ענן מילים הינו פונקציה המקבלת כקלט מספר טקסטים ופולטת רשימה של מילים ממוינת לפי מספר הטקסטים בהם המילה הופיעה. הצגנו שני עננים מילים המתבססים על סט האימון בלבד. האחד חושב על ידיעות אמת (בעלות תיוג 1), והשני על ידיעות כזב (בעלות תיוג 0).



נתבונן בחלק מהמילים הרשומות בכל אחד מהעננים:

	ידיעות אמת	ידיעות כזב
אישיות ציבורית	Barack / Obama - הנשיא ה-44 של ארצות הברית. tramp / donald - הנשיא ה-45 וה-47 של ארצות הברית.	Kim – שם פרטי, עשוי להתייחס לקים ג'ונג און, הנשיא של צפון קוריאה.
רגשות / תחושה / דעה	Deluded- כינוי לאדם "הזוי". Hate – שנאה. Smear - השמצה. apologies - מצטער. worst – גרוע ביותר.	Powerful – חזק. Ambitious – שאפתני.
מקור חדשותי	facebook – אתר אנטרנט/ רשת חברתית.	Reuters – סוכנות חדשות בינלאומית.
מושג	Legacy – מורשת. Acceptable- מקובל.	-
גוף ממשלתי	Administration- ממשל. Senator- חבר הסנאט האמריקאי. Establishment – מוסד. NASA – סוכנות החלל של ארה"ב.	Minister – שר בממשלה. President – נשיא. Republican – רפובליקנים, מפלגה בארה"ב. Congress – קונגרס, גוף פוליטי בארה"ב.
מצב חברתי / אישי	poor – עני. exposed – חשוף, מתאר גילוי או פרסום.	Foreign – זר. Soldier – חייל. Charged – מואשם.
מיקום פיזי / הקשר לאומי.	-	WASHINGTON – עיר הבירה של ארה"ב. BEIJING – בייג'ינג, עיר הבירה של סין. DENVER – עיר בארה"ב. MADRID – מדריד, עיר הבירה של ספרד. Catalonia – קטלוניה, אזור עצמאי בספרד. China – סין. BERLIN – ברלין, עיר בגרמניה. MOSCOW – מוסקבה, עיר הבירה של רוסיה. German – תיאור אדם גרמני.

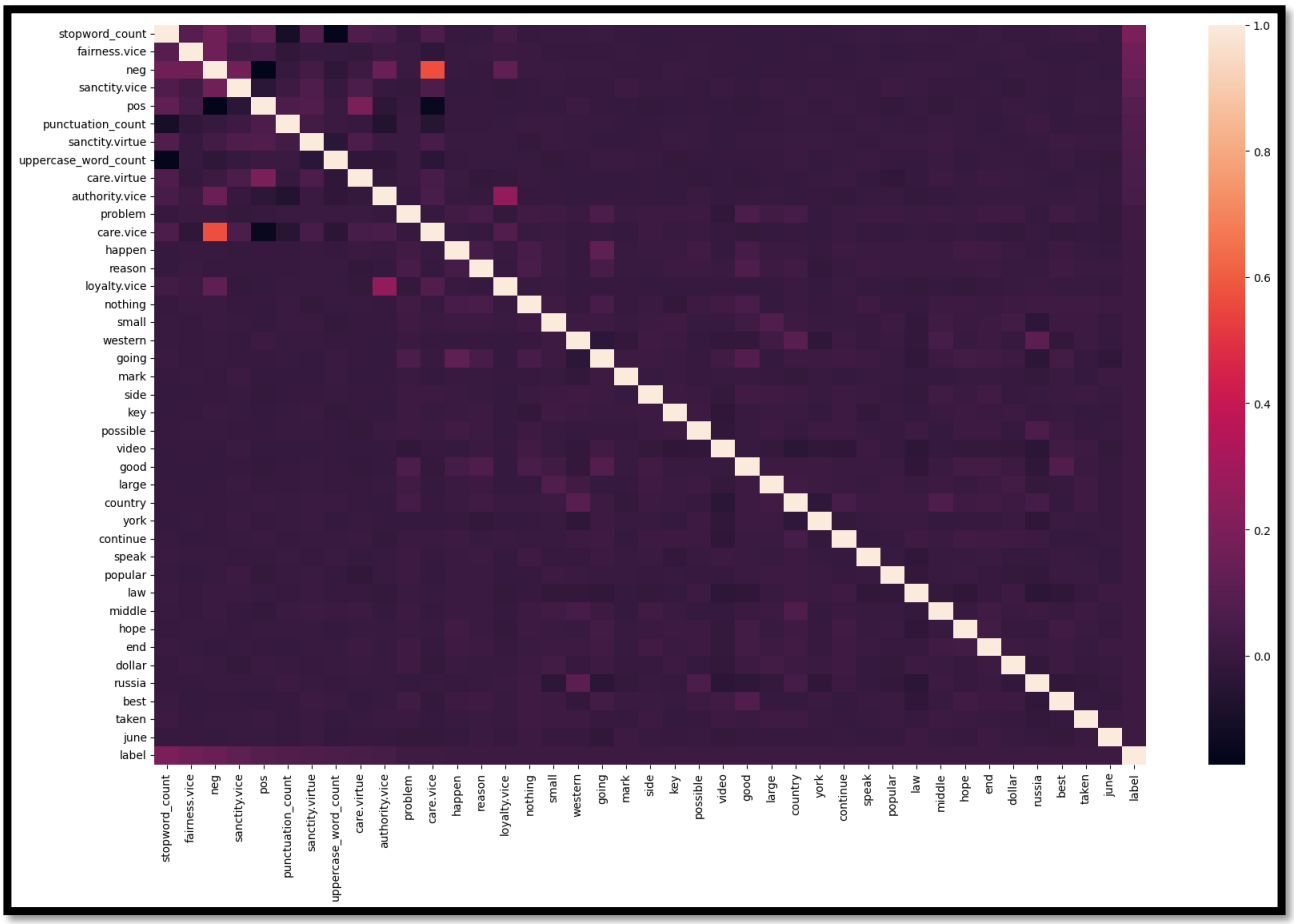
מהתבוננות במילים שקיבלנו כתוצאה מענן המילים, היו מספר מילים שהקטגוריה שבה היו נפוצות הפתיעה אותנו. לדוגמה, ציפינו שנשיא ארצות הברית לשעבר דונאלד טראמפ דווקא יופיע לרוב בידיעות כזב, כי ברשתות החברתיות מופצות המון ידיעות כזב כחלק מהמאבק הפוליטי שבין המפלגות. בנוסף, דווקא הבעת הדעה על אישים כגון "חזק" ו- "שאפתני" אכן הופיעה בידיעות כזב, ייתכן מטעם אנשים שמנסים לאדר את המנהיגים בהם הם בוחרים לדוגמה. הפתיע אותנו שמילים כגון "שנאה", "השמצה", ו- "הזוי" מופיעות תחת ידיעות אמת, שכן היינו מצפות לראות אזכור שלהן דווקא בידיעות כזב כמילים קיצוניות ומתריסות. האזכור הרב של תפקידים פוליטיים ושל גופים בארה"ב בידיעות כזב מדאיג ביותר ויכול לגרום להטיה בדעת האדם התמים שקורא הידיעה ולא מצליח לזהות כי אכן מדובר בידיעת כזב.

2. מטריצת קורלציה

מטריצת קורלציה הינה מטריצה שבעמודותיה ובשורותיה כתובים הפיצ'רים ובתא ה i,j בטבלה מופיעה הקורלציה בין הפיצ'ר ה i לפיצ'ר ה j. הצגנו מטריצות קורלציה לאורך העבודה על מנת לדלות מידע לגבי הפיצ'רים שלנו:

- קורלציה גבוהה בערך מוחלט בין 2 פיצ'רים מעידה על כך שהם מצביעים על אותה תופעה ולכן אולי ניתן לוותר על אחד מהם.
- קורלציה נמוכה בערך מוחלט בין 2 פיצ'רים מעידה על כך שהם לא קשורים ולכן כל אחד מוסיף מידע שהשני לא הוסיף לפניו. לכן לא נוותר על אחד מהם.
- קורלציה גבוהה בערך מוחלט בין פיצ'ר לתיוג מעידה על כך שהפיצ'ר טוב ועוזר בבעיית הסיווג שלנו שכן קשור באופן הדוק לתיוג של הדאטה.

מטריצת הקורלציה של הדאטה שלנו לאחר חישוב הפיצ'רים הראשוני: (מכיוון ויש מאות פיצ'רים, הצגנו במטריצה את 40 הפיצ'רים בעלי הקורלציה הגבוהה ביותר לתיוג)



ניתן לראות כי הפיצ'ר בעל קורלציה מקסימלית לתיוג הינו stopword count. אמנם הקורלציה לא גבוהה אך אינה שלילית עבור אף אחד מהפיצ'רים. ניתן לזהות קורלציה גבוהה ע"י היחס הישר שלה לבהירות התא במטריצה. ניתן לראות זוגות פיצ'רים מסוימים "בהירים" ואחרים כהים כמצופה. לדוגמה הזוג: "care.vice" ו- "neg" בעלי קורלציה גבוהה (בהיר) לעומת הזוג "care.vice" ו- "pos" בעלי קורלציה נמוכה (כהה). מדוגמה זו נסיק שאולי לוותר על הפיצ'ר "care.vice" לא יגרע מאיכות המודל שכן "neg" ו-"pos" תופסים את התופעה שהוא מבקש למדוד- תדירות מילים בטקסט שנוגדות את המונח "דאגה". מנגד, מכיוון שהפיצ'ר הזה תופס קטגוריה ספציפית של מילים בעלות סנטימנט שלילי, אולי שווה להשאיר אותו. לכן לא נמהר לזרוק פיצ'רים מהמודל.

wordcloud קישור המסביר יותר בפירוט על ענן המילים, מופיע גם בביבליוגרפיה.

שלב 4: למידה

אלגוריתמי למידה בהם השתמשנו הם:

1. k שכנים קרובים ביותר (k-NN) בעזרת מטריקה אוקלידית וגם בעזרת מיפוי פולינומי.
2. עצים להחלטות (Decision Trees), בפרט: ID3.
3. מכונות וקטור תמיכה (SVM) עם שימוש בקרנלים שונים: rbf, לינארי ופולינומי.
4. הגברת אדפטיביות (AdaBoost).
5. רגרסיה לוגיסטית (Logistic Regression).
6. יערות אקראיים (Random Forests).
7. XGBOOST.

בחרנו להשתמש באלגוריתמים המתוארים לעיל, לפי הגרף המתואר במאמר [1], אשר הראה את הדיוק שבעבודתו הוא הצליח לקבל מכל מודל:

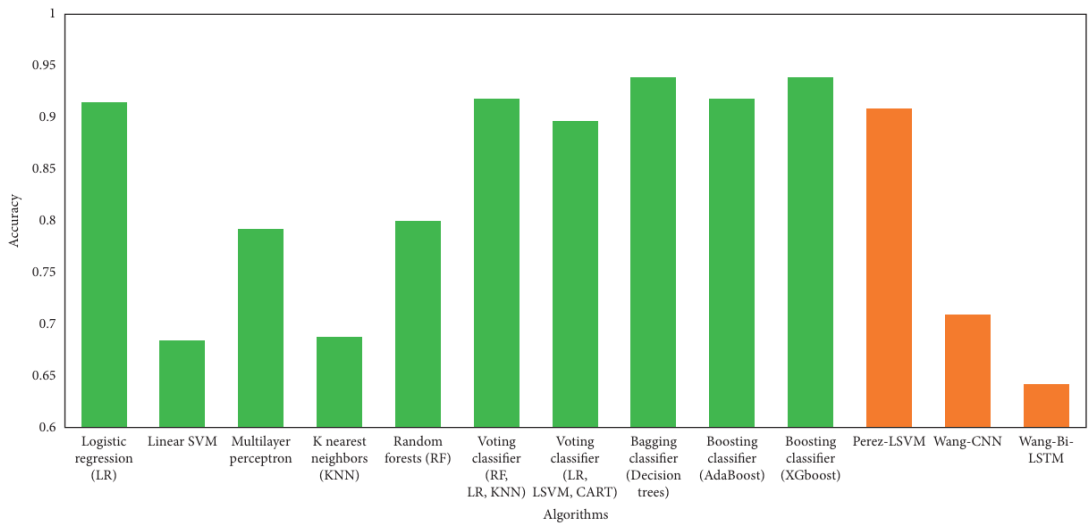


FIGURE 2: Average accuracy over all datasets.

סיקרן אותנו לדעת את התוצאות שאנו נקבל, תוך השוואה למאמר ובתקווה לקבל תוצאות טובות יותר, בעזרת הפיצ׳רים שאנו נבחר. כעת נפרט על כל מודל, את אחוזי ההצלחה שקיבלנו על אימון ועל המבחן, ואת הגרפים המתאימים.

מודל K-nn

מודל זה מסווג את הנקודה לחיזוי לפי התיוג של k נקודות האימון הקרובות אליו ביותר לפי מטריקה שהגדרנו מראש. באלגוריתם זה עם מטריקה אוקלידית, מתבצע חישוב של המרחק האוקלידי בין נקודת החיזוי לכל נקודת אימון, ומחזיר את תיוג רוב הנקודות מבין k הנקודות הקרובות ביותר. בהתחלה, בחרנו ב- k=5, והרצנו לקבלת תוצאות ראשוניות:

Train accuracy: 71.5%

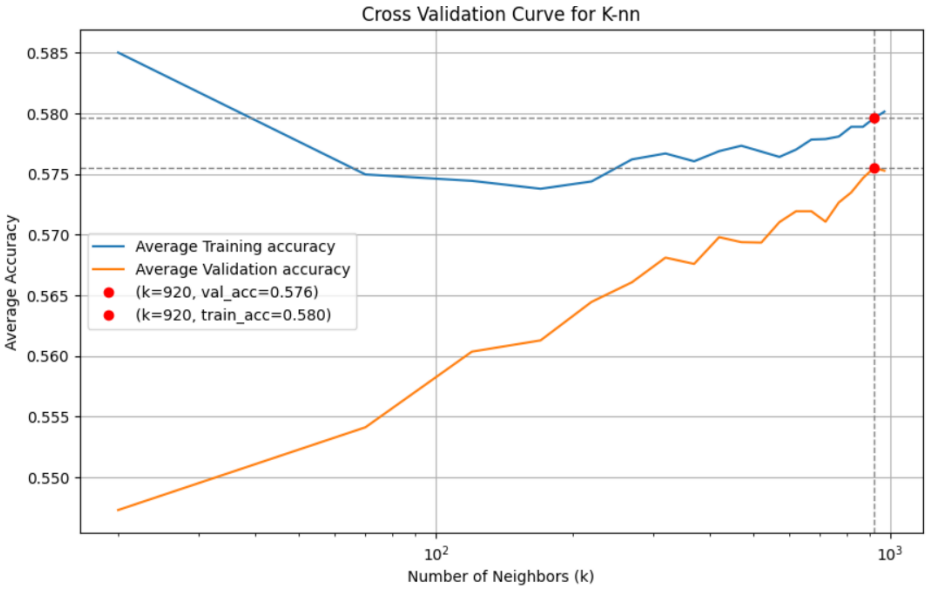
Validation accuracy: 55.6%

כלומר, המידול הראשוני מצליח לשפר את הbase line של ניחוש ב-5.6%. היינו רוצים לשפר זאת.

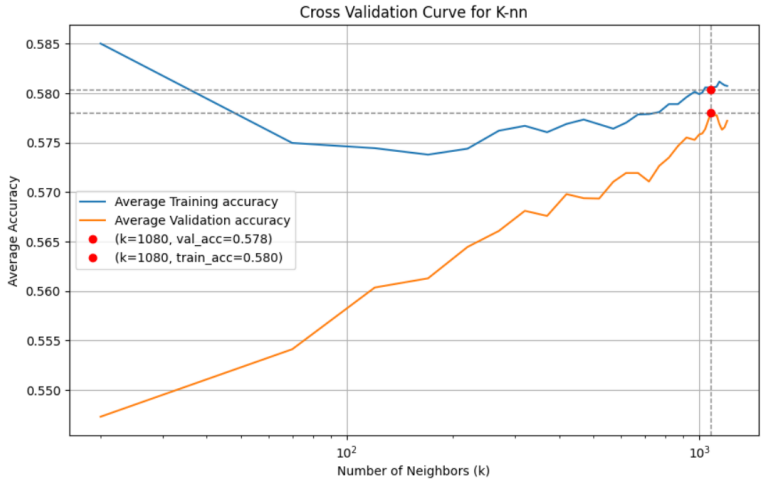
ניסוי: מציאת ה- k האופטימלי (= ממקסם את הדיוק על סט הוולידציה) באלגוריתם ה K-nn.

מהלך הניסוי: לכל k ברשימת הערכים האפשריים שהגדרנו neighbors_list (20 עד 1000 בקפיצות של 50) יצרנו מודל Knn ראשוני וחילקנו את סט האימון ל-5 חלקים. עבור כל חלק אימנו את המודל עם 4 החלקים הנותרים ובחנו אותו על החלק הנוכחי. שמרנו את דיוק האימון והוולידציה בצד. לבסוף, מיצענו את 5 הדיוקים לכדי דיוק אימון ממוצע ודיוק וולידציה ממוצע ושמרנו בצד. לקחנו את הk שקיבל את דיוק הוולידציה הממוצע הטוב ביותר כפרמטר הטוב ביותר שלנו. (שיטת cross validation כפי שהכרנו).

תוצאות הניסוי ומסקנות: קיבלנו את הגרף הבא המתאר את הדיוק הממוצע על סט האימון (בכחול) והדיוק הממוצע על סט הוולידציה (בכתום) כפונקציה של k ואת התוצאות הבאות:



ראשית, ניתן לראות מהגרף כי עקב בחירת רשימת הערכים האפשריים, חתכנו את מגמת העלייה בדיוק הוולידציה הממוצע. נציין שמכיוון שדיוק הוולידציה הממוצע חסום על ידי דיוק האימון, ולפי הגרף זה נראה שהם מתחילים להתקרב, נצפה לשיפור מינורי בתוצאות עקב הרחבת הרשימה. ננסה להרחיב את הרשימה עם ערכים גדולים יותר על מנת לבדוק את השערתנו:

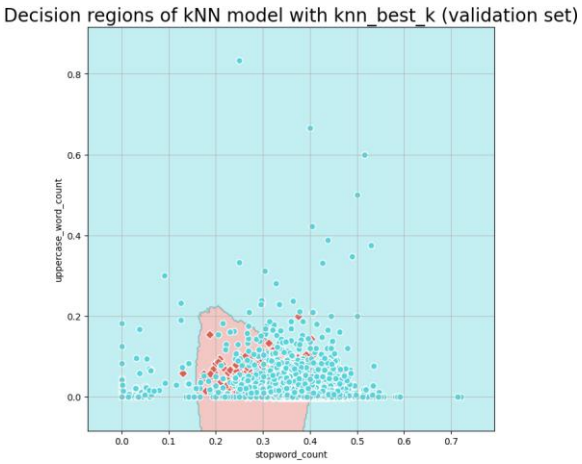


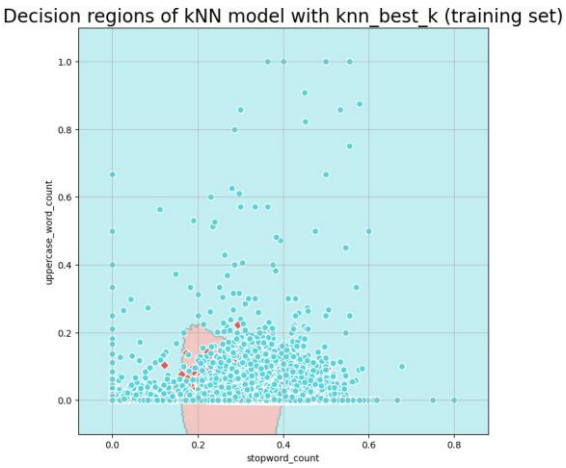
נראה כי צדקנו בהשערתנו. נאמן מודל חדש עם הk האופטימלי שמצאנו ונראה מה הדיוק על סט האימון והווליידציה. ניקח את הדיוק על סט הווליידציה להיות הbase line החדש שלנו אליו נשווה את שאר המודלים:

Train Accuracy: 57.9%

Validation Accuracy: 57.0%

סיבה לתוצאות כאלו יכולה להיות שמודל זה יחסית פשוט ולא מספיק expressive לעולם הבעיה שלנו. הוא לא מסתמך על דברים נוספים מלבד המרחק האוקלידי מנקודות האימון. המטרה שלנו בבחינת מודל פשוט זה היא לשפר את הbase line שלנו מעבר לניחוש פשוט של תיוג הדגימה, כדי לדעת להעריך את טיב התוצאות של מודלים מורכבים ממנו. בנוסף, על מנת להמחיש את הפשטות של מודל זה והתאמתו לבעיה שלנו, ביצענו Knn עם 2 פיצ'רים בלבד עם k האופטימלי שמצאנו מקודם והצגנו את גבולות ההחלטה שלו פעם אחת עם סט האימון ובפעם השנייה עם סט הווליידציה. הפיצ'רים הם: "*stopword_count*" ו "*uppercase_word_count*". בחרנו בפיצ'רים אלו כי הם היו נראים לנו מעניינים ויכולים לעזור לנו לסייע בבעיית החיזוי שלנו ובפרט מהתבוננות בדאטה, הבחנו כי טקסט של ידיעה כוזבת בדר"כ מאופיין בכמות גדולה של אותיות גדולות ובמילות עצירה. קיבלנו את התרשימים: (פונקציית הויזואליזציה לקוחה מתרגיל בית בקורס מבוא למערכות לומדות)

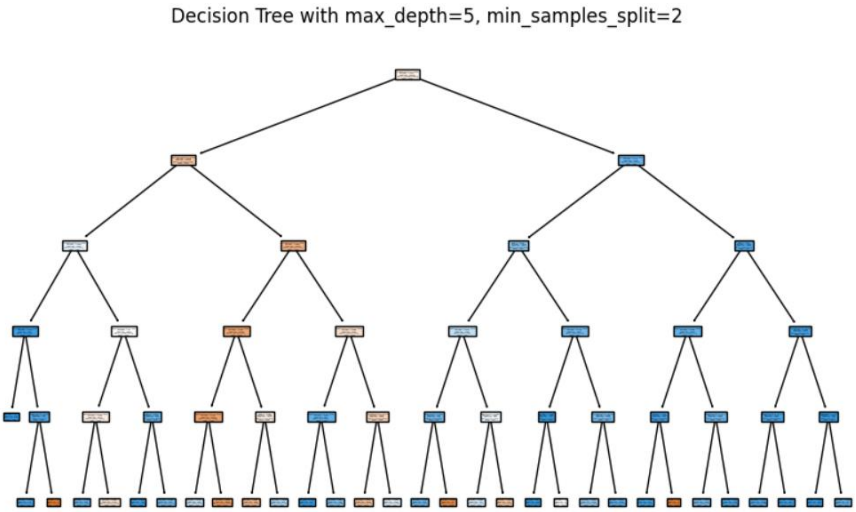




ניתן לראות כי במקרה של 2 פיצ'רים אכן המודל לא מצליח ללמוד את ההתפלגות (וגם לעולם לא יצליח) מכיוון שהמרחקים בין הנקודות בבעיה שלנו לא רלוונטיים לסיווג. ניתן לראות זאת בכך שכל הנקודות יושבות באותו איזור בלי קשר לצבע שלהן. לכן המודל פשוט מידי ונרצה להשתמש במודלים מורכבים יותר.

מודל ID3

מודל ID3 יוצר עץ החלטה ע"י תהליך איטרטיבי בו בכל איטרציה בחירת הפיצ'ר שלפיו מפצלים את הדגימות ב- node הנוכחי הוא זה שמביא למקסימום את ה Information Gain - מדד המעיד כמה פיצול בעץ היה אינפורמטיבי ע"י שימוש באנטרופיה. התהליך עוצר כאשר מתקבלים עלים טהורים (כל הדגימות מסווגות באותו תיוג) או עד שמגיעים למגבלת עומק או מספר דגימות מינימלי בעלה. בהתחלה, בחרנו בהגבלת עומק העץ להיות 5 ומספר מינימלי של דוגמות בעלה להיות 2 (דיפולט). לאחר האימון, קיבלנו מבנה כזה של העץ:



נשים לב כי זהו מבנה של עץ כמעט מלא. כלומר, מבחינת המודל, לא קיים פיצול ברמות העליונות של העץ אשר יביאו לפיצול דאטה כזה שאחד מבניו יהיה עלה. משמעות הדבר היא שאין פיצ'ר או 2 או 3 פיצ'רים אשר לבדם מצליחים לסווג קבוצה מסוימת של דגימות שליליות/ חיוביות. כלומר, העץ אינו מסתפק בפחות מ5 רמות בלבד על מנת לסווג נכונה חלק מהדוגמאות בסט האימון. התוצאות שקיבלנו:

Training Accuracy: 73.6%

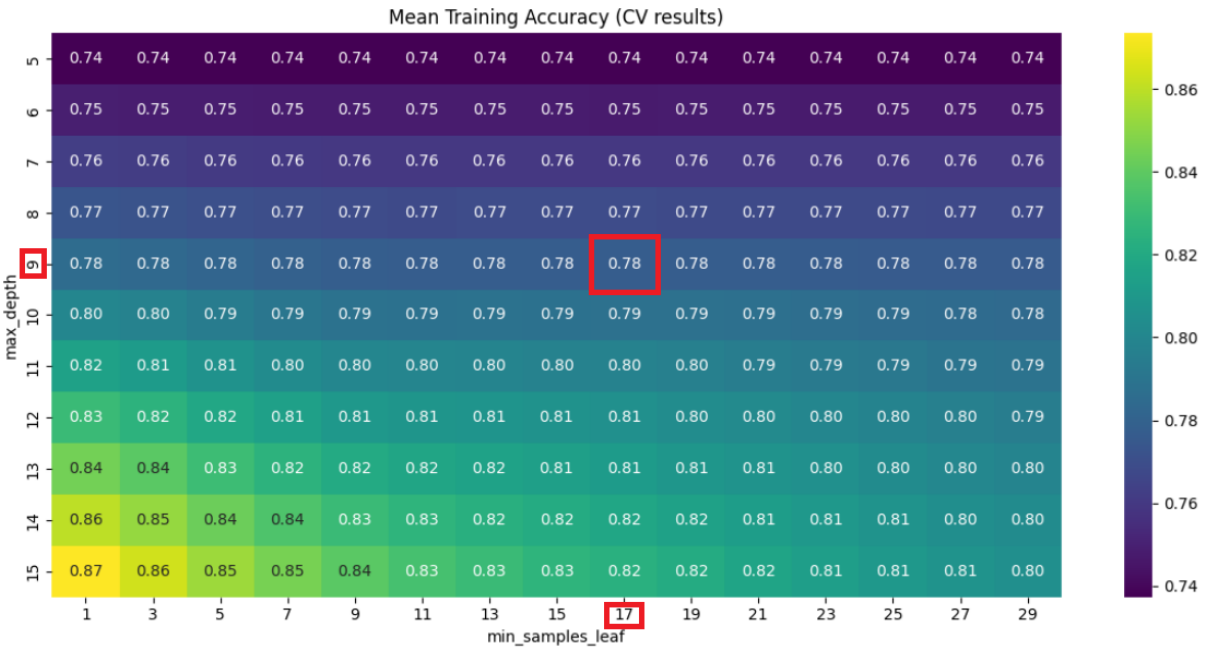
Validation Accuracy: 72.9%

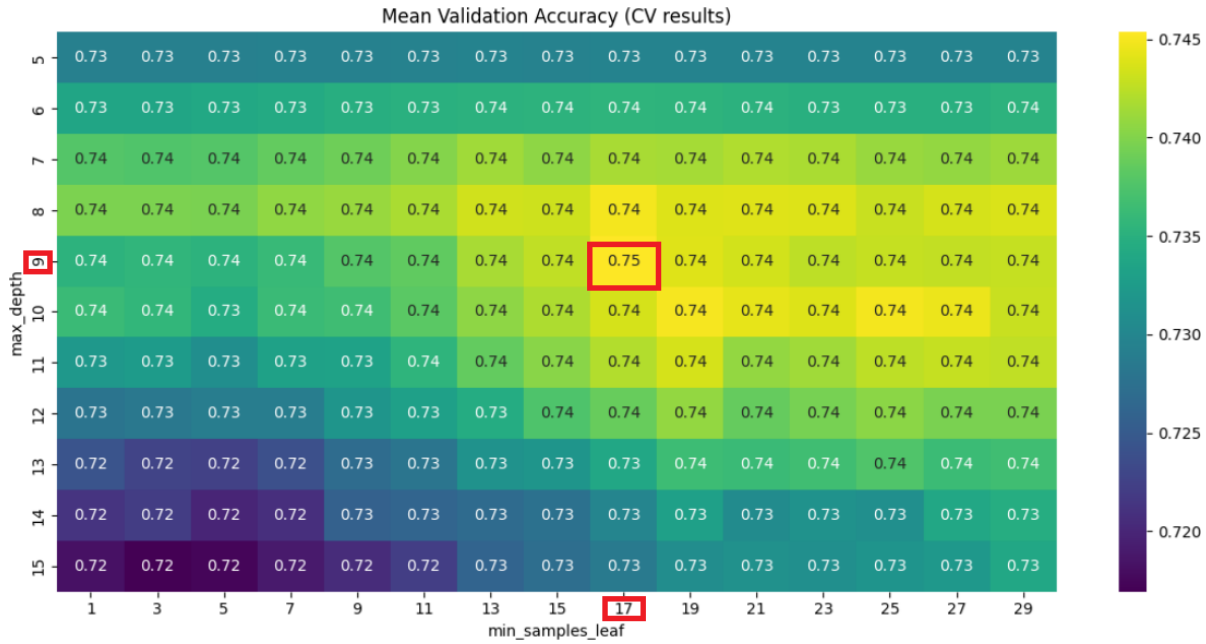
התוצאות שקיבלנו טובות יותר מה base line החדש בכ 16%. נרצה למצוא את הערכים האופטימליים שאיתם נוכל לקבל הכללה טובה יותר על סט הוולידציה.

ניסוי: מציאת ערכי היפר הפרמטרים: max_depth ו- min_samples_leaf האופטימליים.

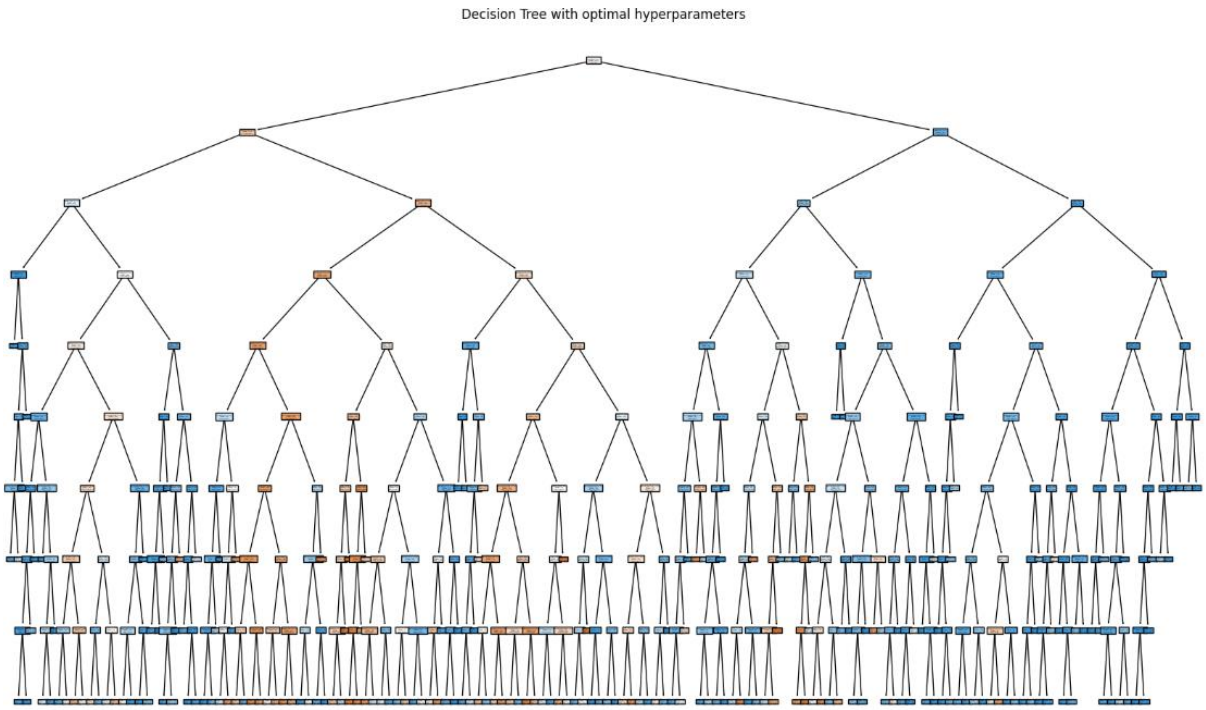
מהלך הניסוי: השתמשנו באותה שיטה כמו בKnn רק שהפעם מצאנו מחלקה הממשת את מה שאנחנו עשינו בKnn ידנית. השתמשנו בתחומים: range(1, 30, 2) עבור min_samples_leaf וב range(5, 16) עבור max_depth.

תוצאות הניסוי ומסקנות: על מנת להמחיש את תוצאות הניסוי כתבנו קוד אשר משתמש ב seaborn על מנת להמחיש מפת חום המתארת לכל זוג פרמטרים את הדיוק הממוצע שקיבלנו על סט האימון ועל סט הוולידציה בתהליך הcross validation. המפה מתארת לכל זוג הפרמטרים את הדיוק ע"י מתן צבע- בהיר יותר ככל שהדיוק גבוה יותר. הדגשנו בריבוע אדום על מפות החום את התאים בטבלה המתאימים לזוג הפרמטרים האופטימלי:





לפי מפות החום נקרא כי באופן גורף דיוק על סט הוולידציה נמוך מדיוק על סט האימון. עם זאת, לפי התוצאות המודל לא בוחר את היפר הפרמטר שהביא לדיוק מקסימלי על סט האימון אלא את זה שהביא לדיוק מקסימלי על סט הוולידציה ובכך משפר את יכולת ההכללה שלו. כעת הרצנו שוב את המודל, הפעם עם הערכים האופטימליים וקיבלנו מבנה של עץ כזה:



נשים לב כי כעת העץ יותר עמוק (עומק 9) וזאת מכיוון שככל הנראה ההיפר פרמטרים ההתחלתיים לא נתנו לעץ מספיק יכולת הכללה. בנוסף, ניתן לראות שקיימים מסלולים בעץ הנגמרים ברמה נמוכה יותר מהרמה המקסימלית שאפשרנו לו. זה אומר שתהליך הלמידה הצליח לסווג קבוצות מסוימות של פיצ'רים בצורה טובה מאחר והוא לא היה צריך לרדת עמוק יותר. סיבה אחרת יכולה להיות בגלל הגבלת מספר הדגימות בעלה. התוצאות שקיבלנו על סט האימון והוולידציה הן:

Train Accuracy: 77.2%
Validation Accuracy: 72.9%

ניתן להבחין כי הדיוק על סט הוולידציה לא השתנה. אמנם הדיוק על סט האימון ירד, מכיוון שממקסמים על דיוק הוולידציה בכל איטרציה של האלגוריתם ולא על דיוק האימון, מה שמפחית את הסיכויים שהמודל יתאים את עצמו טוב מידי לסט האימון. אנחנו מניחות כי התוצאות לא השתפרו למדיי מעבר לדיוק עבור בחירה שרירותית מכיוון שעץ למידה לבדו הוא מודל פשוט מידי שלא מצליח ללמוד את ההתפלגות המורכבת של הדאטה שלנו. לכן שיפור הדיוק ע"י כיוונון הפרמטרים שלו לא ייתן לנו ביצועים טובים יותר, אלא עלינו לבחון מודלים נוספים לשם כך. בנוסף, ע"י "זום אין" בדקנו מה הם פיצ'רים שהופיעו בארבע הרמות הראשונות של העץ. זה ייתן לנו אינדיקציה לזהותם של הפיצ'רים שהצליחו להפריד הכי טוב בין נקודות בעלות סיווג שונה כלומר שהפחיתו הכי הרבה את האנטרופיה. קיבלנו:

features used in first 4 levels of the tree ordered:

'stopword_count', 'authority.virtue', 'neg', 'punctuation_count', 'fairness.vice'

נשים לב כי ב-4 השכבות הראשונות של העץ יש 15 צמתים. לעומת זאת, אנחנו קיבלנו שימוש ב 5 פיצ'רים בלבד. ועל כן נוכל להסיק כי המודל נתן חשיבות יתרה לפיצ'רים אלו. נתייחס לכך בחלק של feature selection.

מודל Soft SVM

SVM (Support Vector Machine) הוא מודל למידה המתבסס על פרידות לינארית במשימתו לסווג דאטה למחלקות בינאריות. הוא ינסה למצוא מישור n ממדי (n - מספר הפיצ'רים) שיכול להפריד בין שתי הקבוצות עם מינימום שגיאה על סט האימון, תוך הסכמה על כך שלא כל הנקודות של האימון יהיו בצד הנכון של קו ההפרדה, וזה בסדר שחלק מהן יקבלו סיווג שגוי או יהיו קרובות מדי לקו ההפרדה על מנת לגרום ליכולת ההכללה של המודל להיות טובה. במקרה שלנו קבוצה עם תיוג 0 היא מידע כוזב, וקבוצה עם תיוג 1 היא מידע אמין. פונקציית המטרה של המודל, אותה מנסה להביא למינימום היא:

$$\underset{\boldsymbol{w}\in\mathbb{R}^d,b\in\mathbb{R}}{\operatorname{argmin}}\ \lambda\|\boldsymbol{w}\|_2^2+\frac{1}{m}\sum_{i\in[m]}\max\{0,1-y_i(\boldsymbol{w}^\top\boldsymbol{x}_i+b)\}$$

Complexity penalty (Regularization)
Hyperparameter λ controls the model complexity

Margin violation penalty
How much the ith example *violates* the margin

כאשר החלק האדום הינו השגיאה של המודל, והחלק הכחול הינו מורכבות המודל (רגולריזציה) המוודא שהמודל לא יתאים את עצמו בצורה טובה מידי לסט האימון (overfitting). במימוש השתמשנו ב SVC של sklearn ובה משתמשים בנוסחה הבאה השקולה לה:

$$\underset{\boldsymbol{w}\in\mathbb{R}^d,b\in\mathbb{R}}{\operatorname{argmin}}\ \|\boldsymbol{w}\|_2^2+C\sum_{i\in[m]}\max\{0,1-y_i(\boldsymbol{w}^\top\boldsymbol{x}_i+b)\}$$

למודל זה אפשרות להתמודד עם datasets מורכבים יותר שאינם פרידים לינארית בהכרח ע"י שימוש בקרנלים. בפועל בשימוש בקרנל מתבצע מיפוי מסוים על הדגימות (למשל עבור דגימות במרחב החד ממדי מיפוי פולינומי ריבועי ימפה את הדגימות למישור ע"י פרבולה) לממד גבוה יותר בתקווה שזה מה שיעזור להפריד את הדגימות מהמחלקות השונות ע"י מישור רב ממדי. השתמשנו במודל זה עם כמה סוגי קרנלים שונים: rbf, polynomial, linear.

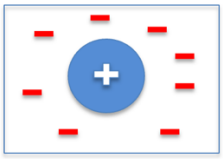
נפרט את התוצאות עם הקרנלים השונים עבור מודל זה בטבלה המצורפת בעמוד הבא.

בנוסף, מצורפת תמונה הממחישה את ההבדלים בין הקרנלים השונים בנספחים.

ניסוי: השוואת קריטריונים בין שימוש בקרנלים: RBF, Polynomial, Linear במודל SVM.

החל ממודל זה והלאה, שינינו את שיטת כיוונון הפרמטרים לשיטה אחרת המפורטת לאחר חלק זה (למידה).

תוצאות:

קרנל	rbf	polynomial	linear
תיאור קצר	קרנל זה ממפה למרחב אינסופי ומפריד בין המחלקות לסיווג בהתאם למרחק מנקודת מרכז, נקודת ייחוס במרחב. הקרנל מודד את המרחק של כל נקודת אימון מכל נקודת מרכז של כל מחלקה, כדי לקבוע לאיזו מחלקה היא קרובה יותר ובכך מסווג אותה, באמצעות פונקציה רדיאלית. קרנל זה מתאים למקרים מורכבים ובפרט במקרים בהם לא ניתן להפריד לינארית. הסבר נוסף מתוך "ויקיפדיה".	קרנל זה מייצר תכונות חדשות באמצעות חישוב מכפלה פולינומית של הפיצ'רים הקיימות עד לדרגה הנקבעת לבחירתנו. הוא מתאים למקרים שבהם הקשר בין המחלקות אינו ליניארי, אך גם אינו דורש מעבר למרחב אינסופי כמו ב- RBF. כאשר הדרגה גדולה, המודל אמנם יכול ליצור קשרים מורכבים יותר בין הפיצ'רים אך עם זאת הוא עלול לנטות ל- overfitting. מצורפת תמונה הממחישה זאת בנספחים מקורס מבוא למערכות לומדות. באמצעות מודל זה ניתן לקבל הפרדה עם פונקציות מורכבות יותר, לדוגמה עם דרגה 2 נוכל לסווג נכונה כאשר יש קשר מעגלי בין המחלקות:  הסבר נוסף מתוך "ויקיפדיה".	קרנל זה נחשב לפשוט ביותר. הוא מתאים למקרים שבהם יש הפרדה ליניארית ברורה בין המחלקות לסיווג, כלומר ניתן להפריד ביניהן באמצעות קו ישר או מישור ובפרט הוא אינו יעיל כלל במקרים בהם אין הפרדה לינארית.
הרצה ראשונית			
ערכי היפר הפרמטרים	C = 1 gamma = 'scale'	C = 1 gamma = 'scale'	C = 1
דיוק על סט האימון (%)	78.0	82.8	71.5
דיוק על סט הוולידציה (%)	67.3	67.4	68.6
ניסוי hyper parameters tuning			
אילו פרמטרים כוונו	gamma ,C	gamma ,C	C
תחום האפשרויות לכל פרמטר	C: [0.5,1,1.5,5,10,20,50,100] gamma: [1e-3,5e-3,0.01, 0.05,0.1,0.5]	C: [0.5,1,1.5,5,10,20,50,100] gamma: [1e-3,5e-3,0.01,0.05,0.1,0.5]	C: range(0.1,1.5,0.01) range(1.5,15.5,0.5) range(15, 50, 5)
דיוק ממוצע על סט האימון (%)	73.4	82.6	72.0
דיוק ממוצע על סט הוולידציה (%)	66.1	65.0	66.4
פרמטרים אופטימליים	C = 50 gamma = 0.005	C = 0.5 gamma = 0.5	C = 0.91
מודל בעל היפר פרמטרים אופטימליים			
דיוק על סט האימון (%)	72.4	80.9	71.4
דיוק על סט הוולידציה (%)	67.9	67.4	68.4

מסקנות: באופן גורף, כל הקרנלים לא הצליחו להגיע לדיוק גבוה כלל על סט הוולידציה: בין 67 ל-68 אחוזים לכולם – שיפור של בין 10-11% ביחס לbase line ואף מבצעים גרוע יותר מעץ ההחלטה גם לאחר כיוונון הפרמטרים. ניתן לשער אם כן כי גם לאחר מתן חוזק למודל הלינארי באמצעות קרנלים שונים, עדיין אופן הסיווג הלא לינארי של עץ ההחלטה מצליח ללמוד טוב יותר את ההתפלגות. ניכר כי אין הבדל משמעותי בדיוק בין הקרנל הלינארי לפולינומי, למרות החוזק הרב שיש לקרנל הפולינומי בהשוואה ללינארי. למרות זאת, נרצה לציין תכונה של הפיצ׳רים שניתן להסיק ע״י למידה ממודל זה (עם קרנל לינארי): במודל זה, פיצ'רים שקיבלו ערכים גבוהים בערך מוחלט בכניסות המתאימות בווקטור המשקולות w, משפיעים משמעותית על המישור הרב ממדי המסווג. לעומת זאת, פיצ'רים שקיבלו ערכים נמוכים בערך מוחלט בכניסות המתאימות ב w ישפיעו פחות על המישור. ננצל תכונה זו בהמשך על מנת לבצע feature selection.

Optimal C=0.91	C=1
-15.36559275045047 authority.virtue 13.855238878957906 fairness.vice 11.974582162076228 stopword_count 11.203110883076377 uppercase_word_count 7.672132666948277 sanctity.vice -6.908407476040826 care.vice -6.512369209374938 loyalty.virtue 5.234426948817056 sanctity.virtue 5.105808186791175 punctuation_count -4.103374420801821 neu -3.6918368253510048 fairness.virtue 3.2413325244880027 care.virtue 2.9284914474193613 authority.vice 2.6076681043550844 pos -1.5726808524779032 fight 1.5305073129318072 happen -1.4302567892700941 began 1.3781059176003674 40	-15.803280653203291 authority.virtue 14.754445277123306 fairness.vice 11.995413287231758 stopword_count 11.363198192606182 uppercase_word_count 8.298999183203222 sanctity.vice -7.251175635073062 care.vice -6.684174374359932 loyalty.virtue 5.454804623941294 sanctity.virtue 5.1118269759389525 punctuation_count -4.110462110399125 neu -3.846718808679695 fairness.virtue 3.4357876222700154 care.virtue 3.1605902972177833 authority.vice 2.549244817598166 pos 1.6029677583454496 happen -1.5735822855835746 fight -1.4593440801600126 began 1.3687711879630717 key

נשים לב כי הרבה מהפיצ׳רים שהתקבלו כמשמעותיים ביותר במודל זה, הינם פיצ׳רים שהתקבלו מהמילון MFD שלקחנו ממאמר [6]. ניתן לראות כי הפיצ'ר בעל המשמעות הגבוהה ביותר בשני המקרים הוא authority.virtue שמקבל סימן שלילי בווקטור המשקולות w אך ערך גבוה בערכו המוחלט. כלומר, בעל השפעה רבה על הסיווג של המודל. המשמעות של פיצ'ר זה הינה האם השפה בה משתמשים בטקסט מבטאת סמכות, נורמות וערכים מסורתיים ובנוסף האם יש שימוש בביטויים שמדגישים היררכיה ומשמעת. בנוסף, שלושת הפיצ'רים המשותפים המשמעותיים ביותר הינם: stopword_count
uppercase_word_count ו- fairness.vice. זה מתכתב עם הפיצ'רים שראינו ב4 הרמות הראשונות של עץ ההחלטה. נשים לב כי ב14 הפיצ'רים הראשונים ישנה הסכמה גורפת על סדר הפיצ'רים שקיבלו את המשקולות עם הערך המוחלט הגבוה ביותר לפני הכיוונון ולאחר הכיוונון לכן נסיק כי עבור הפיצ'רים ה"כבדים" (להם נתן המודל משקל גבוה בערך מוחלט) המודל רק כיוונן את המשקולות ולא בחר פיצ'רים אחרים. זה מעיד על עקביות מעניינת של המודל ביחס לבחירת Cים שונים ונוכל להסתמך עליה בנימוק לחלק בחירת הפיצ'רים בו נתאר את אופן הבחירה בשיטת SVM עם קרנל לינארי. כלומר, המודל נותן אותה חשיבות לאותם פיצ'רים ללא קשר לפרמטרים שלו.

מודל AdaBoost

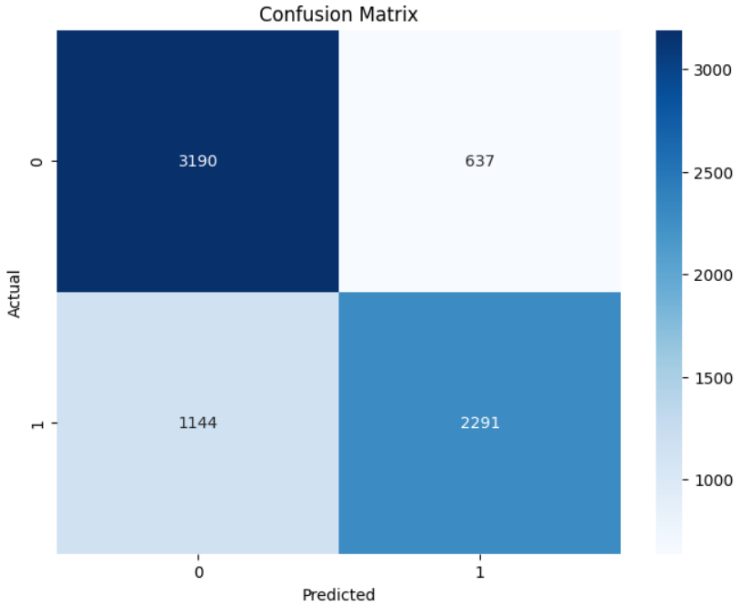
AdaBoost הוא אלגוריתם מסוג אנסמבל. הוא משלב מודלים פשוטים, כאשר כל אחד מהם נחשב חלש ולא מסווג טוב בהכרח. מטרת השילוב היא קבלת מודל אשר יניב סיווג מדויק יותר וטוב יותר של ה- data בעזרת אגרגציה על כלל המודלים החלשים שאומנו. בכל שלב של אלגוריתם הלמידה, אם מודל מסוים טועה בסיווג דוגמה כלשהי, האלגוריתם נותן לדוגמה הזו יותר חשיבות בשלב הבא על מנת לתקן את הטעות. על כן, המודל הבא מתמקד יותר בתיקון הטעויות שנעשו על ידי המודל הקודם. לבסוף, כל המודלים מחוברים יחד לתוצאה אחת משוקללת, שבה הדגש ניתן יותר למודלים שתיקנו את הטעויות. בהתחלה השתמשנו במודל זה, עם מספר המודלים החלשים: T, להיות 20. קיבלנו את התוצאות הבאות:

Train Accuracy: 75.9%

Validation Accuracy: 75.4%

עבור בחירה שרירותית במספר המודלים החלשים, T, קיבלנו שיפור של כ- 18% מה base line של מודל הKnn ושיפור של כ-3% מתוצאות מודל עץ ההחלטה. לאור תוצאות אלו, מודל זה מעניין אותנו ולכן בחרנו להציג מספר תרשימים החוקרים את הביצועים שלו יותר לעומק:

תרשים 1: מטריצת בלבול על סט הוולידציה



תרשים זה מציג את תוצאות המודל על גבי סט הוולידציה. ניתן לראות כי אמנם את רוב נקודות המבחן הוא מסווג נכונה, אך בבחינת נקודות המבחן שהמודל טועה עליהן נמצא כי הוא מסווג 637 ידיעות כזב כידיעות אמת, ומסווג 1144 ידיעות אמת כידיעות כזב.

מסקנה: המודל מדייק יותר בזיהוי ידיעות כזב אך מתקשה בזיהוי ידיעות אמת. זאת ניתן לראות טוב יותר בתרשים הבא.

confusion matrix – זהו קישור המסביר יותר בפירוט על מטריצת הבלבול. הוא מצורף גם בביבליוגרפיה.

תרשים 2: מדדים

	precision	recall	f1-score	support
0	0.74	0.83	0.78	3827
1	0.78	0.67	0.72	3435
accuracy			0.75	7262
macro avg	0.76	0.75	0.75	7262
weighted avg	0.76	0.75	0.75	7262

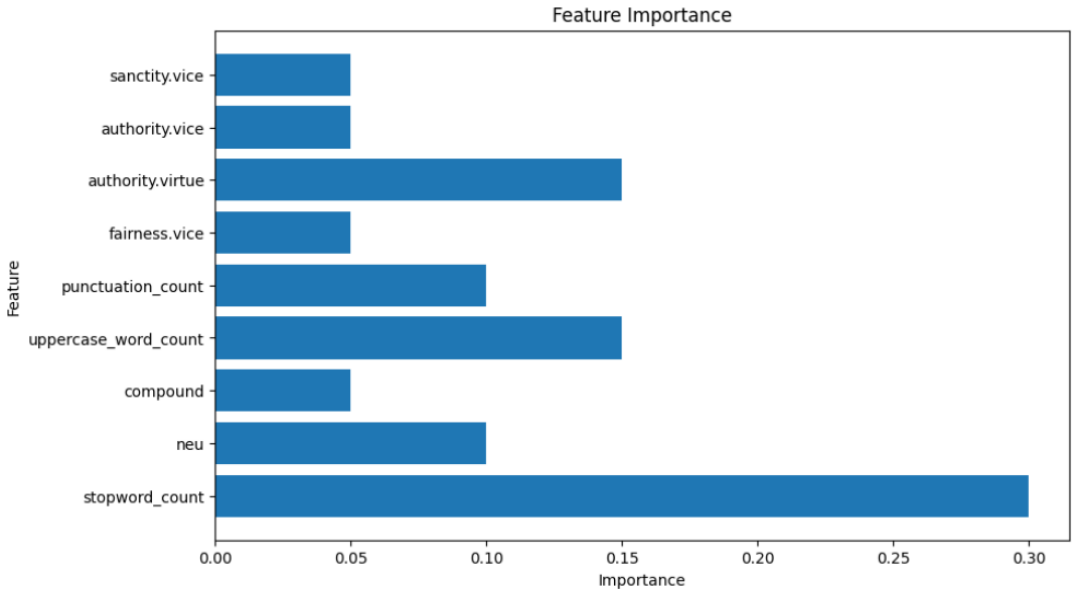
תרשים זה מציג תוצאות של מספר מדדים שונים על גבי המודל הנ״ל.

<u>הסבר התוצאות המוצגות בטבלה:</u>

1. accuracy: אחוז הדוגמאות שהמודל מסווג נכונה מכלל הדוגמאות - 75%.
2. precision: לכל מחלקה, אחוז הדוגמאות השייכות למחלקה מתוך כלל הדוגמאות אותן סיווג המודל למחלקה זו. מדד זה מראה עד כמה התחזיות החיוביות/ השליליות של המודל היו נכונות.
   - עבור המחלקה 0 הדיוק הוא 74%. כלומר, 74% מהתחזיות שסומנו כ-0 ע"י המודל היו נכונות.
   - עבור המחלקה 1 הדיוק הוא 78%. כלומר, 78% מהתחזיות שסומנו כ-1 ע"י המודל היו נכונות.
3. recall (שליפה): מדד זה מראה כמה מהדוגמאות שבאמת היו חיוביות/שליליות בהתאמה זוהו על ידי המודל ככאלה.
   - עבור המחלקה 0 השליפה היא 83%. כלומר, המודל סיווג נכון 83% מכל הדוגמאות שתיוגן 0.
   - עבור המחלקה 1 השליפה היא 67%. כלומר, המודל סיווג נכון 67% מכל הדוגמאות שתיוגן 1.
4. מדד f1-score: מדד זה משלב את מדדי הדיוק והשליפה (2 כפול מכפלתם חלקי סכומם).
   - למחלקה 0 יש f1-score של 78%, מה שמעיד על איזון טוב בין דיוק לשליפה.
   - למחלקה 1 יש f1-score של 72%, מה שמעיד על כך שלמודל יש קצת יותר קושי בזיהוי מדויק של מחלקה זו.
5. ממוצעים:
   - macro avg: הממוצע של הדיוק, השליפה ו- f1-score. כאן הינו 75%-76%.
   - weighted avg: ממוצע משוקלל של כל מדד לפי מספר הדוגמאות בכל מחלקה. במקרה זה שווה ל macro avg.

המסקנה הכללית היא כי אכן המודל מזהה טוב יותר את המחלקה 0 מאשר המחלקה 1. כלומר שיעור גבוה של False negative. ניתן לראות זאת לפי ה-f1-score, הדיוק והשליפה של המחלקה של ידיעות הכזב, שהם מעט גבוהים יותר לעומת המחלקה של ידיעות האמת. כעת ניתן לשאול את עצמנו מה החשיבות של כל מדד מבחינתנו. התשובה היא שזה תלוי מי הלקוח של המוצר וניתן להגדיר אופני שימוש בו על מנת לצמצם את המדדים האלה. לדוגמה, לנסות להוציא אחוז וודאות של המודל לכל סיווג של ידיעה, כך שדוגמאות בעלות וודאות נמוכה תסווגנה ידנית.

תרשים 3: חשיבות הפיצ'רים המשמעותיים ביותר באחוזים



על בסיס תכונת feature_importances של המודל המאומן, ניתן לחלץ את הפיצ'רים החשובים לו ביותר לפי הסדר. בתרשים לעיל הדפסנו לפי הסדר רק את אלו שעברו את הסף 0.005. ניתן לראות כי הפיצ׳רים המשמעותיים ביותר במודל זה הינם stopword_count, uppercase_word_count ו- authority.virtue כפי שראינו כבר במודל עץ ההחלטה ובמודל ה SVM. מעניין לראות את ההסכמה בין המודלים לגבי הפיצ'רים החשובים. נתייחס לכך בהרחבה בחלק של feature selection.

ניסוי: כיוונון היפר הפרמטרים: learning_rate ו- n_estimators (קצב הלמידה ומספר מודלים חלשים)

תוצאות ומסקנות:

Best parameters found: {'learning_rate': 0.7, 'n_estimators': 95}

Average training accuracy for best parameters: 77.4%

Average validation accuracy for best parameters: 76.2%

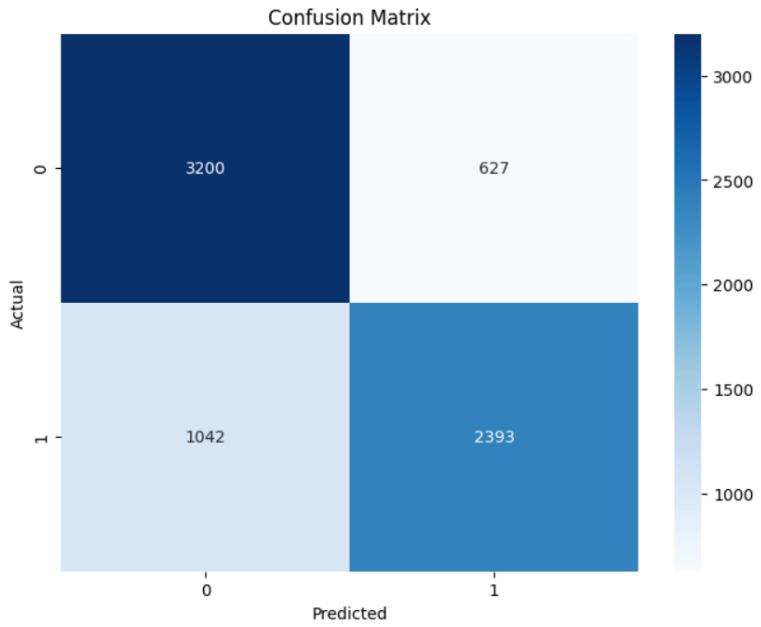
במהלך הכיוונון הדיוק הממוצע על סט הוולידציה השתפר בכ-1% לעומת הדיוק טרם הכיוונון. נאמן כעת את המודל על סט האימון ונבחן אותו על סט הוולידציה. קיבלנו:

Train Accuracy: 77.4%

Validation Accuracy: 77.0%

הכיוונון שיפר את ביצועי המודל בכ-2% בלבד. נצהיר כי אמנם, השימוש בשיטה החדשה שיפרה את זמן הריצה משמעותית (מפורט תחת שיטת כיוונון הפרמטרים) אך אנחנו לא יודעים אם זה השיפור המקסימלי בביצועים שקיים מאחר והשיטה הזו מתפשרת על מציאת המקסימום הגלובלי תמורת ביצועים. מבחינתנו זה מתקבל על הדעת מכיוון שזמן הריצה מאוד חשוב לנו ואיננו מוכנות להתפשר אליו וגם משום שלא הייתה ירידה בביצועים ביחס לבחירה שרירותית ואף עלייה. בנוסף, נעדיף להשקיע את הזמן והאנרגיות בשלב זה בשיפור ביצועי המודל במישורים אחרים כמו למשל בחירת הפיצ'רים.

תרשים 1: מטריצת הבלבול על סט הוולידציה לאחר כוונון הפרמטרים



ישנה ירידה של 100 ידיעות בסיווג False negative וירידה של 10 ידיעות בסיווג False positive. לכן נסיק כי למרות הכיוונון המודל מתקשה בצמצום אחוזי הFalse positive וזאת ניתן לראות טוב יותר בתרשים הבא:

תרשים 2: מדדים לאחר כיוונון הפרמטרים

	precision	recall	f1-score	support
0	0.75	0.84	0.79	3827
1	0.79	0.70	0.74	3435
accuracy			0.77	7262
macro avg	0.77	0.77	0.77	7262
weighted avg	0.77	0.77	0.77	7262

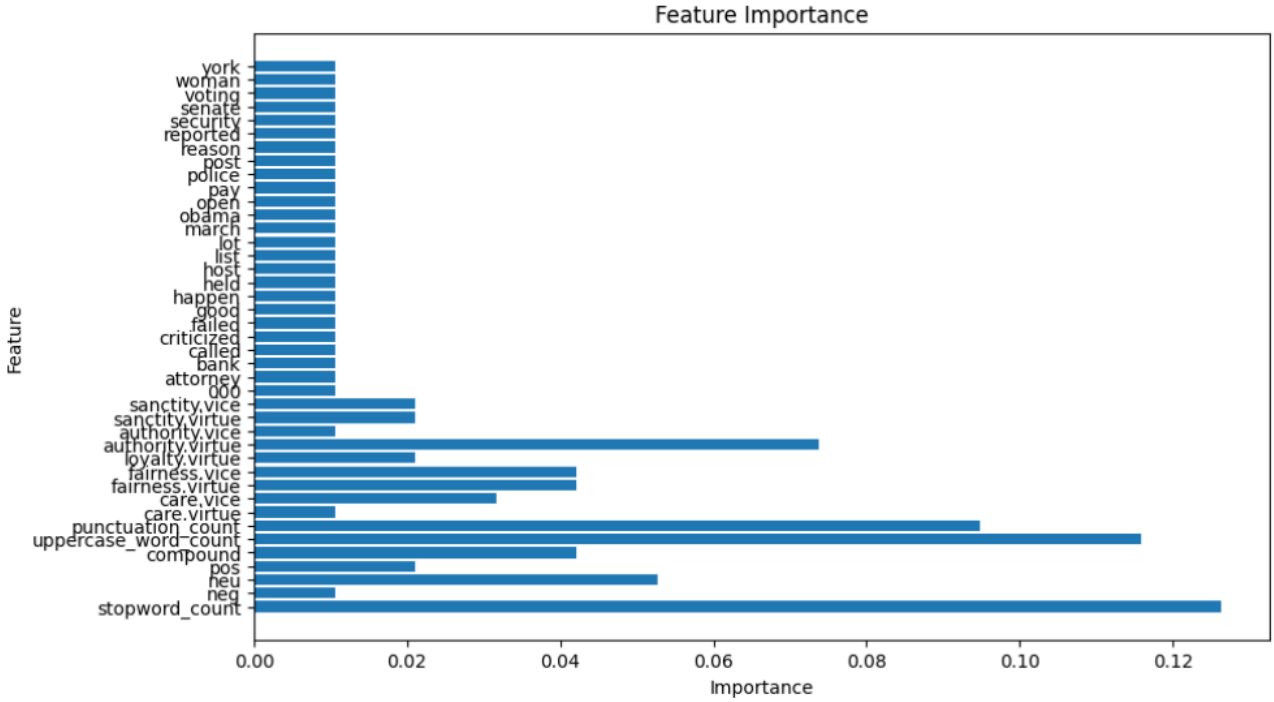
<u>מהתוצאות המוצגות בטבלה ניתן להסיק כי:</u>

1. accuracy: המודל סיווג 77% ידיעות נכונה מכלל הידיעות.
2. precision:
   - מתוך כלל הידיעות שסווגו ככוזבות, 75% מהן היו כוזבות.
   - מתוך כלל הידיעות שסווגו כאמת, 79% מהן היו אמת.
3. recall (שליפה):
   - מתוך כלל הידיעות הכוזבות, 84% מהן סווגו ככוזבות.
   - מתוך כלל ידיעות האמת, 70% מהן סווגו כאמת.
4. מדד f1-score :
   - למחלקה 0 יש F1-score של 79%, מה שמעיד על איזון טוב בין דיוק לשליפה.
   - למחלקה 1 יש F1-score של 74%, מה שמעיד על איזון סביר בין דיוק לשליפה.

5. ממוצעים: שני סוגי הממוצעים עומדים על 77%.

המסקנה הכללית: המודל מזהה טוב יותר את המחלקה 0 מאשר המחלקה 1. ניתן לראות זאת לפי מדד השליפה בו יש פער של 14% בין 2 המחלקות לטובת המחלקה 0. לכן נסיק כי בהינתן ידיעת אמת הסיכוי של המודל לחזות אותה נכונה נמוך יותר מאשר אם הוא היה מקבל ידיעת שקר. ניתן לטעון כי תכונה זו של המודל עדיפה על התכונה ההפוכה (במסגרתה המודל מסווג ידיעות שקר נכונה בהסתברות נמוכה מידיעות אמת) מכיוון שיש פחות חשש להטעיה של משתמשים בקריאת ידיעת שקר. אמנם, יש לנו אינטרס גם באימות ידיעות אמת שכן עליהן מתבססים משתמשי הרשת בקביעת דעתם בנושא מסוים.

תרשים 3: חשיבות הפיצ'ר המשמעותיים ביותר באחוזים לאחר הכיוונון



ראשית נשים לב כי מספר הפיצ'רים להם המודל נתן חשיבות הגבוהה מערך הסף 0.005 גדל לאחר הכיוונון בכ 30 פיצ'רים. כלומר, הכיוונון השפיע רבות על הערכת המודל את הפיצ'רים שנתנו לו בדאטה וכי כעת הוא חילק את החשיבות בין יותר פיצ'רים על חשבון פיצ'רים אותם העריך רבות לפני הכיוונון. ניתן לראות כי עדיין הפיצ׳רים stopword_count , authority.virtue ו- neu הם חשובים אך חשיבותם קטנה פי 2. חשיבות stopword_count מ 30% לבערך 13%, חשיבות authority.virtue מ 15% לבערך 8% וחשיבות neu מ 10% לבערך 5%. על חשבונו, עברו את הסף והתווספו לתרשים פיצ'רים מTF-IDF כמו:

woman, voting, senate, security, york, police, obama

עבור פיצ'רים שכבר היו קיימים בתרשים, חלקם לא שינו את חשיבותם כמו: fairness.vice, compound. כל שאר הפיצ'רים שהיו קיימים בתרשים לא שינו דרסטית את חשיבותם. כעת הפיצ'רים החשובים ביותר הם:

Stopword_count, uppercase_word_count, punctuation_count, authority.virtue, neu, compound, fairness.vice, fairness.virtue.

בדומה לפיצ'רים החשובים ביותר טרם הכיוונון.

מודל Logistic Regression

מודל זה מנבה הסתברות לכך שמשתנה מסוים שייך לכל אחת קטגוריות, במקרה שלנו האם הידיעה אמיתית או כוזבת, ע"י שימוש בפונקציה הנקראת Sigmoid. פונקציה זו ממפה את התוצאה לערכים בין 0 ל-1, המייצגים את ההסתברות לשייכות לכל אחת מהקטגוריות. כדי לקבוע את הסיווג של נקודה, בדר"כ נשתמש בסף של 0.5 ונבדוק: אם התוצאה שלפונקציית ה-Sigmoid גדולה מסף זה, נסווג אותו עם תווית 1, כלומר ידיעה אמיתית. אחרת, נסוגו עם תווית 0, כלומר ידיעת כזב.

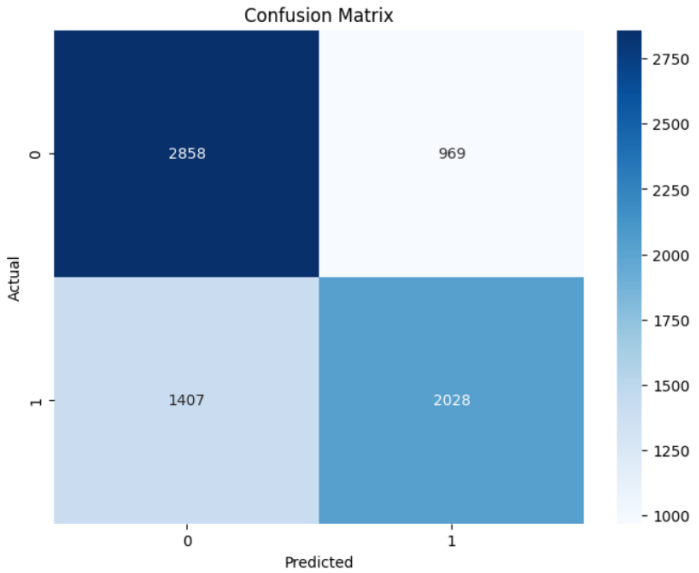
בהתחלה השתמשנו במודל זה עם C=1 וקיבלנו את התוצאות הבאות:

Logistic Regression Training Accuracy: 70.0%

Logistic Regression Validation Accuracy: 67.2%

עבור בחירה שרירותית של היפר פרמטר קיבלנו דיוק של 10% שיפור מעבר ל base line של Knn ונרצה לשפר זאת. גם כאן הצגנו את התוצאות בעזרת אותם תרשימים כמו במודל ה- adaboost.

תרשים 1: מטריצת בלבול על סט הוולידציה



תרשים זה מציג את תוצאות המודל על גבי ה- DATASET שלנו. ניתן לראות כי יש מספר גבוה של נקודות מבחן שהמודל טועה עליהן: מסווג 968 ידיעות כזב כידיעות אמת, ומסווג 1407 ידיעות אמת כידיעות כזב. בנוסף לראות כי המודל מצליח לזהות ולסווג נכונה יותר ידיעות כזב מאשר ידיעות אמת.

תרשים 2: מדדים

Logistic Regression Classification Report:

	precision	recall	f1-score	support
0	0.67	0.75	0.71	3827
1	0.68	0.59	0.63	3435
accuracy			0.67	7262
macro avg	0.67	0.67	0.67	7262
weighted avg	0.67	0.67	0.67	7262

מהתוצאות המוצגות בטבלה ניתן להסיק כי:

1. accuracy: המודל מסווג נכונה 67% מכלל הידיעות.
2. precision:
   - מתוך כלל הידיעות שסווגו ככוזבות, 67% מהן היו כוזבות.
   - מתוך כלל הידיעות שסווגו כאמת, 68% מהן היו אמת.
3. recall (שליפה):
   - מתוך כלל הידיעות הכוזבות, 75% מהן סווגו ככוזבות.
   - מתוך כלל ידיעות האמת, 59% מהן סווגו כאמת.
4. מדד f1-score :
   - למחלקה 0 יש F1-score של 71%.
   - למחלקה 1 יש F1-score של 63%.
5. ממוצעים: שני סוגי הממוצעים עומדים על 67%.

לפי ציוני f1 score ניתן לראות כי אין איזון טוב בין דיוק לשליפה. בהשוואה למודל adaboost לפני כיוונון הפרמטרים ניתן לשים לב להבדלים המשמעותיים באחוזים – מודל הרגרסיה הלוגיסטית נופל בכל אחד מהמדדים בהשוואה למודל adaboost. עובדה זו לא מפתיעה מאחר רגרסיה לוגיסטית היא בעצם סוג של מודל לינארי, אותו כבר בחנו ולא הגענו לתוצאות טובות.

ניסוי: כיוונון היפר הפרמטר C.

בניסוי השתמשנו בטווח הערכים:

range(0.1, 1.5, 0.01), range(1.5, 15.5, 0.5), range(15, 50, 5)

תוצאות הניסוי ומסקנות:

Best parameter found: {'C': 1.27}

Logistic Regression Training Accuracy: 70.1%

Logistic Regression Validation Accuracy: 67.5%

תוצאות הניסוי הן שיפור מינימלי עד כדי אפסי בדיוק המודל על סט הוולידציה. תוצאות אלו דומות לתוצאות של המודל הלינארי SVM כצפוי – כנאמר קודם לכן, רגרסיה לוגיסטית היא סוג של מודל לינארי. לכן תוצאות אלו לא צריכות להפתיע אותנו. המטרה בבחינת מודל זה הייתה ניסיון ביצירת וודאות בסיווג ידיעות. אך עם תוצאות כאלו לא נוכל להסתמך על מודל זה במודל הסופי ולכן לא ניקח בחשבון את הוודאות שלו בסיווג ידיעות.

LogisticRegression – קישור המפרט יותר על המודל.

מודל Random Forest-

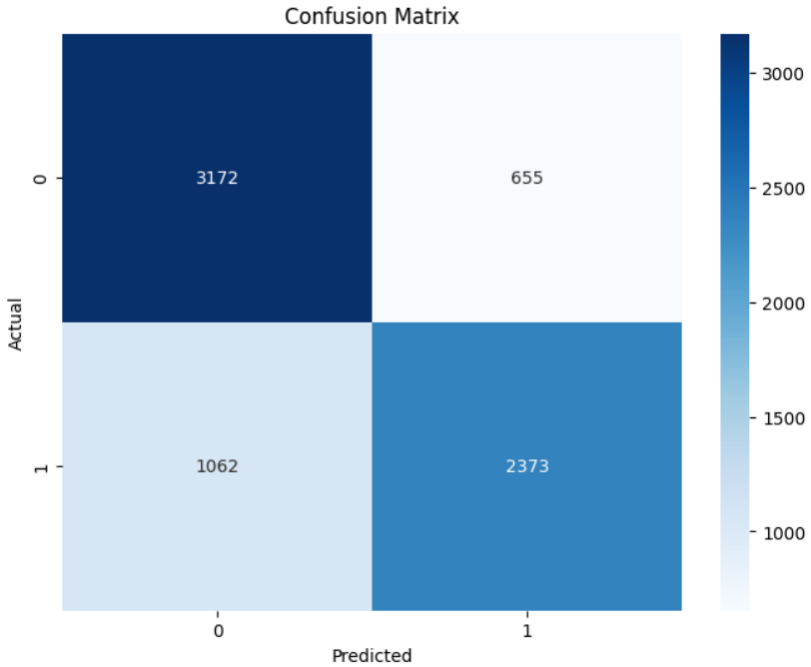
מודל זה הוא גם כן מודל אנסמבל, בדומה ל- ADABOOST, מודל זה מבוסס על שילוב של מספר עצי החלטה כדי לשפר את הדיוק של סיווג הדוגמאות ובפרט על מנת למנוע Overfitting. כל עץ החלטה ביער נבנה על סט אקראי של דוגמאות פיצ'רים מהנתונים, כך שכל עץ מתבסס על חלק שונה של דוגמאות האימון והפיצ'רים. הסיווג הסופי הוא שילוב של כל הסיווגים המתקבלים ע"י כל עץ החלטה, לרוב לפי ה majority במקרה של מודל קלסיפיקציה. בהתחלה השתמשנו במודל זה עם מספר עצים חלשים: n_estimators, להיות 100. קיבלנו כי:

Random Forest Training Accuracy: 100.0%

Random Forest Validation Accuracy: 76.3%

דיוק כזה עבור סט האימון וסט הוולידציה יכול להעיד על overfitting. עם זאת עדיין ישנו שיפור משמעותי של כ20% מעל הbaseline של Knn. ננסה לשפר זאת ע"י כיוונון. גם כאן הצגנו את התוצאות בעזרת אותם תרשימים כמו במודלים הקודמים.

תרשים 1: מטריצת בלבול על סט הוולידציה



ממטריצת הבלבול הנ"ל ניתן להסיק כי המודל אכן מצליח לסווג נכונה את רוב המוחלט של נקודות המבחן, אך טועה בכ-1717 נקודות סה"כ מתוך 7262 נקודות מבחן. מסווג 655 ידיעות כזב כידיעות אמת, ומסווג 1062 ידיעות אמת כידיעות כזב. ניתן להסיק כי המודל מדייק יותר בזיהוי ידיעות כזב אך מתקשה בזיהוי ידיעות אמת. זאת מחזק התרשים הבא:

תרשים 2: מדדים

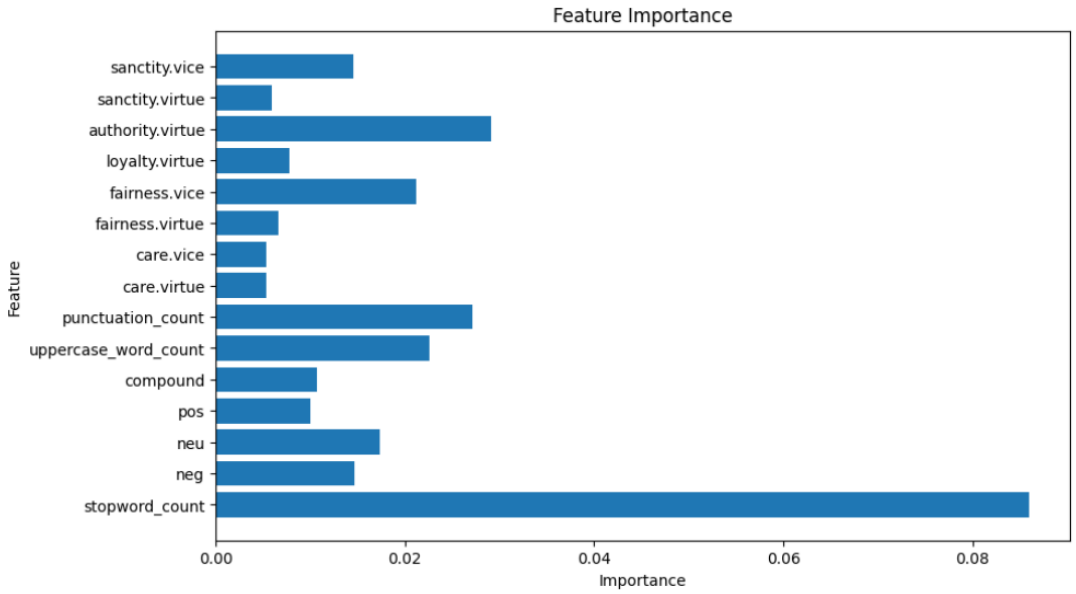
```
Random Forest Classification Report:
              precision    recall  f1-score   support

           0       0.75      0.83      0.79      3827
           1       0.78      0.69      0.73      3435

    accuracy                           0.76      7262
   macro avg       0.77      0.76      0.76      7262
weighted avg       0.77      0.76      0.76      7262
```

<u>מהתוצאות המוצגות בטבלה ניתן להסיק כי:</u>

1. accuracy: המודל מסווג נכונה 76% מכלל הידיעות.
2. precision:
   - מתוך כלל הידיעות שסווגו ככוזבות, 75% מהן היו כוזבות.
   - מתוך כלל הידיעות שסווגו כאמת, 78% מהן היו אמת.
3. recall (שליפה):
   - מתוך כלל הידיעות הכוזבות, 83% מהן סווגו ככוזבות.
   - מתוך כלל ידיעות האמת, 69% מהן סווגו כאמת.
4. מדד f1-score:
   - למחלקה 0 יש F1-score של 79%, מה שמעיד על איזון טוב בין דיוק לשליפה.
   - למחלקה 1 יש F1-score של 73%, מה שמעיד על איזון סביר בין דיוק לשליפה.
5. ממוצעים: שני סוגי הממוצעים עומדים על 76%- 77%.

אלו תוצאות הדומות יותר לתוצאות של מודל adaboost – שניהם הביאו תוצאות הכי טובות עד כה. לכן נתחשב בסיווגו במודל הסופי. ניתן לראות כי בהשוואה לתוצאות של adaboost וביחס לתוצאות של המדדים בין אותן מחלקות, הוא סובל מאותן חולשות ויש לו את אותן החוזקות כפי שתיארנו שם.

תרשים 3: חשיבות הפיצ'ר המשמעותיים ביותר באחוזים



מתרשים זה ניתן להסיק כי הפיצ'ר המשמעותי במיוחד במודל זה הוא stopword_count, עם חשיבות שגבוהה מ- 8%. בנוסף עוד שני פיצ'רים שמשמעותיים הינם punctuation_count ו- authority.virtue. פיצ'רים אלו חזרו גם במודלים קודמים, דבר שמעניין לראות ונתייחס לכך בחלק של feature selection. בנוסף מעניין לראות שאף פיצ'ר לא קיבל חשיבות גבוהה מ8% שלא כמו בadaboost שם קיימים פיצ'רים שהגיעו ל15-30%. ניתן לשער כי כלל הפיצ'רים שלא מופיעים בטבלה מוערכים בחשיבותם באופן די דומה, שכן מרבית אחוזי החשיבות לא מופיעים בתרשים מאחר וכל פיצ'ר בתרשים לא קיבל יותר מ8%.

ניסוי: כיוונון היפר הפרמטר n_estimators.

בניסוי השתמשנו בטווח הערכים 3-200.

תוצאות ומסקנות מהניסוי:

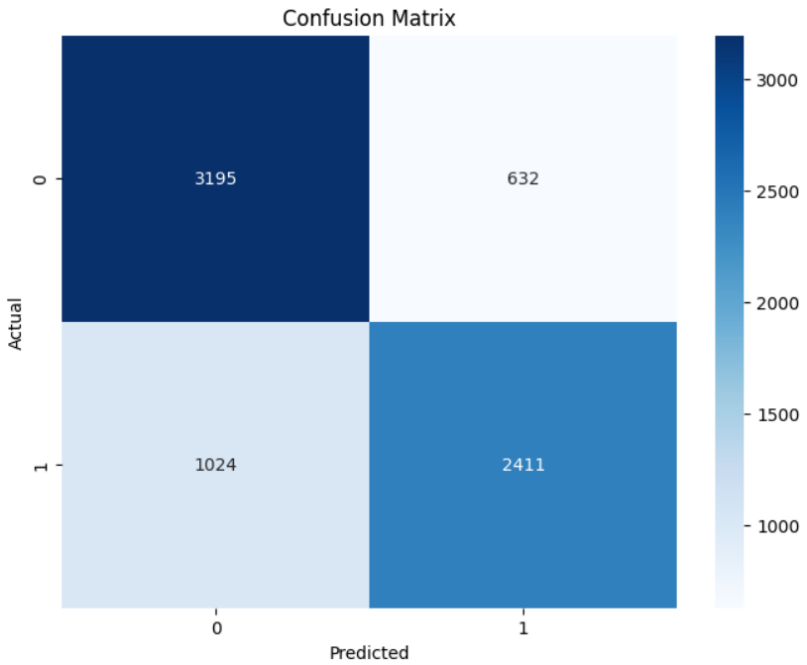
Best parameter found: {'n_estimators': 183}

Random Forest Training Accuracy: 100.0%

Random Forest Validation Accuracy: 77.1%

עדיין ישנה התאמה מלאה לסט האימון ושיפרנו את הדיוק על סט הוולידציה ב1%. ניתן לנסות את הניסוי שוב עם טווח ערכים מורחב על סמך גרפים וזיהוי מגמות עלייה על מנת לשפר את ביצועי המודל על סט הוולידציה.

תרשים 1: מטריצת בלבול על סט הוולידציה



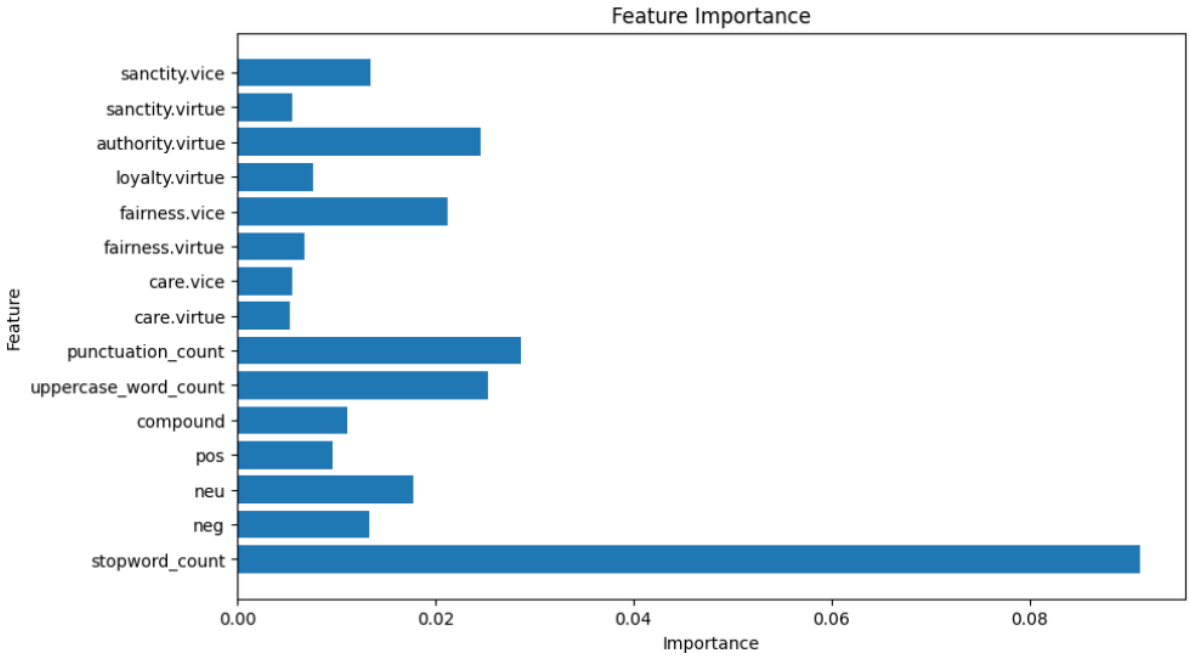
לאחר הכיוונון קיבלנו תוצאות טובות יותר בזיהוי 2 סוגי הידיעות, אך עדיין לא מספק. מסווג 637 ידיעות כזב כידיעות אמת לעומת 655 לפני כיוונון, ו 1011 ידיעות אמת כידיעות כזב לעומת 1062 לפני כיוונון. שוב נציין את הדמיון בתוצאות לאלו של adaboost. שוב, השיפור הינו מינורי לכן נסיק כי שיפור ביצועי המודלים צריך לבוא ממישורים אחרים ולווא דווקא מכיוונון יותר מוצלח של היפר פרמטרים.

תרשים 2: מדדים

```
Random Forest Classification Report:
              precision    recall  f1-score   support

           0       0.76      0.83      0.79      3827
           1       0.79      0.70      0.74      3435

    accuracy                           0.77      7262
   macro avg       0.77      0.77      0.77      7262
weighted avg       0.77      0.77      0.77      7262
```

לסיכום ניתן לראות כי הדיוק הכללי עלה מ- 76% ל- 77%. בנוסף, שאר המדדים גם כן השתפרו אך באחוזים מעטים: מדד ה- precision עלה בכ- 1% עבור כל אחת מהמחלקות. מדדי ה- recall וה- f1 score עלו ב1% עבור מחלקה 1 בלבד. נסיק כי שיפור הביצועים של המודל שופרו במועט עד כדי לא הורגשו כלל.

תרשים 3: חשיבות הפיצ'ר המשמעותיים ביותר באחוזים



גם כעת לאחר הכיוונון הפיצ'רים המשמעותיים והבולטים ביותר הינם: stopword_count, עם חשיבות שגבוהה מ- 8%. בנוסף לכך גם punctuation_count, uppercase_word_count ו- authority.virtue. הכיוונון אינו שינה את הפיצ'רים המשמעותיים עליהם המודל מסתמך, וגם לא את יחסי החשיבות בין הפיצ'רים. ניתן להסביר זאת בחוסר השינוי בביצועי המודל כלומר סה"כ המודל השתנה מינורית בביצועיו ובאופיו בעזרת הכיוונון.

מודל XGBOOST

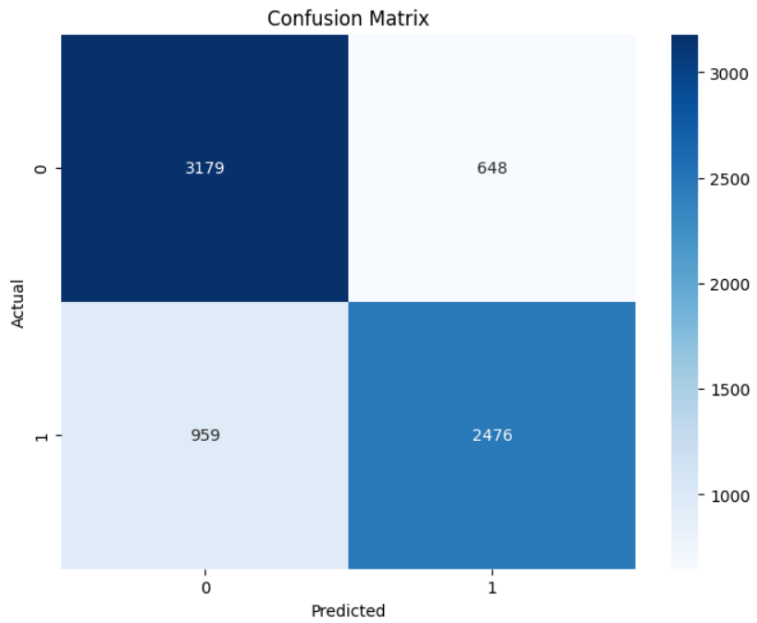
מודל XGBoost, הנקרא Extreme Gradient Boosting, הוא גם כן מודל מסוג אנסמבל. הוא משתמש ב-Boosting לבניית עצי החלטה בצורה סדרתית, כאשר כל מודל מתמקד בתיקון טעויות הסיווג של נקודות האימון עליהן טעה העץ הקודם בסדרה. היתרון המרכזי של XGBoost הוא ביצועיו המהירים והדיוק הגבוה שהוא מצליח לספק. זאת ע"י שימוש באופטימיזציות מתקדמות, הכוללות לדוגמה שימוש בגרדיאנטים למזעור שגיאות, מה שמבטיח שכל עץ חדש מתמקד בתיקון השגיאות של העצים הקודמים. בנוסף, הוא מבצע רגורליזציה על מנת למנוע Overfitting ומבצע חישוב מקבילי ומבוזר שגורם לביצועיו להיות מהירים. אימנו עם מודל זה וקיבלנו את התוצאות הבאות:

XGBoost Training Accuracy: 90.8%

XGBoost Validation Accuracy: 77.8%

קיבלנו תוצאות דומות למודלי ה adaboost ו random forest. עם זאת הוא פחות מתאים את עצמו לסט האימון כמו מודל ה random forest. הצלחתו טובה בכ 20% מה base line של Knn.

תרשים 1: מטריצת בלבול על סט הוולידציה



ניתן לראות כי אכן את הרוב המוחלט נקודות המבחן הוא מסווג נכונה, אך יש מספר גבוה של נקודות מבחן שהמודל טועה עליהן: מסווג 648 ידיעות כזב כידיעות אמת, ומסווג 959 ידיעות אמת כידיעות כזב. אכן המודל מתפקד בצורה טובה יותר משאר המודלים שבחנו אך גם כן מתקשה בזיהוי ידיעות אמת יותר מאשר זיהוי ידיעות כזב. זאת ניתן לראות טוב בתרשים הבא:

תרשים 2: מדדים

```
XGBoost Classification Report:
              precision    recall  f1-score   support

           0       0.77      0.83      0.80      3827
           1       0.79      0.72      0.75      3435

    accuracy                           0.78      7262
   macro avg       0.78      0.78      0.78      7262
weighted avg       0.78      0.78      0.78      7262
```

<u>מהתוצאות המוצגות בטבלה ניתן להסיק כי:</u>

1. דיוק המודל (accuracy): הדיוק הכללי של המודל הוא 78%
2. precision:
   - מתוך כלל הידיעות שסווגו ככוזבות, 77% מהן היו כוזבות.
   - מתוך כלל הידיעות שסווגו כאמת, 79% מהן היו אמת.
3. recall (שליפה):
   - מתוך כלל הידיעות הכוזבות, 83% מהן סווגו ככוזבות.
   - מתוך כלל ידיעות האמת, 72% מהן סווגו כאמת.
4. מדד f1-score:
   - למחלקה 0 יש F1-score של 80%, מה שמעיד על איזון טוב יותר בין דיוק לשליפה.
   - למחלקה 1 יש F1-score של 75%, מה שמעיד על איזון טוב בין דיוק לשליפה.
5. ממוצעים: שני סוגי הממוצעים עומדים על 78%.

המסקנה הכללית: היא כי המודל משיג ביצועים טובים בשתי המחלקות ועם איזון טוב בין דיוק לשליפה עבור מחלקה 0 המייצגת ידיעות כזב, אך עם ביצועים קצת פחות טובים עבור מחלקה 1 המייצגת ידיעות אמת כך ראינו ברוב המודלים החזקים עד כה.

שיטת : Hyper Parameters Tuning

לצורך הכיוונון (tuning) של היפר הפרמטרים השתמשנו בתחילה בשיטת GridSearchCV.
במסגרת שיטה זו בוחרים מראש רשימת ערכים אפשריים לכל היפר פרמטר, ועוברים על כל הקומבינציות האפשריות מתוך הרשימות לסט הפרמטרים שרוצים לכוונן. לכל קומבינציה, מחלקים את הדאטה ל n חלקים (n - פקטור חלוקה נקבע מראש כפרמטר לפונקציה), מאמנים את המודל הרצוי על n-1 חלקים מן הדאטה ועל החלק הנותר מבצעים ולידציה. כך עושים במשך n איטרציות כך שבכל איטרציה חלק אחר מן הדאטה מהווה את סט הוולידציה. שיטה זו ממצה את כל האפשרויות לקומבינציות של היפר פרמטרים ואם מהיכרות עם המודל והדאטה ניתן רשימות בהן נמצאים ערכים הממקסמים את הדיוק על סט הוולידציה, אכן נמצא ערכים אופטימליים. עם זאת, שמנו לב כי שיטה זו לא יעילה ולוקחת זמן רב מידי. לאחר חיפוש באינטרנט מצאנו שיטה יעילה יותר לכוונון היפר פרמטרים.

שיטת HalvingGridSearchCV מתחילה עם כל הקומבינציות אבל משערכת על פני חלק קטן מהדאטה.
לאחר שסיימה לשערך על פני כל הקומבינציות, לוקחת את חצי (או חלק יחסי אחר שמגדירים לה בפונקציה) הקומבינציות שנתנו שקלול טוב ביותר ומשערכת אותן על פני יותר דגימות מהדאטה וכן הלאה.
התהליך מסתיים כאשר נותרת קומבינציה אחת הנחשבת הכי טובה מבין כולן.

במסגרת ההשוואה הזו שנעשתה בין GridSearchCV לבין HalvingGridSearchCV בוצעה הדגמה שבה ניתן לראות כי HalvingGridSearchCV יעיל בערך פי 4 מ GridSearchCV למרות זאת מצליח להגיע לפרמטרים בעלי דיוק דומה על סט הוולידציה כמו שמוצא GridSearchCV. (פקטור חלוקה: 2).

שלב 5: Feature Selection

בחנו כמה שיטות בחירת פיצ'רים למודל הסופי. עבור כל קבוצת פיצ'רים שנבחרה מכל שיטה, אימנו על סט האימון ובחנו על סט הוולידציה מספר מודלים שמהניסיונות הקודמים היו הכי טובים: Random Forests, XGBoost, AdaBoost, Logistic Regression. השתמשנו במודלים אלו עם ההיפר פרמטרים שכיוונו בניסויים הקודמים. אותם היפר פרמטרים כוונו בהתבסס על סט האימון ולא על סט הוולידציה.

כמו בוויזואליזציה הראשונית החלטנו לעשות מפת חום של מטריצת קורלציה בין הפיצ'רים של קבוצת האימון לבין הסיווג של קבוצת האימון, זאת על מנת לזהות אילו פיצ'רים חשובים במשימת החיזוי שלנו ומשפיע על הסיווג שהמודל נותן. נוכל לזהות זאת ע"י פיצ'רים עם מתאם בעל ערך מוחלט גבוה וקרוב ל-1. בנוסף כך גם נוכל לזהות אילו פיצ'רים אינם תורמים למודל בסיווג הנקודות.

ממטריצת הקורלציה הראשונית שעשינו ניתן לראות כי הפיצ'ר בעל הקורלציה הגבוהה ביותר ל-label הוא stopword_count, דבר שאיננו מפתיע אותנו כי אכן ראינו שרוב המודלים משתמשים בו והוא נחשב כמשמעותי ביותר ברובם.

```
print(most_correlated_to_label)
```

```
['stopword_count', 'fairness.vice', 'neg', 'sanctity.vice', 'pos', 'punctuation_count', 's
anctity.virtue', 'uppercase_word_count', 'care.virtue', 'authority.vice', 'problem', 'car
e.vice', 'happen', 'reason', 'loyalty.vice', 'nothing', 'small', 'western', 'going', 'mar
k', 'side', 'key', 'possible', 'video', 'good', 'large', 'country', 'york', 'continue', 's
peak', 'popular', 'law', 'middle', 'hope', 'end', 'dollar', 'russia', 'best', 'taken', 'ju
ne', 'monday', 'left', 'le', 'agenda', 'center', 'population', 'united', 'email', '2014',
'11', 'violence', 'late', 'become', 'russian', 'muslim', 'iraq', 'vice', 'states', 'migh
t', '12', 'bad', 'serious', 'deal', 'soon']
```

אלו הם כלל הפיצ'רים מודפסים לפי הקורלציה שלהם לתווית, כלומר ל- label, מגבוה לנמוך. מעניין לראות כאן פיצ'רים שחוזרים על עצמם בכל המודלים כמו: stopword_count, fairness.vice, neg, punctuation_count וכולי. זה נותן לנו אינדיקציה שפיצ'רים שהמודלים בחרו הם אינדיקטיביים לתיוג. ננסה לבחור אותם ולראות מה קורה.

1. **WordCloud**

לקחנו את ענני המילים מהחלק שבו עשינו ויזואליזציה לדאטה והפקנו מהם פיצ'רים מתוך מחשבה שמילים שחוזרות על עצמן בתדירות גבוהה בידיעות אמת / שקר יחזרו על עצמן בידיעות המבחן המתאימות לאותו תיוג. מכל ענן מילים (יש 2, אחד לכל תיוג) לקחנו את 20 המילים התדירות ביותר בידיעות, כלומר שהופיעו לפחות פעם אחת בהכי הרבה ידיעות. לכל אחת מבין 40 המילים הללו חישבנו פיצ'ר מתאים המתאר את תדירות המילה בידיעה מסוימת. **הערה**: ה- wordcloud חושבו רק על ה- training set.

המילים שהתקבלו:

```
WordCloud features:  ['deluded', 'Jailed', 'hate', 'BEIJING', 'going', 'Steel', 'design',
'division', 'Catalonia', 'Exposed', 'stack', 'DENVER', 'Obama', 'China', 'German', 'John',
'st', 'WASHINGTON', 'nicknamed', 'Foreign', 'leader', 'anyone', 'Tactics', 'patient', 'Min
ister', 'S', 'cancer', 'Reuters', 'acceptable', 'g', 'legacy', 'Trump', 'Establishm', 'Don
ald', 'use', 'BERLIN', 'Oliver', 'MADRID', 'Bull', 'Smear']
```

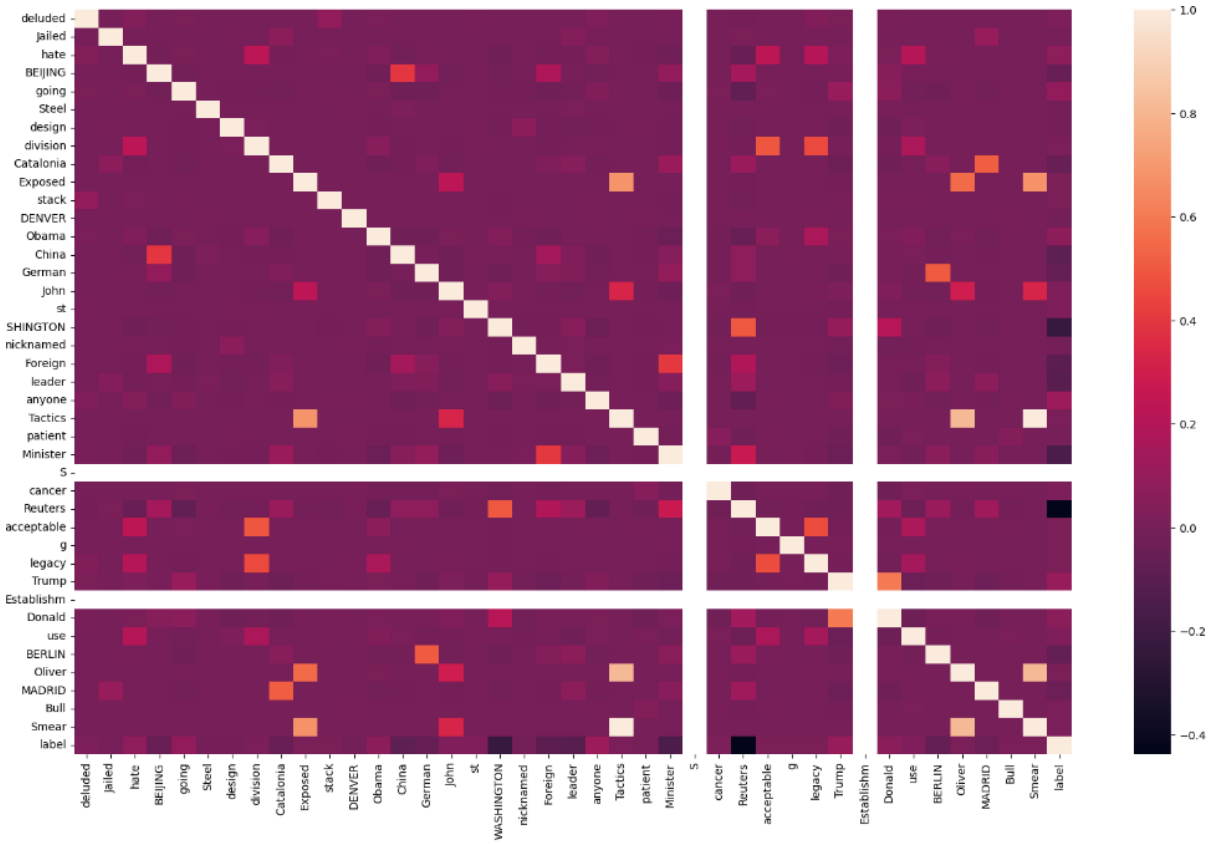
קטגוריית "ידיעות אמת":

```
True category:  ['Obama', 'anyone', 'deluded', 'design', 'Donald', 'Trump', 'going', 'stac
k', 'legacy', 'hate', 'division', 'acceptable', 'use', 'st', 'John', 'Oliver', 'Smear', 'T
actics', 'Exposed', 'Establishm']
```

קטגוריית "ידיעות שקר":

```
Fake category:  ['Reuters', 'S', 'BEIJING', 'WASHINGTON', 'MADRID', 'Jailed', 'Catalonia',
'leader', 'China', 'Foreign', 'Minister', 'DENVER', 'cancer', 'patient', 'nicknamed', 'Ste
el', 'Bull', 'g', 'BERLIN', 'German']
```

מעניין לראות את סוג הפיצ'רים שהתקבלו- שמות של פוליטיקאים כמו דונלד טראמפ ואובמה, שמות של מקומות כמו וושינגטון, סין, מדריד, דנבר וברלין, מילות שמתארות שנאה ופילוג. זה לא מפתיע מאחר ובעולם הפוליטיקה יש הרבה פייק ניוז על מנת לשרת אינטרסים של קבוצות פוליטיות. בנוסף שמות של מקומות מעידים על מה מתרחש בהם לדוגמה: וושינגטון – הבית הלבן נמצא שם, סין – מעצמה שבינה לבין ארה"ב המון התרחשויות כעת. מילות שליליות לעתים יגרמו לקורא לתחושות הזדהות וכעס ולכן נמצא אותם בידיעות המנסות לפנות ללב של הקורא. בנוסף בידיעות השקר יש גם מילים הכתובות באותיות גדולות כפי שחוזר על עצמו לאורך הדוח שלנו. מטריצת הקורלציה:



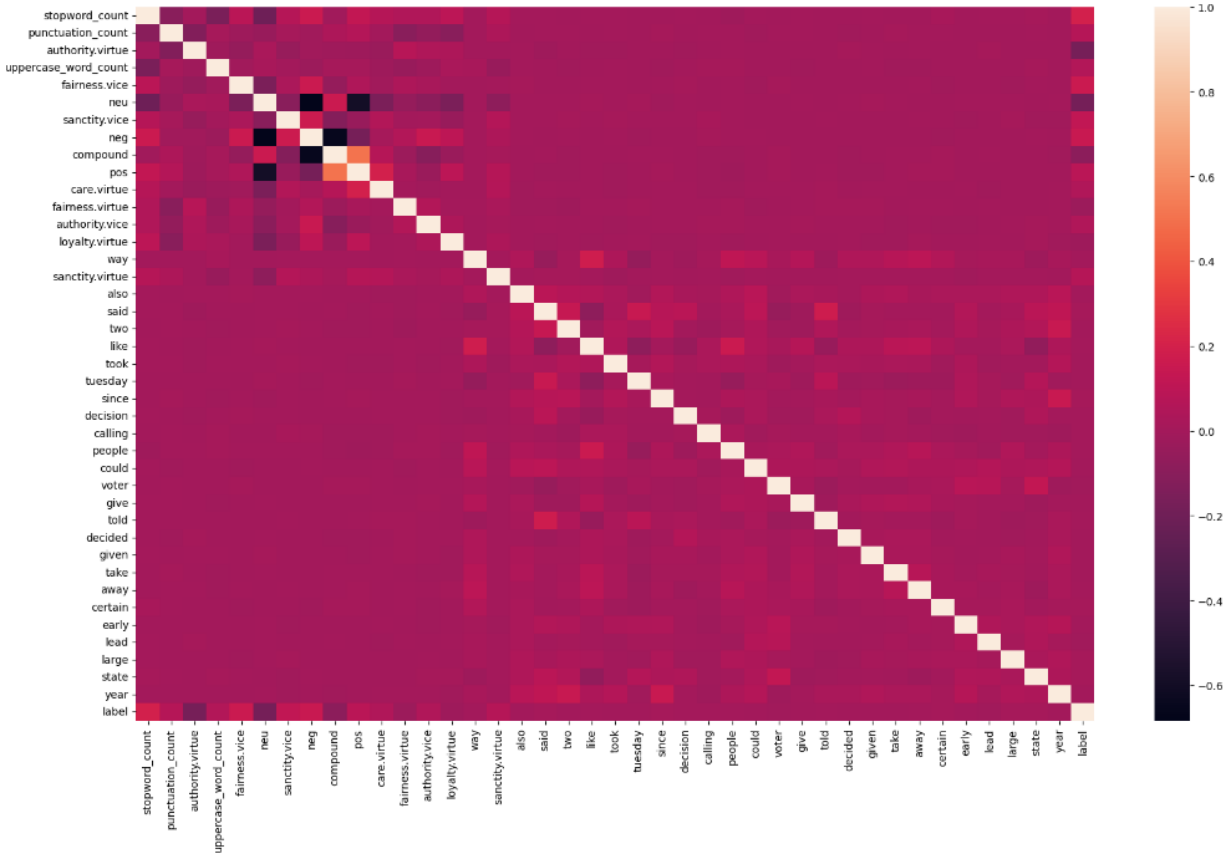
זוהי מטריצת הקורלציה בין כלל המילים שהתקבלו כמעני המילים ונשמרו כפיצ'רים על ידינו. ניתן לראות את הקורלציות בין פיצ'רים ובין פיצ'ר לתיוג. נבחין בכך שקיימים מספר תאים בעלי צבע כהה כלומר קורלציה שלילית גדולה בערך מוחלט (לפי המקרא כ- 0.4-). למשל, בין הפיצ'ר Reuters לתיוג ובין הפיצ'ר WASHINGTON לתיוג ישנה קורלציה כזו. משמעות הדבר היא, שערכי פיצ'רים אלו קשורים לתיוג בקשר הפוך די הדוק (אך לא מוחלט) ולכן יכולים לעזור לנו בסיווג. בנוסף, כמעט כל שאר הפיצ'רים קשורים לתיוג בערכים חיוביים בין 0 ל0.2 ולכן גם יכולים לעזור בתיוג. בנוסף, רוב התאים במטריצת הקורלציה שאינם בעמודת התיוג, אינם בהירים מידי כלומר מראה על אינדיקציה לכך שהפיצ'רים אינם קורלטיביים אחד לשני. התאים הבהירים בטבלה מעידים על קורלציה גבוהה בין 2 פיצ'רים כך שניתן לחשוב על לוותר על אחד מהם, שכן השני "מכיל את המידע שלו". למשל, ניתן לראות כי הקורלציה בין הפיצ'ר Reuters לפיצ'ר WASHINGTON יחסית גבוהה (צבע בהיר) וזה לא מפתיע מאחר ושניהם קשורים לתיוג כמעט באותו פקטור. עוד דוגמה היא שיש קורלציה גבוהה בין פיצ'ר Oliver לבין Tactics. אפשר אולי לפרש את העובדה הזו בכך שאדם בשם אוליבר מצוין בכתבות יחד עם המילה טקטיקה לעתים קרובות, כלומר אוליבר מרבה לעסוק בטקטיקה וכולי.

2. **ID3**

לקחנו את עץ ההחלטה שקיבלנו קודם ורצינו לראות אילו פיצ'רים מקבלים חשיבות גבוהה יותר ולבחור אותם. לשם כך השתמשנו בתכונת _feature_importances המתארת לכל פיצ'ר את חשיבותו לחיזוי המודל. מיינו את הרשימה הזו וקיבלנו בראש הרשימה את הפיצ'רים החשובים ביותר למודל אותם נרצה לבחור. חשיבות כל פיצ'ר נקבעת ע"י חישוב ההפחתה ב- impurity שאותו פיצ'ר מצליח להשיג. פיצ'רים שהצליחו להפריד את הדאטה בכל פיצול כך שההפחתה ב- impurity הייתה הגבוהה ביותר, יימצאו בראש הרשימה ויקבלו חשיבות גבוהה. קיבלנו את הרשימה: (רק 20 פיצ'רים חשובים ביותר מתוכה)

```
Top 40 features by importance chosen by the decision tree:
stopword_count: 0.4486
punctuation_count: 0.1389
authority.virtue: 0.0878
uppercase_word_count: 0.0726
fairness.vice: 0.0472
neu: 0.0325
sanctity.vice: 0.0325
neg: 0.0302
compound: 0.0175
pos: 0.0122
care.virtue: 0.0047
fairness.virtue: 0.0035
authority.vice: 0.0030
loyalty.virtue: 0.0029
way: 0.0026
sanctity.virtue: 0.0026
also: 0.0022
said: 0.0022
two: 0.0022
like: 0.0020
```

בנוסף, נשים לב כי יש קשר בין הופעת אותו פיצ'ר ברמה גבוהה יותר במעלה העץ, לבין חשיבותו. כך למשל 4 הרמות הראשונות (מתוך 10) של העץ האופטימלי מכילות בצמתים רק את חמשת הפיצ'רים המסומנים בתמונה. זו אבחנה שמחזקת את הסיבות שלנו להשתמש במדד חשיבות הפיצ'רים. מטריצת הקורלציה:



גם כאן ניתן אכן לבסס את בחירת המודל ולראות כי הפיצ'רים בעלי קורלציה גבוהה ביותר עם התווית הינם stopword_count, punctuation_count, authority.virtue, uppercase_word_count ו fairness.vice. לstopword_count קורלציה של כ- 0.3 לתיוג ולכן כנראה הרבה מודלים ביניהם עץ ההחלטה בחרו בו. כך גם עם הפיצ'רים האחרים שמצוינים לעיל, ניתן לראות את הקשר בין בחירת המודל לבין בהירות התא בחלק מהתאים בטבלה. כלומר, חלק לא קטן מהפיצ'רים שבחר העץ בעלי קורלציה חיובית לא נמוכה לתיוג.

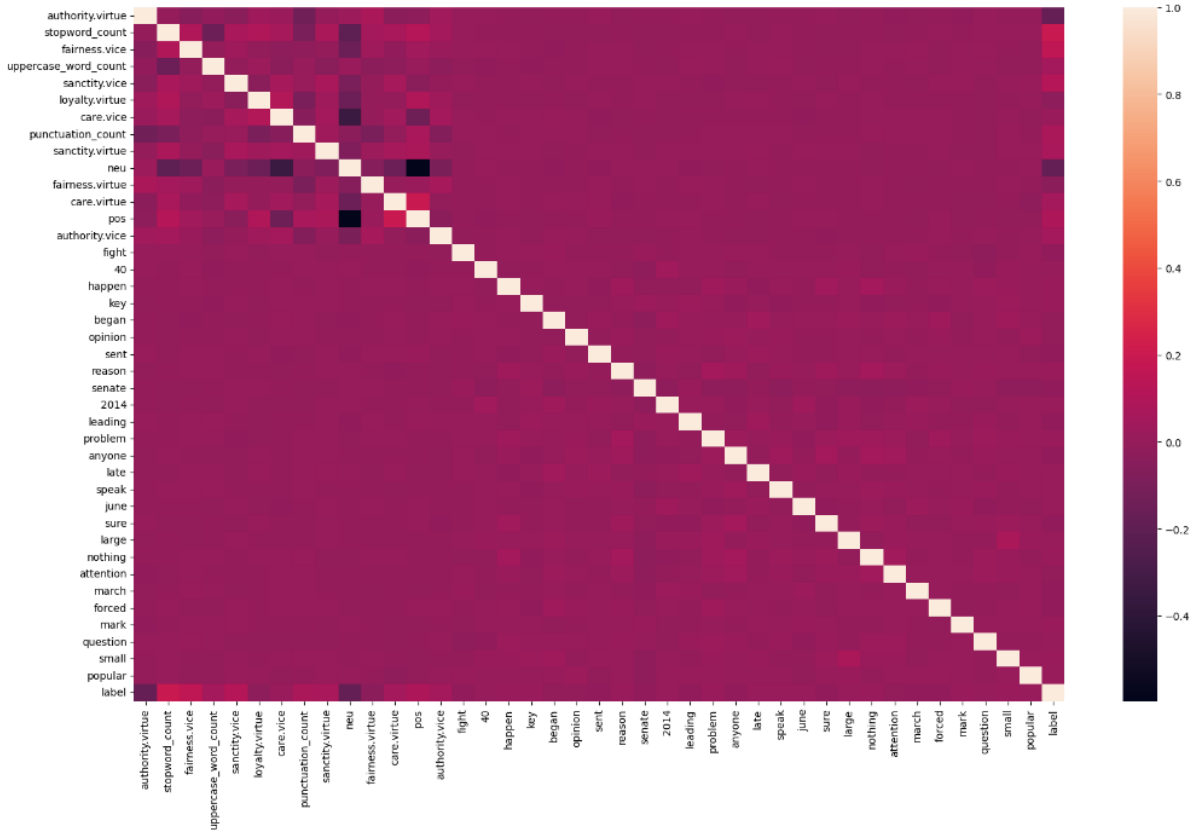
בנוסף ישנם תאים בודדים של זוגות פיצ'רים הקשורים זה לזה בקשר הפוך בעלי קורלציה שלילית של כ- 0.5- למשל neg בעל קורלציה כזו ל compound ול neu ו pos בעל קורלציה כזו ל neu. ניתן להסיק כי אולי ניתן לוותר על אחד מהם כיוון שהם מעידים אחד על המידע של השני ולכן המודלים יוכלו להסתדר בלעדיהם.

3. SVM עם קרנל לינארי

במודל זה בחרנו להסתכל על המשקלים שנתן המודל לכל פיצ'ר ולקחת את 40 הפיצ'רים בעלי המשקל בערך מוחלט הגבוה ביותר. המחשבה מאחורי זה הייתה, שככל שהמשקל בערך מוחלט שנותן המודל לפיצ'ר מסוים גבוה יותר, כך הוא בעל יותר השפעה על המישור הרב ממדי המתיימר להפריד את הדאטה. בחירת הפיצ'רים שהתקבלה:

```
SVM with Linear Kernel choices:  ['authority.virtue', 'stopword_count', 'fairness.vice',
'uppercase_word_count', 'sanctity.vice', 'loyalty.virtue', 'care.vice', 'punctuation_coun
t', 'sanctity.virtue', 'neu', 'fairness.virtue', 'care.virtue', 'pos', 'authority.vice',
'fight', '40', 'happen', 'key', 'began', 'opinion', 'sent', 'reason', 'senate', '2014', 'l
eading', 'problem', 'anyone', 'late', 'speak', 'june', 'sure', 'large', 'nothing', 'attent
ion', 'march', 'forced', 'mark', 'question', 'small', 'popular']
```

נשים לב שרוב הפיצ'רים המצויים בראש הרשימה נבחרו ממילון MFD. מה שמחזק ותומך בשימוש בו ליצירת פיצ'רים העוזרים למודל לקבל החלטות. ניתן להבחין בעקביות בבחירת הפיצ'רים יחסית למודלים אחרים. כמו מודלים אחרים, גם בSVM נבחרו הפיצ'רים stopword_count, authority.virtue, fairness.vice, uppercase_word_count, punctuation_count. כל בחירה של מודל המחזקת בחירה של מודל אחר, מעניינת וחשובה לבחירת הפיצ'רים הסופית. החלק השני של הרשימה, הם פיצ'רים שנוצרו מ- TFIDF. מטריצת הקורלציה:



ניתן לראות כי מטריצת הקורלציה כהה יותר מהקודמות שהצגנו וגם פחות מגוונת. הקורלציה בין רוב זוגות הפיצ'רים זהה ונמצאת סביב 0 כך שלא נוכל לוותר על אף פיצ'ר ביחס לאחר. התאים הכהים מאוד זהים לאלו הפירטנו עליהם במטריצת הקורלציה של עץ ההחלטה מכיוון שאין קשר בין קורלציה בין 2 פיצ'רים לפרמטרים ולהחלטת המודל. זה מדד נפרד ולכן מניב תוצאות זהות בין הצגות שונות. הרעיון בלהציג לכל שיטה את מטריצת הקורלציה הוא לראות את שילוב הקורלציות בזוגות הפיצ'רים הנבחרים ע"י אותו מודל, אבל הערכים

עצמם כאמור לא תלויים במודל אלא ידועים מראש. זה סדר ההצגה שמעניין אותנו. משמעות הדבר היא שהקורלציה בין רוב הפיצ'רים לתיוג נמוכה (ע"י הצבע שמבטא סביבה קטנה של 0) ופיצ'רים בודדים קורלטיביים יותר לתיוג. את הפיצ'רים סידרנו בצירים לפי המשקל שלהם בערך מוחלט שקיבלו מהמודל ואכן ניתן לראות מגמה כללית של בהירות הולכת ועולה כאשר הפיצ'ר קיבל משקל יותר גבוה, ובהירות הולכת וקטנה כאשר הפיצ'ר קיבל משקל יותר נמוך. כלומר, הבהירות (קורלציה) מתרחקת מהאפס ביחס ישר לגודל המשקל של הפיצ'ר והסימן שלו. לדוגמה ניתן להבחין כי הפיצ'ר שקיבל את המשקל הכי גבוה בערך מוחלט- authority.virtue קיבל צבע כהה בתא המתאים לתיוג המעיד על קורלציה של כ- 0.2-. זה גם מעיד על כך שהוא יכול להוות פיצ'ר טוב למודל נשים לב שהסבר לקורלציה הגבוהה בערך מוחלט אך שלילית יכולה להגיע מכך שקיבל משקל שלילי אך גבוה מאוד בערכו המוחלט: כ 15-. לעומתו, הפיצ'ר השני שקיבל את המשקל הכי גבוה בערך מוחלט ממודל SVM הוא stopword_count. פיצ'ר זה קיבל מהמודל משקל של כ-11 ואכן התא שמתאים לו ולתיוג מאוד בהיר.

4. Logistic Regression עם רגולריזציית LASSO (L1)

מהיכרותנו עם רגרסיה לינארית ע"י שימוש בנורמת L1 בפונקציית ה Loss נוכל לבצע בחירת פיצ'רים. שימוש בנורמה זו מקנה לנו את התכונה שלאורך אימון המודל כניסות רבות בווקטור המשקולות w מתאפסות ובכך השפעת הפיצ'רים המתאימים לכניסות אלו על המודל זניחה, ועל כן אינם מסייעים בפתרון בעיית החיזוי שלנו. אותם פיצ'רים נוכל להוציא מקבוצת כלל הפיצ'רים במודל הסופי. קיבלנו את הפיצ'רים הבאים: הפיצ'רים שנבחרו כמשמעותיים במודל זה חושבו בעזרת אלגוריתם SelectFromModel. בנוסף נציין כי היינו צריכים להשתמש ב solver = 'saga' מכיוון שהוא תומך בנורמת L1 ויודע להתמודד עם דאטה סטים גדולים לפי התיעוד של LogisticRegression:

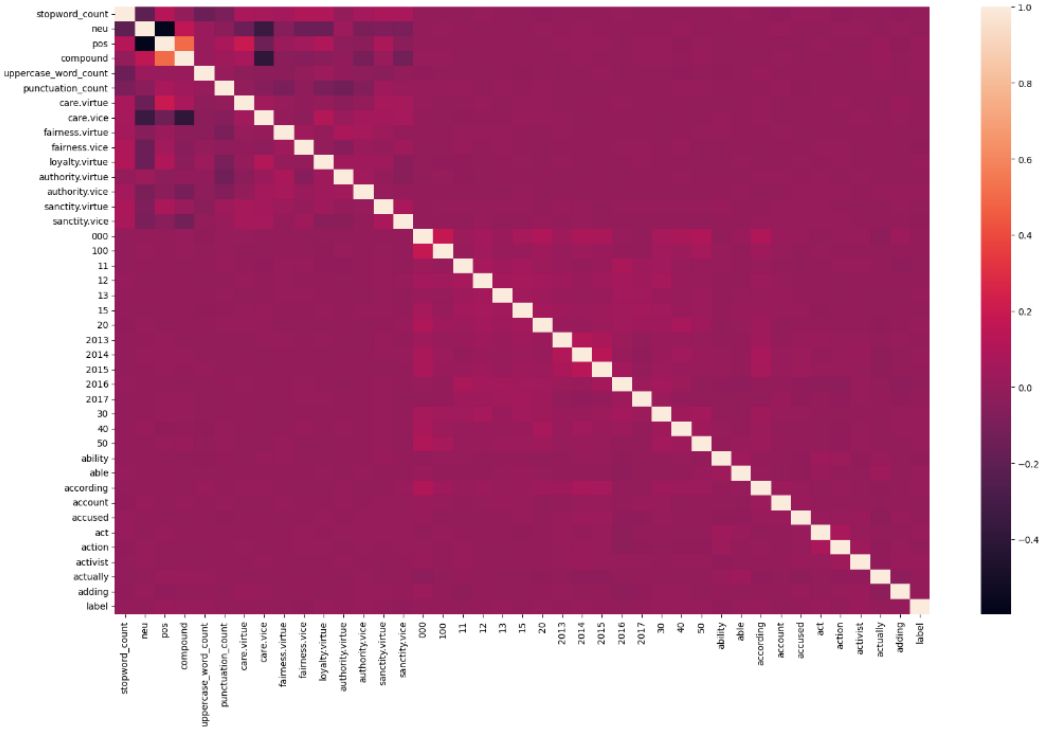
- For small datasets, 'liblinear' is a good choice, whereas 'sag' and 'saga' are faster for large ones;

'saga'	'elasticnet', 'l1', 'l2', None	yes

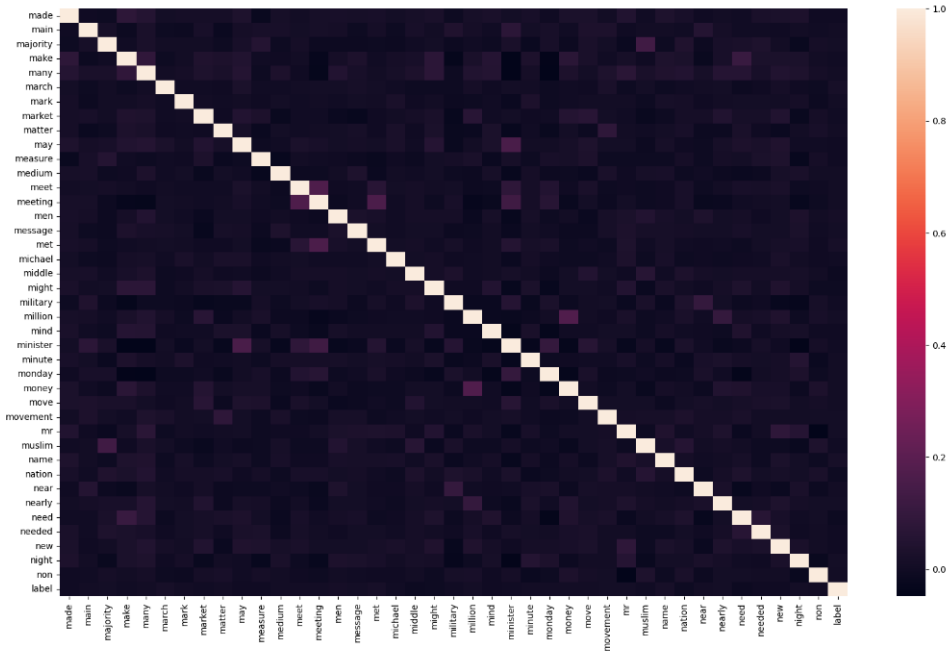
בחירת הפיצ'רים שלו:

```
549  features selected by LASSO:  ['stopword_count', 'neu', 'pos', 'compound', 'uppercase_
word_count', 'punctuation_count', 'care.virtue', 'care.vice', 'fairness.virtue', 'fairnes
s.vice', 'loyalty.virtue', 'authority.virtue', 'authority.vice', 'sanctity.virtue', 'sanct
ity.vice', '000', '100', '11', '12', '13', '15', '20', '2013', '2014', '2015', '2016', '20
17', '30', '40', '50', 'ability', 'able', 'account', 'accused', 'action', 'activist', 'act
ually', 'adding', 'administration', 'adviser', 'agenda', 'agreed', 'agreement', 'ahead',
'ally', 'almost', 'along', 'already', 'also', 'america', 'american', 'americans', 'among',
'another', 'answer', 'anti', 'anyone', 'anything', 'april', 'area', 'army', 'around', 'as
k', 'asked', 'attack', 'attention', 'attorney', 'august', 'away', 'bad', 'ban', 'bank', 'b
arack', 'battle', 'become', 'began', 'behind', 'believe', 'best', 'beyond', 'biggest', 'bi
ll', 'black', 'board', 'body', 'border', 'break', 'build', 'bush', 'california', 'called',
'came', 'campaign', 'candidate', 'capital', 'care', 'cause', 'center', 'certain', 'chairma
```

נשים לב כי בהשוואה לשיטות הקודמות, בחירת פיצ'רים זו מתכתבת עם עקרונות בחירת הפיצ'רים במודלים הקודמים שציינו (SVM ועץ ההחלטה) ומחזקת את הבחירה שלנו. עבור שיטה זו נבחרו המון פיצ'רים ולכן מטריצת קורלציה עבור כולם הייתה גדולה מידי. לכן בחרנו להציג את מטריצת הקורלציה עבור הפיצ'רים בראש הרשימה והתיוג. מטריצת הקורלציה:



המטריצה מאוד דומה לקודמותיה. לשם השוואה, מטריצת הקורלציה עבור פיצ'רים שרירותיים: 332 עד 372 מהרשימה תהיה:



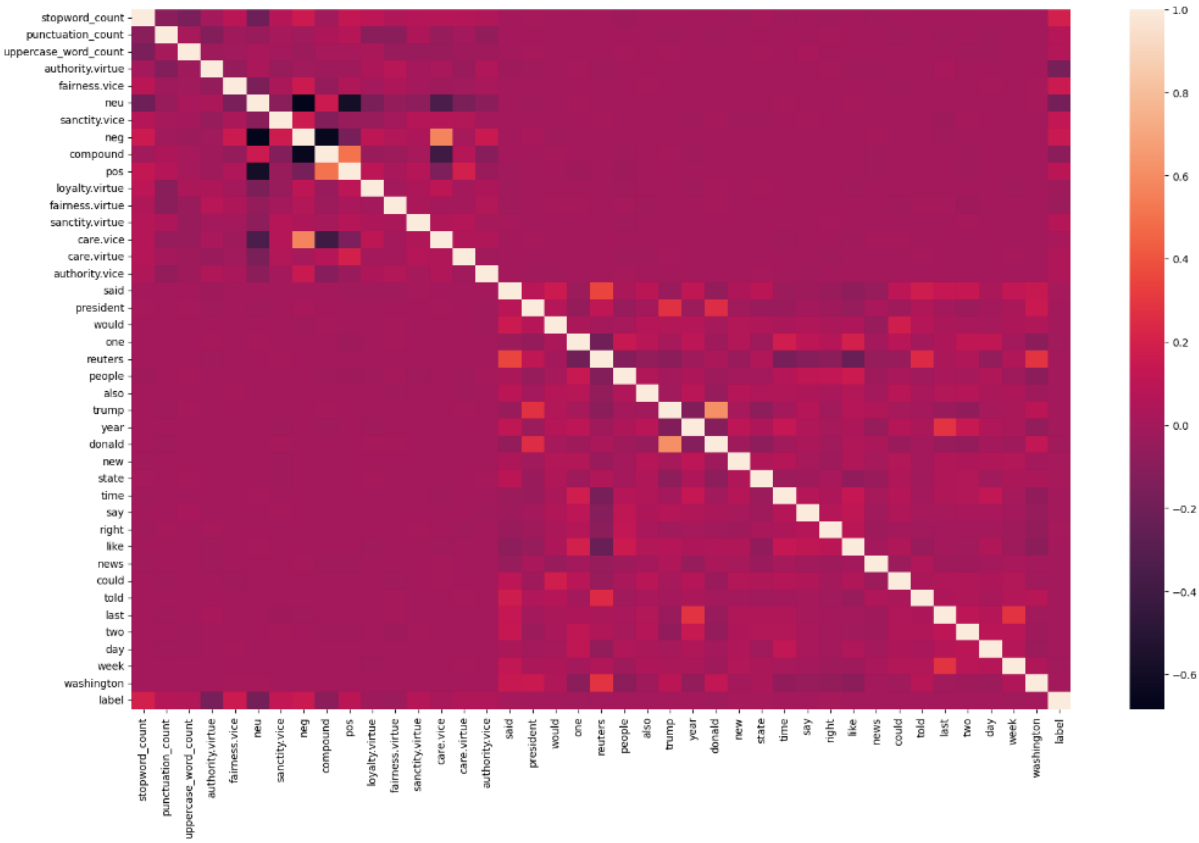
אם נשווה בין שתי המטריצות נשים לב כי גם עם שינוי הscale עדיין המטריצה השנייה כהה הרבה יותר בעמודת התיוג ביחס לראשונה. כלומר, קורלציה הרבה יותר נמוכה לתיוג עם התקדמות ברשימה. לכן נסיק כי הפיצ'רים האלו פחות יסייעו לנו בחיזוי ועדיף לקחת רק חלק קטן יחסית לגודל הרשימה, מהרשימה. תופעה דומה קורית עם הפיצ'רים 40 עד 80 ברשימה. מכאן נסיק כי ההתדרדרות בקורלציה לתיוג יחסית מהירה ולכן החלק שנרצה לקחת לא יעלה על 40 פיצ'רים מתוך הרשימה.

5. Random Forest

גם עבור שיטה זו בחרנו להשתמש בתכונת feature_importances_ של המודל (אותו פירוט כמו בבחירה של ID3) וקיבלנו את רשימת הפיצ'רים:

```
Top 40 features by importance chosen by Random Forest:
stopword_count: 0.0912
punctuation_count: 0.0286
uppercase_word_count: 0.0253
authority.virtue: 0.0246
fairness.vice: 0.0213
neu: 0.0178
sanctity.vice: 0.0135
neg: 0.0134
compound: 0.0111
pos: 0.0096
loyalty.virtue: 0.0077
fairness.virtue: 0.0068
sanctity.virtue: 0.0056
care.vice: 0.0056
care.virtue: 0.0053
authority.vice: 0.0041
said: 0.0034
president: 0.0029
would: 0.0028
one: 0.0026
```

גם כאן יש הסכמה גדולה על הפיצ'רים הנבחרים עם בחירתם של המודלים הקודמים. ניתן לראות זאת בא לידי ביטוי בתוצאות ובדמיון של כל מטריצות הקורלציה. מטריצת קורלציה:



שלב 6: אימון מחדש על המודלים הקיימים ועל מודלים חדשים:

כעת נפרט בטבלה את תוצאות האימון והמבחן על כל מודל והפיצ׳רים שחילצנו **רק** ממנו (באחוזים):

Random Forests		Logistic Regression		SVM		ID3 features		Word Cloud features		
train	val	train	val	train	val	train	val	train	val	
77.3	77.2	77.4	77.1	76.8	77.4	77.2	77.0	82.1	82.3	AdaBoost
69.5	70.0	70.0	67.4	70.2	70.1	69.3	69.8	62.4	61.6	Logistic Regression
100.0	79.3	100.0	77.1	100.0	78.9	100.0	78.7	96.0	81.7	Random Forests
90.1	79.3	90.5	77.9	87.9	79.3	89.4	78.3	85.9	83.6	XGBoost

באופן מובהק ניתן לראות כי הפיצ'רים שעשו הכי הרבה שיפור ביחס לבחינת המודלים עד כה הם של ענן המילים. בממוצע על פני 3 המודלים שנבחנו על סט הוולידציה, אם לא נחשיב את הרגרסיה, שימוש בפיצ'רים אלו בלבד נתנה דיוק של כ-82%. נדגיש כי ענן המילים **לא** חושב על סט הוולידציה אלא רק על סט האימון. ניתן לראות כי באופן גורף (מלבד הפיצ'רים שבחר ה SVM) הרגרסיה נתנה ביצועים הרבה פחות טובים משאר המודלים וכי המודלים Random Forests ו XGBoost היו מעט טובים יותר מ AdaBoost. בנוסף, כל שיטת בחירת הפיצ'רים (שאינה שיטת ענן המילים) שיפרה את ביצועי המודל עליו נבחנה לכל מודל ביחס לביצועים עם כלל הפיצ'רים! (מלבד הרגרסיה) בנוסף, אם נשווה את מטריצות הקורלציה של כלל השיטות, נראה כי חוץ מענן המילים, כלל השיטות נותנות מטריצות יחסית בהירות ודומות, וכי עמודת הקורלציה לתיוג דומה בכולם. ואכן נבחרו בכל השיטות פיצ'רים דומים להם קורלציה חיובית שאינה אפס או קורלציה שלילית לא קרובה לאפס לתיוג.

<u>בנוגע לבחירת הפיצ'רים בחרנו לנסות 2 אפשרויות על סט הוולידציה:</u>

1. לבחור את הפיצ'רים שנבחרו ע"י ענן המילים, עץ ההחלטה, LASSO, SVM ו Random Forests.
2. להוסיף על כלל הפיצ'רים משלב העיבוד המקדים את הפיצ'רים שנבחרו ע"י ענן המילים.

בכך נוכל לראות האם יש יתרון להשמטת פיצ'רים שלא נבחרו על דיוק המודלים שנבחנו. מי מבין 2 האפשרויות שתיתן תוצאה טובה יותר, נבחר בה למודל הסופי.

התוצאות מפורטות בטבלה:

מודל / פיצ'רים	אפשרות 1		אפשרות 2	
	val	train	val	train
AdaBoost	86.6	86.5	86.6	86.6
Logistic Regression	70.5	73.2	71.1	73.4
Random Forests	86.7	100.0	86.1	100.0
XGBoost	86.4	95.9	86.9	96.2

לפי הטבלה ניתן לראות כי הביצועים קופצים לכ- 86% למודלים החזקים עבור 2 האפשרויות ולכ- 70% במודל הרגרסיה. כלומר קיבלנו סה"כ שיפור של כ-30% מה baseline איתו התחלנו של Knn! בנוסף, נשים לב כי הבחירה בפיצ'רים והוספת פיצ'רים של ענן המילים שיפרה בכ- 10% את הביצועים יחסית לבחירה שרירותית בהיפר פרמטרים במודלים אלו, לעומת השיפור שקיבלנו בכוונון היפר פרמטרים של אחוזים בודדים. כלומר, אכן השיפור המשמעותי נובע מבחירת הפיצ'רים ופחות מכוונון היפר הפרמטרים. נשים לב שאין יתרון משמעותי להשמטת הפיצ'רים מהדאטה בביצועים. לכן ניתן לחשוב האם אנחנו רוצים לוותר על אותם פיצ'רים בתמורה לזמן עיבוד מקדים טוב יותר, או שמא ניקח את כולם. נימוק ללקחת את כל הפיצ'רים יהיה כמה אנחנו מסתמכים על איכות הדאטה שלנו. אם מלכתחילה הדאטה לא היה מספיק מגוון, ולא ייצג נאמנה את ההתפלגות האמיתית של ידיעות חדשותיות בעולם, אז לא יהיה נכון לזרוק לפח חלק מהפיצ'רים. מאחר ואנחנו לא בטוחות במאת האחוזים שהדאטה איכותי מאוד, לא נרצה לוותר על אותם פיצ'רים מאחר ולא ראינו יתרון לכך בטבלה הנ"ל.

שלב 7: יצירת מודל סופי ובחינתו

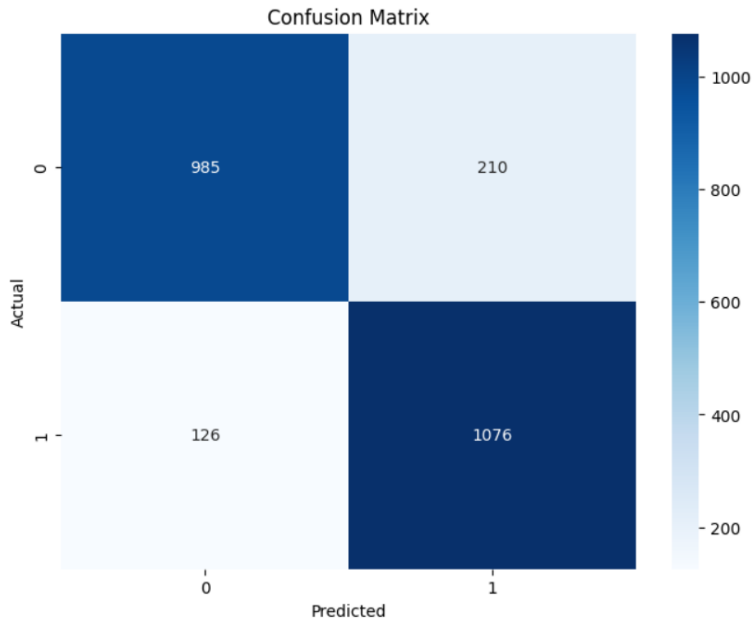
המודל הסופי שלנו הוא כלל majority מבין סיווגי המודלים Random Forests, XGBoost, AdaBoost. לאחר בדיקה שלנו עלה כי הוספת הפיצ'רים של ענן המילים הוסיפה משמעותית לביצועים על סט המבחן (נזכיר כי עליו לא חושבו עננים המילים). בנוסף ראינו כי מבין 3 האפשרויות, כאשר עבור שלושתן הוספנו את הפיצ'רים של עננים המילים, לא היה הבדל משמעותי בביצועים. לכן, החלטנו להשתמש בכלל הפיצ'רים במודל הסופי. ניתן לטעון כי ישנה יתירות בשל הממצאים האלה, ועדיף לבחור פחות פיצ'רים, אבל כאמור חבל להפחית מידע אם בסוף מגיעים לאותה תוצאה ואין ביטחון מלא באיכות סט האימון. מגוון זה אולי יכול לסייע בחיזוי ידיעות מגוונות יותר ממה שקיים אצלנו בדאטה סט. בנוסף, לאחר עיבוד הטקסטים שמנו לב כי לאחר פילטור ולמטיזציה ישנם ערכים חסרים (NaN). השלמנו את הערכים החסרים באמצעות הטקסט המקורי לפני הפילטור כדי לא לאבד את המידע שלו (מקור עזר: מילוי ערכים חסרים עי DF אחר).

בחנו אותו על ה- test set וקיבלנו את התוצאות והתרשימים הבאים:

Test Accuracy: 85.9%

די דומה לאחוזי הדיוק על סט הוולידציה. נדגיש כי לא נגענו בכלל בסט המבחן לאורך העבודה עד עכשיו כאשר בחנו את המודל הסופי עליו.

תרשים 1: מטריצת בלבול על סט המבחן



ניתן לראות ממטריצת הבלבול כי אכן המודל הסופי שלנו מסווג נכונה את רוב נקודות המבחן, אך מתקשה יותר בזיהוי ידיעות כזב מאשר זיהוי ידיעות אמת. ישנם 247 ידיעות כזב אשר סווגו כידיעות אמת לעומת 104 ידיעות אמת שסווגו כידיעות כזב. זאת בניגוד לקושי שלו שעלה בניתוח המודלים קודם לכן. על כן המודל אכן אינו מסווג בצורה מושלמת אך עדיין יכול לתת אומדן טוב עבור ידיעות שאיננו יודעים מה סיווגם האמיתי.

תרשים 2: מדדים

	precision	recall	f1-score	support
0	0.89	0.82	0.85	1195
1	0.84	0.90	0.86	1202
accuracy			0.86	2397
macro avg	0.86	0.86	0.86	2397
weighted avg	0.86	0.86	0.86	2397

<u>מהתוצאות המוצגות בטבלה ניתן להסיק כי:</u>

1. accuracy: הדיוק הכללי של המודל הוא 85%.
2. precision:
   - מתוך כלל הידיעות שסווגו ככוזבות, 89% מהן היו כוזבות.
   - מתוך כלל הידיעות שסווגו כאמת, 84% מהן היו אמת.
3. recall (שליפה):
   - מתוך כלל הידיעות הכוזבות, 82% מהן סווגו ככוזבות.
   - מתוך כלל ידיעות האמת, 90% מהן סווגו כאמת.
4. מדד f1-score:
   - למחלקה 0 יש F1-score של 85%, מה שמעיד על איזון טוב בין דיוק לשליפה.
   - למחלקה 1 יש F1-score של 86%, מה שמעיד על איזון טוב בין דיוק לשליפה.
5. ממוצעים: שני סוגי הממוצעים עומדים על דיוק של 86% - מעיד על ביצועים יציבים ב2 המחלקות.

המסקנה הכללית היא כי המודל מציג ביצועים טובים ומאוזנים, ומצליח לזהות בצורה נכונה את רוב הדוגמאות בשתי המחלקות, עם ביצועים מעט יותר טובים במחלקה 1, כלומר של ידיעות אמת בהיבט של שליפה.

שלב 8: בדיקת המודל הסופי על תמלילים של דובר צה"ל ועל ידיעות כזב

רעיון נוסף שעלה לנוכח המצב, היה לבדוק את ביצועי המודל על ידיעות שדובר צה"ל מפרסם בשפה האנגלית. ידוע לנו כי אלו הן ידיעות אמת ממקור מוסמך ולכן וקטור התיוגים הינו וקטור 1ים.
אספנו 24 ידיעות שונות מזמנים שונים לאורך השנה. ה- data מצורף כמסמך. ביצענו עליו predict אשר בפנים הפעיל עליו את תהליך העיבוד המקדים של ה- data ליצירת הפיצ'רים הנכונים באותו תהליך שנוצרו הפיצ'רים של סט המבחן קודם לכן. אלו הן התוצאות שקיבלנו:

Test Accuracy: 91.6%

מודל הצליח לסווג נכונה 22 ידיעות מתוך ה 24 שנתנו לו.
הידיעות אותן לא הצליח לסווג הן:

	text
10	The Nahal Brigade Combat Team continues to ope...
17	Hezbollah is an internationally recognized ter...

וניתן להתבונן בהם בקובץ המצורף. ביחס לשאר הידיעות לא מצאנו מאפיינים בידיעות אלו שיכולים להסביר את הכישלון של המודל עליה, לכן ניתן לשער כי אולי וודאות המודל בסיווג הידיעות לא הייתה גדולה.

תרשים 1: מדדים

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         0
           1       1.00      0.92      0.96        24

    accuracy                           0.92        24
   macro avg       0.50      0.46      0.48        24
weighted avg       1.00      0.92      0.96        24

[1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1]
```

<u>כעת נפרט על מדדי המודל:</u>

נציין כי הדאטה לא הכיל כלל ידיעות כזב ולכן הערכים בטבלה שקשורים למחלקה זו הינם אפסים.

עבור מחלקה 1, שהינה מחלקת ידיעות אמת, המודל מציג precision של 1.00 ו-recall של 0.92, מה שמוביל ל- f1-score של 0.96. זה מראה כי המודל הסופי שלנו מצליח לזהות נכון את המחלקה הנ"ל כמעט ללא טעויות. נציין כי זה מספר מועט של ידיעות ועלול לא לייצג נכונה את ביצועי המודל על דאטה שבא מאתר זה. למרות זאת, אנחנו כן רואות בכך הצלחה של המודל בהכללה על דאטה מתחום ספציפי שאולי היה פחות מיוצג בדאטה עליו התאמן, ואם כן אז בוודאות לא מגורם רשמי ומקורב למציאות כמו דובר צה"ל.

כעבודת המשך נוכל לקחת יותר ידיעות ולנסות עליהן את המודל, וגם לקחת ידיעות מאתרים שונים, תחומים שונים ומגוונים יותר. היה לנו קושי למצוא ידיעות כזב מכיוון שמצאנו רק דיווחים על ידיעות כזב ולא את הידיעות עצמם. לכן לא הצלחנו בבניית דאטה סט דומה על ידיעות כזב.

דיון:

בעקבות המלחמה שהחלה לפני כשנה והתופעה השכיחה ברשתות החברתיות של ידיעות כזב, רצינו לייצר כלי שבעזרתו בהינתן טקסט, הוא יפלוט לנו האם הידיעה היא אמיתית או ידיעת כזב (שקרית).

באופן אישי, היה לנו מאוד קשה לראות ברחבי הרשתות את ידיעות הכזב המופצות על ישראל ולכן היה לנו מאוד חשוב לבנות מודל שיפתור בעיה זו. מעבר לכך, מאוד עניין אותנו להבין אילו מאפיינים מזוהים בעיקר עם ידיעת כזב ועם ידיעת אמת, כדי לנסות להפעיל ראייה ביקורתית ואף לנסות להבין בעצמנו ולפענח את טיב הידיעה. השתמשנו במספר מודלים שונים של למידת מכונה קלאסית, באמצעות חלקם הרכבנו מודל סופי ובנוסף חילצנו את המאפיינים שעוזרים למודלים שלנו לסווג נכונה. רצינו במסגרת הפרויקט לא רק לעשות תחזית אלא גם להבין יותר את הדאטה והמשימה עצמה ע"י פרשנות של אילו פיצ'רים רלוונטיים ומשמעותיים בכל מודל.

כחלק מהפרויקט בחנו מספר מודלים, לכל אחד יש יתרונות וחסרונות. מצאנו כי המודלים החזקים שאותם בדקנו מתקשים בזיהוי ידיעות אמת מאשר זיהוי ידיעות כזב. מהמודלים שהיו נראים החזקים ביותר רצינו לקחת מכל אחד את היתרונות הייחודיים לו ע"י כלל majority בין כולם ובכך לנסות לחפות על חסרונותיו של כל מודל מה שהיה נראה לפי התוצאות שעובד. לשם השוואה, בתחילה, במודל XGBoost ראינו כי הדיוק על סט הוולידציה היה 77%. לאחר בחירת הפיצ'רים של ענן המילים הדיוק על סט הוולידציה עלה ל- 83% ולאחר מידול של כלל majority הדיוק על סט המבחן היה .86%

מעבר להצלחה בסיווג הידיעות עסקנו בשאלת המחקר: מהם הפיצ'רים הכי משפיעים ותורמים לדיוק המודל הסופי? גילינו כי הפיצ'רים הכי משמעותיים לרוב המודלים היו:

Stopword_count, uppercase_word_count, punctuation_count, neu, authority.virtue.

מה היה אפשר לעשות אחרת: ניתן להשתמש במודלי למידה עמוקה שהם state of the art בתחום הNLP. למצוא דאטה מגוון יותר מעוד מקורות או לבנות דאטה סט בעצמנו. נציין כי הדבר חשוב ללמידת דפוסים אמיתיים המתאפשרת רק באמצעות דאטה מגוון. לכן אנו רואות בכך צעד חשוב וראשון לעבודת המשך. בנוסף, הוספת פיצ'רים נוספים כמפורט מטה. כל אלו אמנם היה ניתן לעשות אך עקב מגבלות הזמן והיקף הפרויקט לא עשינו זאת. כמובן שאת כל הדברים הללו ניתן לעשות כעבודת המשך וכמחקר לעתיד על מנת לשפר את ביצועי המודל שלנו. בעתיד היינו גם מבצעות מודל שיצליח לסווג לפי כותרת של הידיעה ולא רק לפי תוכנה כפי שעשינו בעבודה שלנו.

נדגיש כי אנחנו רואות בעבודתנו את הצעד הראשון לפתרון הבעיה שהצגנו וכי היא לא מסתיימת כאן. אמנם הצלחנו להגיע לביצועים טובים על סט המבחן אך אין הדבר מבטיח את הביצועים על כל סט מבחן שהוא. לכן כיווני מחקר לעתיד חשובים מאוד לפתרון הבעיה.

היתרונות של השיטה: הכלים שעשינו בהם שימוש מוכרים ומובנים בספריות למידה הנגישות לכולם, השתמשנו בפיצ'רים שבנינו בעצמינו ולכן קל להתחבר למשמעות שלהם לעולם הבעיה (בניגוד לשימוש במודלי למידה עמוקה בהם המודל מפיק פיצ'רים שלא תמיד נראים הכי הגיוניים למשתמש) בנוסף יש חופש בבחירת הפיצ'רים ולכן היכרות עם עולם הבעיה תורם רבות לדיוק המודל.

החסרונות של השיטה: זמן עיבוד גדול, הדאטה נלקח ממקור אחד ולכן עלול לייצג באופן מוטה את כלל הידיעות הקיימות בעולם (אך הנחנו בעבודה שעקב כך שהוא גדול מאוד הוא מייצג בצורה סבירה את העולם), שיטות למידה קלאסיות אינן הstate of the art לבעיה זו (אמנם עשינו בהם שימוש עקב תימוך ממאמרים), שיטת כיוונון היפר פרמטרים נתונה לשיפור.

בנוסף, אנחנו יודעות שיש בעייתיות בשיטת כיוונון הפרמטרים שבה השתמשנו. למשל ב- adaboost נתנה תוצאות פחות טובות מבחירה שרירותית של הפרמטרים (אמנם לא בהרבה). עם זאת לא הייתה לנו ברירה

כיוון שהשיטה היסודית יקרה מאוד בזמן ריצה. לכן כחלק מעבודת ההמשך, היינו חוקרות שיטות אחרות לכוונון הפרמטרים כך שגם יעילות בזמן ריצה, וגם נותנות לנו תוצאות טובות.

<u>בעתיד כעבודת המשך, יש לנו רעיונות לעוד פיצ'רים שניתן לעשות שיכולים לעזור למודל שלנו לסווג נכונה:</u>

1. כמות/ תדירות של: סימני קריאה צמודים/ סימני שאלה/ תיוגים (@ או #) / אמוג'י / מילות השוואה / מילים שליליות / מילים מתריסות.
2. כמות ציטוטים בטקסט.
3. מילות ניגוד.
4. מילים "אמוציונליות".
5. מילים לא פורמליות/ קללות.
6. מילות שאלה (איך/ כמה/ למה/ מתי וכו)
7. היחס בין השימוש במאזכרים (כינויי גוף, כינויי שייכות, מילות יחס בהטיות שונות, ביטויים מכלילים, כינויי רומז) לעומת השימוש בשמות של אנשים / מקומות.
8. מילים טוטאליות "never", "always", "beyond doubt", "unanimously" .
9. תדירות של מספרים.
10. רמת השפה של הידיעה / כמות הבהרות ממוצעת במילה.

כפי שנאמר, בעתיד ניתן לשלב למידה עמוקה עם רשתות נוירונים, ובכך לגרום למודל להיות מורכב יותר ויוכל אולי לסווג בצורה נכונה אף יותר. בנוסף, אנחנו רוצות לחקור יותר לעומק מאפיינים שאינם בתוכן הידיעה אלא מחוצה לה כמו למשל מי פרסם אותה, מתי, בסמיכות לאילו אירועים חשובים וכדומה. לעבור על עוד מאמרים ועבודות שלא הספקנו לעבור עליהם ולהפיק מהם עוד פיצ'רים מעניינים עבור המודל הקלאסי.

ביבליוגרפיה:

מאמרים בנושא:

[1]: Fake News Detection on Social Media:
https://arxiv.org/pdf/1708.01967

[2]: A Survey of Fake News:
https://arxiv.org/pdf/1812.00315

[3]: Fake News Detection Using Machine Learning Approaches:
https://iopscience.iop.org/article/10.1088/1757-899X/1099/1/012040/pdf

[4]: Fake News Detection Using Machine Learning Ensemble Methods
https://onlinelibrary.wiley.com/doi/pdf/10.1155/2020/8885861

[5]: An overview of online fake news: Characterization, detection, and discussion
https://www.globacademy.org/wp-content/uploads/2023/03/Zhang- -Ghorbani-2020.pdf

מילון:

[6]: Moral Foundations Dictionary
https://osf.io/ezn37/
https://provalisresearch.com/products/content-analysis-software/wordstat-dictionary/moral-foundations-dictionary/

עבודות בנושא:

https://github.com/kapilsinghnegi/Fake-News-Detection/tree/main

קישורי עזר לכתיבת הקוד: (נוסף על כך, כל התיעודים של כל המחלקות ופונקציות שבהן השתמשנו)

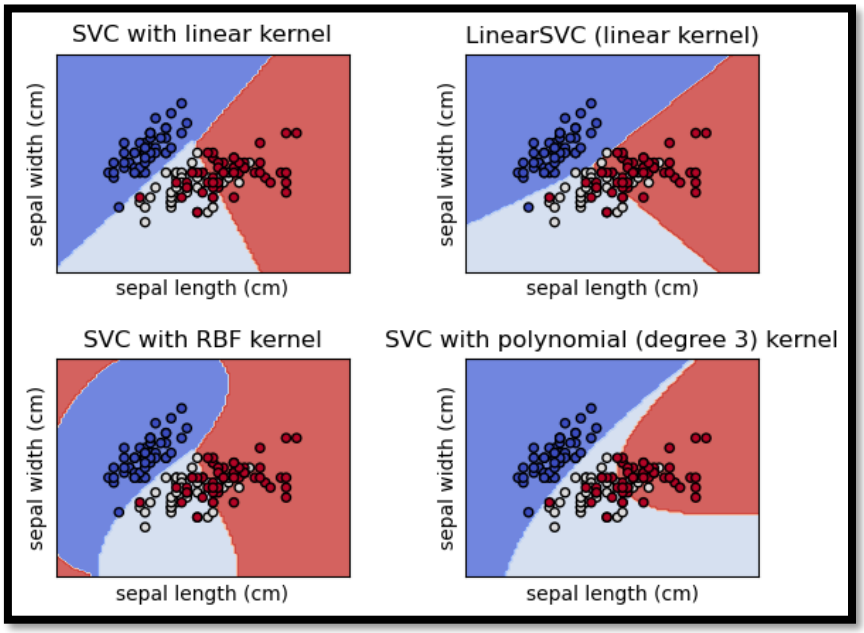
1. pickle docs
2. Tuning the hyper-parameters of an estimator
3. plot ID3 tree
4. SelectFromModel
5. confusion matrix
6. word cloud
7. xgboost
8. saving-the-state-of-a-program-to-allow-it-to-be-resumed
9. precision handling

עוד שיטות לבחירת פיצ'רים כעבודת המשך: feature selection, VarianceThreshold
עוד טכניקות לבחירת פיצ'רים לעתיד: (אחת שהשתמשנו בה - קורלציה) feature selection techniques

פרטים טכניים: מפורטים בקבצי הREADME
ספריות: מפורט בקובץ libs.txt

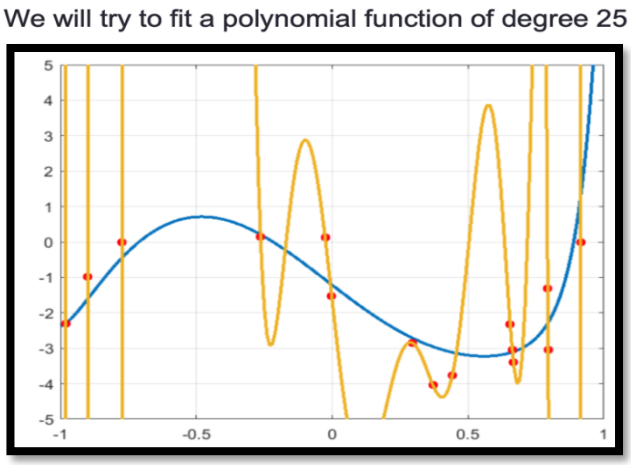
נספחים:

1. המחשה על ההבדלים בין סוגי ה-kernels השונים:



ניתן לראות כי אכן באמצעות Linear kernel ניתן לבצע הפרדה לינארית בלבד שאיננה טובה עבור ה-data הניתן לעיל כלל. עם Polynomial kernel מדרגה 3 ניתן לבצע הפרדה טובה יותר מאשר הקרנל הלינארי. ניתן לראות כי הוא מכליל באזורים בהם אין נקודות אימון בצורה שיתכן ואינה מיטבית ושאולי אינה נכונה עבור כלל ה-data, שאינו זמין כולו בבניית המודל המסווג. כאן עם RBF kernel ניתן לראות כי ניתן לבצע הפרדה טובה יותר ומפרידה טוב יותר. ניתן לראות כי הגבול בין המחלקות הרבה יותר גמיש ובעל צורה מעוקלת שמנסה להפריד בין הקבוצות בצורה מדויקת יותר. ייתכן כי מתקיים כאן overfit, לדוגמה באיזורים בהם יש גם נקודות אדומות וגם לבנות. לא נוכל לדעת זאת כי אכן הוא כל ה- data אינו זמין, ונוכל לבדוק זאת ע"י קבוצת המבחן. נציין כי התמונה נלקחה מתוך אתר scikit-learn

2. המחשה על סכנת ה- overfitting כאשר הדרגה גבוהה:



נשים לב שכבר כאשר דרגת הינה 25, נקבל מהמודל את הפונקציה הצהובה עם שגיאת אימון 0, אך כמובן אינה מכלילה טוב ואינה כלל תסווג כנדרש. הפונקציה שתסווג נכונה היא הפונקציה הכחולה ואכן ניתן לראות את השוני הרב ביניהם. זה ממחיש את חשיבות בחירת דרגת הפולינום. נציין כי התמונה נלקחה ממצגת התרגול של קורס מבוא למערכות לומדות.

הצהרה על מודלי שפה:

בנינו את כל הקוד לבד ולא העתקנו שום פסקה ממודלי השפה. כל דרכי הפתרון, הרעיונות לבניית הפיצ׳רים, השימוש במודלים וכל שאר הרעיונות הינם פרי יצרתינו ומחקרינו **בלבד** חוץ מאלה שהדגשנו שלקחנו ממאמר שקראנו וציינו את שם המאמר. כלומר, לא השתמשנו כלל במודלי שפה על מנת לג'נרט דרכי פתרון או להביא רעיונות יצירתיים אלא אך ורק עבור פרטים טכניים למשל: נעזרנו במודל שפה chatgpt על מנת להצליח להדפיס נכונה גרפים ב matplotlib וב seaborn, לסדר את הקווים והצירים ועבור פרטים טכניים דומים נוספים המהווים אחוזים בודדים מהקוד בלבד. בנוסף התבססנו על ידע מקורסים קודמים שלקחנו כמו מבוא למערכות לומדות.

נדגיש כי מבחינת הדוח היבש, לא נעשה שימוש כלל במודלי שפה וכי הכל אנחנו כתבנו.

אנחנו ממש מתנצלות שפספסנו ולא שמרנו את כל ה- prompts בהם השתמשנו, אך הנה חלק מהprompts שתיארנו מזיכרוננו (לא המקוריים אלא דברים שאנחנו זוכרות ששאלנו ודברים שנשאלו לאחרונה) בהם השתמשנו בשאילתות למודל השפה chat gpt בעת כתיבת הקוד:

- What is random_state
- How to open in python .dic file
- Mark a point in matplotlib graph with dashed lines
- How to access the part of the corr matrix which contains only the rows & columns of certain features
- Show an example usage of pd applay on df
- Check if object is None python
- Remove all samples (=rows) from a df which has nan in certain columns
- general construction of class of classifing model with fit and predict only scikit learn

אנו מתחייבות כי שאר ה prompts זהים באופיים לאלו הכתובים ברשימה לעיל.