

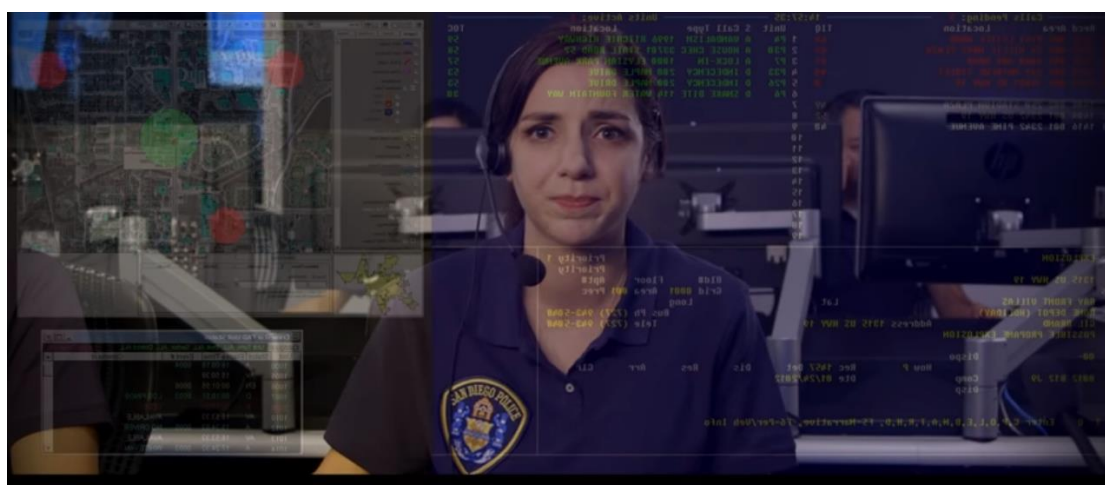
Sound-based Distress analysis

סיווג מצבי מצוקה על סמך (קבצי) שמע

פרויקט בבינה ממלאכותית

236502

הפקולטה למדעי המחשב, טכניון



מגישים:

1. לב טוניק (307366450, tuniklev@campus.technion.ac.il)

2. מורי לוינזון (308328467, smorlevi@campus.technion.ac.il)

תחת הנחיתו של:

מר יניב נמקובסקי

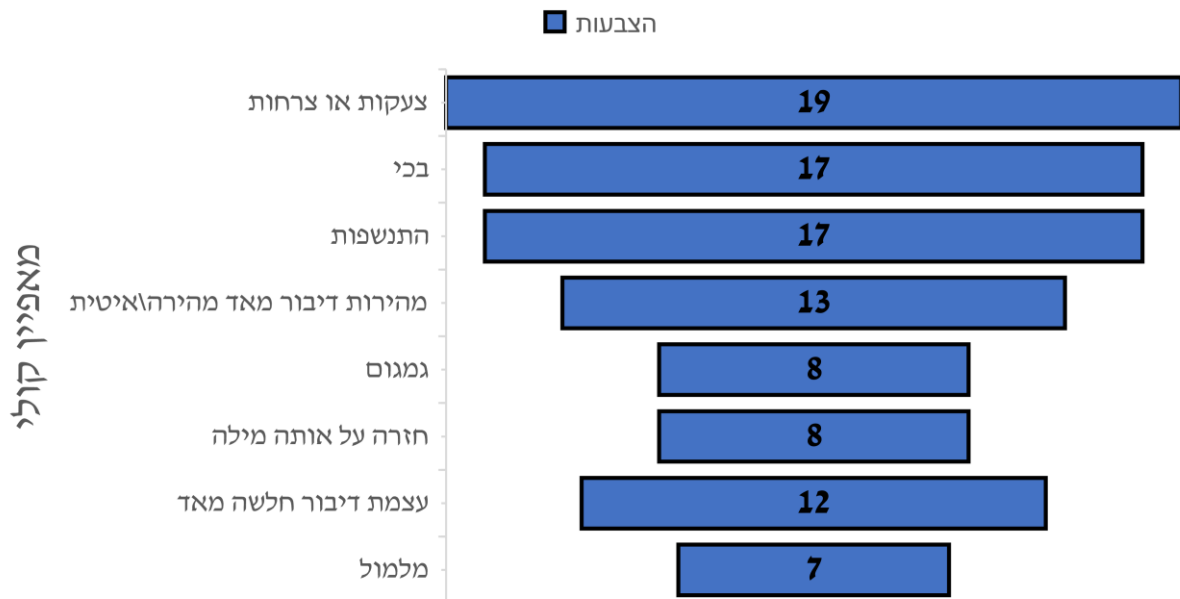
מבוא

1. הצגת הבעיה :
בהתבסס על קבצי שמע, האם ניתן להשתמש באלגוריתמי למידה (של בינה מלאכותית) על מנת לקבוע כי הקובץ מייצג אדם הנמצא במצוקה פיזית?
2. רקע כללי ומוטיבציה ראשונית לפרויקט :
לפני מספר שנים, התרחש בישראל אירוע חטיפה של נערים מאזור יהודה ושומרון ([לחץ פה לקריאת הערך בויקפדיה](#)). במהלך חטיפתם אחד הנערים התקשר למשטרה, אך **המוקדן לא הצליח להבין כי מדובר במצב חירום כיוון שהנער דיבר בקול מוחלש ולא ברור**, דבר אשר עיכב את תחילת הפעולה להשבת הנערים.
בפרויקט זה, אנו בונים וחוקרים כלי ממוחשב אשר **מבליט מאפיינים קוליים** של מצבי מצוקה פיזיים, ובכך מאפשר לצמצם את הסבירות שאדם במצוקה המתקשר לבקש עזרה ידחה על הסף.
3. כיווני מוטיבציה נוספים :
 - שיתוף קרובים במצב מצוקה- במידה והמערכת שלנו תותקן כאפליקציה על גבי מכשירים ניידים, קשישים אשר ייקלעו למצב מצוקה או נשים גברים אשר יקלעו למצב שבו הם חוששים שמה יותקפו ע"י אדם אחר יוכלו להפעיל את האפליקציה מראש, ובמידה והמערכת תזהה **סממן קולי של מצוקה** פיזית היא תוכל להתריע בפני קרובי משפחתו של בעל האפליקציה.
 - רובד אבטחה נוסף בקהילה- במידה וערים שונות יתקינו חיישני קול במקומות מועדים לפרענות, הם יוכלו לקבל רובד נוסף של מידע אשר יוכל להשלים תמונת מצב במרחבים שבהם מצלמת אבטחה ברחוב לא נותנת פתרון מלא.
4. על התהליך שעברנו :
במהלך הפרויקט יצא לנו להיחשף לרובד נוסף של למידה, כאשר נחשפנו ללמידה הנעשית באמצעות ניתוח קטעי קול, בעזרת ספריית Keras. דרך דוגמאות לפרויקטים אחרים שנעשו בעזרת הספרייה הזאת, ובעזרת הדוקומנטציה של הספרייה הצלחנו לבנות מסווג שייתן לנו תשובה בינארית לשאלה אותה אנו מנסים לפתור, תוך כדי כיוונון ההיפר-פרמטרים שאיפשרו לנו להגיע לתוצאה הטובה ביותר עבור כל אחת מתתי השאלות שעליה ניסינו לענות ובכך לספק תשובה לשאלה העיקרית- האם האדם בהקלטה נמצא במצוקה פיזית?
5. האם הצלחנו?
הצלחנו לזהות בהקלטת הנערים החטופים ז"ל מצב של מצוקה, כיוון שהאלגוריתם זיהה שם "צעקה", ו"לחישה", אך מנגד לא הצלחנו להגיע לתוצאות מדויקות בשאר ההקלטות שבדקנו- ועל כך בפרק הסיכום.

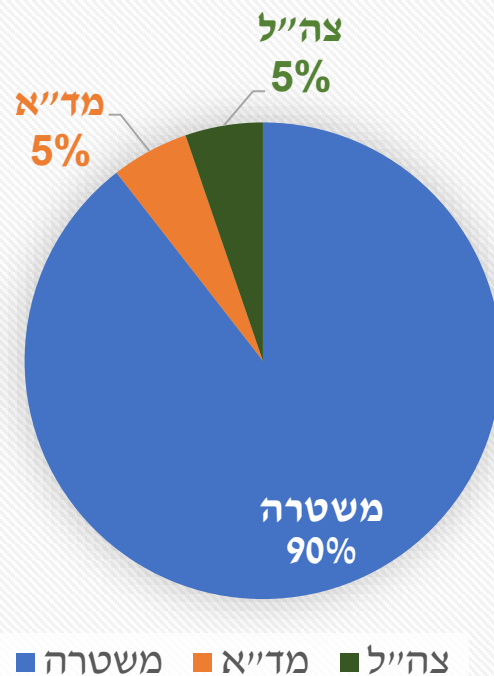
תיאור פתרון הבעיה

1. הדגשת הקושי בקביעת תשובה לשאלה סובייקטיבית :
השאלה "האם אדם נמצא במצוקה פיזית" הינה שאלה קשה מאוד, לא רק למחשב, אלא גם לבן-אדם. נמחיש ע"י דוגמאות :
מוקדן בעיריית חיפה רואה שני מבוגרים מדברים בלהט בפארק עירוני, כאשר פתאום אחד מהשניים שולף סכין, ודוקר את פלוני. פלוני נופל ומתחיל להתפתל על הדשא. האם דרוש במצב זה להזמין אמבולנס ומשטרה לזירת האירוע? או שאולי השניים מתאמנים לקראת הצגה על "יוליוס הקיסר" והסכין הייתה סכין מתקפלת מפלסטיק? באותה הרוח, חיישן קול של המשטרה אשר הותקן בסימטה חשוכה בה בעבר נעשו מעשי פשיעה קולטת פתאום צעקת "הצילו" ומייד לאחריה רעש של ריצה. האם דרוש להזמין מהר ניידת למקום האירוע או שאולי הצעקה קרתה עקב עכבר או חזיר-בר שעבר בסימטה?
הסיטואציות הנ"ל מובילות אותנו לנקודה שחשוב לציין.
2. הצהרה (Disclaimer) :
הפיתרון שלנו בפרויקט זה לוקה בכך שהוא אינו לוקח בחשבון את "ההקשר הרחב" של הסיטואציה שבה נמצאו המאפיינים הקוליים. אכן נתייחס לכך בפרק הסיכום, בו נציג כיוונים אפשריים לפיתרון נקודת תורפה זו. עם זאת, הפרויקט שלנו מהווה בסיס טוב לפיתרון השאלה האם אדם נמצא במצוקה פיזית, שכן הוא מכוון לזיהוי מאפיינים קוליים שבני האדם מייצרים בעת שהם במצוקה פיזית.
3. מהם המאפיינים הקוליים של אדם הנמצא במצוקה פיזית?
 1. ניסיון ראשון בדרך להבנה : תחקור פסיכולוגים.
לאחר שהבנו את הקושי בסוגיה, הגענו למסקנה (שגויה) שעל אף שהתשובה הינה סובייקטיבית, ניתן להגיע לקירוב טוב ע"י התייעצות עם מומחים בתחום הפסיכולוגיה. למרבה הפתעתנו, כאשר פנינו לפסיכולוגים ולסטודנטים לפסיכולוגיה, הם הסבירו לנו שמה שאנו מבקשים לדעת אינו נמצא בתחום אחריותם, שכן הם מטפלים באנשים שנמצאים במצוקה פיזית מתמשכת, בעוד שאנו מבקשים לגלות מאפיינים קוליים של אנשים הנמצאים במצב מצוקה מידי אשר ניתן לפעול לפותרו ע"י הזעקת עזרה.
 2. ניסיון שני : מוקדי חירום טלפוניים.
יצרנו סקר שאותו הפנינו למוקדים טלפוניים של כוחות הביטחון וההצלה. הפעם התשובות היו חד משמעיות ורלוונטיות שכן אנשים אלה מתעסקים באופן יום-יומי בסיווג ותיעוד של שיחות עם אזרחים אשר נמצאים במצוקה פיזית מידית. (לינק לסקר). על בסיס תוצאות הסקר יצרנו תיעוד של מאפיינים קוליים נפוצים במצבי מצוקה גופניים. (בפרק הסיכום נציע גישה אלטרנטיבית לסקר נוסף, שלטעמנו יוכל לשפר את המערכת שיצרנו). להלן תוצאות הסקר (עמוד הבא) :

מהם המאפיינים הקוליים העיקריים של אנשים במצב מצוקה פיזי?



פילוג המגיבים לסקר לפי רשות הצלה וביטחון



3. הקושי בקביעת ה"קלט": לאחר שהבנו את המאפיינים הקוליים שבהם נרצה להתרכז, היה דרוש להבין כיצד ניתן להפוך את השאלה הסובייקטיבית (האם אדם נמצא במצוקה פיזית)

לשאלה שקולה, שעליה ניתן יהיה לאמן מסווג מתחום הבינה המלאכותית. האופציה "פשוט לתת קבצי קלט של שמע שבהן מוקלטות שיחות טלפון של מוקדי חירום ולתייג אותן" ירדה די מהר מהשולחן. הסיבה טמונה בכך שאורכן עולה בדרך כלל על 5 שניות, ואף עלול להגיע לדקות שלמות. כך שהערכנו שהמחשב לא ילמד באופן יעיל את המאפיינים הקוליים החשובים בהקלטה שיש בה הרבה פרטים שאינם מייצגים מאפיין קולי חשוב (קיימת התייחסות נוספת לבניית מסווג על בסיס שיחות שלמות עם מוקדי חירום בפרק המסכם את העבודה). לכן הגענו למסקנה כי עלינו לקבל כקלט קטעי שמע קצרים (3-5 שניות) עם מאפיינים קוליים רלוונטיים, עליהם לבצע את הלמידה, ולאחר מכן, כשהמסווג יהיה מאומן ומוכן לבדיקה, לתת לו כקלט שיחות שלמות, אשר פורקו למקטעים שאורכן גם כן 3-5 שניות. (פירוט נוסף על הפירוק יינתן בפרק על רכיבי המערכת).

לסיכום, תוצאות הסקר, והמסקנה בנוגע לאופן שבו המחשב אמור לקבל את הקלט, הביאו אותנו לחשוב על הפתרון הבא:

4. הפתרון המוצע:

הסבר אינטואיטיבי: נפרק את תת השאלה הסובייקטיבית לתתי שאלות אובייקטיביות (השאלות האובייקטיביות מבוססות על תוצאות הסקר).

1. קלט המערכת:

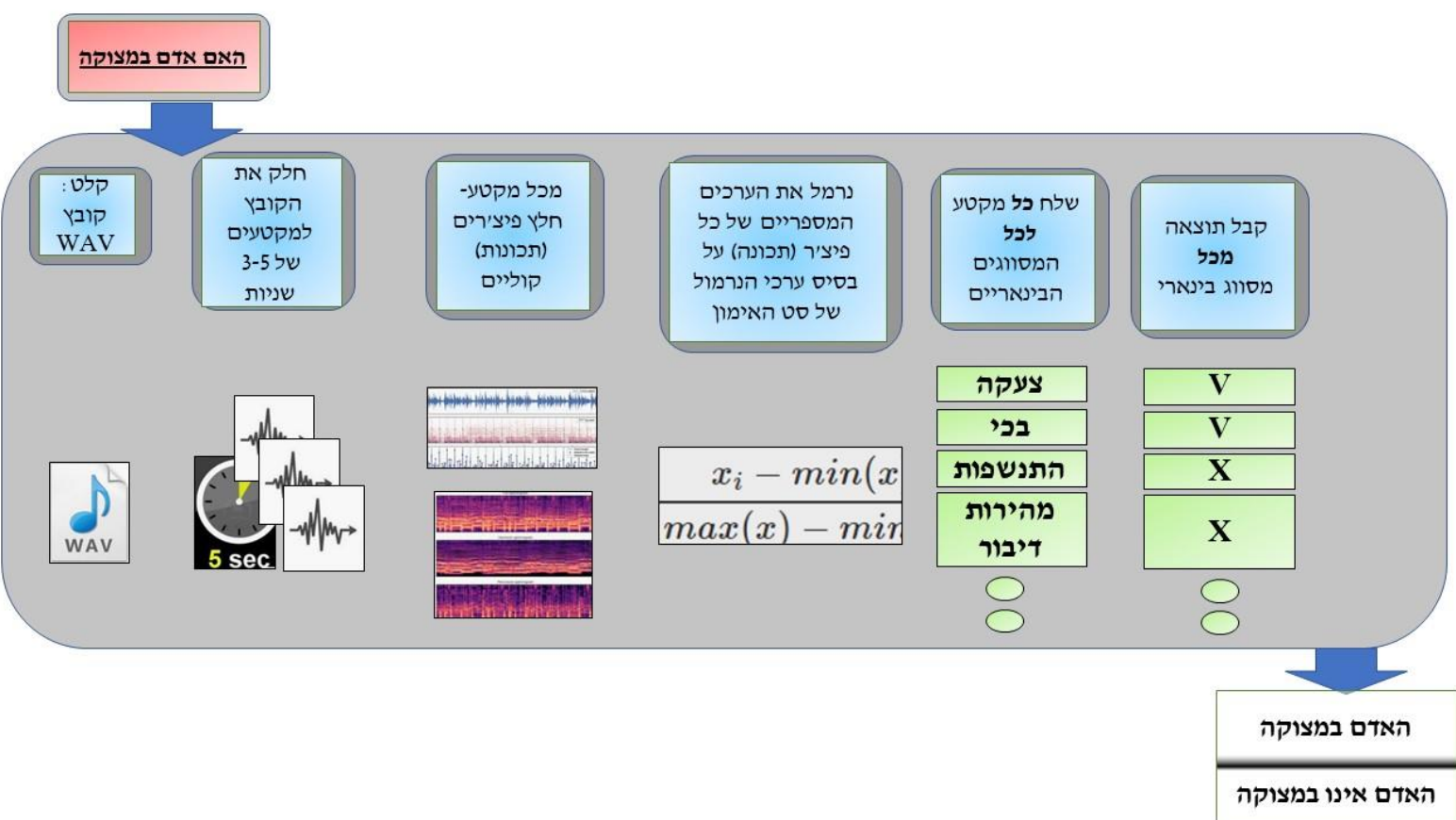
- קובץ שמע.

2. פלט המערכת:

- הקובץ מכיל לא מכיל מאפיינים קוליים של אדם במצוקה פיזית.
- בהתבסס על התוצאות של המאפיינים שזוהו נוכל לכתב גם המלצה לפעולות שיש לנקוט בהן.

3. להלן המחשת ה"פסאודו-קוד" של האלגוריתם:

(פירוט על רכיבי האלגוריתם יינתן בפרק על רכיבי המערכת)



תיאור המערכת לפי שלבי בנייתה

איסוף המידע הגולמי (scrape data)

1. משטרת ישראל - ניסיון שלא צלח לעת עתה.
בתחילה פנינו למשטרת ישראל כדי לקבל גישה לשיחות טלפון שנעשו למוקד החירום "100", אך המשטרה סירבה לספק לנו את ההקלטות. (הנימוק לסירוב והבקשה מצ"ב בנספחים, פירוט נוסף על מהות הבקשה יינתן בפרק הסיכום). לכן נאלצנו להסתמך על מקורות אחרים:
2. חברת גוגל (google) - בחרנו לבסוף שלא להשתמש במאגר זה.
לחברת גוגל קיים מאגר שנקרא AudioSet [\(לחץ פה להפניה למאגר\)](#). זוהי הייתה האופציה השנייה שחשבנו עליה, עם זאת בחרנו לבסוף שלא להשתמש במאגר זה עקב הסיבות הבאות:
 1. אמינות - המאגר התבסס על סרטוני "יו-טיוב" (YouTube) שסווגו ע"י מכונות ונבדקו אקראית ע"י בני-אדם, כאשר הדיוק בסיווג שם עומד על כ-80% לפי מה שכתוב באתר. מבחינתנו דיוק זה אינו מספיק טוב למטרה רגישה כמו סיווג אנשים במצבי מצוקה.
 2. איכות - כיוון שהמידע נלקח מ סרטוני "יו-טיוב" (YouTube), האיכות של מאפייני השמע שם הייתה בפורמט שאינו WAV ולכן היה שם אותנו במצב התחלתי פחות טוב מבחינת היכולת לבדל בין הקלטות שונות. (הסבר על חשיבות הפורמט תינתן בפסקאות הבאות).
 3. פיצורים מוכנים מראש - המאגר ניתן להורדה בפורמט "טנזור-פלו רקורדס" (tensorFlow records) כאשר הפיצורים בו כבר חולצו מקטעי הוידאו. עצם המחשבה על כך שנשתמש בפיצורים מוכנים מראש, מקטעי וידאו ולא אודיו, באיכות שאינה האיכות הכי טובה שניתן למצוא, גרמה לנו סופית להבין שעדיף לנו לחפש מקורות אחרים. חשוב לנו לציין את קיומו של מאגר זה שכן ההתלבטות לגביו הייתה רצינית ושקולה ונמשכה זמן לא מועט.
3. מאגרי המידע שהשתמשנו בהם (לחץ על כל שם לטובת כניסה לאתר):

[FSD](#)

[SHOCKWAVE-SOUND](#)

[Acoustic Event Dataset](#)

[FreeSound](#)

[soundbible](#)

[google AudioSet](#)

בחרנו באתרים הנ"ל מהסיבות הבאות:

1. איכות - קבצי שמע בפורמט WAV.
פורמט זה הוא הפורמט הכי איכותי, כלומר הכי פחות מכווץ, אשר זמין במאגרי מידע קוליים ברשת. [\(לחץ כאן ללינק שמסביר את ההבדלים בין פורמטי wav ל-mp3\)](#)
2. מיקוד - קצר ולעניין.
הקבצים המוצעים הינם בין 5-1 שניות ברובם, כאשר קבצים ארוכים מכך נוחים לסינון אוטומטי כחלק מהממשק המוצע באתרים.
החשיבות בקבצים קצרים הינה קריטית לאלגוריתם שלנו. אנו הרי מחפשים מאפיינים קוליים ייחודיים, כאשר אנו רוצים לאמן מסווג שיזהה מאפיינים אלה. לכן ההשערה שלנו

הינה שכל שנפסק קבצים קצרים ואיכותיים יותר, המסווג שלנו יהיה יעיל יותר בזיהוי, כאשר לקטעים אלה נוכל בהמשך להוסיף רעשי רקע שונים ע"י data-augmentation. (פירוט נוסף על כך בפרק הניסויים).

4. אופן איסוף המידע: (חצי אוטומטי)

[Acoustic Event Dataset](#)

הורדה פשוטה של מאגר שלם עם עשרות קבצים רלוונטיים ("צעקה") בלחיצת כפתור בודדת, רוב הקבצים הלא רלוונטיים נמחקו באופן ידני.

[SHOCKWAVE-SOUND](#)

הורדה של עשרות קבצים ע"י כלים להורדה אוטומטית כגון (לחץ על השם לכניסה ללינק) [Batch Link Downloader](#) [Simple mass downloader](#). הכלים דורשים שהמשתמש ייכנס לדף האינטרנט הספציפי, ורק אז יוכל להוריד ממנו קבצים בפורמט מסוים.

[FreeSound](#) [FSD](#)

בתחילה ניסינו להשתמש ב-API להורדה מסיבית של קבצים איכותיים לפי סינון שקבענו. הדבר אפשרי טכנית, ובילינו מספר שעות בניסיון להשתמש ב-API, אך כשהתקלות הנפוצות הפנו אותנו ללמידת אלגוריתמי OAuth2 (לינק לעמוד ההסבר שעקבנו אחריו), החלטנו שעדיף לא לבזבז זמן נוסף בהבנת פרוטוקולי הזדהות ופשוט להוריד קבצים ידנית. ההחלטה לא הייתה פשוטה שכן המשמעות שלה הייתה שבפועל בילינו שעות בהורדת קבצי שמע, עם זאת אנו שלמים עם ההחלטה שנלקחה בזמנה, שכן היא אפשרה לנו להתמקד בדברים יותר מהותיים מבחינתנו ולהתקדם לשלבים הבאים. יתרון נוסף הוא שהמאגר שלנו מאד איכותי שכן הוא וודא עד רמת הקובץ הבודד על ידינו. (פירוט נוסף על ראייתנו על דליית מידע מהאינטרנט בפרק הסיכום).

5. דגשים על לוגיקת איסוף המידע

בראייתנו, החשיבות **לאיכות** הקבצים הייתה קריטית- לכן נבחר הפורמט wav. בנוסף, הקבצים עצמם היו חייבים להיות כמה שיותר **ממוקדים** על המאפיינים הקוליים שחיפשנו. כך הגענו למסקנה שהקבצים עם הסיווגים החיוביים ("צעקה", "בכיי" וכו') צריכים להיות באורך 3-5 שניות בממוצע. דגימות שליליות כמו רעשי רקע שונים ("רחוב, אוניברסיטה, כביש, חיות וכו') אשר אורך הקלטתם היה מעל לזמן זה, חולקו אוטומטית למקטעים של 5 שניות ע"י הכלי NCH (לחץ לכניסה לאתר הכלי) כדי ליצור אורכי מקטעים אשר תואמים לדגימות החיוביות. המטרה באחידות יחסית של זמני המקטעים נבעה מהרצון לשלול מקרים בהם המסווג יקבל מקטע ארוך יותר ויסיק על סמך כך שהקטע הינו בעל סיווג שלילי.

יצירת מבנה הנתונים (generate data)

1. אלגוריתם:

פונקציה זו קובעת את הפרמטרים שיישלחו לפונקציה עוקבת שתחלץ את הפיצורים ותבצע את העיבוד המקדים לקטעי השמע שהורדו מהאינטרנט. פונקציה זו לוקחת נתיב שבתוכו נמצא עץ תיקיות שבתוכו מסודרים כל קבצי השמע, כאשר התיקיות ממוינות קודם כל לפי סיווג (חיובי/שלילי), ולאחר מכן לפי התווית של הסיווג ("צעקה", "בכי" וכו'). כל קובץ נשלח בתורו לפונקציה העוקבת. חשיבותו של קובץ זה בכך שהינו מעטפת שמאפשרת לפעול בצורה מודולרית על כל קובץ שמע בנפרד, כך שכל הקבצים יעובדו באותו אופן (לדוגמא- נקלוט מכל קובץ שמע זמן מקסימלי של 5 שניות ראשונות).

2. פרטים טכניים יבשים:

קובץ	הצהרה	קלט	פלט	הערות
evaluate_screamClf.py	def create_csv():		קובץ CSV עם מבנה הנתונים הרצוי ייפלט לנתיב היחסי (לדוגמא) data.csv	משתמש במשתנה גלובלי מהמחלקה class global_For_Clf() שמספק פרטים חשובים (יפורט באלגוריתם) הפונקציה נכתבה על ידינו.

עיבוד מידע מקדים (pre-process)

1. אלגוריתם:

פונקציה זו מבצעת עיבוד מקדים לקבצי השמע טרם דליית הפיצורים ע"י כך שהיא מקצה זמן מקסימלי לכל קובץ שמע (5 שניות כהגדרת ברירת מחדל), בודקת שהוא תקין, וקובעת קצב שבו יידגמו קבצי השמע (22050 הרץ).

2. פרטים טכניים יבשים:

קובץ	הצהרה	קלט	פלט	הערות
evaluate_screamClf.py	extract_feature_to_csv(wav_path, label, data_file_path, min_wav_duration, fcc_amount):	קובץ שמע	הוספת שורה לטבלת מבנה הנתונים	הפונקציה נכתבה על ידינו.

יצירת וקטור התכונות (extract features)

1. אלגוריתם :
פונקציה זו מוציאה את הפיצ'רים מקובץ השמע, אשר בשלב זה כבר נמצא לאחר עיבוד מקדים ולכן הינו באורך של 0.5-5 שניות.
2. הפיצ'רים :
ראשית יש לציין כי כדי לבחור באיזה פיצ'רים ראשוניים להשתמש, היינו צריכים לרכוש הבנה בנושא שלא הכרנו עד כה – צלילים ותנועתם באוויר, אפיוניהם השונים של הצלילים, והדרכים המומלצות להבדלה בין צלילים שונים. רק על הנושא הנ"ל ניתן לחקור חיים שלמים, כך שהגבלנו את עצמנו לרמת הבנה מסוימת שמצד אחד תאפשר לנו לבחור באופן משכיל פיצ'רים, ומנגד לא תעצור את הפרויקט שלשמו התכנסנו.
להלן רשימה של הפיצ'רים שבחרנו להשתמש בהם, כאשר מתחת לכל פיצ'ר מתומצת האופן שבו הוא עשוי לתרום להשגת המטרה שלנו, ולאחר מכן משפט קצר שמסביר מה הפיצ'ר הזה מציג. לאחר פירוט הפיצ'רים צירפנו מספר גרפים בסיסיים שנועדו אך ורק לתת אינטואיציה כללית על הנושא של אפיון צלילים שונים.
נדגיש כי בפרק הניסויים נפרט באופן מקיף על השפעות הפיצ'רים והשפעת שינויים על איכות המסווג

1. spectral centroid
תורם בהבדלה בין בני אנוש לחיות, ולהבדלה בין בני אדם שונים. מראה את הטווח שבו נמצא מרכז המסה של הצליל.

2. zero-crossing rate
תורם בזיהוי "דיבור" של בני אנוש והבדלתו ממוזיקה. מראה כמה פעמים עצמת הקול חוצה את קו האפס (שמייצג מחסור בצליל).

3. roll-off frequency
סוג של "חתימה קולית". מייצג את התדר ש-85% מכלל התדרים באותו הרגע נמצאים בו או מתחתיו.

4. Chromagram
תורם באבחנה בין סוגי מוזיקה, ובכך יסייע בעקיפין בהבדלת רעשי רקע מדיבור של בני אדם. מייצג את הגובה של הצליל.

5. Mel-frequency cepstral coefficients (MFCCs)
תורם באבחנה בין צלילים של בני אנוש לצלילי רקע, ותורם באבחנה בין אותיות שונות שבני אדם מבטאים. מציג את המנעד הקולי. זהו בעצם ייצוג קצר-טווח של מנעד הצליל (ריכוז האנרגיה של הצליל).

6. root-mean-square (RMS)

תורם באבחנה בין קטעי שמע שונים. ערך שמייצג את חציון עצמת האנרגיה של קטע השמע.

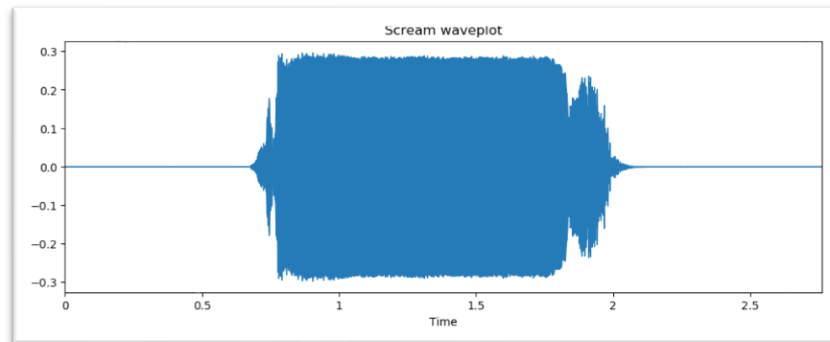
7. mel-scaled spectrogram

תורם בדומה לנקודה 5, אך בייצוג מתמטי שונה. כלי נוסף שיעזור לנו להבדיל בין קטעי שמע שונים.

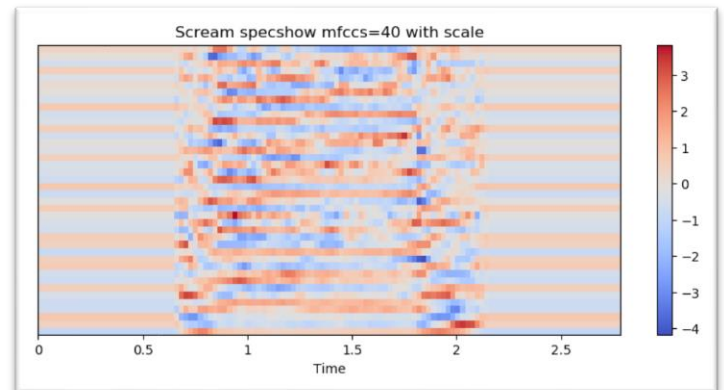
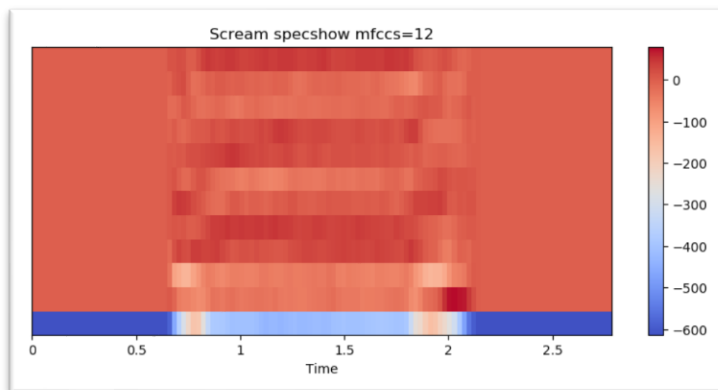
3. המחשות ויזואליות של הפיצורים על קובץ שמע של צעקה :

- אין להתייחס לגרפים הבאים בתור "גרף מדעי איכותי שמסביר את עצמו", שכן מטרת הגרפים בהמשך נקודה זו הינה רק לצורך אינטואיציה כללית.

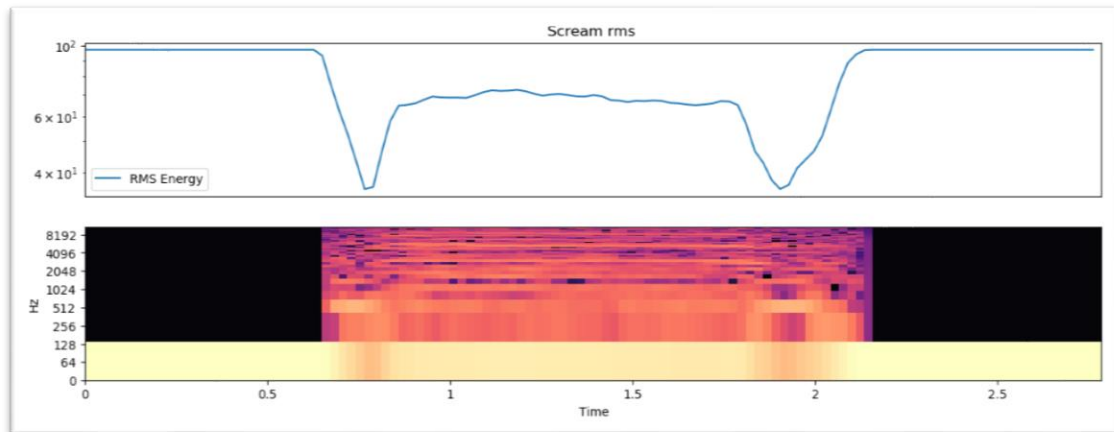
1. מטרת הגרף הבא היא לתת אינטואיציה לכך שניתן להסתכל על עצמת צליל כעל **המרחק** שהצליל גורם לחלקיק בודד של אוויר לנוע ממקומו "קדימה" עד שהוא שב למיקומו המקורי. בגרף הבא ניתן להסיק כי "צעקה" מאופיינת בכך שבמשך רוב הצליל הנפלט- החלקיק מוזז מאד ממקומו המקורי, ולמשך רוב הצעקה.



2. מטרת הגרפים הבאים היא לתת אינטואיציה לכך שגם בניתוח צליל ניתן לרדת ל"רזולוציות הגדלה" שונות (בהקבלה למיקרוסקופ- עד כמה להגדיל את מה שרואים). את הפיצור MFCC ניתן לכוון לרמות רזולוציה שונות. ב-2 הגרפים הבאים, מוצגות שתי רמות רזולוציה שונות. הימני ברמה של 40, והשמאלי ברמה של 12. ניתן לראות בבירור כי ככל שרמת הרזולוציה עולה- המשבצות בגרף נהיות קטנות יותר- כלומר יורדים יותר לפרטים (בהקבלה למיקרוסקופ- חוסר ירידה לפרטים יהווה צפיה בעור ובשערות שעליו, בעוד שירידה נוספת לפרטים תאפשר להסתכל על מיקרובים או חיידקים קטנים). בפרק הניסויים נפרט על חשיבות כיוון הרזולוציה בפיצור זה ועל המשמעויות שלו לאימון המסווגים.



3. מטרת הגרפים הבאים הינה להראות באופן ויזואלי עוד ייצוג של "צעקה". הגרף התחתון מייצג ספקטוגרמה, וניתן להבחין כי בכל שניה קיימים תדרים שונים שמושמעים באותו הזמן, עם זאת קיים מבנה תדרים מסוים שחוזר על עצמו במשך הצעקה. בנוסף בגרף העליון ניתן לראות כי האנרגיה גם כן משתנה בצורה מסוימת לאורך הצעקה (נבחר שלא להיכנס בשלב זה לעומק הגרפים).



4. פרטים טכניים יבשים :

קובץ	הצהרה	קלט	פלט	הערות
evaluate_screamClf.py	extract_feature_to_csv(wav_path, label, data_file_path, min_wav_duration, fcc_amount):	קובץ שמע	הוספת שורה לטבלת מבנה הנתונים	הפונקציה נכתבה על ידינו. הספרייה העיקרית שאנו נעזרים בה בפונקציה זו נקראת Librosa

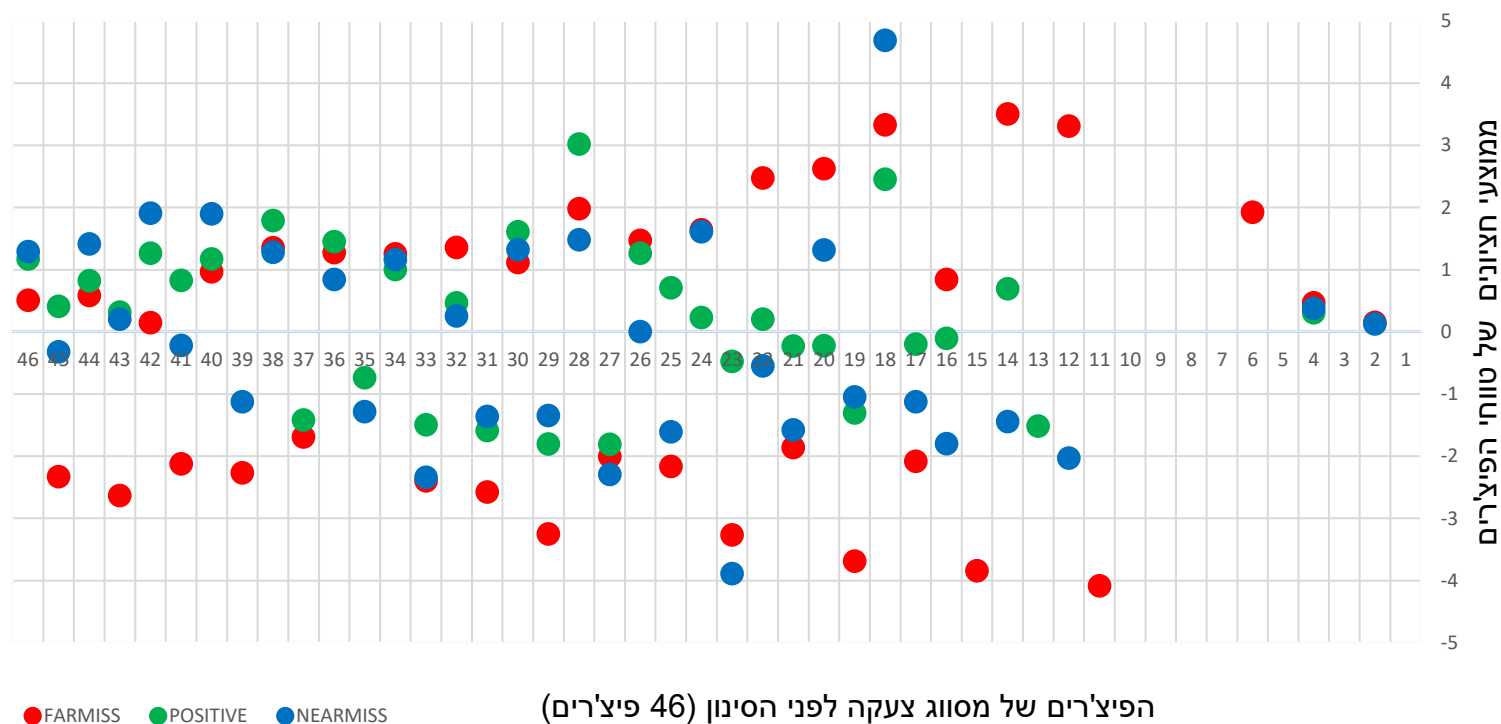
נתונים גולמיים על מאגר המידע

הערה: הנתונים הגולמיים על מאגר המידע אינם "רכיב ישיר במערכת", ולכן מיקומו של פרק זה יכול להימצא בפרק שנמצא מחוץ לרכיבי המערכת. עם זאת אנו בחרנו למקמו כאן כדי שיופיע לפני פרק האימון, בחירת הפיצ'רים והערכת הביצועים, כדי שלקורא תהיה כבר היכרות עם המידע שעבדנו עימו וכך נקל על הקוהרנטיות של קריאת המסמך, ובפרט עם ההחלטות שנפרט בהמשך לגבי רכיבי המערכת.

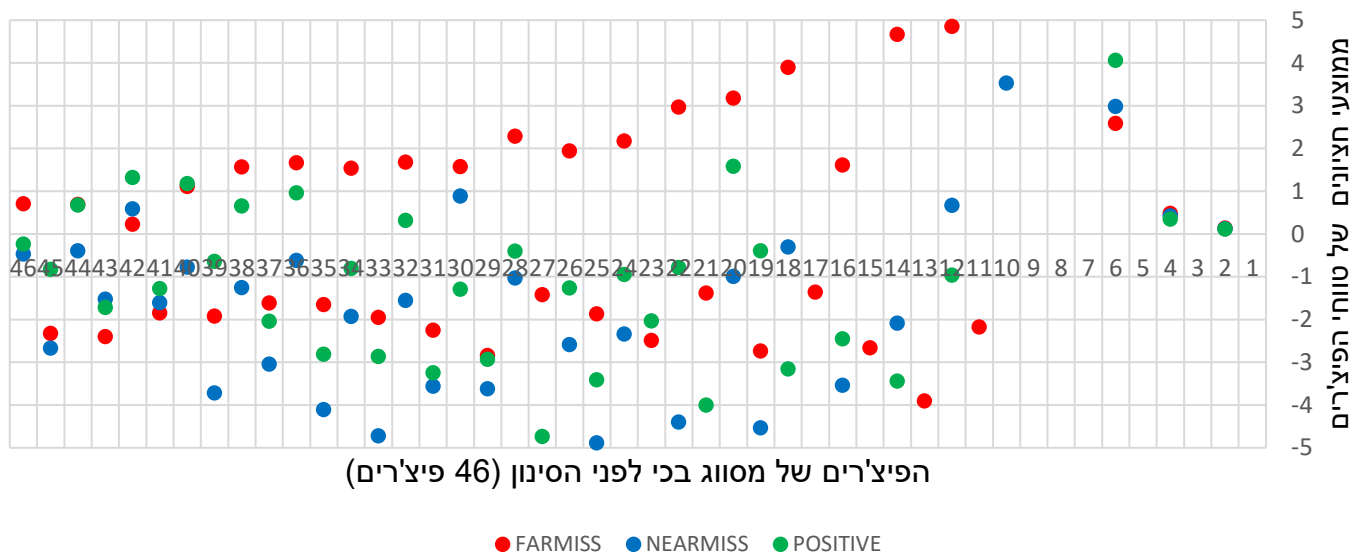
ניתוח ראשון של הנתונים הגולמיים

מטרה: מהם הפיצ'רים המועילים להבחנה בין הסיווגים השונים?
תוצאת הגרף (ציר Y מנורמל):

ממוצעי חציונים של הפיצ'רים של מסווג צעקה מחולקים לפי סוג הלייבלים



ממוצעי חציונים של הפיצ'רים של מסווג בכי מחולקים לפי סוג הלייבלים



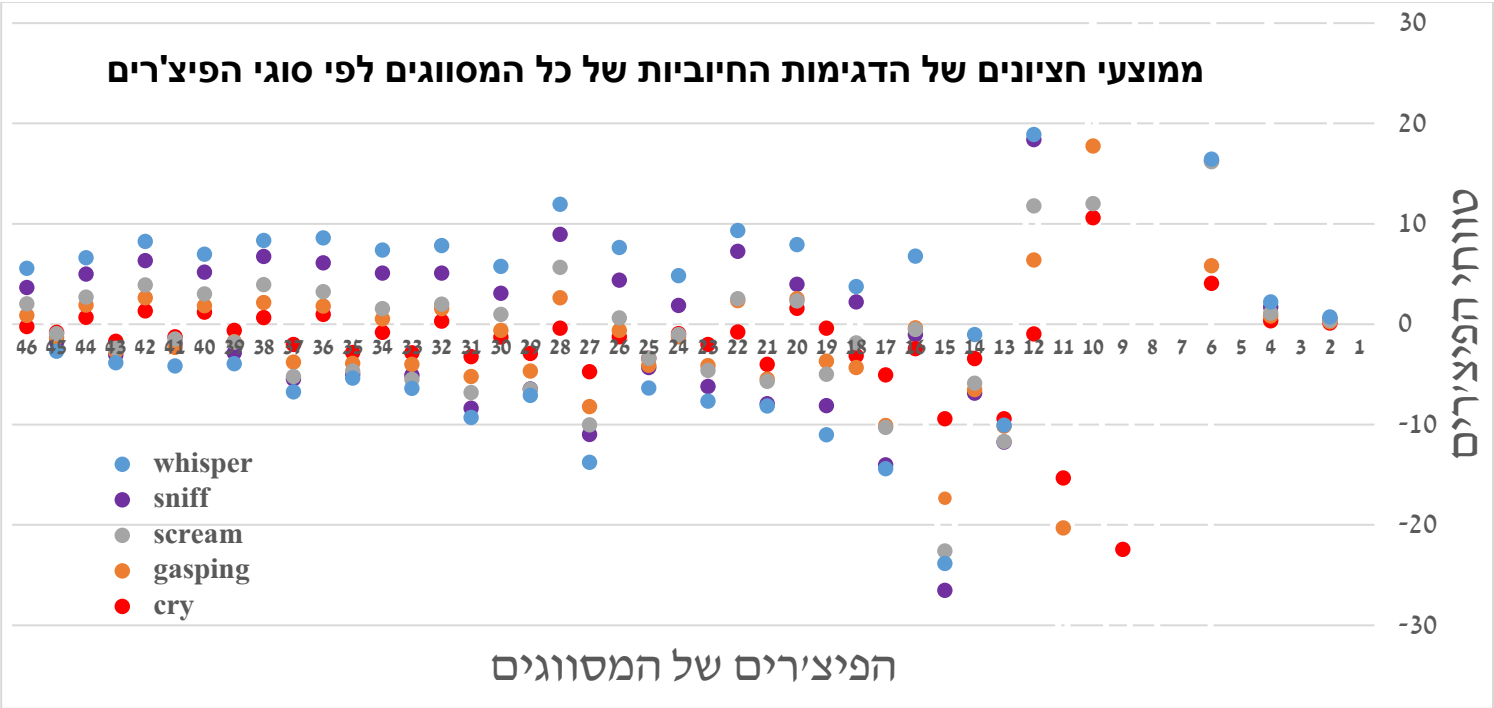
מסקנות :

ניתן לראות שבעוד שבחלק מהפיצ'רים קשה להבחין בין תכונה של קטע צעקה לעומת קטעים בהן אין צעקה כלל (FARMISS) או קטעי קול הדומים לצעקה, אך אינם צעקה (NEARMISS), ברוב הפיצ'רים ניתן להבדיל בין הסוגים השונים ואת הפיצ'רים הללו נבחר בהמשך מתוך כלל הפיצ'רים בכדי לאמן את המסווג (בדיעבד אלגוריתם אוטומטי בחר אותם בשבילנו). כמו כן, ניתן לראות קרבה מסוימת בין ממוצעי הערכים בפיצ'רים של קטעי הקול האמיתיים (POSITIVE) ל-NEARMISS, לבין אלו של FARMISS- הפיצ'רים הללו יעזרו לנו לחדד את ההבדלים בין הדגימות החיוביות לשליליות ולשפר את אחוזי ההצלחה שלנו.

ניתוח שני של הנתונים הגולמיים

מטרה : הבנת הבדלי חציונים בין כלל הסיווגים החיוביים שלנו

תוצאת הגרף :



מסקנות :

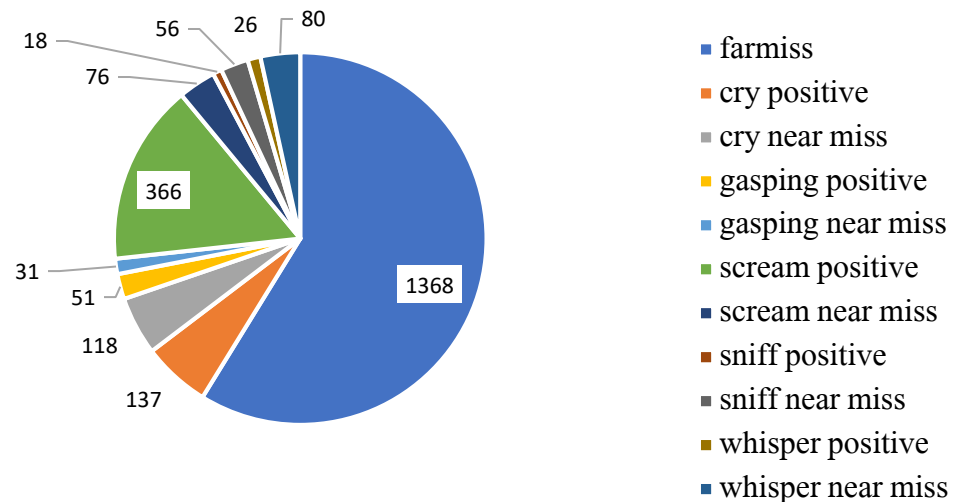
בנוסף לאימון מסווגים ע"י דוגמאות שליליות של אותו מאפיין (FARMISS, NEARMISS), ניתן להשתמש בדגימות החיוביות משאר המסווגים על מנת לשפר את המסווג. ניתן לראות מהגרף כי בפיצ'רים מסוימים ישנם הבדלים משמעותיים בין דגימות חיוביות של מסווג אחד לאחר. שימוש בדגימות חיוביות של מסווגים אחרים בכל מסווג כחלק מתהליך אימון המסווג יאפשר לנו לחדד את ההבדלים בין המאפיינים השונים ולא לסווג לדוגמה התנשפות בתור צעקה.

ניתוח רביעי של הנתונים הגולמיים

מטרה : להציג כמויות ופילוח של המידע שאספנו

תוצאת הגרף :

פילוח כלל הדגימות בחלוקה לפי סיווגם



מסקנות :

התרשים מצביע על כמות גדולה של דגימות רחוקות ביחס לדגימות חיוביות ודגימות קרובות עבור כל מסווג. ריבוי הדגימות השליליות מאפשרות לנו לשרטט מעין חוצץ בינם לבין הדגימות החיוביות והקרובות ובכך לשפר את אחוזי הדיוק שלנו. בעוד שבמסווגים מסוימים כדוגמת מסווג צעקה ומסווג בכי, יש לנו די דגימות על מנת להבחין באיכות גבוהה בין קטע קול שיש בו צעקה או בכי או כזה שלא, הדגימות היחסית מועטות במסווגים של התנשפות, לחישה ורחרוח (SNIFF) גורמות לנו לשרטט קו יותר כללי של התכונות הללו שפחות יבדיל בדקויות מסוימות.

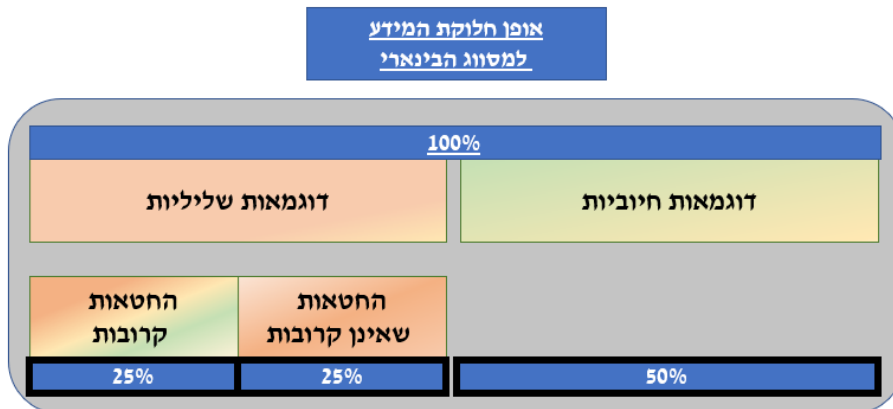
בנוסף לזאת, ככל שכמות הדגימות יותר גדולה כך המרווח שלנו לבצע בדיקות הנוגעות לכמות הדגימות שיש לאסוף בכדי להגיע לאחוז דיוק גבוהה בחיזוי ולא לסיטואציה של התאמת יתר (over-fitting).

בפרק הניסויים ניווכח כי לא מתקיים אצלנו over-fitting וכי למעשה על אף כמות הדגימות שאספנו- כדי לשפר את אחוזי הדיוק עלינו להגדיל את מאגר המידע של הדגימות החיוביות. עוד על כך בפרק הסיכום.

אימון (train)

1. המידע בתהליך האימון :

1. דימוי סט האימון לעולם האמיתי : נתקלנו באתגר ביצירת מאגר אימון שישקף את יחס ה"סיווג חיובי : סיווג שלילי בעולם האמיתי", כיוון שאיננו מתעסקים במפעל שבו ידוע כי כמות המוצרים הפגומים הינה X ולכן יש ליצור מאגר נתונים עם אחוז זהה של מוצרים פגומים, אלא אנו מתעסקים בזיהוי של מאפיין קולי של מצוקה. בהינתן מצב זה, החלטנו כי נאמן את המסווג על יחס של 1:1 כדי למנוע "הטיה" של המסווג לכיוון מסוים על בסיס הסקה סטטיסטית. לדוגמא : במסווג הבינארי שמייצג "צעקה", בסט האימון על כל דגימה חיובית של צעקה, תהיה גם דגימה שלילית של צעקה. (עוד על החלטה זו ומשמעויותיה בפרק הניסויים והסיכום).
2. שימוש בהחטאות קרובות (Near Misses) : החטאה קרובה מבחינתנו הינה סיווג שלילי, אך כזה שנמצא קרוב מאד במאפייניו לסיווג החיובי. לדוגמא : במסווג "צעקה", החטאה קרובה יכולה להיות "צעקה של שמחה והתרגשות", או "קול של חיה שנשמע כמו צעקה של אדם", או "שאגה של מפלצת". אכן קולות אלה הינם צעקות אשר נשמעים יחסית דומה לצעקה שאנו מחפשים, אך היא אינה צעקה במובן של "צעקת מצוקה". לכן החלטנו שמתוך כל הסיווגים השליליים, 50% יהיו "החטאות קרובות", כאשר שאר הדוגמאות השליליות יילקחו בצורה רנדומלית מתוך מאגר הדגימות השליליות. (ההמלצה לשימוש בהחטאות קרובות הגיעה ממרצה הקורס (פרופ' שאול מרקוביץ). (עוד על ההחלטה לשימוש באחוזים אלה ומשמעויותיה בפרק הניסויים והסיכום)



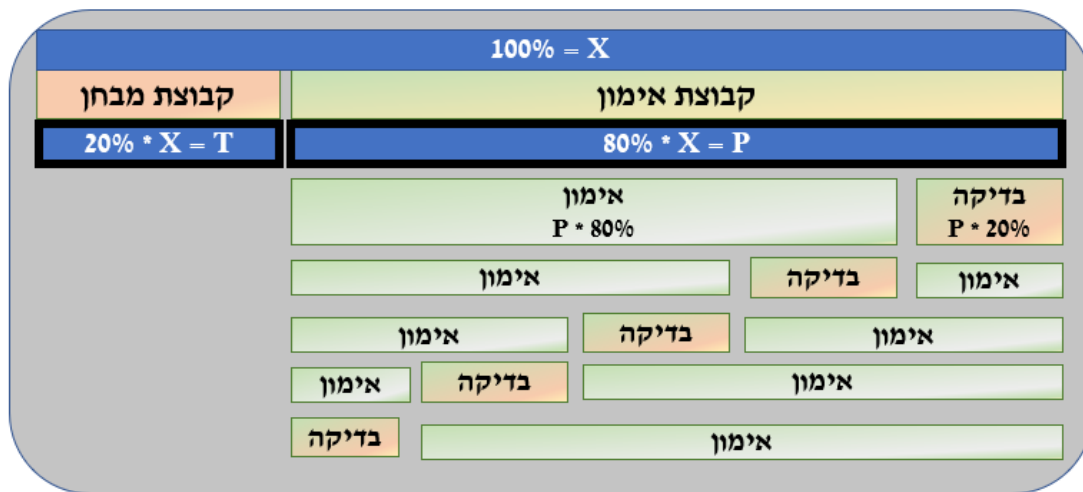
3. נרמול המידע : עקב הבדלים בסדרי הגודל בין הערכים המספריים של הפיצורים השונים, היה דרוש לנרמל את המידע. כיוון שנתקלנו בגישות שונות לגבי האופן שבו דרוש לנרמל את המידע, החלטנו לקבל חוות דעת גם מעמיתים ומנחים. לבסוף החלטנו כי הנרמול עבור קבוצות האימון התבצע באופן הבא : כיוון שהשתמשנו ב**בוריאציה שלנו על-stratified repeated k-fold** , בכל פעם שקיבלנו חלוקה של קבוצת אימון וקבוצת בדיקה (לפעמים נקראת גם ולידציה או מבחן), השתמשנו ב**כלי אוטומטי שמנרמל את המידע על קבוצת האימון**. כלי זה שומר את הערכים שעל בסיסם התבצע הנרמול, ועל בסיס ערכים אלה נרמלנו את **קבוצת הבדיקה**. (ההמלצה לנרמל כך את

המידע התקבלה מתגובות [לפוסט](#) שפרסמנו. מצ"ב בנספחים). בפרק החיזוי נתייחס לכך שגם שמרנו את אובייקט הנירמול.

2. תהליך האימון :

האימון בוצע **בנפרד עבור כל מסווג בינארי**, כלומר אימון מסווג ל"צעקה", ל"בכיי" וכו' הינם תהליכי אימון **נפרדים אך זהים**. כל מסווג פוטנציאלי קיבל מידע באופן שפורט בנקודה הקודמת. כחלק מהתהליך פיצלנו את המידע לקבוצות של אימון ובדיקה, ולאחר מכן שלחנו את קבוצת הבדיקה לעוד פיצולים בהתאם לאלגוריתם [stratified k-fold](#)

אופן פיצול המידע למסווג הבינארי



3. המסווגים בתהליך האימון :

בחרנו לעבוד עם מסווגים בינאריים מתוך הנחה שע"י פיצול השאלה שלנו ל"תתי מחלקות אובייקטיביות" (הוסבר בפרק תיאור הבעיה) אנחנו בעצם מספקים מידע נוסף שבאמצעותו נגיע לתוצאות מדויקות יותר (עוד על כך בפרק הניסויים והסיכום). המסווג שנבחר לאחר תהליך של מחקר ובחינת מסווגים שונים הינו מהספרייה [keras](#), מסוג [Sequential](#). (עוד על הסיבות לכך- בפרק הערכת המסווגים)

4. הערות לתהליך האימון :

1. תהליך האימון והערכת המסווגים אצלנו כרוכים זה בזה ולכן פרטים נוספים שאולי

מבחינת הקורא שייכים לפרק זה, יימצאו בפרק על הערכת המסווגים.

2. יש לציין כי במסווגים מסוימים, לא הגענו להתכנסות בתהליך האימון שלהם עקב

מגבלות של כוח מחשוב- עוד על כך בפרק הערכת המסווגים והסיכום.

3. הקורא יכול לשאול את עצמו "מדוע בפרויקט זה לא בחרו לעבוד עם מודלים שלומדים

על התמונות שמייצגות את קטעי הקול, כפי שהוצגו בפרק על יצירת וקטור התכונות?".

התשובה לכך היא שעל סמך דעות שונות שקראנו באינטרנט, מאנשים שביצעו בדיקות

בדיוק על נושא זה, הגיעו למסקנות שאינן חד משמעיות- כלומר אין יתרון חד משמעי

לעבודה על התמונות, ולכן מבחינתנו זוהי הייתה פשוט החלטה שהסתמכה על "במה

יותר מעניין אותנו להתעמק בפרויקט זה ; בנתונים של קבצי שמע כנתונים מספריים

שעליהם נוכל לבצע מניפולציות שונות? או בתחום עיבוד התמונה?" וכיוון שנושא "עיבוד הצלילים וניתוחם" היה מספיק מענין ורחב בשבילנו, בחרנו שלא להתרחב לתחום "עיבוד התמונה".

5. פרטים טכניים יבשים :

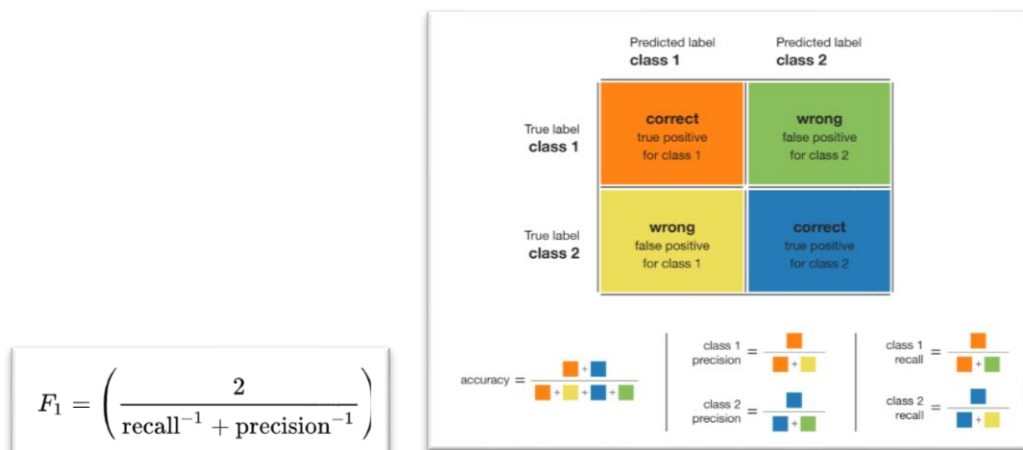
קובץ	הצהרה	קלט	פלט	הערות
evaluate_screamClf.p y	def get_stratified_results(k, X_for_k_fold, y_for_k_fold, IntPositive, screamGlobals):	סט אימון עם סיווגים וקלטים היקפיים נוספים	מסווג מאומן	הפונקציה נכתבה על ידינו. נציין כי הפונקציה שונתה הספריות העיקריות שאנו נעזרים בה בפונקציה : sklearn keras

בחירת פיצ'רים (feature selection)

בפרק "נתונים גולמיים על מאגר המידע" הצגנו גרפים על הטווחים והחציון של כל פיצ'ר, ובפרק "הניסויים" נציג ניסוי שמראה כיצד בחירת פיצ'רים שונים משפיעה על תוצאות הדיוקים שלנו.

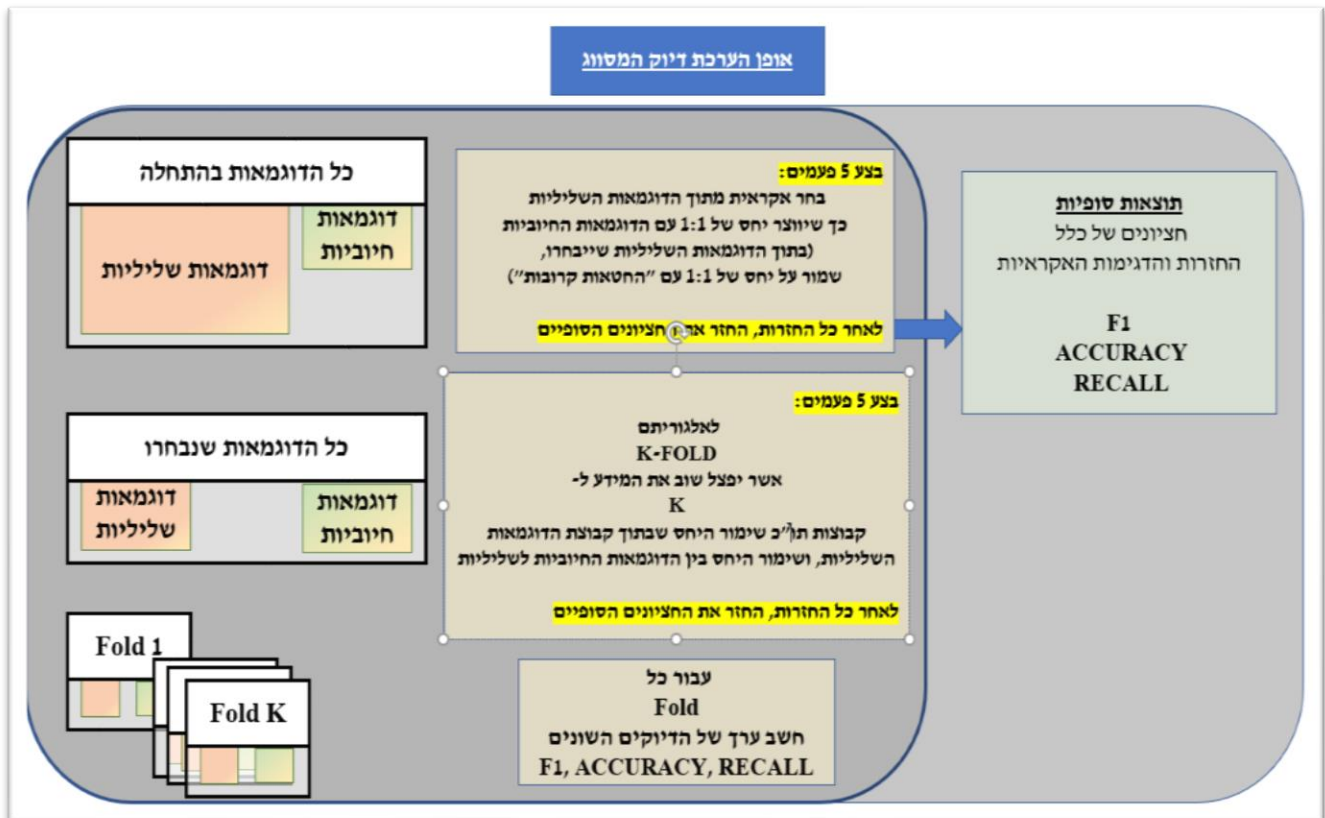
הערכת המסווגים (evaluate)

1. חשיבות הגדרת אופן "הערכת התוצאות" של המסווגים השונים :
 בהתחלה הגדרנו את המסווג הכי טוב בתור המסווג שקיבל את אחוז הדיוק הכי גבוה, אך לאחר שקראנו [כתבה מעולה ב-Towards Data Science](#) (ראה תמונה מהכתבה בסוף פסקה זו, [הנוסחה לחישוב נלקחה מויקיפדיה](#)) הבנו שיש מדד לא פחות חשוב שלמדנו בקורס המבוא של בינה מלאכותית, אך עד אותו הרגע התעלמנו מקיומו- מדד אחוזי ה-recall. מדד זה עוזר לנו להבין את יכולתו של המסווג שלנו לענות על דוגמאות שהסיווג האמיתי שלהן הינו "צעקה", בצורה נכונה. חשיבותו של המדד נובעת מכך שעלות סיווג של "צעקת מצוקה" בתור אזהקת שווא, גבוהה בהרבה מעלות סיווג "רעש שאינו צעקת מצוקה בתור צעקת מצוקה". כלומר אנחנו מוכנים לספוג נטייה מסוימת לכיוון של "אזהקות שווא", במידה והדבר יקנה לנו אחוזי דיוק גבוהים יותר ב"זיהוי צעקות מצוקה". כיוון שבסופו של דבר גם אזהקות שווא חשובות בעינינו, ניתן להסתכל על מדד ה-f1, אשר עושה שיקלול גם לאחוזי הדיוק הכלליים, וגם ל-recall. לכן בסופו של דבר בחרנו להסתכל על מדד ה-F1 עבור המחלקה "צעקה" (עבור מסווג הצעקה הבינארי) בתור המדד המכריע של הערכת המסווגים.



$$F_1 = \left(\frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} \right)$$

2. הערכה על בסיס בדיקות חוזרות ונשנות של מידע אקראי: כדי להקל על הקורא, להלן תרשים של התהליך שבו נקטנו כדי להעריך את תוצאות המסווגים שבדקנו. במידה והתרשים אינו מפורט מספיק, תוכלו לעקוב אחרי ההסבר המפורט שמתחתיו, אשר כולל גם הסברים לסיבה שבגללה זה התהליך שבנינו.



כל ההערכות שלנו למסווגים מתבססות על לקיחת ערך החציון של הדיוקים מתוך הרצות חוזרות ונשנות לכל מסווג. כיוון שכמות הדגימות החיוביות שלנו הייתה נמוכה מכמות הדגימות השליליות (ביחס כללי של כ- 1:4 לטובת הדגימות השליליות), כל תוצאה סופית על דיוק של מסווג כללה בתוכה מעל ל- 5 הרצות של דגימות אקראיות בהן נלקחה קבוצת דגימות שליליות שגודלה היה כגודל הקבוצה עם הסיווגים החיוביים. מעבר לכך, את קבוצת הדגימות החיוביות והשליליות, שהיו כעת ביחס של 1:1, שלחנו ל-5 חזרות שונות של תוצאות שהתקבלו מאלגוריתם [stratified k-fold](#), כאשר אלגוריתם זה בעצמו סיפק לנו חציון מתוך חמשת ההרצות שהוא ביצע. (משמעות נוספות מתהליך זה וערכי החזרות והאקראיות שלו בפרק הניסויים והסיכום).

הערה על כמות החזרות - עקב מגבלות חישוביות ותוצאות שקיבלנו, לא העלנו מעבר לכך את כמות החזרות כיוון שבהעלאתן המחשב לא החזיר לנו תוצאות בזמנים סבירים (גם לא במשך לילה שלם). התייחסות נוספת לכוח מיחשוב תינתן בסיכום).

3. המודלים שביצעיהם נמדדו: על מנת למטב את היכולת שלנו להגיע לתוצאות מדויקות, השתמשנו בספריית Keras – ספריה אשר עוטפת את tensor-flow flow כך שהשימוש בה נהיה נוח יותר. ספריה זו מאפשרת

למידה וניתוח של קטעי אודיו, תמונה וטקסט ע"י המסווגים שהיא מאפשרת לאמן. השתמשנו באפשרויות המתקדמות שהיא מציעה בכדי לשפר את יכולת הניבוי של כל אחד מהמסווגים שלנו.

נתאר להלן את תהליך הלמידה שלנו על המודל הסדרתי של קראס, כאשר בניסוי 5 נבצע השוואה של מודל הסדרתי לעומת ספריות למידה אחרות בכדי להיווכח בנכונות הבחירה בספרייה זאת.

המודל הנבחר - המודל הסדרתי של "קראס"

המודל הסדרתי הוא מחסנית לינארית של שכבות

אפשר ליצור מודל סדרתי ע"י העברת רשימה של שכבות לבנאי של המודל

לדוגמא :

```
from keras.models import Sequential
from keras.layers import Dense, Activation

model = Sequential([
    Dense(32, input_shape=(784,)),
    Activation('relu'),
    Dense(10),
    Activation('softmax'),
])
```

לחלופין אפשר בפשטות להוסיף שכבות ע"י שימוש במתודה ADD :

```
model = Sequential()
model.add(Dense(32, input_dim=784))
model.add(Activation('relu'))
```

פירוט של צורת הקלט :

המודל צריך לדעת לאיזה סוג של קלט הוא צריך לצפות. מסיבה זו השכבה הראשונה (ורק הראשונה כי כל השכבות עושות התאמת קלט אוטומטית) צריכה לקבל מידע על סוג הקלט. ישנם מספר אפשרויות לעשות את זה :

- להעביר ארגומנט של `input_shape` לשכבה הראשונה-זהו טאפל של מספרים שלמים או של NONE כאשר כל NONE מציין מספר חיובי שעתיד להיות בתא זה.
- שכבות דו ממדיות כמו DENSE שתומכות בספציפיקציה לסוג הקלט בעזרת הארגומנט `input_dim` או שכבות תלת ממדיות דרך `input_dim` ו-`input_length`.
- אם צריך לציין גרסה שעתידה להשתנות לקלט שלנו אפשר להעביר את הארגומנט של `batch_size` לשכבה. אם מעבירים `batch_size=32` and `input_shape=(6, 8)` אז המודל יצפה לכל סוג של קלט שיהיה לו את סוג הקלט מצורת `(32, 6, 8)`.

קומפילציה

לפני שמאמנים את המודל צריך להגדיר את תהליך הלמידה, זאת נעשה דרך המתודה COMPILE שמקבלת שלושה ארגומנטים :

- אופטימיזר : זה יכול להיות מזהה מחרוזת של אופטימיזר קיים כמו `rmsprop` OR `adagrad` או מופע של מחלקת אופטימיזר.

אנחנו נשתמש במודל ADAM מאחר והוא האופטימיזר שלומד הכי מהר ביחס לשאר הוא הכי יציב ולא סובל מירידות משמעותיות ברמת הדיוק שלו בהתאם לעלייה בכמות המידע שהוא "מעכל" לאורך זמן (לאור העובדה שהוא מתקן את עצמו ככל שיינתן לו יותר זמן לעבוד) ההגדרה היחידה שנרצה להסתכל עליה בהקשר של ADAM הוא קצב הלמידה-LR אבל הקצב הדפולטיבי המוגדר -0.001 הוא זה שבהינתן גודל מודל מספק את התוצאות הטובות ביותר ולכן לא נשנה אותם (כל הגדלה של קצב הלמידה הגדילה את זמן האימון ולכן לא נגדיל יותר מערך זה).

- פונקציית תמצות- תפקידה הוא לצמצם את המידע של המודל. כאן יכולה להיות פונקציית תמצות מוכרת כמו `mse` OR `categorical_crossentropy` או בפונקציית תמצות אחרת (<https://keras.io/losses/>)
אנחנו נשתמש ב- `binary_crossentropy` היות והסיווג שאנו מחפשים להשיג הוא סיווג בינארי מהסוג של או שזה צעקה או שלא, או שזה בכי או שזה לא וכו'.

- רשימה של מדדים. לכל בעיית אימון צריך לסדר את המדד להיות : `metrics=['accuracy']`. מדד יכול להיות מחרוזת של מזהה של מדד מוכר או של פונקציית מודד שבנינו.

```
# For a multi-class classification problem
model.compile(optimizer='rmsprop',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# For a binary classification problem
model.compile(optimizer='rmsprop',
              loss='binary_crossentropy',
              metrics=['accuracy'])

# For a mean squared error regression problem
model.compile(optimizer='rmsprop',
              loss='mse')

# For custom metrics
import keras.backend as K

def mean_pred(y_true, y_pred):
    return K.mean(y_pred)

model.compile(optimizer='rmsprop',
              loss='binary_crossentropy',
              metrics=['accuracy', mean_pred])
```

אימון

מודלים של קראס מאומנים על מערכים של Numpy המכילים מידע ותווית האימון (הלייבל). בשביל לאמן מודל צריך להשתמש בפונקציית `.fit`. (לקריאת הדוקומנטציה של `.fit`: <https://keras.io/models/sequential>)

```
# Train the model, iterating on the data in batches of 32 samples
```

```
model.fit(data, labels, epochs=10, batch_size=32)
```

```
# For a single-input model with 10 classes (categorical classification):
```

```
data = np.random.random((1000, 100))
```

```
labels = np.random.randint(10, size=(1000, 1))
```

```
# Convert labels to categorical one-hot encoding
```

```
one_hot_labels = keras.utils.to_categorical(labels, num_classes=10)
```

```
# Train the model, iterating on the data in batches of 32 samples
```

```
model.fit(data, one_hot_labels, epochs=10, batch_size=32)
```

מודל שכבות סדרתי מוערם

במודל הזה אנו עורמים שלוש שכבות אחת על השנייה כדי לגרום למודל להיות מסוגל ללמוד ברמה גבוהה של ייצוגי ביניים

בעוד ששני השכבות הראשונות מחזירות את הפלט הסדרתי המלא של המודל השכבה השלישית מחזירה את הצעד האחרון בפלט הסדרתי ולמעשה שומטת את כל ממדי הביניים-כלומר הופכת את הפלט לוקטור יחיד.

```
from keras.models import Sequential
from keras.layers import LSTM, Dense
import numpy as np

data_dim = 16
timesteps = 8
num_classes = 10

# expected input data shape: (batch_size, timesteps, data_dim)
model = Sequential()
model.add(LSTM(32, return_sequences=True, input_shape=(timesteps, data_dim))) # returns a sequence of vectors
of dimension 32
model.add(LSTM(32, return_sequences=True)) # returns a sequence of vectors of dimension 32
model.add(LSTM(32)) # return a single vector of dimension 32
model.add(Dense(10, activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='rmsprop', metrics=['accuracy'])

# Generate dummy training data
x_train = np.random.random((1000, timesteps, data_dim))
y_train = np.random.random((1000, num_classes))

# Generate dummy validation data
x_val = np.random.random((100, timesteps, data_dim))
y_val = np.random.random((100, num_classes))

model.fit(x_train, y_train, batch_size=64, epochs=5, validation_data=(x_val, y_val))
```

מודל שכבות סדרתי מוערם רב מצבי

במודל זה מצבי הביניים נוצרים רק לאחר עיבוד של מקבץ של דוגמאות בתור מצבים התחלתיים על מנת שנוכל ליצור מהם את המצבים הבאים. זה מאפשר לעבד יותר סדרות של מידע ולשמור על הסיבוכיות החישובית בצורה שהיא ניתנת לביצוע

```
from keras.models import Sequential
from keras.layers import LSTM, Dense
import numpy as np

data_dim = 16
timesteps = 8
num_classes = 10
batch_size = 32

# Expected input batch shape: (batch_size, timesteps, data_dim)
# Note that we have to provide the full batch_input_shape since the network is stateful.
# the sample of index i in batch k is the follow-up for the sample i in batch k-1.
model = Sequential()
model.add(LSTM(32, return_sequences=True, stateful=True,
              batch_input_shape=(batch_size, timesteps, data_dim)))
model.add(LSTM(32, return_sequences=True, stateful=True))
model.add(LSTM(32, stateful=True))
model.add(Dense(10, activation='softmax'))
model.compile(loss='categorical_crossentropy',
              optimizer='rmsprop',
              metrics=['accuracy'])

# Generate dummy training data
x_train = np.random.random((batch_size * 10, timesteps, data_dim))
y_train = np.random.random((batch_size * 10, num_classes))

# Generate dummy validation data
x_val = np.random.random((batch_size * 3, timesteps, data_dim))
y_val = np.random.random((batch_size * 3, num_classes))

model.fit(x_train, y_train, batch_size=batch_size, epochs=5, shuffle=False, validation_data=(x_val, y_val))
```

השכבות של קראס

לכל השכבות של קראס יש מספר מתודות משותפות:

- `layer.get_weights()` : מחזירה את המשקל של השכבה בתוך מערך Numpy
- `layer.set_weights(weights)` : מסדרת את המשקלים של השכבה מרשימה של מערכים (באותו סדר גודל כמו של הפלטים של `get_weights`)
- `layer.get_config()` : מחזיר מילון המכיל את ההגדרות של השכבה. השכבה יכולה להיות מוגדרת מ-CONFIG בצורה הבאה:

```
layer = Dense(32)
config = layer.get_config()
reconstructed_layer = Dense.from_config(config)
```

Or:

```
from keras import layers
config = layer.get_config()
layer = layers.deserialize({'class_name': layer.__class__.__name__, 'config': config})
```

אם לשכבה יש מצב אחד כלומר היא אינה חולקת שכבות אז אפשר לקבל את טנזור הקלט והפלט שלה יחד עם קלט ופלט הצורה שלה דרך:

```
layer.input
layer.output
layer.input_shape
layer.output_shape
```

אם לשכבה יש מספר מצבים אז נשתמש ב:

```
layer.get_input_at(node_index)
layer.get_output_at(node_index)
layer.get_input_shape_at(node_index)
layer.get_output_shape_at(node_index)
```

השכבות המרכזיות של קראס

רוב השכבות של קראס עובדות על מטריצות בעלות 1-2-3 ממדים. אנו נתעסק באלה שמתאימות לנו.

סוגי שונים של שכבות מספקים שינויים שונים לשכבות הנורוניים:

Convolutional layers מתאימים בעיקר לעבודה עם עיבוד תמונה

Recurrent layers מתאימים לעבודה עם מודל שמעבד מידע סדרתי תלוי זמן

Dense layers מקשר לכל קלט פלט המתאים לו

DENSE

מסוגל לקשר מטריצה של $n \times n$ שכבות. הוא למעשה עושה פעולה של DOT על הקלט עם איזשהו וקטור גרעין שמאותחל לפי מה שאנחנו מגדירים (בתור ברירת מחדל Glorot Uniform initialize) זה מוגדר להיות משקול התחלתי לפי מרחק הנתונים מהאפס, מה שעשוי להיות הדבר הנכון היות ואנחנו עובדים עם תכונות המגדירות את מיקומם של רכיבים באמפליטודה סביב ציר ה-X).

הארגומנטים של DENSE:

UNITS - ערך חיובי שאומר את גודל הפלט של השכבה. זה הפרמטר הכי חשוב. גודל הפלט למעשה מגדיר את הגודל של המטריצה הממושקלת.

Activation == - איזה פונקציית אקטיבציה נשתמש. במידה ולא אז זה מאותחל ל-NONE ואז יש אקטיבציה ליניארית שיתכן ועובד אך עדיין קראס נותן פונקציית אקטיבציה שונות.

BIAS - וקטור של "אפליה מתקנת" לערכים מסוימים במטריצה. היות ואנו לא יודעים איזה עדיפות לתת לאיזה תכונות נשתמש בערך ברירת המחדל שהוא הריק.

INITIALIZERS - אומר לקראס איך לאתחל את הערכים של השכבה שלנו כלומר גם את המטריצה הממושקלת וגם את וקטור העדיפות (אם יש כזה). יש כל מיני מאתחלים אפשריים. ברירת המחדל של המאתחל הוא לאפס את ערכי העדיפות.

Glorot Uniform initialize - מבצע משקול לפי התפלגות אחידה סביב האפס שגבולותיה משתנים בהתאם למרחק של הערכים מהאפס:

$$\text{limit} = \sqrt{6 / (\text{fan_in} + \text{fan_out})}$$

כאשר fan_in זה גודל ה-UNIT's של הקלט ו- fan_out זה אלו של הפלט.

REGULARIZERS - פרמטרים אופציונליים שיכולים להשפיע על המטריצה ממושקלת או וקטור העדיפות או על כל המודל (אם נפעיל זאת אחרי ה-ACTIVATION) שווה בדיקה לשם שיפור אחזי ביצוע:

Constraints - מגביל את הערכים למטריצה או וקטור העדיפות. כברירת מחדל זה לא מופעל.

RNN

כולל את השימוש בשכבות:

SimpleRNN, LSTM (Long Short-Term Memory layer), ConvLSTM2D, ConvLSTM2DCell, RNN

וכל המקבילים שלהם.

משתמשים בסוג הזה כאשר יש משמעות לסדר של המידע אותו אנו מעבדים. היות ואנו לא מחפשים להבין משמעות אלא מחפשים תוכן מסוים בקובץ האודיו אין חשיבות לאיזה סדר נקלוט דברים אלא לאם נקלוט אותם או לא.

איזה פונקציות ACTIVATION להשתמש

המסקנה מבדיקת פונקציות האקטיבציה הקיימות היא שיש להשתמש ב- RELU כי הוא הכי טוב לשכבות הביניים, אך אם יש אובדן מידע לשקול לבדוק תוצאות עם LEAKY RELU או SOFTMAX:

כל הערכים בוקטור יעברו דרך פונקציית האקטיבציה שאמורה לתת להם ערך בין 0 ל 1 כאשר ככל שהם קרובים לאחד אז הנוירון של התכונה יותר מופעל. בפועל לא כל פונקציית האקטיבציה עובדות בצורה הזאת.

SIGMOID – לוקח את הערכים בוקטורי התכונות ומעביר אותם למספרים בתחום שבין 0 ל 1.

הפונקציה שבה הוא משתמש:

$$f(x) = \frac{1}{1 + e^{-\beta x}}$$

יש בזה שני בעיות:

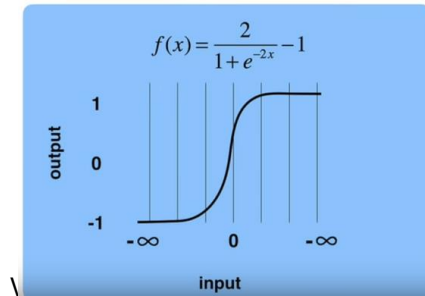
1. בגלל מבנה הפונקציה- חסומה זה גורם לחלק מהגארדיאנטים להיעלם ולאבד מידע חשוב

2. הפלט של הפונקציה אינו מרוכז סביב האפס. זה מתחיל מאפס ומסיים ב 1.

זה גורם לערכים מסוימים להיות רק שליליים או רק חיוביים. זה אומר שחלק מהערכים יהיו מנוגדים חזק מידי ויהיה קשה לשלוט בהם.

-HYPERBOLIC TANGENT FUNCTION (TANH)

hyperbolic tangent activation function

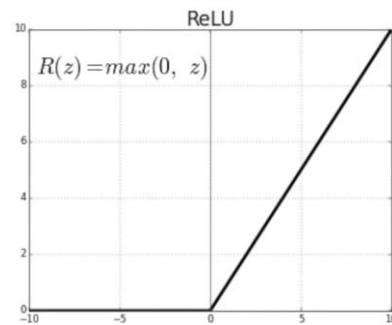


זה משנה את הערך של התכונה להיות בין -1 ל-1

מה שגורם לאופטימיזציות יותר טובות.

חסרון: עדיין סובל מהבעיה של SIGMOID בנוגע לאיבוד גרדיאנטים מסוימים.

-RELU



פופולרי בשנים האחרונות.

הוא נותן את הערך 0 לערכים שנמוכים מאפס ושומר על הערך כאשר הוא חיובי.

זאת לא פעולת חישוב יקרה, אז היא לומדת מהר ונמנעת מהבעיה אובדן המידע בשל העובדה שהיא לא חסומה.

השימוש ב-RELU הוא לשכבות המוסתרות

הבעיה ב-RELU היא שלפעמים חלק מה-UNITS יכולים להיות שבירים המהלך אימון "ולמות"

כלומר שהמידע לא יעבור שוב דרך אותו NODE ברשת הנוירית, לכן התיקון לבעיה זאת נקראת LEAKY RELU

LEAKY RELU – כעת במקום שהערך יהיה 0 כאשר הערך שלילי, ניתן ערך שלילי מתון.

MAXOUT – גרסה שנבנתה מ-RELU ומ-LEAKY RELU בשביל לתקן את החסרונות של אובדן המידע.

הבעיה-מספר הפרמטרים שיש לכל נוירון מוכפל אז זה לוקח יותר זמן לעבד.

-SOFTMAX

כאשר יש להשתמש ב-SOFTMAX בשכבות הפלט בכדי לפלוט קלאסיפיקציה של הרשת.

מאחר והיא נותנת הסתברות לסיווגים שונים

LINEAR FUNCTION – כאשר משתמשים בגרסיה בתוצאה (לא המקרה שלנו)

כמה שכבות להשתמש

כל שכבה שבין שכבת הקלט והפלט משתמש כ-HIDDEN LAYER למעשה כל נוירון בשלב לפני מקושר לנוירון בשכבה מעליו בסוג של קשר:

$$output = activation(weighted\ sum\ of\ inputs)$$

לכל שכבה נחשב את הקשר של כל נוירון עם הנוירונים הקודמים לה בעזרת הנוסחה הזאת עד אשר נגיע לשכבת הפלט.

*רק השכבה הראשונה מקבל INPUT_shape היות והיא זו שיוצרת מהם את הנוירונים שיחוברו לנוירונים שיהיו בכל שכבה מעליה- זאת הופכת את השכבה הראשונה למעשה לשני שכבות: השכבה של ה-INPUT והשכבה שגודלה ציון ב-UNITS.

מן הסתם ככל שיש לנו יותר שכבות נסתרות כך זמן החישוב הולך וגדל.

כשזה מגיע לכמות השכבות ומספר ה-NODES בכל שכבה יש לבצע ניסוי שיבדוק את הגדלים המתאימים (יעשה בהמשך).

מבחן הפרמטרים המתאימים למודל

כשזה מגיע לפרמטרים של מספר השכבות המוסתרות ומספר הנוירונים בכל שכבה כזה ופונקציית האקטיבציה הנכונה לכל שכבה, אין הערכה כללית שתעזור כאן ויש להתאים את הפרמטרים לפי המקרה.

עבורנו יש לעשות חיפוש ממצה עבור הפרמטרים האלה, והכלי שעומד למשימתנו הוא gridsearchCV של ספריית SKICITLEARN שצורת העבודה מתנהלת כך:

המחלקה מקבלת את המודל עליו אנו מחפשים את הפרמטרים. את מקבץ הטווחים שאנחנו רוצים לבדוק לכל פרמטר ומבצעת חיפוש ממצה בו היא עוברת על כל סוגי האפשרויות עד אשר היא מוצאת קומבינציה שמניבה את התוצאות הכי טובות.

דברים שהבנו מהרצות חוזרות של המחלקה:

- כשזה מגיע לפונקציית אקטיבציה לשכבות הנסתרות – התוצאות הטובות ביותר הגיע עם פונקציית האקטיבציה RELU. מכאן שניסיק שהחיסרון של אובדן מידע ופגיעה בנוירונים מסוימים לא משפיע ולכן ההעדפה היא לפונקציה שהפעלתה היא יותר פשוטה.
- ישנה העדפה לשימוש בגדלי שכבות שהם חזקות של 2 מאחר והם נותנות תוצאות בקצב מהיר ביחס לשאר המספרים.
- מספר ה-EPOCHS: אחרי 20 סבבים המודל מספיק להתייצב ואין צורך ביותר EPOCHS מכמות זאת.

מספר השכבות וגודל כל שכבה - בדיקה מספר השכבות האופטימלי וגודל כל שכבה נעשה ע"י ניסוי מס' 4 המפורט לעיל. באמצעותו נגיע לפרמטרים המתאימים לכל אחד מהמודלים שלנו.

4. פרטים טכניים יבשים:

קובץ	הצהרה	קלט	פלט	הערות
evaluate_screamCif.py	save_evaluate_bestMode l()		שמירת מסווג מאומן	הפונקציה נכתבה על ידינו. הספריות העיקריות שאנו נעזרים בה בפונקציה: sklearn keras

חיזוי (predict)

1. תיאור תהליך החיזוי והפעלתו :

- תזכורת- הפרויקט עובד כרגע על WINDOWS, ויש להתקין מבעוד מועד את החבילות שבו הוא משתמש.

א. כניסה לנתיב שבו נמצא הפרויקט

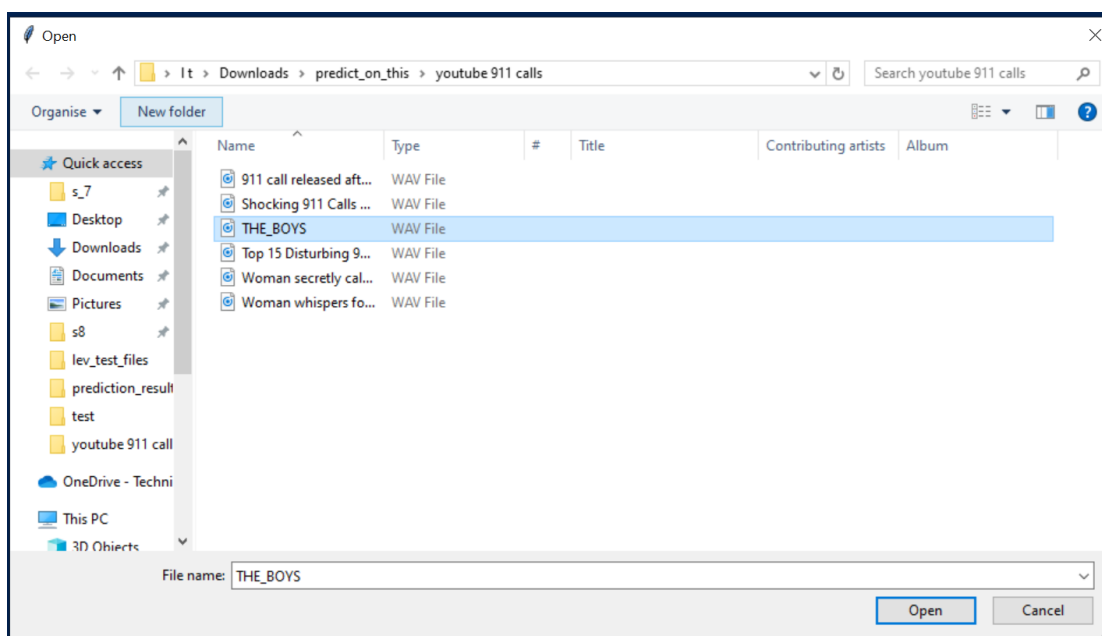
```
Administrator: Windows PowerShell
```

```
PS C:\Users\tunik> cd C:\Users\tunik\PycharmProjects\AI_Distress_3
```

ב. הפעלת קובץ הניבוי

```
PS C:\Users\tunik\PycharmProjects\AI_Distress_3> python predictor_ver_2.py
```

ג. ייפתח חלון שיש לבחור בו קובץ שמע עם שם באנגלית מסוג WAV בלבד, וללחוץ **Open**



ד. ההרצה תיקח מספר שניות, במהלכה האלגוריתם ידפיס למסך את הפעולות העיקריות שהוא מבצע, כגון : פיצול הקובץ שנבחר למקטעים של 3 שניות

```
exporting examined_files/THE_BOYS_sec_start_0.wav
exporting examined_files/THE_BOYS_sec_start_3.wav
exporting examined_files/THE_BOYS_sec_start_6.wav
exporting examined_files/THE_BOYS_sec_start_9.wav
exporting examined_files/THE_BOYS_sec_start_12.wav
```

טעינת המודלים שבהם אנו משתמשים לצורך החיזוי (יש להתעלם מהדפסות של אזהרות הנוגעות ל-tensor-flow)

```
loading model from models/best_from_final_models/gasp_clf_mfcc_15.h5
loading model from models/best_from_final_models/cry_clf_mfcc_20.h5
loading model from models/best_from_final_models/scream_clf_mfcc_20.h5
loading model from models/best_from_final_models/whisper_clf_mfcc_20.h5
```

יוצג לבסוף פלט למסך אשר יראה את המקטעים שבהם קיימת סבירות של מעל ל-80% למציאת מאפיין קולי של מצוקה. במידה ולא נמצע מאפיין מסוים, יודפס : Empty

```
Labeled as scream:
              filename  label_scream
5  THE_BOYS_sec_start_15.wav          0.84
Labeled as sniff:
Empty DataFrame
Columns: [filename, label_sniff]
```

בהדפסה הנ"ל, במקטע שמתחיל בשניה מספר 15, זוהתה סבירות של 84% לצעקה. (זהו הקטע שבו אחד הנערים ז"ל צועק בזמן הנסיעה).

ה. בנוסף להדפסות, יעלה אוטומטית חלון של Excel עם כל ניתוח ההסתברויות לכלל הסיווגים. (קיימות עמודות ללא נתונים- אלה עמודות לפיתוח עתידי וכרגע אינן בשימוש).

	A	B	C	D	E	F	G	H	I	J
1	date	filename	length(sec)	label_cry	label_gasp	label_scream	label_sniff	label_whisper	score	label
2	16_10_2019-9:44:31	THE_BOYS_sec_start_0.wav	3	1	0	0.75	0	0.119999997		
3	16_10_2019-9:44:31	THE_BOYS_sec_start_3.wav	3	1	0	0.610000014	0	0.140000001		
4	16_10_2019-9:44:31	THE_BOYS_sec_start_6.wav	3	1	0	0.200000003	0	0.100000001		
5	16_10_2019-9:44:31	THE_BOYS_sec_start_9.wav	3	1	0	0.340000004	0	0.129999995		
6	16_10_2019-9:44:31	THE_BOYS_sec_start_12.wav	3	1	0	0.090000004	0.01	0.090000004		
7	16_10_2019-9:44:31	THE_BOYS_sec_start_15.wav	3	1	0	0.839999974	0	0.170000002		

2. העמקה בתיאור האלגוריתם (מעבר לני"ל)

א. האלגוריתם משתמש במספר אובייקטים שמורים, אשר הוכנו מבעוד מועד :

1. Scaler - זהו העצם שמנרמל את המקטעים שניתנים לקובץ הניבוי. עצם זה נבנה מבעוד מועד על מאגר המידע שלנו שעימו אימנו את המסווגים.

2. `Scream_clf_mfcc_20.h5` - מודלים עם סיומת `h5` הם בעצם המסווגים המאומנים שלנו, אשר אימנו מבעוד מועד על מאגר המידע שברשותנו. (לפירוט נוסף על החלק בשם עם `mfcc_<15/12/20>` קראו את הפרק על בחירת הפיצורים).

- ב. הכנו שתי וריאציות על הניבויים, אחת שפשוט פולטת 0/1 בהתאם לסיווג, והשנייה (זו שמתוארת למעלה) אשר פולטת אך ורק סיווגים עם סבירות גבוהה מאחוז מסוים (בדוגמא זו- מעל ל-80%). כאשר בכל מקרה נפתח קובץ שמראה את כלל הניבויים.
- ג. כלומר התשובה לשאלה "האם אדם נמצא במצוקה פיזית מידית" מתרגמת למציאת מצוקה במקטע של כ-3 שניות, ובמידה ונמצא אחד המאפיינים, האדם במצוקה.

3. איכות הניבויים

- א. המנבא שלנו הצליח לזהות בהקלטה של שלושת הנערים שנחטפו ז"ל את הקטע שבו הם צועקים, אך לצערנו המסווגים אינם פועלים מספיק טוב עדיין.
- ב. הסיבות לכך מגוונות, ונפרט עליהן בסיכום. נציין כעת שהדבר קשור לקטעים שבחרנו בתור FALSE, כלומר הקטעים שאינם מייצגים מצוקה, ולכך שהיינו מוגבלים בקטעים שמצאנו בתור TRUE, כלומר הקטעים שכן מייצגים מצוקה (ולעוד סיבות שיצינו בסיכום).

2. פרטים טכניים יבשים :

קובץ	הצהרה	קלט	פלט	הערות
predictor_ver_2.py	-	-	קובץ אקסל עם כלל תוצאות הניבוי, ופלט למסך עם קטעי השמע שסווגו כחיוביים	

ניסויים

1. שחזור התוצאות:

1. אנא עקוב ראשית אחרי ההוראות בפרק הנספחים תחת הכותרת "הקמת סביבת עבודה לפרויקט".

2. בקובץ `evaluate_screamClf.py`, בתוך הפונקציה `experiments_for_report` תוכל למצוא ולשחזר את כל הניסויים שבוצעו. לדוגמא הפעלת `experiment1` תפעיל את הניסוי הראשון לפי הסדר הכתוב במסמך זה.

3. תוצאות הניסוי נפלטות לטבלת אקסל בנתיב היחסי `results/experiments_results.csv` שבה נשמרות כל התוצאות באופן אוטומטי, כאשר קיימת עמודה בטבלה ששמה `userInput` שערכיה הינם "שם הניסוי + תאריך".

4. לרוב הניסויים בנינו טבלאות מושקעות מאד באקסל, כלומר העתקנו את התוצאות מסעיף 3 לקובץ אחר בשם הניסוי שבוצע, למשל `experiment2.csv`, ובתוכו יצרנו ידנית את הטבלאות שבסוף העתקנו למסמך זה.

2. מתודולוגיה ניסויית:

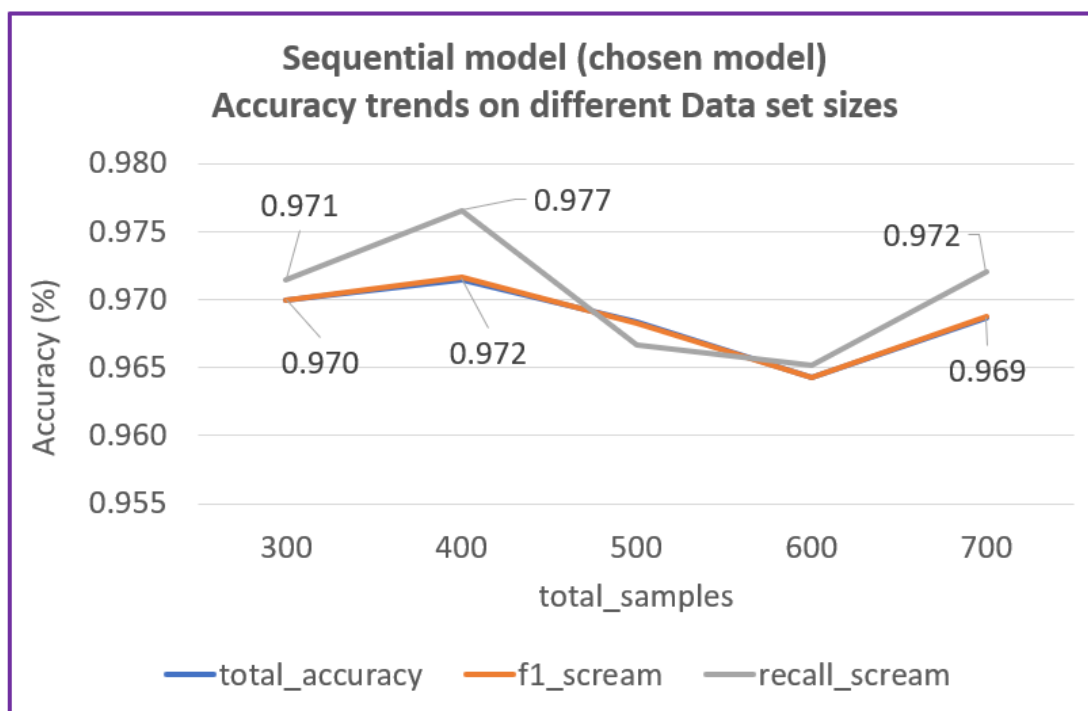
בחרנו ליצור לכל ניסוי פונקציה משלו. בכל תחילת פונקציה נקבעים הפרמטרים הרלוונטיים לניסוי, כדוגמת הקובץ שממנו נלקחים הנתונים, הפרמטרים השונים של המסווג ושל אופן עיבוד קובץ המידע טרם העברתו הלאה. חשוב לציין כי כל הניסויים עוברים את אותו התהליך שצוין בפרק על "אימון", כלומר כל ניסוי מחזיר תוצאות שכוללות דגימה אקראית, חוזרת ונשנית של המידע, וביצוע כ-5 חזרות על תהליך ה-"K-fold". כאשר K שווה ל-5. נציין כי רוב הניסויים כוללים שני גרפים- גרף ממוקד על המסווג הסופי שלנו, וגרף נוסף שכולל חציון או ממוצע על כ-10 מודלים שונים (יפורטו בהמשך). הסיבה לכך היא שקודם כל מעניין אותנו איך המסווג שלנו מתפקד בניסויים השונים, ולאחר מכן מעניין אותנו מהי המגמה כאשר משתמשים בכמה מסווגים שונים באותו הניסוי.

3. ניסוי ראשון

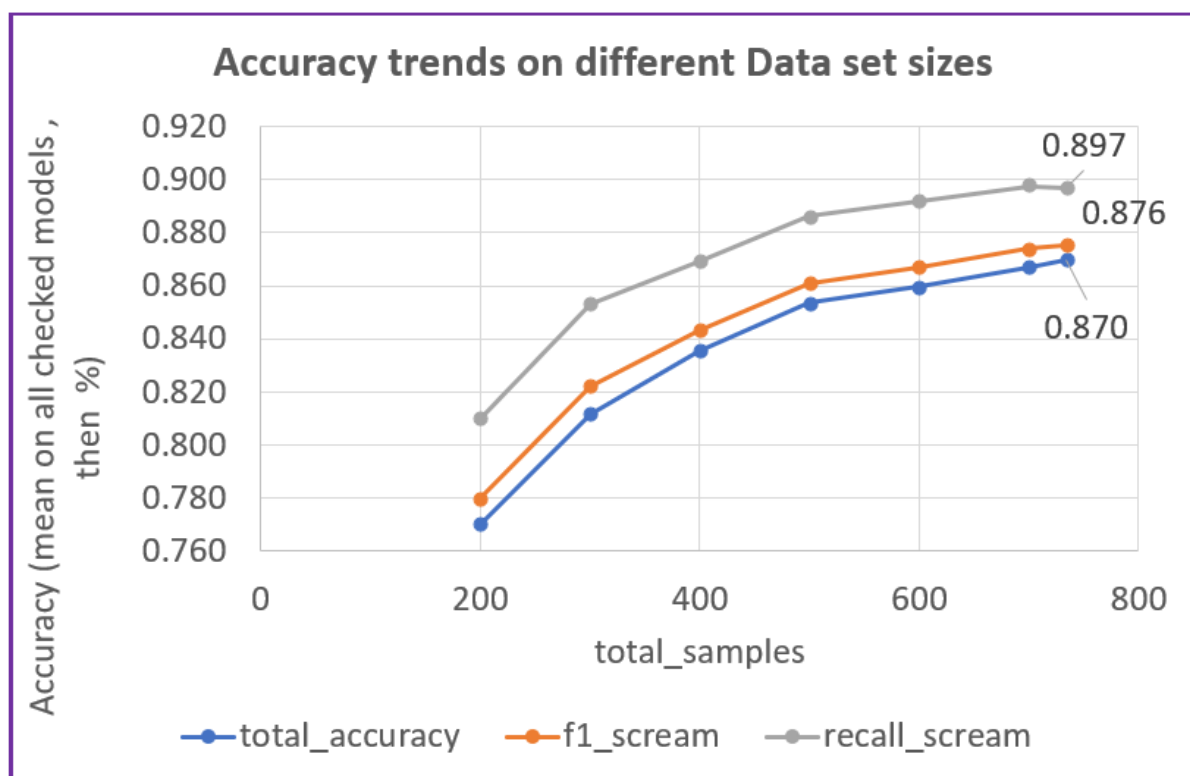
מטרה: האם בהינתן מאגר דוגמאות גדול יותר, הדיוק ישתפר? או יהיה גרוע יותר? ובמידה והדיוק משתפר- האם קיימת קבוצת דוגמאות בגודל מסוים אשר מביאה את רמת הדיוק להתכנסות מיטבית?

הצדקה: אחד האתגרים שהצגנו במסמך, היה שסברנו כי לא הצלחנו לחלץ מאגר נתונים "מספק מבחינתנו". כלומר הייתה לנו דאגה ממשית כי גודל המאגר שלנו ימנע מאיתנו להגיע לרמות דיוק טובות. עם זאת, הופתענו לטובה כאשר כבר בניסוי הראשוני הגענו לרמה של כ-90% דיוק (במדד ה-accuracy). לכן החלטנו שהדבר הכי קריטי בשלב הראשון הינו להבין האם בעצם גודל הקבוצה שסיפקנו הביא להתכנסות מסוימת בדיוק של המסווג, או שאולי קיים מקום לשיפור נוסף בהנחה ונמצא דרך להגדיל את מאגר הנתונים. בנוסף רצינו לשלול כי הגדלת קבוצת הדוגמאות לא פגמה בדיוק המסווג.

תיאור הניסוי: נציג שני גרפים- הראשון על המודל הספציפי שבחרנו בשביל המחלקה "צעקה", והשני על חציון הדיוקים של כלל המסווגים השונים שנבדקו (ראה פרטים נוספים על המודלים בפרק הנספחים תחת הכותרת "נספחים לניסויים"). :



- בגרף הבא, מגמות הדיוקים הינם הממוצע (חציון) על כלל המודלים שמימין לגרף (כלומר כל קו מייצג אחוז דיוק לאחר שנלקח החציון של כלל תוצאות המודלים על גודל קבוצת דוגמאות מסוימת):



AdaBoostClassifier
DecisionTreeClassifier
GaussianNB
GaussianProcessClassifier
Id3Estimator
KNeighborsClassifier
MLPClassifier
RandomForestClassifier
Sequential
SVC

תיאור התוצאות: בגרף התחתון ניתן לראות כי ככל שנותנים יותר דוגמאות כך הדיוקים נהיים טובים יותר. בנוסף ניתן לראות כי השיפוע נהיה פחות חד ככל שמתקרבים ל-800 דוגמאות. **בגרף העליון** ניתן לראות כי באיזור ה-400 דגימות קיים שיא מסוים ברמת הדיוק, ובנוסף ניתן לראות כי באיזור ה-800 דגימות השיפוע עדיין חד וחיובי.

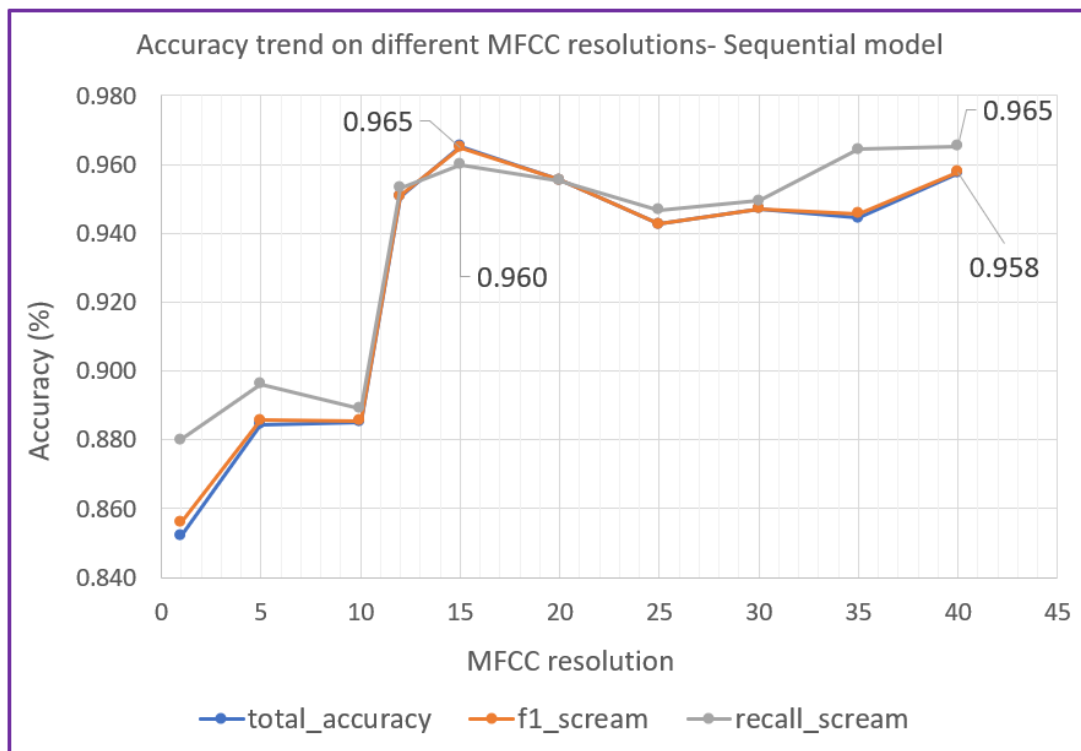
מסקנות מהניסוי:

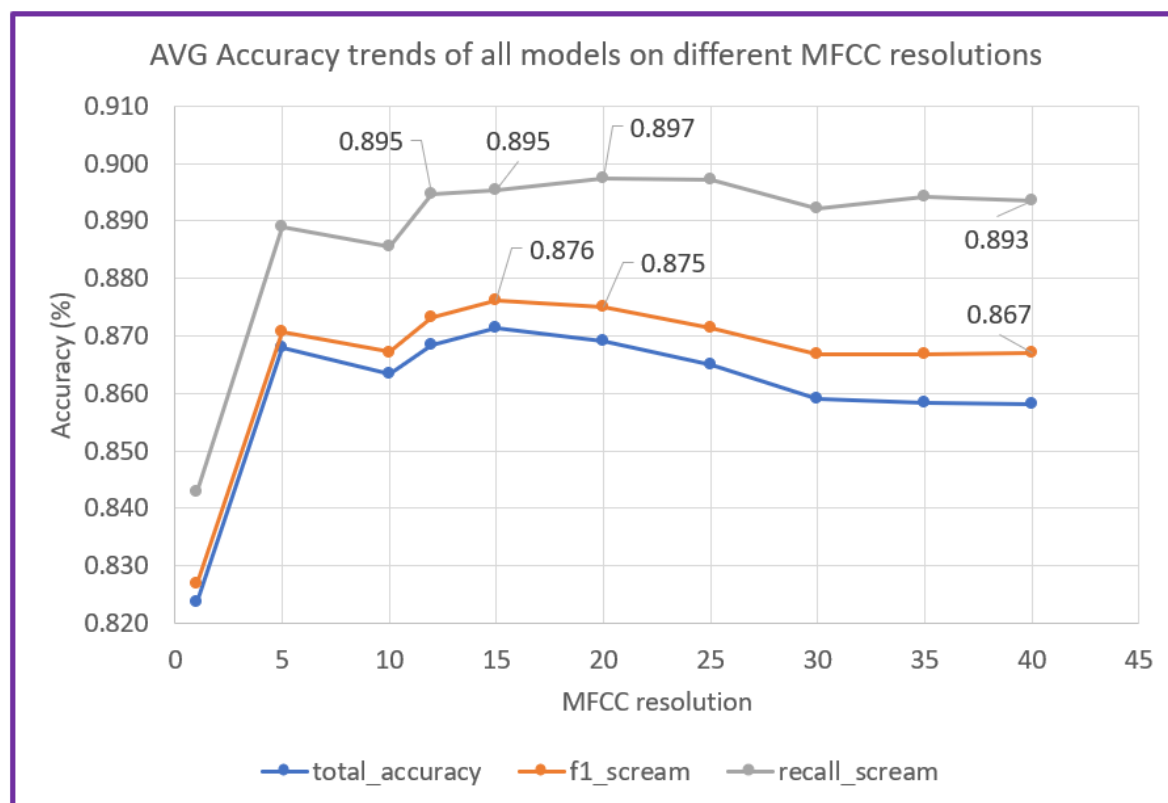
1. הדיוק עולה ככל שקבוצת הדוגמאות גדלה, כלומר להערכתנו במידה ויכולנו לספק עוד דוגמאות בהתאם ליחס שהסברנו בפרק "תיאור המערכת (אימון המערכת והערכת המסווגים), הדיוק יכול עוד להשתפר.
2. המסווג שבחרנו בעל דיוק גבוה מהממוצע המוצג בשאר המסווגים. (השוואה ממוקדת בין המודלים תתבצע בניסויים הבאים).
3. לא ניתן להכריע כי הגענו להתכנסות ברמת הדיוק ככל שהגדלנו את קבוצת הדוגמאות. (עוד על כך בפרק הסיכום).

4. ניסוי שני

מטרה: מהי הרזולוציה המיטבית עבור המאפיין [mfcc](#) (feature) ?

הצדקה: בשלבים הקודמים בעבודה הסברנו את **חשיבותו** של מאפיין זה, ופירטנו כי ניתן להקבילו למיקרוסקופ שניתן לכוונו לרזולוציות שונות. עקב כך שבמאמרים השונים שקראנו הדעות לגבי ערך הרזולוציה נעו בין 12-20, ובין ערכים שמתקרבים יותר ל-40, עניין אותנו מאד להכריע מה תהיה הרזולוציה הכי טובה בשביל המסווגים שלנו, ועד כמה באמת שינוי הרזולוציה ישפיע על תוצאות הדיוקים השונים. (ראה הערה על מתודולוגית הניסוי הספציפי הזה בפרק נספחים תחת "נספחים לניסויים")





תיאור התוצאות: בשני הגרפים ניתן לראות כי הנקודות הבולטות עם הדיוקים הכי גבוהים נמצאות כאשר הרזולוציות שוות ל-15, 20 ו-40. בנוסף ניתן לראות כי השוני בין הנקודות הנ"ל מבחינת הדיוקים אינו עולה על 1.5%.

בגרף העליון הנקודה בה הרזולוציה שווה ל-15 היא בעלת ערך ה-F1 הכי גבוה, אך הנקודה בה הרזולוציה שווה ל-40 היא בעלת ערך ה-RECALL הכי גבוה. בנוסף בנקודה בה הרזולוציה היא 20 ערכי הדיוקים נמצאים באותו ערך. **בגרף התחתון** ניתן לראות כי המגמה עד לרזולוציה של 15 הינה עליה כללית עם מינימום מקומי ברזולוציה ששווה ל-5. וכי החל מרזולוציה של 15 מתחילה מגמה הדרגתית של ירידה עד לרזולוציה של 30, ממנה והלאה הדיוקים יחסית אחידים.

מסקנות מהניסוי:

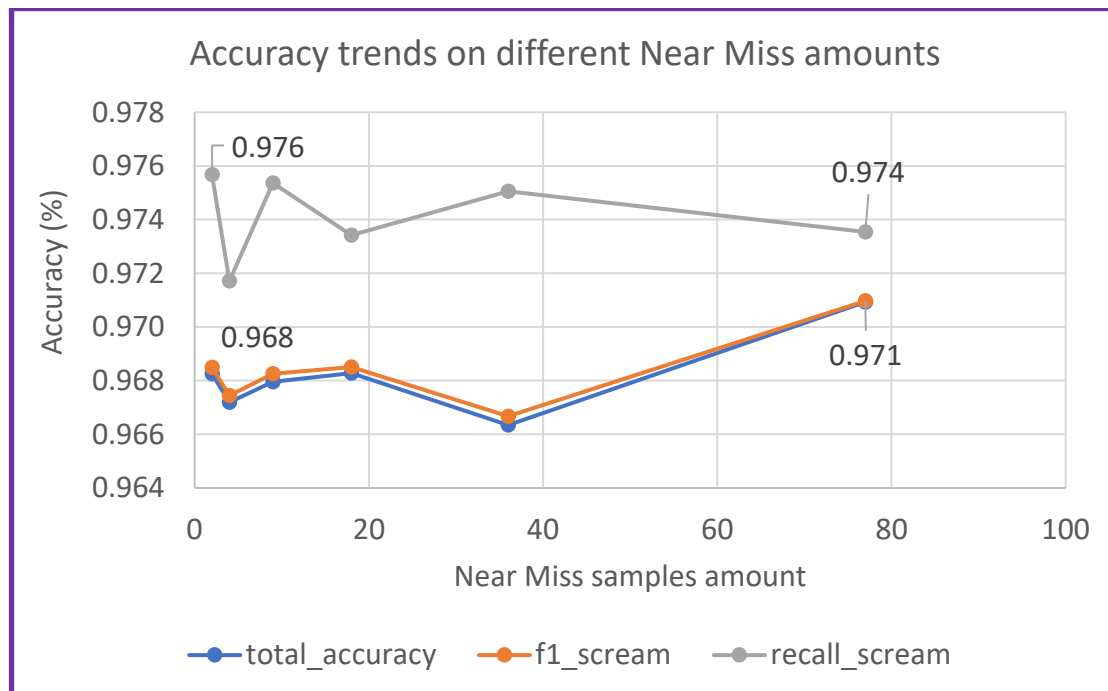
- קיימת קירבה בתוצאות הדיוק בין רזולוציה של 15 לרזולוציה של 20, לכן נצטרך לבצע ניסוי דומה לניסוי זה כדי להכריע באיזה רזולוציה להשתמש בסופו של דבר במסווג הסופי ("production").
- הרזולוציה במודל Sequential שמוזהה הכי טוב "צעקה" היא 40, עם זאת המחיר הוא דיוק כללי נמוך יותר, בעוד שברזולוציה של 20 נוותר על 0.5% ב-RECALL כדי להרוויח 1% בערך ה-F1 אשר משקלל גם את הדיוק בתוכו.
- אנו רואים בתוצאות הגרף העליון, בהשוואה בין רזולוציה של 40 לרזולוציה של 20 דוגמא לעיקרון "התער של אוקס". כיוון שברזולוציה של 20 אנו משתמשים ב-20-

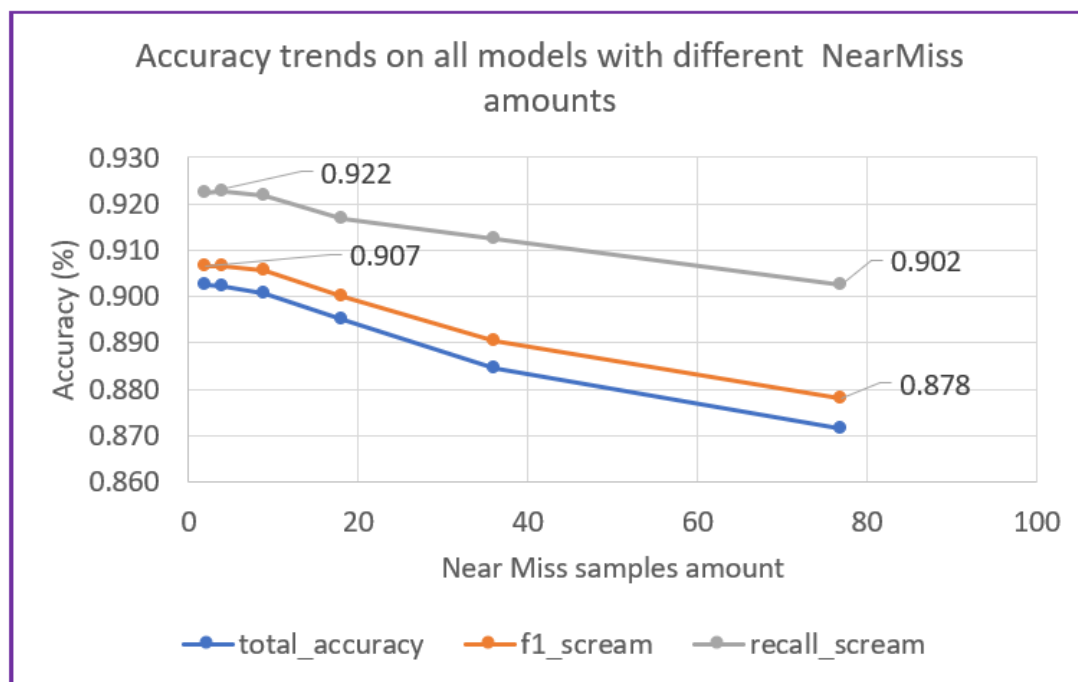
עמודות פחות בוקטור המאפיינים ועדיין מגיעים לתוצאות מאד דומות בדיוקים, נעדיף להשתמש ב"הסבר הפשוט יותר" כלומר ברזולוציה של 20 ולא של 40. (כאשר הרזולוציה היא 40- המשמעות היא שיש 40 עמודות שמייצגות את הרזולוציה של MFCC כיוון שכל עמודה מייצגת טווח מסוים ברזולוציה שנקבעה, אך כאשר יש 20 עמודות אז טווחי הערכים גדלים פשוט- כמו במיקרוסקופ, עד כמה תהיה מעוניין לראות לפרטים?).

4. ניתן לראות כי המסווג שלנו מספק תוצאות טובות יותר בדיוקים מאשר הממוצעים של כל המסווגים ביחד.

5. ניסוי שלישי

מטרה: האם השימוש ב"החטאות קרובות" (Near misses) משפר את הדיוק של המסווגים?
הצדקה: הדגשנו בעבודתנו את חשיבות השימוש בתיחום הסיווגים השונים ע"י שימוש ב"החטאות קרובות". בהקבלה לאופן שבו מלמדים ילד על החפצים שסביבו- כשרוצים ללמוד "מהו שולחן?"- מראים לילד שולחן ואומרים: "זהו שולחן", ואז מראים לו כיסא ואומרים "זהו כיסא. כיסא זה לא שולחן". כלומר מנסים ללמדו על דוגמאות שליליות שקשורות לסיווג האמיתי של העצם הרצוי. בניסוי זה רצינו לבדק את ההשערה כי ע"י שימוש ב"החטאות קרובות" רמת הדיוק שלנו באמת תהיה גבוהה יותר.





תיאור התוצאות:

הערה: **בשני הגרפים** כמות הדגימות הכללית בניסויים הייתה 734, כאשר היו 367 דוגמאות חיוביות ומספר זה עבר הדוגמאות השליליות. מתוך הגרפים ניתן לראות בציר האופקי את כמות הדגימות השליליות מתוך כלל הדגימות השליליות. חשוב לזכר כי השתמשנו כאן באלגוריתם סטרטפורד אשר שמר על היחסים של הכמויות בין הסיווגים {של חיובי, שלילי שאינו החטאה קרובה, והחטאה קרובה} בין סט האימון לסט המבחן.

בגרף העליון ניתן לזהות מגמות הפוכות בין דיוק ה-RECALL לדיוק הכללי. מנגד נשים לב כי התוצאות בין שימוש ב-Near-Miss לבין אי שימוש בהן אינן משפיעות מעבר ל-0.05 באחוזי הדיוק השונים. **בגרף התחתון** ניתן לראות כי קיימת מגמה בולטת של ירידה, אשר בניגוד לגרף העליון, הפעם הינה בהחלט משמעותית שכן ההבדלים ברמות הדיוק מגיעים ל-0.3 אחוזים שוני.

מסקנות מהניסוי:

1. במסווג הסופי שלנו השינוי במגמת הדיוק אינו משמעותי, אינו עולה על 0.05 באחוזי הדיוק, אך עדיין יש מקום לבחון האם בהינתן קבוצת דוגמאות גדולה יותר עם אותם יחסים בין הסיווגים השונים, המגמות יתכנסו לתוצאת דיוק מסוימת. (עוד על כמות הדוגמאות שהייתה ברשותנו בפרק הסיכום).
2. להערכתנו עליה ביחס של "החטאות קרובות" מתוך כלל הדגימות השליליות כן שיפרה את הדיוק שכן דיוק של 97% כשיש כמעט 80 "החטאות קרובות" שווה בעינינו הרבה יותר מדיוק של 96.8% כשיש רק 2 "החטאות קרובות".
3. מסקנה חשובה נוספת הינה שבעצם המסווג הסופי שלנו מגיב טוב יותר להוספת דגימות שליליות מהמגמה הכללית של שאר המסווגים שנבדקו.

6. ניסוי רביעי

מטרה: להבין האם מסווג בודד ידע לתת דיוקים גבוהים יותר מאשר כל אחד מהמסווגים הבודדים שלנו

הצדקה: במידה והניסוי יראה כי המסווג הבודד נותן דיוקים גבוהים יותר מהמסווגים הבודדים שלנו, נסיק כי "ההתאמה האישית" שעשינו לכל מסווג בנפרד לא הייתה הכרחית.

תיאור התוצאות:

הדיוק הממוצע לאחר 100 הרצות הינו 84.45%, עם סטיה סטנדרטית של כ-2%.

Baseline: 84.45% (1.78%)

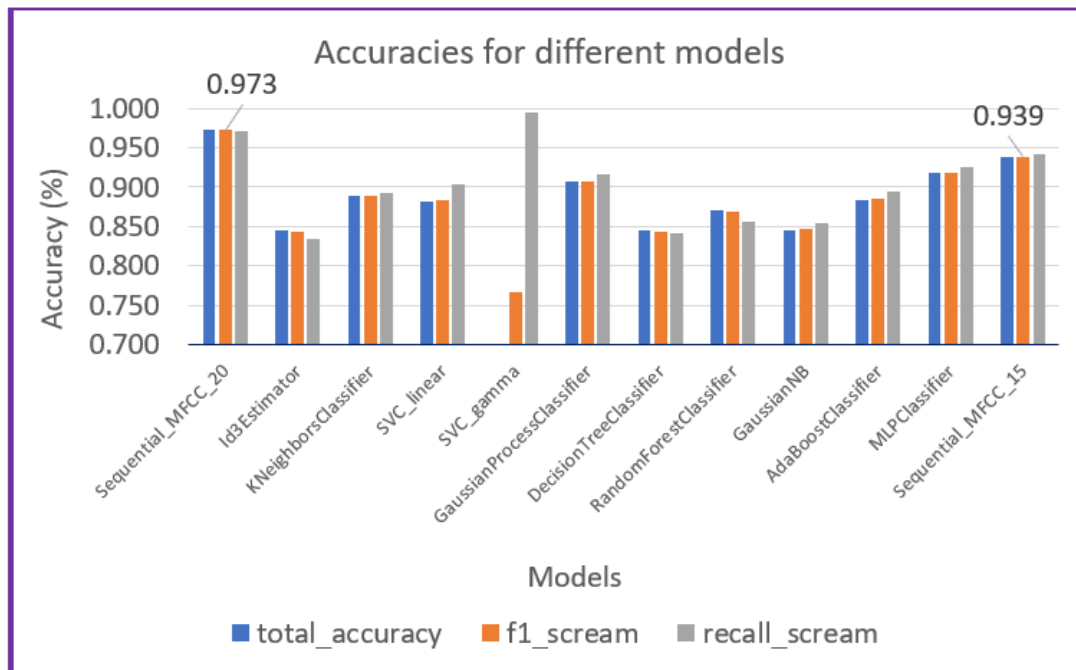
מסקנות מהניסוי:

1. הדיוקים של המסווגים הבינאריים שלנו הרבה יותר גבוהים. (מעל ל-90% במסווגים הבינאריים לעומת 84% במסווג הבודד שפעל על כלל המחלקות)
 2. אוסיף התייחסות עמוקה יותר על מסקנה מספר 1 בפרק הסיכום תחת "דיון בתוצאות הניסויים".
- כדי להריץ את הניסוי הרביעי יש להפעיל את הפונקציה הראשית בקובץ `check_sequential_categorical.py`

7. ניסוי חמישי

מטרה: וידוא כי המודל שבחרנו למציאת הסיווג הרצוי (הגרף הבא מייצג את הדיוק על הסיווג "צעקה") הינו המסווג בעל הדיוק הכי גבוה למשימה זו.

הצדקה: ניסוי זה אולי אינו מעניין כמו הקודמים, אך הינו הכרחי. בסופו של דבר אנו רוצים להשתמש במסווג הכי טוב, ולכן היינו חייבים להשוות בין המסווגים השונים כדי לוודא כי אנו באמת משתמשים במסווג שמחזיר את התוצאות הכי טובות עבור הדיוק ועבור זיהוי הסיווגים שלנו. בנוסף בהתחשב בניסויים הקודמים, ראינו כי רזולוציה של 15 בתכונה MFCC מחזירה תוצאות די גבוהות, ולכן היה הכרחי להשוות בין מסווג מסוג Sequential עם 15 וגם עם 20 ברזולוציה של MFCC.



תיאור התוצאות:

המסווג עם רמת דיוק הגבוהה ביותר בכל שלושת הדיוקים שנמדדו הינו מסווג מסוג Sequential בעל רזולוציה של 20 בתכונה MFCC. הבא אחריו הינו אותו סוג מסווג אך עם רזולוציה של 20 ב-MFCC, ולאחריהם יתר המסווגים.

מסקנות מהניסוי:

1. המסקנה העיקרית היא שבחרנו את המסווג היעיל ביותר מבחינת כלל מדדי הדיוק הרלוונטיים. (פרטים נוספים על מסווג זה מופיעים בנספחים תחת "המודלים שנבדקו בניסויים").

8. ניסוי שישי

מטרה: מציאת מספר השכבות המתאים והגודל המתאים לכל שכבה במודל.

הצדקה: בניית המודל שניב את אחוזי הדיוק הגבוהים ביותר לכל מאפיין

תהליך הניסוי:

הספרייה HYPRAS מהווה ספריית מעטפת המאפשרת לבצע אופטימיזציה לפרמטרים במודל כך שלכל מאפיין המודל יניב את אחוזי הדיוק הגבוהים ביותר. תחילה, בנינו פונקציית תבנית כללית שבה יש אפשרות להתאים כל פרמטר בכדי לבדוק אם שינוי בו מעלה את אחוז הדיוק שלנו:


```

def create_final_model(X_train, y_train, X_test, y_test):
    model = models.Sequential()
    model.add(Dense({choice([32, 64, 128, 256, 512, 1024])}, activation='relu', input_shape=(X_train.shape[1],)))
    # choices={choice(['one', 'two', 'three', 'four', 'five'])}
    choices = {choice(['four'])} # based on prior checks
    if choices == 'two':
        model.add(Dense({choice([32, 64, 128, 256, 512, 1024])}, activation='relu'))
    elif choices == 'three':
        model.add(Dense({choice([32, 64, 128, 256, 512, 1024])}, activation='relu'))
        model.add(Dense({choice([32, 64, 128, 256, 512, 1024])}, activation='relu'))
    elif choices == 'four':
        model.add(Dense({choice([32, 64, 128, 256, 512, 1024])}, activation='relu'))
        model.add(Dense({choice([32, 64, 128, 256, 512, 1024])}, activation='relu'))
        model.add(Dense({choice([32, 64, 128, 256, 512, 1024])}, activation='relu'))
    elif choices == 'five':
        model.add(Dense({choice([32, 64, 128, 256, 512, 1024])}, activation='relu'))
        model.add(Dense({choice([32, 64, 128, 256, 512, 1024])}, activation='relu'))
        model.add(Dense({choice([32, 64, 128, 256, 512, 1024])}, activation='relu'))
        model.add(Dense({choice([32, 64, 128, 256, 512, 1024])}, activation='relu'))
    model.add(layers.Dense(1, activation='sigmoid')) # the 1 means binary classification
    model.compile(optimizer='adam',
                  loss='binary_crossentropy',
                  metrics=['accuracy'])
    model.fit(X_train, Y_train,
              batch_size={choice([128, 256, 512])},
              nb_epoch=20,
              verbose=0,
              validation_data=(X_test, Y_test))
    score, acc = model.evaluate(X_test, Y_test, verbose=0)
    print('Test accuracy:', acc)
    return {'loss': -acc, 'status': STATUS_OK, 'model': model}

```

הפרמטרים שנבדקו הם אלו אשר תחומים בסוגריים מרובעים [] בתוך פונקציה CHOICE. כאשר הפונקציה get_optimised_model_final מקבלת בכל פעם את וקטורי האימון והטסט עבור המאפיין הספציפי אותו באנו לבדוק ומבצעת הרצות חוזרות עם פרמטרים שונים בכל פעם.

```

def get_optimised_model_final(x_for_k_fold, x_test, y_for_k_fold, y_test):
    """
    return: best classifier after hyper parameter search
    receives: split dataset for train and test and classifier label (scream, cry)
    for pickle files directory navigation
    """
    save_data_to_pickle(x_for_k_fold, x_test, y_for_k_fold, y_test)
    best_run_inner, best_model_inner = optim.minimize(model=create_final_model,
                                                       data=data_func,
                                                       algo=tpe.suggest,
                                                       max_evals=100,
                                                       trials=Trials(),
                                                       eval_space=True)
    print(f"best_run= {best_run_inner}")
    best_model_inner.summary() # print summary
    return best_model_inner

```

כמו כן, אנו מסוגלים לשלוט במספר ההרצות שנעשות ולשמור את התוצאות שכל מודל עם הפרמטרים שאיתן רץ הניב לנו.

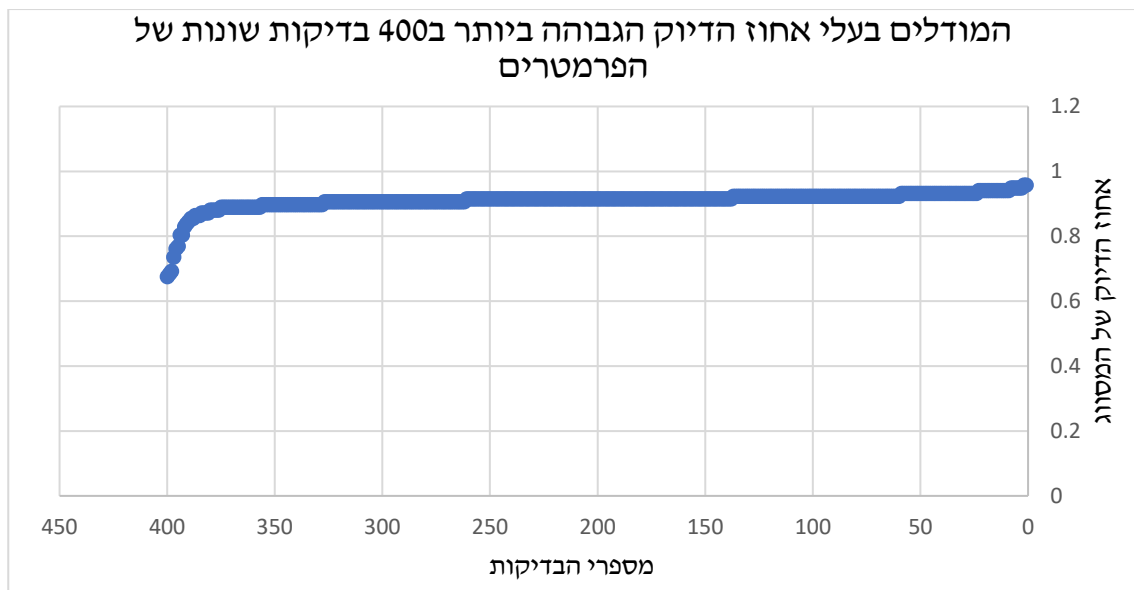
לכל מאפיין שמרנו את תוצאות הריצות שלו בקובץ EXCEL נפרד שבו לקחנו את המודל שהניב את אחוזי הדיוק הגבוהים ביותר (נמצאים בתיקייה RESULTS)
הנחות שהנחנו במהלך הניסוי לאור מידע שקראנו באינטרנט:

1. פונקציות האקטיבציה שנכון לעבוד איתן בכל שכבה היא הפונקציה RELU כאשר בשכבה האחרונה נעבוד עם הפונקציה SIGMOID על מנת לקבל סיווג בינארי (הסבר מצוי בפרק על הערכת המסווגים, בפירוט על המסווג שבחרנו).
2. ערך ה-EPOCH יהיה 20 היות ולא נדרשים יותר EPCHS מזה על מנת שהמודל יגיע לידי חישוב.

3. גודל השכבה הראשונה לא יהיה קטן מ-128 -נובע מניסוי מקדים שנעשה על גדלים שקטנים מערך זה והניבו ערכי דיוק נמוכים מאוד ביחס לערכים הללו.
4. נשתמש בגדלי שכבות שהן חזקות של 2 מאחר והן נותנות תוצאות יותר מהירות ביחס לערכים אחרים.
5. לא נדרש יותר מ-5 שכבות במודל לכל מאפיין – שימוש במספר גדול של שכבות גורם ל-over fitting של מידע ולהפחתה בערכי הדיוק של המודל.

תוצאות הניסוי:

תוצאות בחינת פרמטרים עבור מאפיין צעקה:



1. הדיוק הגבוה ביותר עבור מאפיין צעקה התקבל ככל שהתקרבונו למודל המכיל 5 שכבות.

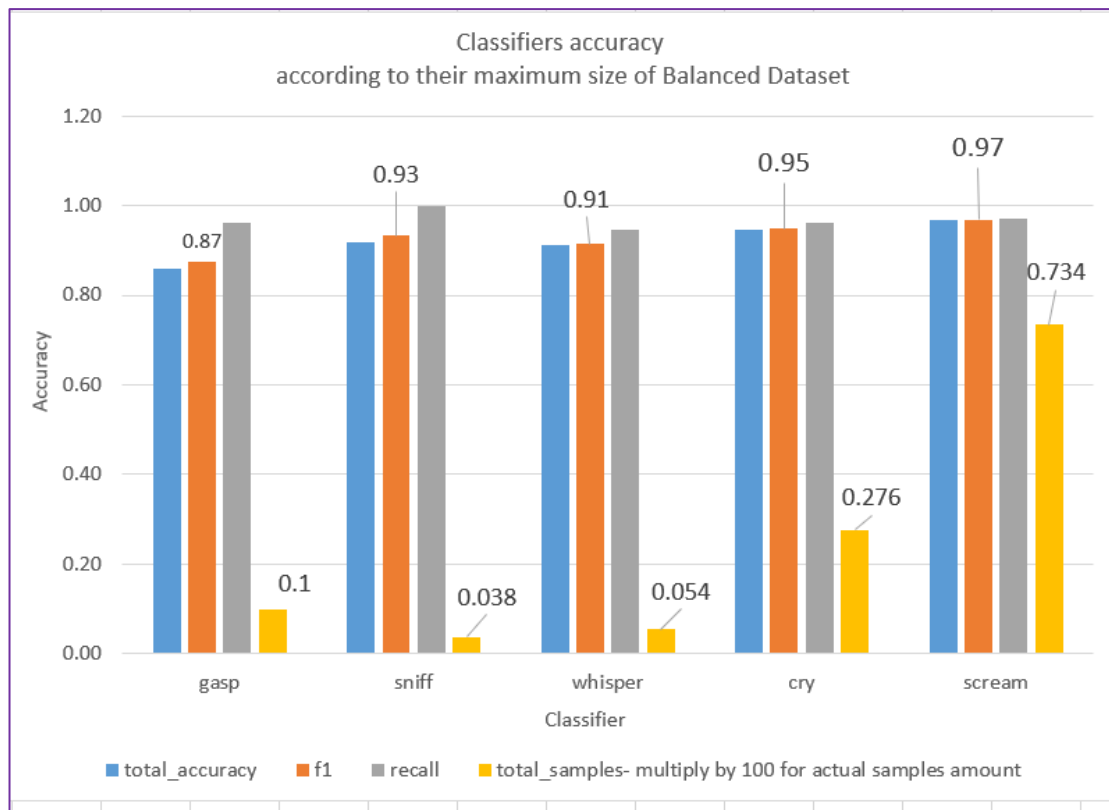
מסקנות מהניסוי:

לאחר בדיקה נפרדת של כל אחד מהמסווגים הגענו למספר השכבות האופטימלי שכל מאפיין נדרש אליו ולגדלי השכבות המתאימים לכל אחד מהשכבות של המודל:

המסווג	בכי	התנשפות	צעקה	רחרוח (SNIFF)	לחישה
גודל הקלט	26 (MFCC)	21 (MFCC 15)	26 (MFCC 20)	18 (MFCC 12)	26 (MFCC 20)
שכבה 1	256	1024	32	32	256
שכבה 2	1024	256	512	512	32
שכבה 3	256	64	512	128	256
שכבה 4	1024	512	32	64	32
שכבה 5	1	1	1	1	1

9. ניסוי שביעי

מטרה: להראות כיצד כמות הדגימות שהצלחנו לאסוף משפיע באופן ישיר על הדיוקים.
הצדקה: ניסוי זה גם כן אינו עיקרי בעינינו, אך הוא חשוב כדי להסביר מדוע הפוטנציאל בפרויקט זה עוד לא מומש במלואו, כיוון שלא זכינו לשיתוף פעולה מצד רשויות החוק לעת עתה.



תיאור התוצאות:

הדיוק הכי גבוה נמצא במסווג מסוג "צעקה" אשר זכה לאיסוף של מעל ל-730 דגימות מאוזנות (50% חיובי ו-50% שלילי). הדיוק הכי נמוך דוקא במחלקה "gasp" (לקיחת אוויר חדה לריאות) אשר זכתה ל-100 דגימות מאוזנות.

מסקנות מהניסוי:

- בהסתמך גם על גרפים קודמים, המסקנה העיקרית הינה שככל שנוסיף דגימות, הדיוק שלנו יגדל. הדבר איננו תמיד נכון, אך באמת במקרה שלנו ולפי הניסויים הקודמים גם שעשינו, הוספת דגימות רק שיפרה את הדיוקים, לכן אנו סבורים כי עוד לא הגענו לנקודה שבה קיים "עודף מידע שגורם ל-overfitting".
- מסקנה מפתיעה היא שיכולת זיהוי הסיווג אצלנו היא יחסית גבוהה על אף שכמות הדגימות נחשבת נמוכה יחסית, הרי כלל המחלקות קיבלו פחות מאלף דגימות. הדבר עלול לנבוע מעוד סיבות אשר ייבדקו בהמשך כאשר נבחן את המסווגים שלנו גם על קטעי שמע מתוך סדרות, סרטים וכו'. (התייחסות לכך בסיכום).

סיכום

1. דיון בתוצאות

1. הניסויים הראו שיש מקום לנסות להגדיל את מאגר הדוגמאות שכן לא הגענו להתכנסות באחוז הדיוק בניסוי שבו הגדלנו בהדרגה את כמות הדוגמאות.
2. המודל הסדרתי עם כיוון הפרמטרים סיפק אחוזי דיוק גבוהים יותר ביחס לעשרת המודלים האחרים שנבדקו.
3. אחת השאלות שיצאנו איתן לדרך בתחילת הפרויקט, הייתה "האם בהינתן מידע נוסף- אחוזי הדיוק יעלו?" כלומר המטרה הייתה לבחון האם בניית מסווגים מאד ספציפיים- בינאריים, תשיג אחוזי דיוק גבוהים יותר מאשר מסווג בודד שמנבא תשובה על כלל הסיווגים. עקב הניסויים שביצענו ראינו שמודלי למידה שמקומפלים עם פונקציית למידה בינארית- binary crossentropy נותנים תוצאות דיוק גבוהות יותר ביחס למודל בודד אשר נדרש להחזיר תוצאות עבור כלל הסיווגים, ומשתמש בפונקציית למידה מסוג categorical_crossentropy. להערכתנו הביצועים של המסווגים הבינאריים טובים יותר מסיבה מאד בסיסית- כל מודל "תפור למידות" של הסיווג שאותו הוא מנסה למצוא- הן בכמות ה- mfcc (מאפיין שפירטנו עליו במהלך העבודה), הן בכמות ה- layers של המודל, והן בגודלה של כל layer.
4. סעיפים 1-3 למעלה כוללים את הדברים העיקריים שרצינו להתייחס אליהם בסיכום, נקודות נוספות אשר ספציפיות יותר לכל ניסוי נמצאות בפרק "ניסויים".

2. נקודות לשיפור

1. אי התעסקות ב-Data Augmentation

1. כיוון שקיבלנו תוצאות דיוק גבוהות יחסית בהתחלה (מעל 0.90), לא מצאנו לנכון להתעסק בריבוד המידע. בדיעבד קיים סיכוי טוב שטעינו, ושכן היה צריך להתעסק בנושא זה, כלומר שכן היינו צריכים לקחת הקלטות עם סיווגים חיוביים ולהוסיף להן רעשי רקע שונים. להערכתנו הריבוד יכול היה לשפר את תוצאות הדיוקים עוד יותר. ולעזור לנו לגלות מוקדם יותר כיצד המודל שלנו מתמודד עם רעשי רקע בזמן הקלטת שמע וכו'. בראיה המקורית שלנו רצינו שהדגימות החיוביות יהיו איכותיות- וההגדרה שלנו לאיכותי היה שהצעקה נשמעת באופן מושלם ללא הפרעות, עם זאת בדיעבד אנו מבינים שדווקא כן קיימת חשיבות רבה להכנסת סיווגים חיוביים ביחד עם רעשי רקע, שכן בסופו של דבר המסווג יפעל על הקלטות שהינן מלאות ברעשי רקע.

2. עבודה על מאגר דוגמאות חיוביות שאינו "גדול"

1. כיוון שלא הצלחנו לחלץ הקלטות אמיתיות ממשטרת ישראל, נאלצנו לחלץ את המידע בעצמנו מהאינטרנט ממקורות מגוונים. הצלחנו לחלץ מאות של דגימות חיוביות רלוונטיות לסיווגים השונים, ועשרות של "החטאות קרובות", אך איננו שלמים עם

מאגר המידע, במובן שלפי תוצאות הניסויים שביצענו, בהנחה וכן היה מאגר גדול יותר, יש סיכוי שהדיוק היה עולה.

3. בחירת קטעי שמע שליליים בצורה ממוקדת יותר לסביבה שבה תפעל המערכת

לאחר שקיימנו פגישה עם מרצה הקורס על הרצון להמשיך עם הפרויקט גם בסמסטר הבא, כבדרך אגב שיתפנו אותו באופן שבו אספנו דגימות שליליות. סיפרנו שאספנו דגימות שליליות ממקורות מגוונים, ושסיפקנו לאלגוריתם צלילים של ציפורים, רכבים נוסעים, דיבורים של אנשים בחללים פתוחים, צלילי סביבה וכו'. ההערה של המרצה הייתה שלדעתו לא מיקדנו מספיק את הדוגמאות השליליות, שכן במערכת שאמורה לתפקד קודם כל במוקדי שיחות, הדוגמאות השליליות אמורות להיות **שיחות טלפון** רגילות שאינן מצוקה. בראייתנו הכנסנו דגימות אלה כי רצינו שהמערכת תפעל גם כאפליקציה מתוך הפלאפון לאנשים שמסתובבים ברחוב, אך הלכח שלנו הוא שדרוש תמיד להתחיל במשהו ממוקד יותר.

4. למידת אלגוריתמי OAuth2 (לינק לעמוד ההסבר שעקבנו אחריו)

ציינו במהלך הדו"ח כי זהו אלגוריתם שניסינו להשתמש בו, אך לא הצלחנו בסופו של דבר למצות את כל אפשרויותיו עקב תקלות טכניות שלא רצינו להתעמק בפתרון. עם זאת חשוב לציין כי לפי ההסבר עליו באינטרנט בלינק המצ"ב בכותרת, ניתן להשתמש בו ולדלות בעזרתו יותר קבצי שמע רלוונטיים בפחות זמן. הסיבה שאנו שמים נקודה זו בחלק ה"לשיפור", היא שאם בפרויקטים הבאים שנעשה ניתקל שוב במגוונים מסוג זה, נגיע למסקנה שזה **כן** נושא חשוב שדרוש להבין כיצד לפצחו. בפרויקט ספציפי זה אנו שלמים עם ההחלטה שלא להשקיע עוד זמן בפיתרון התקלות שנתקלנו בהן בעת העבודה עימו, אך חשוב לנו לציין את האלגוריתם כדי שמי שיקרא את עבודתנו ידע על אתגר טכני זה וידע להחליט בעצמו האם שווה או לא שווה לזולה לדעת לעבוד עם אלגוריתמי OAuth2.

5. מגבלות כוח מיחשוב ועבודה במכונה וירטואלית

בעבודה זו נתקלנו במגבלות של כח המחשוב שעמד לרשותנו. קצה הקרחון היה בזמני האימון, ורוב הקרחון היה בניסויים שביצענו, בהם היה דרוש לבצע הרבה חזרות כדי לקבל תוצאות אמינות. כיוון שעבדנו עם מחשבים ביתיים, בכל פעם שהרצנו אלגוריתם שלקח זמן, הדבר הקפיא לנו כמעט לגמרי את היכולת לעשות דברים נוספים במחשב במקביל. בשלב מסוים התחלנו לשחק עם חשבונות ענן שמספקות חברות כמו google ו-Kaggle, אך זה כבר היה בשלב מאוחר מדי שבו כבר "ספגנו" את זמני הריצות הארוכים על המחשבים הביתיים שלנו. כלומר לכל מי שיקרא שורות אלה ויעשה פרויקט של בינה מלאכותית בהמשך, אנו ממליצים כחלק מהשלב הכי בסיסי של עבודתכם להקים סביבת עבודה במכונה וירטואלית בענן. להלן דוגמאות של מגבלות שנתקלנו בהן: אי יכולת לעשות כמות גדולה יותר של folds באלגוריתם k-fold, אי יכולת להגיע להתכנסות במודל מסוים שהרצנו כיוון שהוא דרש עוד חזרות, אי יכולת להריץ פונקציות עוקבות ארוכות כיוון שהמחשב הגיע ל"מיצוי משאבים".

3. נקודות לשימור

1. תיעדוף מאפיינים קוליים של אנשים במצוקה פיזית
הדבר הכי בסיסי שהתחלנו ממנו בעבודה זו הינו התייעצות עם אנשי מקצוע בתחום הביטחון וההצלה. ללא שלב זה לא היינו יודעים על איזה מאפיינים קוליים הכי חשוב לעבוד. נעמיק בנקודה זו בסעיף "כיוונים להמשך מחקר".
2. שימוש באלגוריתמי מציאת "היפר-פרמטרים"
כפי שהראינו בפרק הניסויים, השלב שבו נקבעים ההיפר-פרמטרים הוא שלב שאצלנו לפחות מאד השפיע על רמות הדיוקים השונים.
3. התחשבות בערכי הדיוקים השונים ולא רק ב"דיוק הכללי"
עקב ההבנה כי השאלה "מה המסווג שמהזה הכי טוב מאפיין קולי מסוים" והשאלה "האם אני מוכן לספוג אזעקות שווא באותה מידה כמו סיווג מצב מצוקה אמיתי בתור מצב שגרה?" הינן שאלות שבדיעבד ערכן יסולא בפז. אנו סבורים כי ההתחשבות בערכי ה-RECALL וערכי F1 היא אחת ההחלטות הכי טובות שלקחנו בפרויקט זה, כיוון שבסופו של דבר אזעקת שווא אחת היא אינה טובה, אך עדיף אזעקת שווא אחת על מקרה אמת שלא זיהינו.
4. ביצוע הרצות חוזרות ונשנות על מדגמי קבוצות אקראיים
אחד הדברים שלדעתנו עושים את המחקר שביצענו אמין, הוא העובדה שכל תוצאה שמוצגת בניסויים מבוססת על הממוצע חציון של 5 דגימות אקראיות של המידע, וכסה"כ-25 חלוקות אקראיות לקבוצות שונות. האסימון שבחשיבות ההרצה באופן הזה נפל לנו כאשר ביצענו בהתחלה כ-2 ניסויים פשוטים אחד אחרי השני, וקיבלנו הפרשים של מעל ל-3% בשוני בתוצאות. (פירוט נוסף על אופן הדגימה והחלוקה לקבוצות בפרק על מבנה המערכת).
5. השימוש ב"החטאות קרובות"
למי שקורא רק את הסיכום- החטאות קרובות (Near Misses) הינן דוגמאות בעלות סיווג שלילי אשר דומות מאד במאפייניהן לדוגמאות בעלות סיווג חיובי (פירוק נוסף בפרקים קודמים). **להערכתנו** השימוש בדוגמאות אלה איפשר לנו להגיע לדיוקים גבוהים על אף שלא היו לנו הרבה דוגמאות חיוביות או שליליות (לא הרבה – כלומר פחות מאלף), ע"י כך שבעצם ההחטאות הקרובות תיחמו למסווג שלנו את הצורה של הסיווג החיובי.
6. הצמדות לפתרון הפשוט
מעבר לחיפוש אחר המודלים המדויקים ביותר עם הפרמטרים שיניבו את התוצאות הטובות ביותר, בהתייחסותנו לכמות המידע אותו אנו מעבדים ולרזולוציית הנתונים אותם אנו מחלצים, דאגנו תמיד לבצע השוואה של איכות התוצאות ביחס לגדלי המידע שאנו מעבדים, ובאם שינוי במימדי המידע לא היווה שינוי באיכות התוצאות אז היינו בוחרים להיצמד לעקרון "התער של אוקס" ולבחור בעבודה עם פחות נתונים מיותרים.

4. כיוונים להמשך המחקר

1. שינוי השאלה הכללית ששאלנו לשאלה ספציפית יותר
במקום לשאול "מה המאפיינים הכי נפוצים במקרי סכנה פיזית", עדיף לשאול "איזה מאפיינים צריך לזהות כדי למנוע מקרה מסוג מסוים" (למשל: אונס, הילכדות בשריפה וכו'). הסיבה לכך היא שהגענו למסקנה שהשאלה ששאלנו אינה יעילה דיה כדי לעזור בהכלת מצב מסוכן בצורה מהירה. במידה ואנו מזהים מאפיין מסוים, או מספר מאפיינים, נוכל רק להכווין לכך שככל הנראה אדם נמצא במצוקה כלשהי, אך כדי באמת לעזור לרשויות או לקרובי משפחה לפעול לפתרון באופן יעיל יותר, כדאי שנספק להם מידע נוסף בנוגע לתרחיש שבו נמצא האדם שבמצוקה.
2. יצירת הקשר
אחד הדברים העיקריים שהצגנו בתחילת העבודה, הוא שחסר לנו ההקשר הרחב שבו אנו מזהים מאפיין קולי של מצוקה פיזית מיידית. (לדוגמא: זיהיתי צעקה- האם היא קרתה עקב זיהוי חרק ברחוב או עקב מצוקה אמיתית...). העבודה שהכנו מעולה בתור בסיס, אך מכאן דרוש לחשוב על פתרונות שיכולים ליצור לנו הקשר. אחד הפתרונות שכדאי לחקור הינו "יצירת חלון מקדים שגם בו מנתחים קולות, ונעזרים בו על מנת ליצור הקשר", או "שילוב עם למידה ממוחשבת שמתבססת על הקלטת וידאו" או על "נתוני מיקום", או כל פיתרון אחר שישלב ביניהם. בכל מקרה, מה שבטוח הוא שניתן לשפר את המטרה שלנו ע"י חשיבה על פתרונות שיספקו הקשר רחב לסיטואציה שבה נמצאו מאפייני המצוקה.
3. ציין פתרונות אפשריים ופתח נקודה זו. למשל למצב נקודה מסוים.
חשיבות שיתוף הפעולה עם ארגונים ורשויות רלוונטיים
אנו מזהים אינטרס משותף בנושא שחקרנו, כאשר לנו יש את הכלים הטכניים להתקדם במחקר, אך חסרים לנו הנתונים הדרושים על מנת לבנות מערכת שפועלת באופן רציני ויעיל. נתונים אלה ניתנים להשגה במידה ונדע להציג את האינטרס המשותף בפני מקבלי החלטות במקומות הרלוונטיים. התשובה שקיבלנו ממשטרת ישראל טענה כי כדי לספק לנו את המידע יידרשו משאבים רבים. תשובה זו נוטעת בנו תקווה כי ניתן להקצות משאבים ברגע שהנושא מועבר להחלטת בכיר אשר מבין את פוטנציאל השת"פ. בנוסף ארגוני מחלצים לבטח ישמחו לספק ללקוחותיהם מערכת את תוכל לעזור להם לזהות מצבי מצוקה בטבע.
4. כיוונים נוספים מבחינת המידע המועבר לתהליך הלמידה
אנו מעריכים כי ניתן בנוסף לצלילים המאפיינים אנשים במצוקה פיזית, להכניס לתהליך הסיווג גם מילים ספציפיות אשר מעידות על מצוקה, ובנוסף להוסיף רעשי רקע רנדומליים לדגימות החיוביות שלנו כדי לגרום למסווג לעבוד טוב יותר בכל סביבה עם מאפיינים אקוסטיים משלה
5. מתן אומדן יותר מדויק לגבי מצבו המנטלי של הדובר
מעבר להערכה בינארית אם אדם נמצא במצוקה ניתן אפשרות למדוד את מידת המצוקה הנפשית שבה נמצא הדובר על מנת לאפשר קדימות עבור מקרים יותר דחופים. (קיימת קורלציה בין מחלות נפשיות מסוימות לאופן דיבור).

6. ניתוח מאפיינים נוספים בהקלטת הדובר

ניתוח של מאפיינים נוספים לדוגמא, רעשי רקע יאפשרו לנו לקבל יותר מידע על מיקומו של הדובר (רעשי תחבורה לדוגמה לשם הבחנה כי הדובר מתקשר מחוץ לביתו) אבחון רעשי רקע נוספים כגון רעשי יריות, אנשים נוספים ברקע יאפשרו לנו לקבל מידע על הסיטואציה בה נמצא הדובר.

7. עדכון הדוגמאות השליליות לדוגמאות ספציפיות יותר

ניתן להחליף את מאגר הדוגמאות השליליות שלנו בשיחות טלפון שאינן מוגדרות כמצוקה- דבר זה ייתן למסווגים שלנו מידע קרוב ורלוונטי יותר באשר לאנשים שאינם נמצאים במצב של מצוקה, ויועיל יותר לתהליך הלמידה מאשר העברת "צלילי רקע מהסביבה" בתור סיווגים שליליים. (לא בוצע בפרויקט זה כיוון שהבנו זאת מאוחר מדי).

5. סיכום

במהלך הפרויקט באנו לענות על השאלה: "האם ניתן לזהות אדם הנמצא במצוקה על סמך הקלטה קולית שלו?". במענה על השאלה הזאת ניסינו לפרק את השאלה הזאת לתתי שאלות בינאריות בהן ניסינו למצוא האם בקלטה אנו מוצאים מאפיין מסוים בשיחה או לא, מאפיינים כדוגמת: בכי, צעקה, התנשפות וכדומה. על מנת לגלות האם מאפיין מסוים נמצא עלינו היה לבנות מסווג לכל אחד מהמאפיינים.

תהליך בניית המסווג נעשה דרך ניסויים המשווים בין כמה מהמודלים היותר מוכרים לבין המודל הסדרתי של קראס שעפ"י מאמרים שונים שקראנו נותן תוצאות יעילות על ניתוח של קטעי קול. ובאופן לא מפתיע, קיבלנו את התוצאות המדויקות הטובות במודל של קראס. ניסויים יותר פרטניים נכנסו לעומק הדיוק של המודל ע"י בחינה של הפרמטרים של המודל ושל הגדלים של וקטורי התכונות ואת גודלן של קבוצות האימון. דרך תהליכי הניסוי הללו הצלחנו לבנות מסווג בינארי שיאפשר לתת לנו הערכה לקיומה של כל תכונה. את המסווג הכללי בנינו תוך שימוש במסווגים של כל המאפיינים ע"י שימוש בשקלול סך התוצאות.

בסופו של דבר המסווג שלנו ניבה את התשובה לכל **מקטע** קצר (3 שניות) בקובץ השמע שניתן לו, כאשר המטרה היא **להבליט סממנים של מצוקה פיזית מיידי**. המסווג שלנו הצליח לסווג באופן חיובי את המקטע שבו אחד הנערים ז"ל צועק בזמן החטיפה, אך עדיין להערכתנו ניתן לשפר רבות את דיוק המסווגים על ידי הכיוונים שהעלינו למעלה, כאשר הדבר הכי בסיסי הינו קיום שת"פ עם רשויות החוק וקבלת גישה למאגרי מידע עם שיחות חירום של אנשים במצוקה.

תם ולא נשלם...

ראשית נרצה להודות למנחה שלנו ולמרצה הקורס, אשר עזרו לנו בהתמודדות עם דילמות טכניות, ועזרו למקד אותנו בפרויקט הראשון שאנו עושים בטכניון. אנו שמחים כי ניתנה לנו האפשרות ללמוד כלכך הרבה על בינה מלאכותית כחלק מנושא שקרוב לליבנו. הפרויקט שלנו רותם את הבינה המלאכותית לטובת ייעול עבודת מוקדי החירום, ואנו גאים בו מאד.

נספחים

מקורות מידע

כתובת למקור	תיאור
https://he.wikipedia.org/wiki/%D7%A8%D7%A6%D7%97%D7%A9%D7%9C%D7%95%D7%A9%D7%AA%D7%94%D7%A0%D7%A2%D7%A8%D7%99%D7%9D	ערך ויקיפדיה על חטיפת שלושת הנערים
https://docs.google.com/forms/d/1nAr1Vg xenk4wCI_bTRoS_6Jgiux_NWxietiGnFdRhU/edit#response=ACYDBNgaVpHUGkV2NDCX24ckYFdAD8Pm_jlUcigfeD3hQy-QaveBIFrV2Uth-gmcsBBUjAo	הסקר שביצענו לטובת תיעדוף מאפיינים קוליים
https://research.google.com/audioset/dataset/screaming.html	מאגר קולי של צעקות מחברת גוגל (מבוסס YouTube)
https://annotator.freesound.org/fsd/explore/%252Fm%252F09l8g/?page=882 https://www.shockwave-sound.com/free-sound-effects/scream-sounds https://data.vision.ee.ethz.ch/cvl/ae_dataset/ https://freesound.org/search/?q=scream	מאגרי מידע של קבצי השמע שהשתמשו בהם
https://vox.rocks/resources/wav-vs-mp3	חשיבות בחירת קבצי שמע בפורמט WAV על פני mp3
https://chrome.google.com/webstore/detail/batch-link-downloader/aiahkbnnpafepcgnhhecilboebmmolnn https://chrome.google.com/webstore/detail/simple-mass-downloader/abdkkegmcbiomijcbdaodaflgehfffed	כלים חצי-אוטומטיים להורדת מוזיקה מדפי אינטרנט
https://freesound.org/docs/api/authentication.html	דליית מידע ע"י API מאתר עם קבצי שמע
https://www.nch.com.au/splitter/index.html	כלי לחלוקת קבצי שמע למקטעי זמן שווים באופן אוטומטי
<p>API design for machine learning software: experiences from the scikit-learn project, Buitinck <i>et al.</i>, 2013</p> <p>https://scikit-learn.org/stable/</p> <p>https://librosa.github.io/librosa/index.html</p> <p>https://pandas.pydata.org/</p> <p>https://www.tensorflow.org/</p> <p>https://docs.python.org/3/library/pathlib.html</p> <p>https://github.com/maxpumperla/hyperas</p>	ספריות בשימוש (כולל כתובות שהספריות ביקשו שנצטט במקרה שמשתמשים בהן)

https://github.com/hyperopt/hyperopt	
http://www.cs.technion.ac.il/he/people/shaulm/ http://www.cs.technion.ac.il/he/people/yanemcovsky/	<p>אנשים שתרמו באופן ישיר לאלגוריתמים שהשתמשו בהם</p> <ul style="list-style-type: none"> מתנצלים מראש אם שכחנו מישהו. אנא פנה/י אלינו ונוסיף/י לרשימה
https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28 https://en.wikipedia.org/wiki/F1_score	כתבות שתרמו להבנה שלנו בנושאים רלוונטיים לפרויקט
https://github.com/ https://keras.io/	ניהול גרסאות
https://www.dlology.com/blog/quick-notes-on-how-to-choose-optimizer-in-keras/ https://medium.com/octavian-ai/which-optimizer-and-learning-rate-should-i-use-for-deep-learning-5acb418f9b2 https://towardsdatascience.com/types-of-optimization-algorithms-used-in-neural-networks-and-ways-to-optimize-gradient-95ae5d39529f <p>ההבדל בין כל פונקציית תמצות:</p> https://www.dlology.com/blog/how-to-choose-last-layer-activation-and-loss-function/ https://riptutorial.com/keras/topic/10674/custom-loss-function-and-metrics-in-keras <p>משווה בין פונקציות תמצות שונות:</p> https://towardsdatascience.com/custom-loss-functions-for-deep-learning-predicting-home-values-with-keras-for-r-532c9e098d1f https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html <p>דוגמא לפונקציית תמצות שניתן לבנות ידנית:</p> https://riptutorial.com/keras/example/32022/euclidean-distance-loss https://towardsdatascience.com/learn-how-recurrent-neural-networks-work-84e975feaaf7 https://towardsdatascience.com/beginners-ask-how-many-hidden-layers-neurons-to-use-in-artificial-neural-networks-51466afa0d3e	המדריך למודל הסדרתי של קראס

<https://stackoverflow.com/questions/36950394/how-to-decide-the-size-of-layers-in-keras-dense-method>

<https://dzubo.github.io/machine-learning/2017/05/25/increasing-model-accuracy-by-tuning-parameters.html>

<https://towardsdatascience.com/a-guide-to-an-efficient-way-to-build-neural-network-architectures-part-i-hyper-parameter-8129009f131b>
(<https://github.com/maxpumperla/hyperas>

בקשה למידע ממשטרת ישראל



- בלמס -
לשכת
מדור תלונות ציבור ארצי וחופש מידע
המחפשי
טלפון: 02-5426020
פקס: 02-5898823
כ"ד ניסן, תשע"ט , 29 אפריל, 2019
סימוכין: 40109619
מס' תיק פניות (מש"י): 55080
מס' חופש מידע 285/19

לכבוד
מר לב טוניק
באמצעות דוא"ל:
tuniklev@gmail.com
שלום רב,

הנדון: בקשת חופש מידע 295/19: קבלת הקלטות שיחות טלפון למוקד המשטרה
בסימוכין ל: פנייתך מיום 14/04/2019

1. בפנייתך שבסימוכין ביקשת לקבל לידיך המידע הבא:
א. אלף הקלטות של שיחות טלפון אשר המוקד הטלפוני הגדירם כשיחות של אנשים במצוקה.
ב. אלף הקלטות של שיחות טלפון אשר המוקד הגדירם כשיחות שאינן שיחות במצוקה.
2. לאחר שבחנו את הבקשה ושקלנו את השיקולים הרלבנטיים החלטנו לדחותה.
3. חשיפת ההקלטות תהווה פגיעה בפרטיות, שכן המתקשרים מוסרים פרטים מזהים אודותם, וזאת בהתאם לסעיף 9 (א) (3) לחוק חופש המידע (התשנ"ח-1998) (להלן: "החוק"), הקובע כי רשות ציבורית לא תמסור מידע אשר בגילויי יש פגיעה בפרטיות.
4. זאת ועוד, על מנת להיענות לבקשתך ולשלוח הנתונים המבוקשים- יהא כרוך בהקצאת משאבים בלתי סבירה. כפי האמור בסעיף 8 (1) לחוק, הקובע כי רשות ציבורית רשאית לדחות בקשה לקבלת מידע כאשר הטיפול בה מצריך הקצאת משאבים בלתי סבירה.
5. מובא לידיעתך כי הנך רשאי לערור כנגד החלטה זו בפני ביהמ"ש המחוזי בירושלים, בשבתו כביהמ"ש לעניינים מנהליים, תוך 45 יום ממועד קבלת הודעה זו.

בברכה,
רפ"ק יניב ארקוס, עו"ד
קצין מדור תלונות ציבור ארצי
ע/ הממונה על חופש המידע

פוסט בנוגע לאופן נרמול המידע

Lev Tunik

4 באוגוסט ב' 17:49

אשמח לכמה טיפים מהנסיון שלכם... האם קודם פיצלתם את הנתונים שלכם למאגר של טסט/ולידציה ואז נרמלתם בנפרד? או שקודם נרמלתם את הכל ורק אז פיצלתם? ...הגוגל לא אחיד בדעותיו בנושא.

7 תגובות נראה על ידי 109

תגובה

לייק

Itai Peri

לדמות מידע חדש שמגיע למערכת (נניח בזמן אמת, כשהמערכת באוויר), ואותו הרי אתה מנרמל לפי הנרמול שנקבע ע"י ה-train set.

לייק

השב 2 שב'

Itai Peri

validation set ואת ה-validation set מנרמלים לפי הנרמול של ה-train set.

לייק

השב 2 שב'

Shaul Markovitch

בתגובה שלו את המושג validation set במושג test set. קבוצת המבחן מדמה אובייקטים שיגיעו למסווג בעתיד, ולכן אינה יכולה לשמש לשום מטרה בתהליך הלמידה (כולל נרמליזציה). קבוצת הולידציה היא חלק מקבוצת הדוגמאות המתווגג... ראה עוד

לייק

השב 2 שב'

Leon Anavy

רק שיהיה ברור, כשאומרים לפצל ולנרמל בנפרד הכוונה היא Itai Peri שמנרמלים את נתוני האימון ואז כל תצפית חדשה מנרמלים לפי אותם פרמטרים של נתוני האימון. נניח שהנרמול הוא לחסר ממוצע ולחלק בסטיית תקן. אז שומרים את הערכים הללו מנתוני האימון לפני הנרמול ואז את נתוני המבחן מנרמלים ע"י חיסור וחילוק בערכים ששמרנו.

לייק

השב 2 שב' נערך

Itai Peri

לדמות את ה-test set כמה שניתן. כי המטרה שלו היא לשמש כ-test set לקביעת ההיפר פרמטרים. לכן עד כמה שאני יודע קביעת הנרמול (נניח ממוצע וסטיית תקן) צריך להתבצע בלעדית, לא?

לייק

השב 2 שב'

Itai Peri

כן, בדיוק Leon Anavy

לייק

השב 2 שב'

Shaul Markovitch

לנרמל לפי קבוצה הכוללת את קבוצת המבחן - זה לא לגיטימי ומזהם את תהליך הלמידה. לנרמל יחד עם קבוצת הולידציה - זה לגיטימי, אבל, כפי שתקנת אותי, עלול לפגוע בתהליך כיוונון הפרמטרים. לסיכום עדיף לכוון רק לפי קבוצת האימון. שים לב שפעמים רבות אנחנו מכוונים פרמטרים בקרוס ולידציה ואז המקפידים ינרמלו מחדש בכל פולד.

לייק

השב 2 שב'

אתגרים משניים שבחרנו להוסיף כנספח

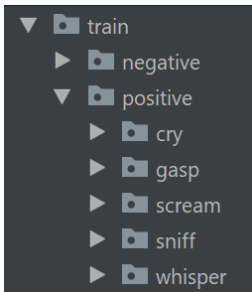
1. השתמשנו ב git בתור מערכת לניהול גירסאות.
2. בדיקת שרתי kaggle ושרת google cloud בתור מכוונות וירטואליות

הקמת סביבת עבודה לפרויקט

1. הפרויקט צריך להתבצע בסביבת עבודה של windows
2. רשימת הספריות הדרושות להרצת הפרויקט נמצאת בקובץ `libs.txt` בתקיה הראשית של הפרויקט.

הסבר על מבנה תיקיית הפרויקט

הוסבר במהלך העבודה במיקומים רלוונטיים כיצד לשחזר תוצאות ניסויים וכו', כאן נפרט בקצרה על מבנה התיקיות כדי להקל על מי שפותח את הפרויקט ורוצה להבין מה נמצא איפה

משמעות	נתיב יחסי
פה שמורים קבצי ה-CSV, אשר שימשו אותנו לשימור וקטורי מאפיינים. כיוון שהתעסקנו עם מוזיקה, לא יכלנו לשמור 3 אלף קטעים על המחשב... קיים פה כפל של קבצים דומים עם שמות שונים פשוט כי כך עבד האלגוריתם. ניתן לייעל זאת בהמשך	csv
בנתיב זה שמרנו לכל קובץ שניתן למסווג הסופי שלנו את תוצאות הניבוי. קובץ ספציפי מתוך תיקיה זו נפתח אוטומטית בסוף הרצת אלגוריתם הניבוי	csv/prediction_results
בתקיה זו נמצאים מקטעים שווים באורכם של קטע השמע שניתן למסווג שלנו- כלומר אם נתת לנו לנבא קובץ באורך 30 שניות, הוא ישמור בתקיה זו 10 קבצים, כ"א באורך 3 שניות	examined_files
בתקיה זו נמצאים המודלים הכי טובים שהצלחנו לאמן, ממנה נטענים המודלים בתהליך הניבוי	models/best_from_final_models
בתקיה זו השתמשנו כדי לשמור תוצאות של הניסויים שהצגנו בעבודה זו	Results
כאן נשמרים האובייקטים שעוזרים לנרמל את הקבצים שניתנים למסווג הסופי	saved_scalers
לכאן מועתק כל קובץ שניתן למנבא הסופי שלנו, מבלי שנעשה עליו עיבוד כלשהו	source_files
בנתיב זה שמרנו את הקבצים שלימדנו עליהם את המסווגים- כלומר בתוכו היו שמורים כל קבצי השמע בפורמט WAV	<p>Train</p> 

aux_functions.py	פונקציות שתומכות אלגוריתמים שונים
check_sequential_categorical.py	פונקציה שעזרה לבדק אחוזי דיוק של מסווג בעל פונקציית categorical_crossentropy שהצגנו בחלק הניסויים למידה
evaluate_genericClf.py	קובץ שיכול לבנות ולבדק מסווגים
libs.txt	כלל הספריות שהיו בשימוש במהלך הפרויקט
models.py	כאן אנחנו מבצעים את כיוונון ההיפר-פרמטרים
predictor_ver_2.py	הקובץ שיש להפעיל כדי להריץ את המסווג הסופי

נספחים לניסויים

המודלים שנבדקו בניסויים

להלן המודלים שנבדקו בניסויים:

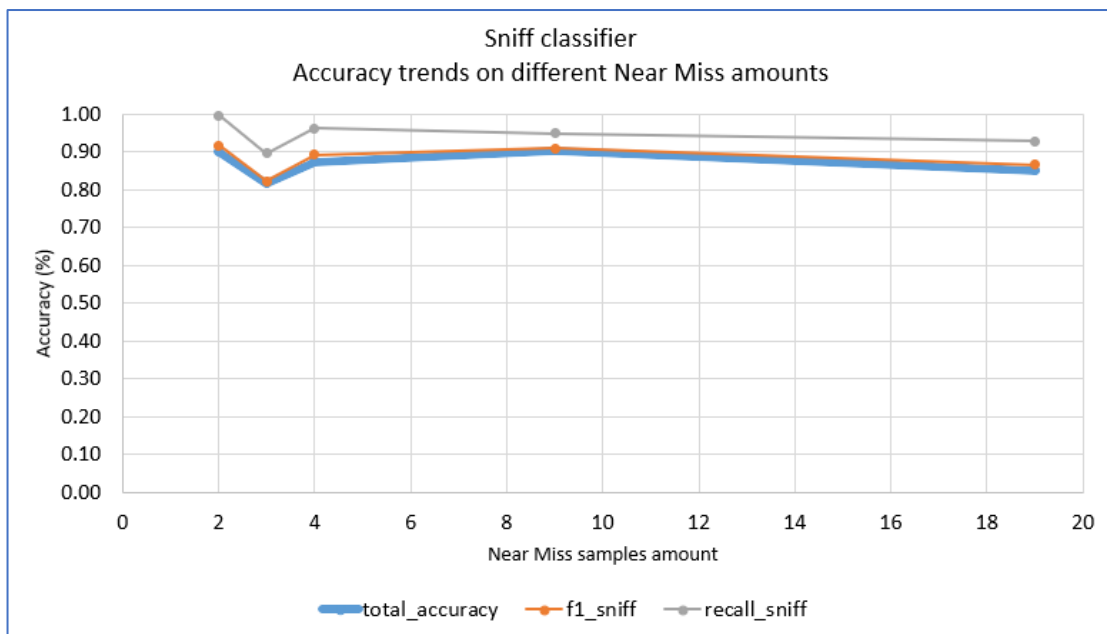
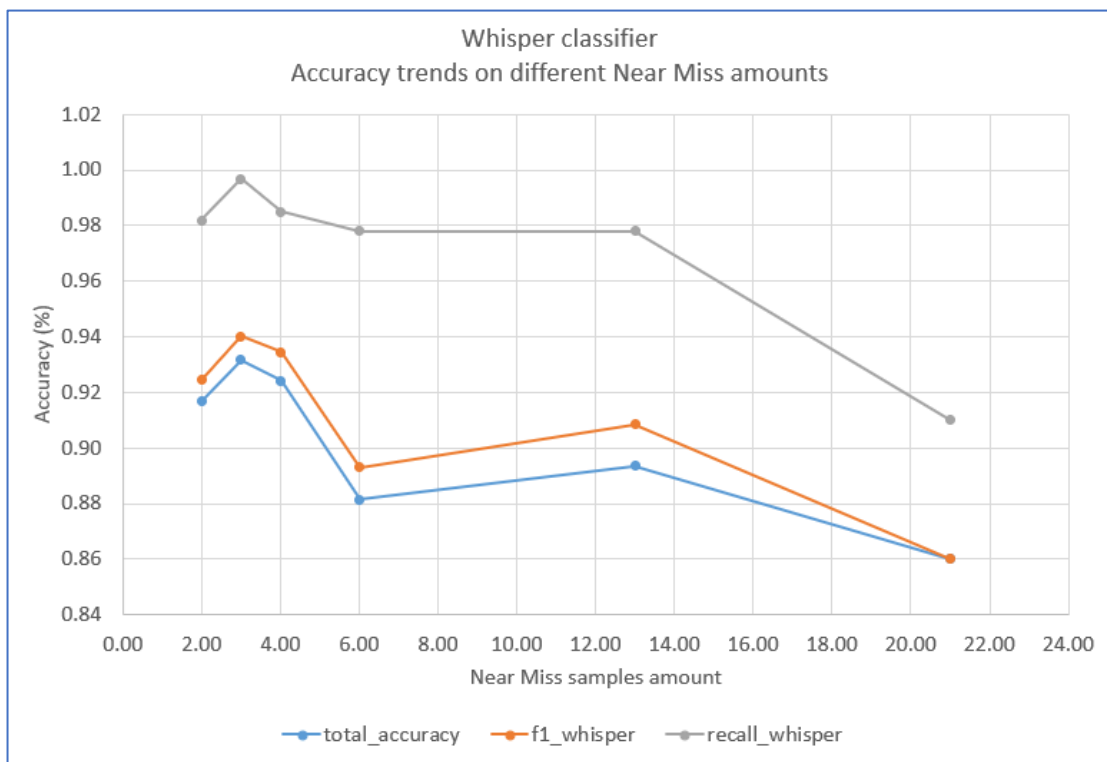
```
best_model = load_bestModel()
id3 = Id3Estimator()
knn = KNeighborsClassifier(3)
svc1 = SVC(kernel="linear", C=0.025)
svc2 = SVC(gamma=2, C=1)
gauss = GaussianProcessClassifier(1.0 * RBF(1.0))
decisionTrees = DecisionTreeClassifier(max_depth=25)
rand = RandomForestClassifier(max_depth=25, n_estimators=10, max_features=1)
adaboost = AdaBoostClassifier()
gaussNB = GaussianNB()
mlp = MLPClassifier(alpha=1, max_iter=100)
models_to_check = [best_model, id3, knn, svc1, svc1, svc2, gauss, decisionTrees, rand, gaussNB, adaboost, mlp]
```

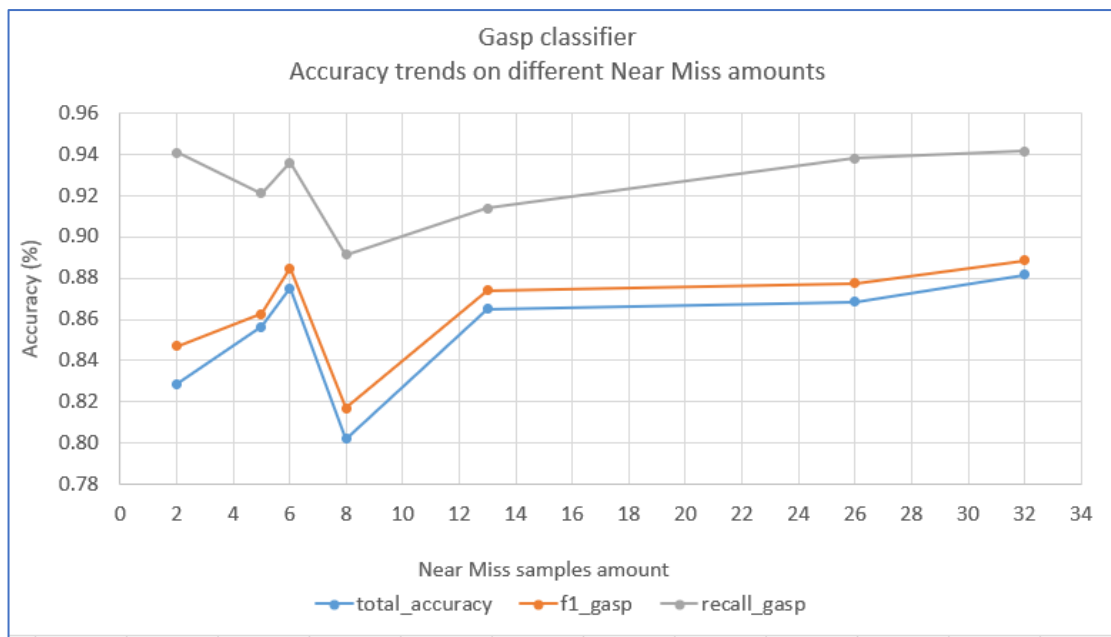
- Best_model - הינו מודל סדרתי מספריית keras אשר לאחר מחקר נבחר למסווג הבינארי עבור "צעקה":

Model: "sequential_46"

Layer (type)	Output Shape	Param #
dense_226 (Dense)	(None, 32)	864
dense_227 (Dense)	(None, 512)	16896
dense_228 (Dense)	(None, 128)	65664
dense_229 (Dense)	(None, 64)	8256
dense_230 (Dense)	(None, 1)	65

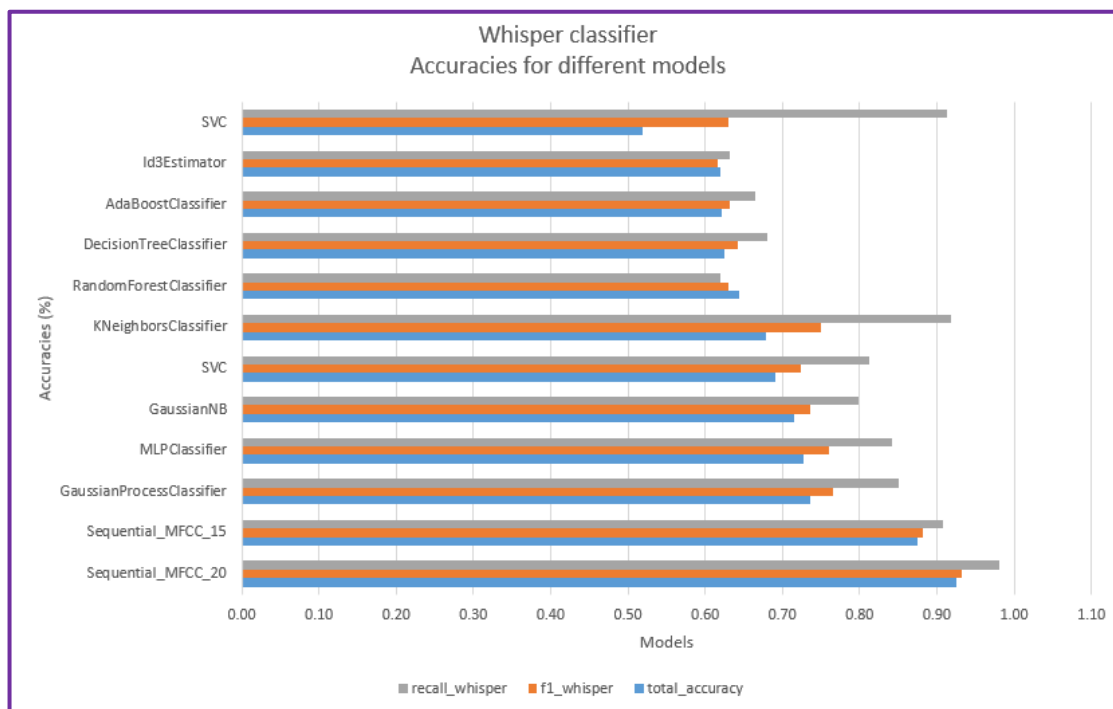
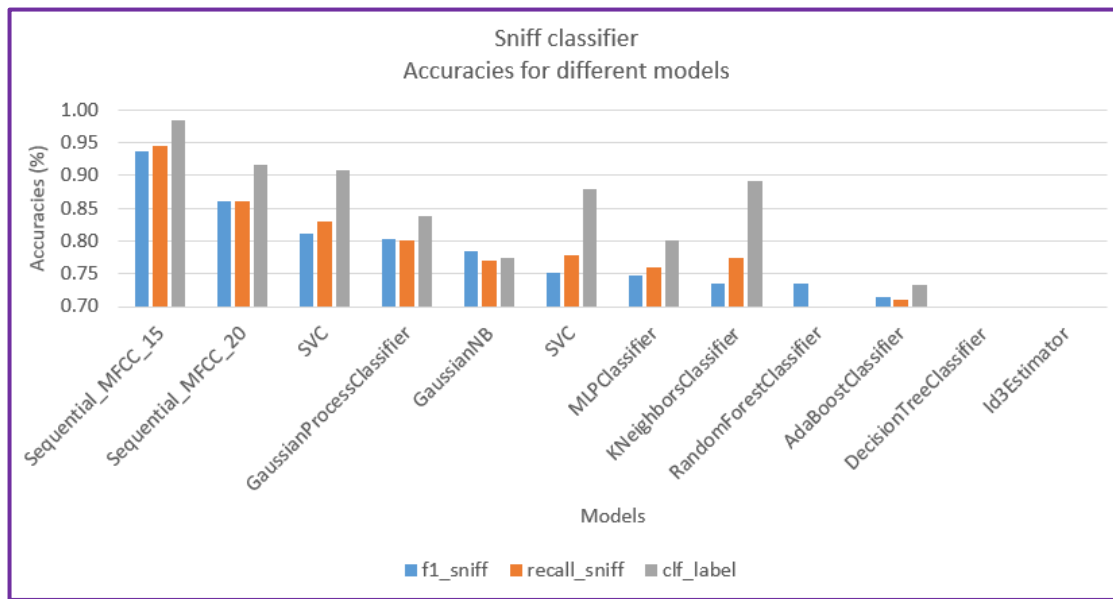
נספח לניסוי על כמותי ה"החטאות הקרובות" – סיווגים שונים עם כמויות שונות של "החטאות קרובות"

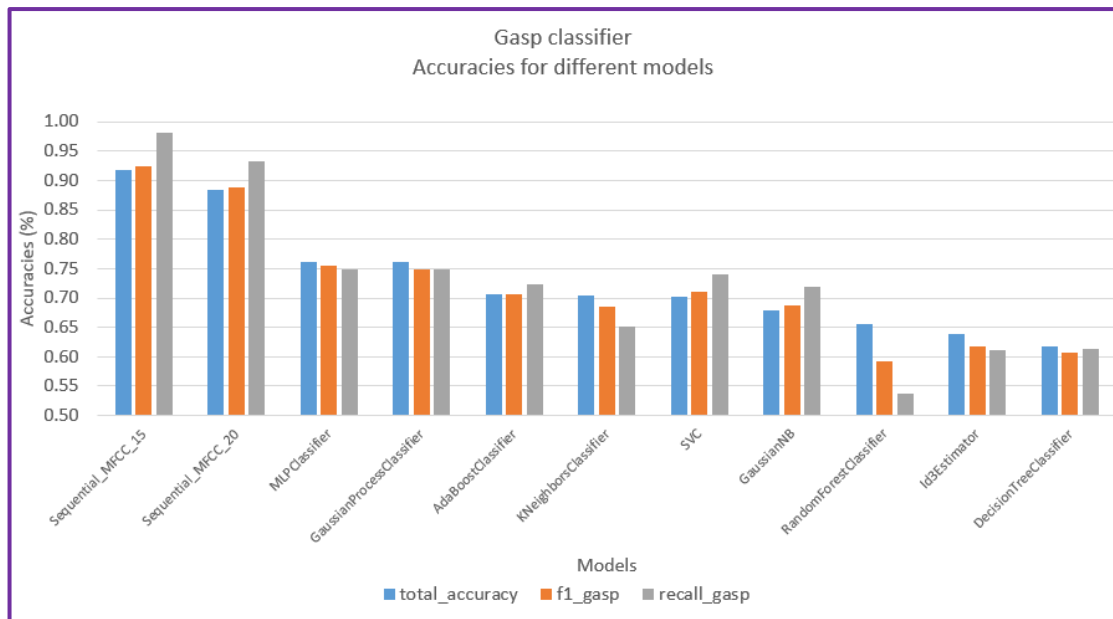




הערות על הגרפים הנ"ל: להערכתנו עקב כמויות קטנות - (ביחס לכמויות של מעל 100 בגרפים של המסווג "צעקה") של "החטאות קרובות" לא ניתן לקבוע בוודאות האם במידה והיו יותר זיהויים חיוביים לסיווג "לחישה" אז היה ניתן להגיע לתוצאות דיוק גבוהות יותר. כלומר הכוונה בפרויקט המשך למשל תהיה להשיג יותר דגימות מכל סוג ובכך להמשיך ולבדק האם "החטאות קרובות" מעלות את הדיוק....".

נספח לניסוי על דיוקי מודלים שונים – השוואת תוצאות של מודלים שונים על סיווגים שונים





הערות על הגרפים הנ"ל: להפתעתנו גילינו כי בהחלט קיימים סיווגים אשר מביאים תוצאות גבוהות יותר ביותר מ-0,04 (הבדל משמעותי מ-0.88 ל-0.92 אחוזי דיוק למשל) כאשר רמת הרזולוציה שונה מ-MFCC 20. למשל מסווג "שאיפה" (sniff) דורש רמת רזולוציה של 12 לטובת שיפור הדיוק, בעוד שמסווג "התנשפות" (gasp) דורש רזולוציה של 15 לטובת שיפור הדיוק.

הערה על מתודולוגיית הניסוי לקביעת כמות ה-mfcc המיטבית

עקב מגבלת כוח מחשוב- נאלצנו להגביל את ה-fit של המודל Sequential לפעם אחת בלבד, וללא מציאת ההיפר פרמטרים הכי טובים עבור כל כמות של mfcc. כלומר הדיוק של מודל זה אינו מייצג באמת את המודל הכי טוב שלנו. עם זאת מתודולוגיה זו אינה פוגמת במטרת הניסוי שכן הדבר היחידי שעניין אותנו בניסוי זה הינן המגמות של הדיוק עבור ערכי mfcc שונים. לסיכום, כיוון שאנו עקביים עם דרך הפעולה בכל האיטרציות של כמויות ה-mfcc תוצאות ומסקנות הניסוי עדיין תקפות.