אפקה המכללה האקדמית להנדסה בתל־אביב

בית הספר להנדסת חשמל

עיבוד אותות אקראיים

עבודת מחשב 2

זיהוי מין על בסיס דגימות קול

תוכן עניינים

הצעת הפרויקט

מוטיבציה

הצורך בזיהוי מין הדובר הינו צורך נפוץ מאז תחילת עידן התקשורת המודרנית. כאשר אנו מתקשרים ללא אמצעים ויזואליים, אנו נאלצים לשער את מינו של הצד השני באותה התקשרות, תהליך אותו המוח שלנו מסוגל לבצע בדיוק רב (כיום, עם התפתחות החברה המודרנית במאה ה-21, אנו עדים למקרים בהם גם כאשר קיימים אמצעים ויזואליים, קיים קושי לזהות את מין הדובר).

זיהוי מין הדובר הינו כלי שימושי מאוד בחיי היום יום שלנו, החל מטלמרקטינג ועד לאמצעי אבטחה (היכולים להשתמש בזיהוי זה כסיווג ראשוני).

רקע תיאורטי נדרש

1. הגישה שנבחרה לניתוח וסיווג המידע הינה MFCC – Mel Frequency Cepstral coefficients, אשר נפוצה בעיבוד וסיווג אותות דיבור.
2. נדרש סט נתונים רחב המכיל דגימות קוליות לפי מין, נחלק ל-train/dev/test.
3. ידע קודם מקורסי המבוא של עיבוד אותות -
   a. תיאורית הדגימה.
   b. מעבר מאותות אנלוגים לאותות דיגיטליים.
4. הכרת תכונות הקול האנושי במישור הזמן והתדר -
   a. תחום התדרים המוגדר כ-"קול אנושי".
   b. עוצמת הקול (אמפליטודה).
   c. רציפות ה-"דיבור" בזמן.
5. התמרות ממישור הזמן למישור התדר, ולהפך.
6. שימוש במסננים לקבלת מקדמים המתארים את אות הדיבור.
7. שימוש ב-SVM לטובת הפרדה בין קבוצות המידע אותן נסווג.
8. כיול ה-SVM בעזרת סט הנתונים מקבוצת ה-dev.

הניסויים שיבוצעו

1. עיבוד המידע – שינוים כמות המסננים (מקדמים) בהם נעשה שימוש.
2. שינוי גרעין ה-SVM – לינארי, RBF, פולינומי.
3. כיול ה-SVM לפי סט ה-dev – מציאת ערכי סף (בהתאם לגרעין) בשיטת "grid search"[1], וקביעת הערכים האופטימליים על margin, classification errors.
4. על סמך הניסויים הקודמים, בדיקת המכונה על סט ה-test.

[1] 2003 Last updated: May 19, 2016, Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin - A Practical Guide to Support Vector Classification

רקע תיאורטי

עיבוד אותות דיבור

המטרה בעיבוד אות דיבור היא להשיג דרך נוחה ויעילה לייצוג המידע שנמצא באות הדיבור. בחירה טובה של מרחב המאפיינים תאפשר השגת תוצאות סיווג טובות יותר תוך שימוש במערכות פשוטות יותר. אות הדיבור עצמו אינו מתאים לזיהויי דיבור כיוון שהינו תהליך אקראי, בעל שונות יחסית גבוהה, רועש, וממד גבוה, ולכן נדרש ייצוג אחר של האות כך שיהיה כמה שיותר קומפקטי, וכמה שפחות רגיש לרעש.

תכונות אות דיבור

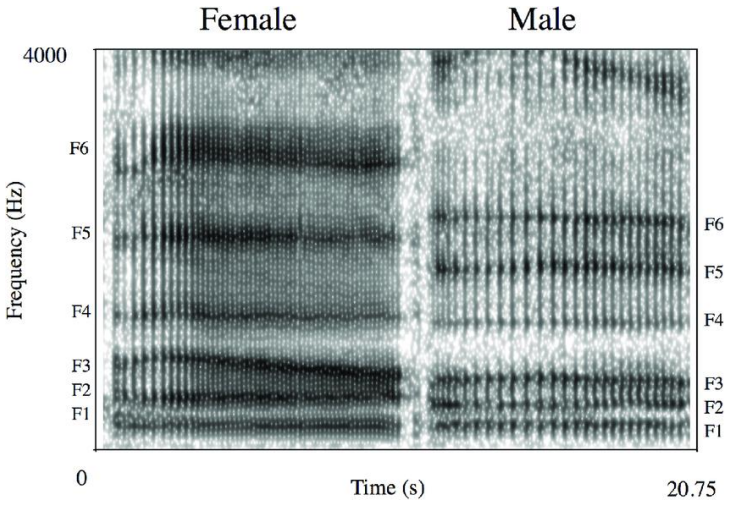
אות דיבור הוא אות שתכנותיו הסטטיסטיות משתנות בזמן (לא סטציונרי), אך בהיבט על חלונות זמן קטנים בתוך אות זה (5-100ms), ניתן לראות שהוא בקירוב בעל מאפיינים סטציונריים. לעומת זאת, בקטעים ארוכי, מאפייני אות דיבור משתנים ומשקפים את צלילי הדיבור השונים אשר נהגו.

מאפיינים במישור הזמן כוללים אנרגיה, אמפליטודה ממוצעת של כל חלון, מעטפת האות, פונקציית האוטו-קורלציה של כל חלון, קצב חציית האפס (חילוף סימן בין שתי דגימות צמודות עוקבות), ועוד. באופן כללי, קול הדיבור האנושי נמצא ברובו בין התדרים 100-6,800Hz.

גברים ונשים

לקול האנושי יש תדר יסוד נמוך - אצל גברים התדר נע בין 85Hz ל-155Hz, אצל נשים בין 165Hz ל-255Hz. בנוסף לתדר היסוד ישנם צלילים עיליים, המשמשים כמרכיבי תדר משמעותיים להבנת דיבור. תדרים קוליים אלה, פורמנטים (Formants), הם תדרי תהודה של מערכת הדיבור האנושית, הכוללת בין היתר את בית הקול וחלל הפה. פורמנטים אלו מתקרבים לתדרים של עד כ-4000Hz.

שני מאפייני הקול העיקריים בין גברים לנשים הם עוצמה ותדר. לקול של אישה יש יותר מרכיבי תדר בהשוואה לגברים. נשים מדברות באוקטבה אחת יותר מגברים. בגרפים שלמטה ניתן לראות שלגברים ישנים מרכיבים תדר רבים בתדרים הנמוכים, בעוד שלקולה של אישה הספקטרום נפרש לתדרים גבוהים יותר.



איור 1 - ספקטוגרמת קול של גברים ונשים

MFCC

תיאוריה

הצעד הראשון בכל מערכת זיהוי דיבור אוטומטית הוא לחלץ תכונות, כלומר לזהות את הרכיבים של אות האודיו הטובים לזיהוי הדובר או התוכן, ולהשליך את כל הקטעים האחרים הנושאים מידע כמו רעשי רקע או קטעים שקטים בהם לא נאמר דבר.

אות קול משתנה כל הזמן, אז כפי שנאמר, אנחנו מניחים שבדגימות קצרות אות הקול לא משתנה הרבה. זו הסיבה שנרצה למסגר את האות בחלונות זמן קטנים, ולעבד כל אחד מהם. במידה והחלון קצר מדי, לא יהיו לנו מספיק דגימות כדי לקבל אומדן ספקטרלי אמין, ובמידה והחלון ארוך מדי, האות ישתנה יותר מדי לאורכו.

חלק מהאוזן האנושית האחראי על "קליטת" הצלילים הנכנסים לאוזן נקרא שבלול. עצמות השמע מרעידות את החלון שבפתחו, כאשר רעידות אלו מרעידות את הנוזל הממלא את השבלול. תנודת הנוזל מניעה את השעריות שעל תאי השערה, ואלו יוצרים פוטנציאל חשמלי המפעיל את תאי העצב של השמיעה. אותם תאי עצב עוברים ישירות למוח. מבנה השבלול מסייע לפרק את צלילי הסביבה לתדרים, אך לא יכול להבחין בהבדל בין שני תדרים במרווחים קרובים. השפעה זו הופכת בולטת יותר ככל שהתדרים עולים. משמעות הדבר היא ששינויים גדולים באנרגיה עשויים שלא להישמע כל כך שונה אם הצליל חזק מלכתחילה.

מימוש

Short Time Fourier Transform

הבחירה ב-STFT מאפשרת לנו לשמור גם על מישור הזמן, וגם על התדר, לעומת FFT המעביר בין מישור הזמן למישור התדר. בעזרת הפעולה בתוך חלון זמן קצר, ובעזרת פונקציית חלון (למשל Hanning), אנו מקבלים פעולה בהתאם לשבלול הרוטט, אשר גורם לפעילות חשמלית בעצבים שונים, המסמנים למוח שישנם תדרים מסוימים. מספר החלונות שיתקבלו בפועל יקבעו את אורך הוקטור שיתקבל כתוצאה.

Mel-spaced Filter bank

בדומה לאוזן האנושית, אשר לא יכולה להבחין בהבדל בין שני תדרים מרווחים קרובים, אנו מייצרים סדרת מסננים, בתחום תדרים נבחר, ומבצעים מכפלה וקטורית של הוקטורים שהתקבלו עם חלקי הצפיפות ספקטרלית של המסננים, כדי לדעת כמה אנרגיה קיימת באזורי תדר שונים.

המסנן הראשון צר מאוד ונותן אינדיקציה לכמות האנרגיה שקיימת ליד 0 הרץ. ככל שהתדרים הולכים וגדלים, המסננים שלנו מתרחבים, כלומר, ככל שאנו מודאגים פחות לגבי יכולת ההבחנה שלנו. אנו ברזולוציה פחות טובה. סקלת Mel קובעת עבורנו בפועל את רוחב המסננים.

$$M(f) = 1125 \cdot \ln\left(1 + \frac{f}{700}\right) \qquad M^{-1}(f) = 700 \cdot \exp\left(\frac{m}{1125}\right) - 1$$

המרה ממישור התדר למישור Mel, ולהפך

פעולת הדחיסה הזו גורמת לוקטור שקיבלנו שלנו להתאים יותר למה שבני אדם שומעים בפועל.

Discrete Cosine Transform

השלב האחרון הוא חישוב ה-DCT, אשר משמש אותנו כשלב לצמצום כמות המידע הנדרשת.
התמרה זו בעלת יכולת טובה של "דחיסת ספקטרום", כלומר רוב האנרגיה מרוכזת במספר קטן של מקדמים, בהשוואה ל-DFT למשל. בנוסף, ההתמרה מבצעת דה-קורלציה למידע, כלומר מקטינה את האוטו-קורלציה של וקטורי הפרמטרים.

לאחר שלבים אלו, אנו מקבלים מטריצת פרמטרים עבור כל אות קול, המתארת מאפייני תדר ועוצמה בחלונות זמן שונים, וזאת בהתאמה לעקרונות הפעולה של האוזן האנושית.

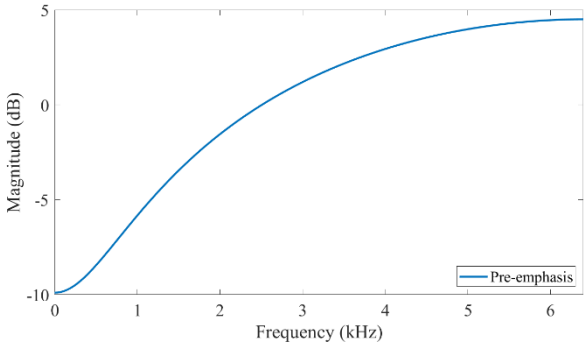
כפי שידוע לנו, מטרתה של למידת המכונה היא לחקות את המוח האנושי, כך שבחיקוי אופן השמיעה שלנו, אנו צופים לקבל תוצאות טובות באימון המודל ובזיהויים עתידיים.

מסננים וכלים נוספים

מסנן Pre-Emphasis

מסנן HPF (סדר ראשון), אשר היה בשימוש בעיקר בעבר במערכות לעיבוד אותות דיבור.

$$y(t) = x(t) - \alpha \cdot x(t-1)$$



איור 2 - דוגמה למסנן Pre-emphasis

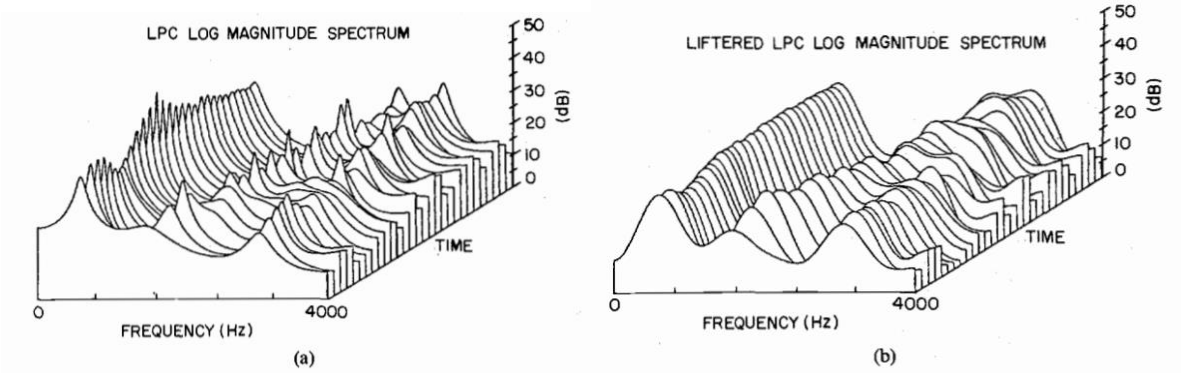
באותות דיבור, רוב האנרגיה במישור התדר מרוכזת בחלק התחתון – עד 6kHz, ובקירוב ניתן להגיד שההספק דועך ב-2dB/kHz. בעבר, ההבדל ב-דיוק הקיים עבור התדרים הנמוכים, לעומת התדרים הגבוהים, גרר בעיות במימוש התמרת פורייה. עבור המקדם, נהוג לבחור $\alpha = 0.95\ or\ 0.97$.

ההבדל בהספק אינו משפיע על המימוש באלגוריתמי FFT, ולכן בעל תועלת זניחה עד לא קיימת.
בעזרת נרמול המידע נקבל תועלת גדולה יותר, ולכן אין לנו צורך במסנן זה.

מסנן Liftering

מימוש מסנן במישור ה-cepstrum (בו אנו עובדים ב-MFCC). מטרתו של מסנן זה היא להקטין את ערכיהם של הפרמטרים המתקבלים מהמסננים הראשונים (תדרים נמוכים), לעומת ערכיהם של המקדמים בפילטרים מסדר גבוה, אשר קטן יותר מלכתחילה (בהתאם לנאמר ב-Pre-emphasis – בקול אנושי העוצמה בתדר דועכת).

$$C_n' = \left(1 + \frac{L}{2} \cdot \sin\left(\frac{\pi n}{L}\right)\right) \cdot C_n$$



איור 3 - לפני (a) ואחרי (b) מימוש Liftering על מקדמים במישור ה-Cepstrum

המימוש יעיל במיוחד לזיהוי טקסט באותות דיבור, בדגש על אותות רועשים.
אנו נשתמש בערכי ברירת המחדל הקיימים בפונקציה אותה נבחר.

נרמול – Scaling

נרמול המידע הינה פעולת מתמטית פשוטה, המקבלת וקטור של ערכים, ומחזירה אותו כך שהחציון יהיה 0, וסטיית התקן תהיה 1.

$$x' = \frac{x - \mu}{\sigma}$$

מטרת הנרמול היא לאפשר גמישות של המכונה לערכים חדשים, ללא תלות בערך bias מסויים שמסית את וקטור הפרמטרים ללמידה/לחיזוי הניתן למכונה. אנו מאפשרים גמישות זו על ידי "הזזת" כלל וקטורי הפרמטרים לאותו התחום, דבר אשר משמעותי מאוד במכונות אשר נשענות בחישוביהן על המרחקים בין הנקודות המתקבלות מכל וקטור פרמטרים.

כיוון שאנו נעבוד עם SVM, נשתמש בנרמול זה בעת עיבוד המידע ואימון המכונה.

למידת מכונה – Machine Learning

כללי

למידת מכונה היא ענף בעולם מדעי המחשב, המשתמש במידע ובאלגוריתמים מתקדים על מנת לחקות את אופן החשיבה של בני האדם.

קיימים ענפים שלמים בתוך עולם למידת המכונה, ביניהם למידה עמוקה ורשתות נוירונים.
כל ענף מאפשר מידת התערבות שונה של האדם היוצר את המודל, כאשר, למשל, בלמידה עמוקה, המודל יכול לסווג בעצמו את המידע שניתן בשלב הלמידה, ללא צורך ב"תיוג" שלו.

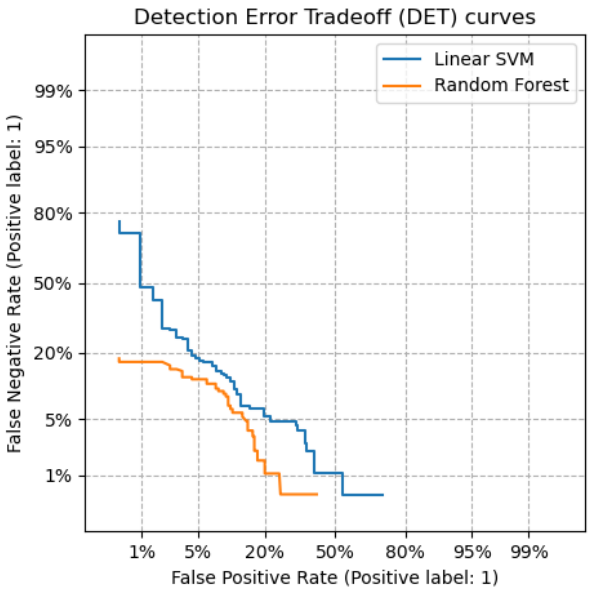
למידת המכונה מתחלקת בעיקרה לכ-3 חלקים:

1. **תהליך ההחלטה** – האלגוריתמים משמשים לחיזוי/סיווג של מידע, על סמך ידע קודם. המודל ידע להשתמש ב-"תבניות" אותן למד בעבר, ויחפש בעזרתן הקשר למידע החדש שניתן.
2. **פונקציית השגיאה** – פונקציה המשמשת כהערכה להצלחת החיזוי/סיווג של המודל
3. **תהליך אופטימיזציה** – כיול המידע שניתן ללמידה באופן אופטימלי, כך שיספק אחוזי חיזוי/סיווג גבוהים יותר מהנתונים כרגע.

מדדי הצלחה ודיוק

Det Curve

גרף המתאר את יחסי הגומלין בין שגיאות מסוג false positive, (סיווג של דגימה מסוג '0' כ-'1'), לבין false negative (סיווג של דגימה מסוג '1' כ-'0').
הגרף מוצג על סקלה של סטיית התקן (נרמול ביחס לסטיית התקן של כל אחת מהתפלגות השגיאות), ולכן מתקבל באופן לינארי ונוח לעבודה. כאשר ישנה משמעות לסוג השגיאה, כלומר יש תעדוף לסוג שגיאה אחד על פני אחר, נקודת העבודה על פני הגרף מהווה מדד לביצועי המודל.

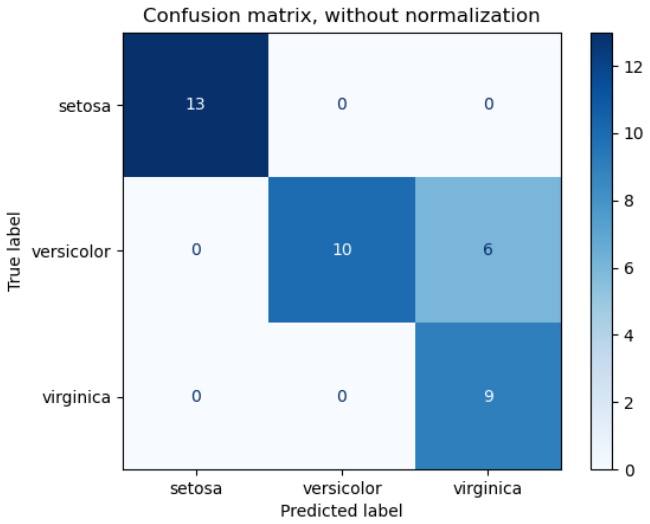


איור 4 - דוגמה ל-Det curve

Confusion Matrix

מטריצה זו מציגה באופן כמותי את תוצאות סיווג המודל על סט נתונים.
המטריצה מתייחסת לסיווג נכון או לא נכון, וגם ל-false positive/negative. במודלים המסווגים בין קבוצות רבות, המטריצה מספקת מידע גם עבור סיווגים לא נכונים בין הקבוצות השונות.

אופן הפעולה של יצירת המטריצה הינה הוספת '1' בכל זיהוי שמתבצע, בתא המתאים לסיווג זה במטריצה. ניתן לקבל מטריצה זו רק כאשר המידע הניתן למודל מסווג מראש (סט dev, test), כך שניתן לדעת האם מדובר בשגיאה, ובאיזו.

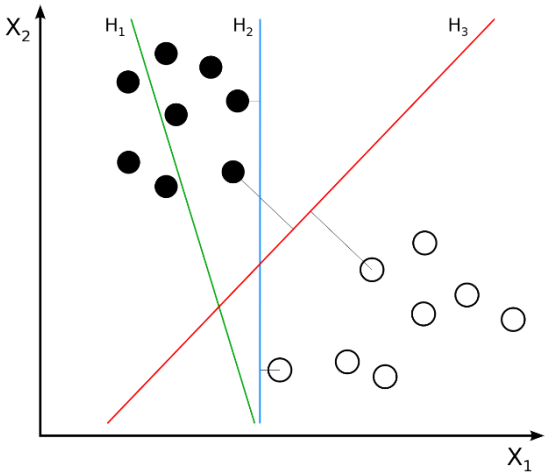


איור 5 - דוגמה ל-Confusion matrix עבור סיווג בין 3 קבוצות

Support vector machines (SVMs)

סט של מתודות למידה מפוקחות המשמשות לסיווג מידע.

באופן כללי, השיטה מייצרת Hyperplane (או סט של מישורים כאלו) המפריד בין הקבוצות אותן יש לסווג. היפר-מישור זה בעל כמות גדולה משהו של ממדים, בהתאם למידע אותו ניתן למודל.



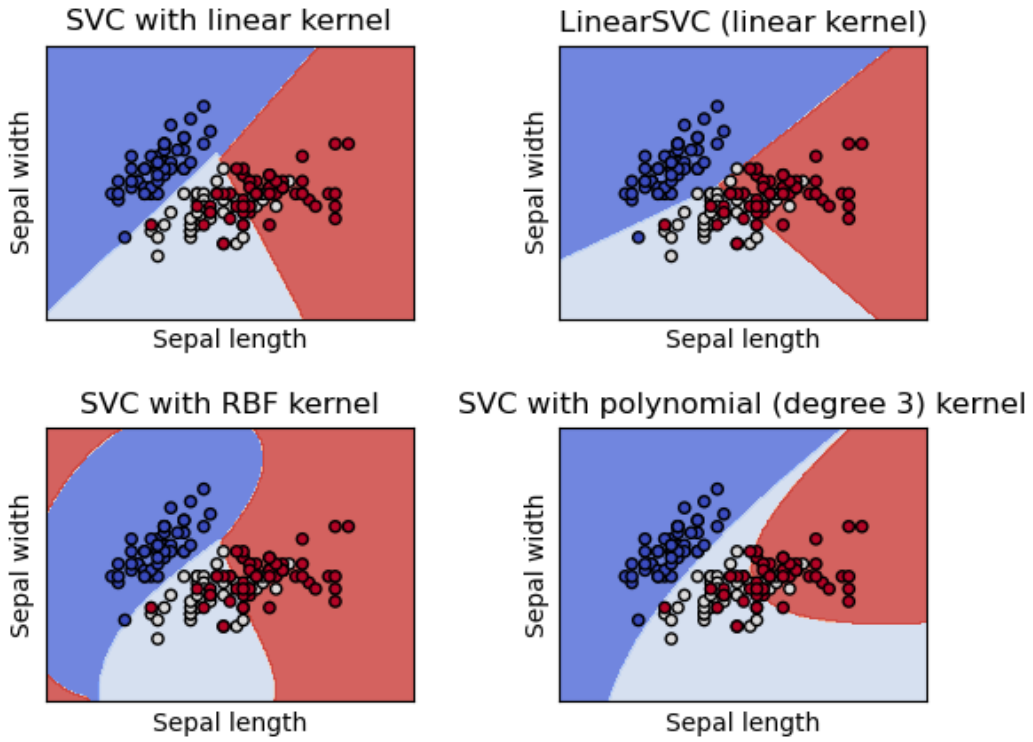
איור 6 - דוגמה דו-ממדית לפונקציית גרעין (היפרמישור - קו ישר) אשר מפרידה בין 2 קבוצות.
ניתן לראות כי H_1 ו-H_2 אינם מפרידים כראוי, ו-H_3 בעל הפרדה מקסימלית.

באופן כללי, נרצה פונקציית גרעין לינארית, אך כיוון שלא ניתן לייצר כזאת עבור כל סט נתונים, ניתן להשתמש בפונקציות גרעין נוספות – פולינומית, רדיאלית (RBF) ועוד.
כמו כן, נרצה גרעין בעל מרחק מקסימלי מקבוצות המידע, כך שיבצע את ההפרדה הטובה ביותר שניתן לקבל על נתונים אלו.

בעזרת פונקציית הגרעין שנקבל, המשמשת כחוצץ בין קבוצות המידע, המודל ידע לסווג כל וקטור מידע חדש שיתקבל על ידי מיקומו במרחב.

פונקציות הגרעין ו-Support Vectors

כפי שצוין, קיימות פונקציות גרעין רבות, אשר כל אחת יכולה לספק מישור שונה עבור אותו הסיווג.



איור 7 - דוגמה לפונקציות גרעין שונה עבור סט נתונים זהה

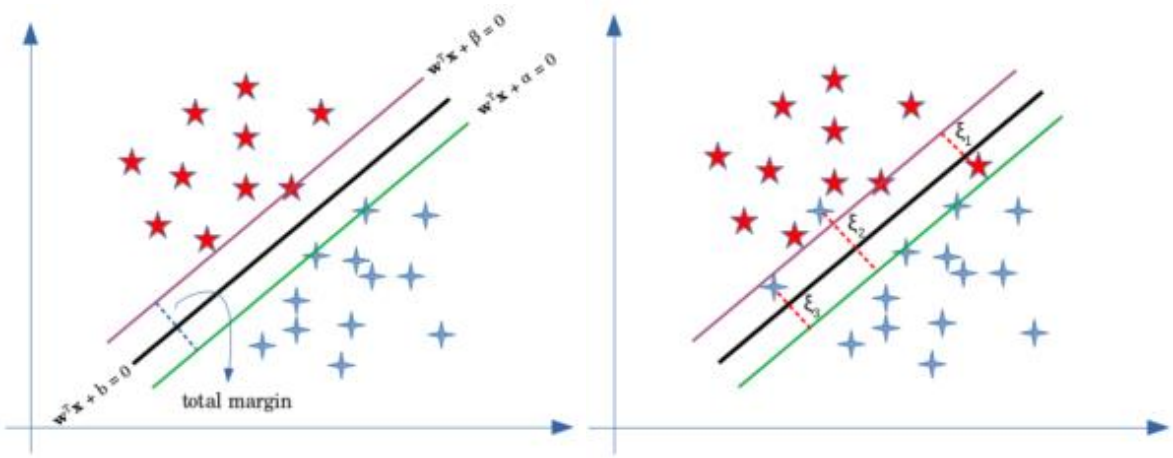
כחלק מיצירת פונקציית הגרעין, אנו לעיתים נתקל בבעיות אשר לא ניתן להפריד באופן מוחלט בין קבוצות המידע. כפי שניתן לראות באיור, קיימת חפיפה של נקודות מהקבוצות אדום ולבן, כך שלא נוכל לייצר פונקציית גרעין שתפריד בדיוק את הקבוצות.

במקרים אלו, בהם לא ניתן להפריד באופן מוחלט בין הקבוצות, נאלץ לאפשר שגיאות ביצירת המישור, למרות שגיאות ידועות מראש. שגיאות אלו נקראות margin errors, וניתן לייצר פונקציות גרעין אשר מוכנות לספוג שגיאות אלו ברמות שונות, כתלות בפרמטר הניתן לשינוי במודל.

השוליים נקבעים בעזרת Support Vectors, אשר נוספים לפונקציית הגרעין.

נבדיל בין 2 מקרים:

1. **Hard margin** – דרישה להפרדה מוחלטת בין קבוצות המידע לסיווג.
2. **Soft margin** – הפרדה בין הקבוצות, התעלמות משגיאות הנמצאות בתוך תחום ה-"שוליים".



איור 8 - דוגמה ל-Soft margin (ימין) ו-Hard margin (שמאל)

היפר-פרמטרים של המודל

למודל קיימים 2 פרמטרים עיקריים – C ו-ɣ, כאשר להם משמעות גדולה ביצירת פונקציית הגרעין.

הפרמטר C קובע את מידת הגמישות של פונקציית הגרעין לשגיאות margin, ומקיים יחס הפוך לשגיאות אלו. באופן זה, אנו מאפשרים יצירת פונקציית גרעין המתעלמת מוקטורי פרמטרים הנמצאים בתוך תחום וקטורי התמיכה (שוליים).
כלומר - ככל שנבחר ערך C גדול יותר, כך נאפשר פחות margin errors.

הפרמטר ɣ קובע את גודל המרחב עליו וקטור פרמטרים בודד ישפיע, ביחס הפוך, כלומר, ניתן להתייחס לפרמטר זה כיחס הפוך לרדיוס ההשפעה של דגימה על המודל.

כיוון שעל ערכים זה להקבע בעת יצירת פונקציית הגרעין, לא ניתן לדעת את ערכם האופטימלי בעת אימון המכונה בפעם הראשונה. נהוג להשתמש בשיטת Grid Search על מנת למצוא את זוג הערכים הנותן את הדיוק הגבוה ביותר.

חלק מעשי

הכנות לניסוי

סט נתונים

ראשית, עלינו היה למצוא מאגר נתונים מתאים למטרותינו:

1. מאגר המכיל קטעי קול רבים, המסווגים כ-זכר/נקבה.
2. מאגר בעל מידע מאומת, כלומר המידע הקיים בפנים תקין וללא טעויות סיווג.
3. מאגר בעל הקלטות באיכות טובה – תדר דגימה בהתאם לתדר הקול האנושי, רעש מועט.
4. מאגר בעל הקלטות מגוונות – גילאים שונים, מבטאים שונים.

בהתאם לסעיפים שהצבנו, מצאנו את המאגר הבא: Common Voice

המאגר מכיל כ-400,000 קטעי קול שונים, מחולקים לקטגוריות לפי רמת האימות שהמידע עבר, כשלכל קטע קול שדות רבים, המלאים באופן חלקי: מין, מוצא, קבוצת גיל, מלל, דירוג.

filename	text	up_votes	down_votes	age	gender	accent
cv-valid-train/sample-000000.mp3	learn to recognize omens and follow them the old king had said	1	0			
cv-valid-train/sample-000001.mp3	everything in the universe evolved he said	1	0			
cv-valid-train/sample-000002.mp3	you came so that you could learn about your dreams said the old woman	1	0			
cv-valid-train/sample-000003.mp3	so now i fear nothing because it was those omens that brought you to me	1	0			
cv-valid-train/sample-000004.mp3	if you start your emails with greetings let me be the first to welcome you to earth	3	2			
cv-valid-train/sample-000005.mp3	a shepherd may like to travel but he should never forget about his sheep	1	0	twenties	female	us
cv-valid-train/sample-000006.mp3	night fell and an assortment of fighting men and merchants entered and exited the tent	3	0			
cv-valid-train/sample-000007.mp3	i heard a faint movement under my feet	2	1			
cv-valid-train/sample-000008.mp3	put jackie right on the staff	3	0	seventies	male	us

איור 9 - מקטע מתוך מאגר הנתונים: קובץ CSV בעל שדות המידע, קבצי MP3.

כיוון שהמידע מגוון, ולא כולו מתאים לצרכינו, נבצע סינון ראשי של המאגר טרם תחילת העבודה:

1. בחרנו את המידע המוגדר כ-"valid".
2. סינון קטעי קול לפי הפרמטרים הבאים:
   - **א.** בעלי סיווג זכר/נקבה
   - **ב.** בעלי 0 הצבעות שליליות
   - **ג.** בעלי לפחות הצבעה חיובית אחת

לאחר סינון זה נותרנו עם כמות לא מאוזנת של הקלטות גברים/נשים, וכיוון שאוכלוסיית העולם בקירוב טוב הינה ביחס 1:1 של גברים ונשים, בחרנו לבצע Resampling (Undersampling). פעולה זו מבצעת חיתוך של הקבוצה הגדולה בסט הנתונים לגודל של הקבוצה הקטנה.

באופן כללי, היינו מעדיפים להשתמש ב-weights בעת אימון המכונה, כך שחוסר האיזון בין הקבוצות לא ישפיע, ובמקביל לא נוותר על מידע. כיוון המידע שברשותנו עצומה, וזמן העיבוד של קבצי הקול היה ממושך מאוד, העדפנו לבצע איזון למידע באופן זה.

כעת, נותרו עם כ-18.5 אלף הקלטות איכותיות, ביחס 1:1 של גברים ונשים.

קבצי שמע

פורמט דיגיטלי

הקבצים שברשותנו הינם קבצי MP3, כלומר קבצים בפורמט דחוס. על מנת לעבד קבצי קול, עלינו לקרוא אותם כ-waveforms, כלומר לפענח את דחיסת הקובץ, פעולה שאורכת מס' שניות. כיוון שמדובר בכ-18.5 אלף הקלטות, ושהניסויים שתכננו דורשים עיבוד של הקבצים מספר פעמים באופן שונה, ביצענו מספר ניסויים מקדימים על משך טעינת קובץ MP3 לעומת טעינה קובץ WAV (גולמי).

ניסויים אלו הושמטו מהפרויקט, כיוון שלבסוף הצלחנו לייעל את אופן העבודה בשיטה אחרת, אך החלטנו שכן לציין את אופן וביצוע הבדיקה:

1. השתמשנו בספרייה Pydub לטעינת קבצי שמע.
2. בחרנו באופן אקראי 1000 קבצי MP3 והמרנו אותם לקבצי WAV.
3. ביצענו, בלולאה, טעינה של הקבצים בפורמט MP3 ובפורמט WAV, תוך כדי שימוש ב-time().
4. חילקנו את משך הזמן הנדרש לטעינה הקבצים בכל פורמט ב-כמות הקבצים שנטענו, וקיבלנו את משך הזמן הממוצע לטעינת קובץ.
5. תוצאות הבדיקה הראו כי טעינת קובץ WAV מהירה פי **~175** מטעינת קובץ MP3.

תוכן ההקלטה – מקטעים שקטים

בשונה מהתיאור התיאורטי של עיבוד אותות דיבור, בהקלטות אמיתיות ישנם מקטעים שקטים רבים, בהם הדובר אינו מדבר, וההקלטה מכילה "רעש" התלוי בציוד ההקלטה ובסביבת הדובר. כיוון שמקטעים אלו **אינם** מאפיינים את מין הדובר, נרצה להסיר את תכונותיהם והשפעתם על תוצאות עיבוד המידע, ולכן כפעולת קדם לעיבוד המידע בחרנו להסיר "מקטעים שקטים".

בחרנו להסיר מקטעים שקטים הגדולים באורכם מכ-500 מילישניות.

עבודה ב-Google Colab – ממשק notebook

בחלק השני של הניסוי, לאחר עיבוד המידע, עברנו לעבודה ב-Google colab. הממשק מאפשר להציג באופן פרוצדורלי את התהליך ואת תוצאותיו הטקסטואליות והגרפיות. באופן זה ניתן לעבוד במשותף על פרוייקט יחיד, לבצע שינויים בזמן אמת ולראות את תוצאותיהם בצורה נוחה.
המחברת מאפשרת גם חלוקה למקטעים ותיעוד ברור ונקי של כל מקטע.

Test #3 - Grid search

In [ ]:
```
# Define the paramter grid for C, gamma from 10^-3 to 10^1
C_grid = [0.001, 0.01, 0.1, 1, 10]
gamma_grid = [0.001, 0.01, 0.1, 1, 10]
param_grid = {'C': C_grid, 'gamma' : gamma_grid}

# best case - kernel = 'RBF', filters = 40
# best C and Gamma will calc'd and the model will be refitted
x_train, x_dev, x_test, y_train, y_dev, y_test = get_data('feature_vectors_n40.csv', test_prc)
lucas = GridSearchCV(SVC(kernel="rbf"), param_grid, cv=2, scoring="accuracy", n_jobs=-1)
lucas.fit(x_train, y_train)

# Find the best model
print(lucas.best_params_)
```

```
{'C': 10, 'gamma': 0.01}
```

איור 10 - מקטע מתוך מחברת הפרויקט

עיבוד המידע

פונקציית ה-MFCC

כפי שהוסבר ברקע התיאורטי, אנו משתמשים ב-MFCC על מנת לייצר וקטור פרמטרים לכל קובץ שמע שברשותנו. השתמשנו בפונקציית ה-MFCC מספריית Librosa. הפונקציה מממשת את ה-MFCC באותה הצורה אותה תיארנו בחלק התיאורי, בעזרת מספר פרמטרים לבחירתנו:

פרמטר	ערך נבחר
גודל החלון (STFT)	2048 [Samples]
הזזה בחלונות (STFT)	512 [Samples]
סוג החלון (STFT)	Hanning
כמות הפילטרים (FILTER BANK)	13, 26, 40 (Test #1)
תדר תחתון	100 [Hz]
תדר עליון	6800 [Hz]
סוג ה-DCT	Type 2

את פעולת עיבוד המידע ביצענו בעזרת הספרייה Dask, המאפשרת לנו לבצע multiprocessing ובכך לקצר את משך העבודה. ביצענו את העיבוד ב-PC הביתי, וקיצרנו את התהליך מכ-140 דקות לכ-12 דקות. ייעול זה קיצר עבורנו את זמן ההמתנה הנדרש בין המקרים.

וקטור פרמטרים (Feature Vectors)

כעת, קיבלנו מטריצת מקדמים עבור כל קובץ שמע. מטריצה זו בעלת M שורות, בהתאם ל-M פילטרים שנבחרו, ובעלת K עמודות, בהתאם לאורך קטע הקול שעובד, ביחס לגודל החלונות ולהזזה ביניהם.

עבור SVM, אנו נדרשים להכניס וקטור פרמטרים, כלומר מימד "שורות" = 1, ומימד עמודות = C, כאשר C קבוע לכל וקטור פרמטרים. כיוון שאורך מקטעי הקול שונים, עלינו להתאים את המידע הקיים לצורה הנדרשת.

בחנו מספר אפשרויות, ביניהן שימוש ב-PCA לחילוץ הפרמטרים המשמעותיים ועיבוד סטטיסטי של המידע. בחרנו לבצע עיבוד סטטיסטי לכל פילטר, הכולל את הפרמטרים הבאים:

1. Mean – ממוצע
2. STD – סטיית תקן
3. Skew – אסימטריה של הנתונים ביחס לממוצע
4. Max – ערך מקסימלי
5. Median - חציון
6. Min – ערך מינימלי

קיבלנו 6 ערכים מכל פילטר, והצבנו את התוצאות אחת לאחר השנייה בתוך הוקטור, כך שקיבלנו וקטורים באורך של 6 * M, כתלות בכמות השורות (פילטרים) בניסוי.

על מנת לא לבחון באופן ידני אלפי קבצים, עבור כל מקרה בו התגלתה שגיאה בעת הריצה – בחרנו להתעלם מתוצאות מקטע קול זה, והחזרנו כתוצאה וקטור "אפסים", אותו נסיר בהמשך.

למידת מכונה

הכנת המידע לאימון

כעת יש בידינו מטריצה המכילה וקטורי פרמטרים רבים, ווקטור תוצאות בהתאמה.

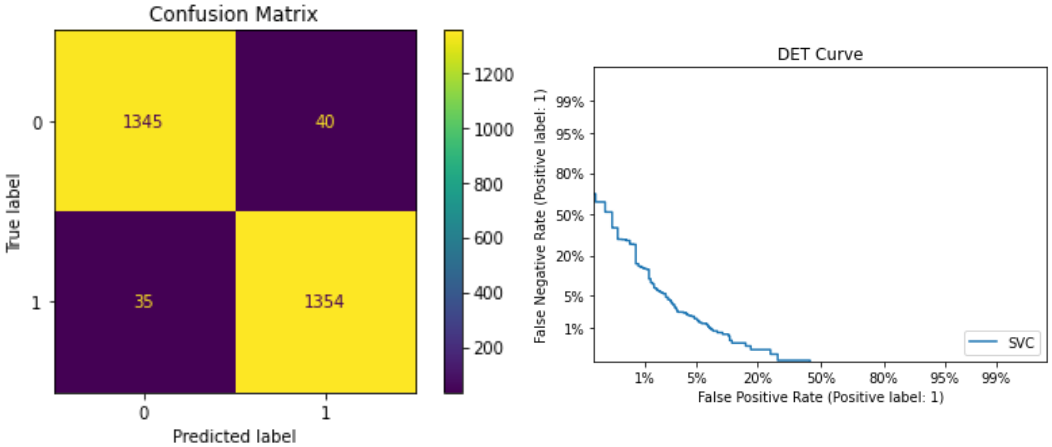
נבצע חלוקה לכ-3 קבוצות – Train, Development, Test.

1. Train – הקבוצה תשמש לאימון המכונה בלבד. כ-70% מסך הנתונים.
2. Dev – הקבוצה תשתמש לכיול המכונה במהלך הניסויים. כ-15% מסך הנתונים.
3. Test – הקבוצה תשמש לבדיקה סופית של המכונה. כ-15% מסך הנתונים.

ראשית, אנו מבצעים אימון למודל עם פרמטרים דיפולטיביים (26 פילטרים, RBF, C=1)

אימון המודל

בבדיקה על סט ה-Dev אנו מקבלים כ-97.3% אחוזי הצלחה בסיווג, באופן הבא:



לאחר בדיקה זו אנו רואים כי השגיאות הינן סימטריות, כלומר המכונה (decision function) מאוזנת ולא מוטית לכיוון זיהוי גבר/אישה. אחוזי ההצלחה הגבוהים מחזקים את הבחירה ב-MFCC לעיבוד המידע.

נציג בכל ניסוי את ה-DET Curve ואת ה-Confusion Matrix, כך שנקבל מדדים לביצועי המודל.

ניסויים – סט Dev

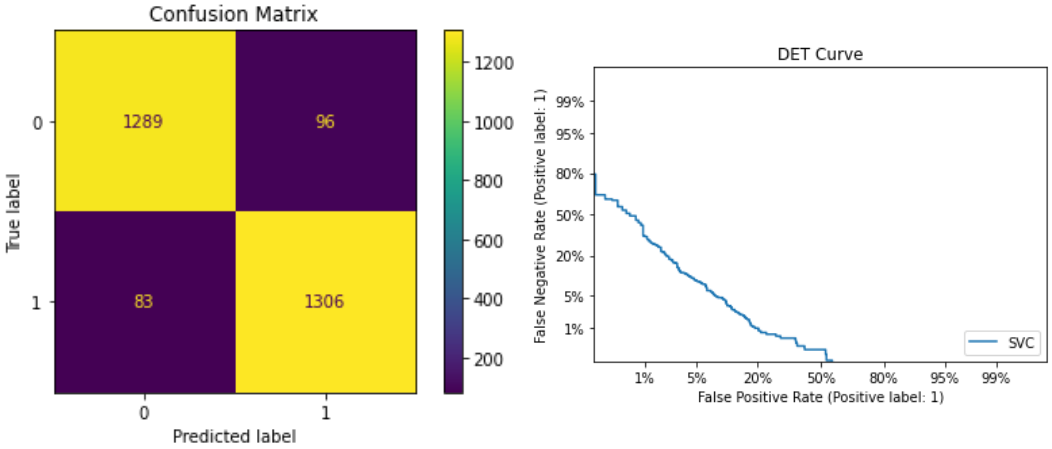
ניסוי 1 – שינוי כמות המסננים

בעת עיבוד המידע, ב-MFCC, אנו בוחרים כמות פילטרים אותה אנו רוצה לקבל כתוצאה.
הגדלת כמות הערכים אותם המכונה לומדת עלולה לגרום ל-over/under-fitting על סט ה-train.

בחרנו 3 ערכים שונים: 13, 26, 40, ונבדוק את תוצאות המכונה בהתאם לערכים אלו.
שאר הפרמטרים נותרו כפרמטרים הדיפולטיביים שהוגדרו באימון המכונה הראשוני (גרעין RBF).

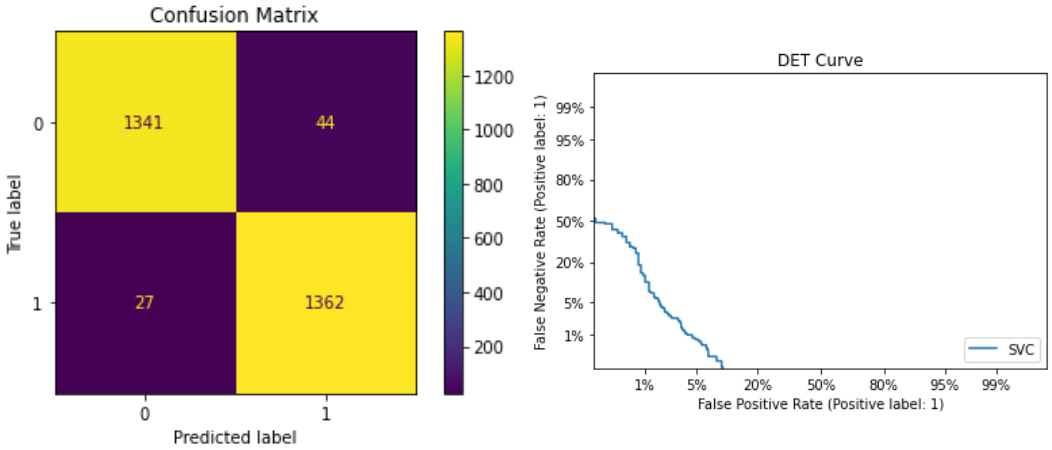
13 מסננים

קיבלנו כ-93.5% אחוזי הצלחה בסיווג, כאשר השגיאות הינן מאוזנות.



40 מסננים

קיבלנו כ-97.45% אחוזי הצלחה בסיווג, כאשר השגיאות הינן מאוזנות.



מתוך שלושת הניסויים אנו רואים כי עבור **40 מסננים** התקבלו התוצאות הטובות ביותר. ניתן לראות כי השיפור בין 40 ל-26 מסננים קטן מאוד, בעוד השיפור בין 26 ל-13 מסננים גדול בהרבה.
כמות הפילטרים קובעת את הרזולוציה בתדר בה נבחן את האות, לכן ניתן להסיק כי קיים שיפור בעת הגדלת הרזולוציה, עד נקודה מסויימת בה ההבדל זניח. מנגד, אנו משלמים בביצועים (זמן) – על המכונה ללמוד כמות גדולה יותר של מידע.
ייתכן כי עבור כמות גדולה מאוד של פילטרים נקבל over-fitting על סט האימון.

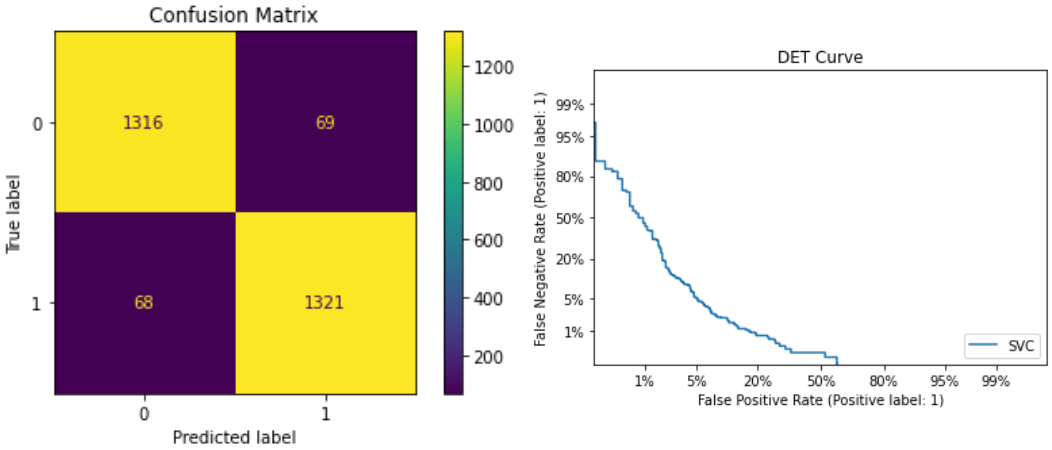
ניסוי 2 – שינוי סוג הגרעין

גרעין המכונה ניתן לבחירה, כפי שתואר בחלק העיוני.

בחרנו 3 גרעינים שונים להשוואה: RBF, Linear, Polynomial, נבדוק כל מקרה ונבצע השוואה.
שאר הפרמטרים נותרו כפרמטרים הדיפולטיביים שהוגדרו באימון המכונה הראשוני (26 פילטרים).

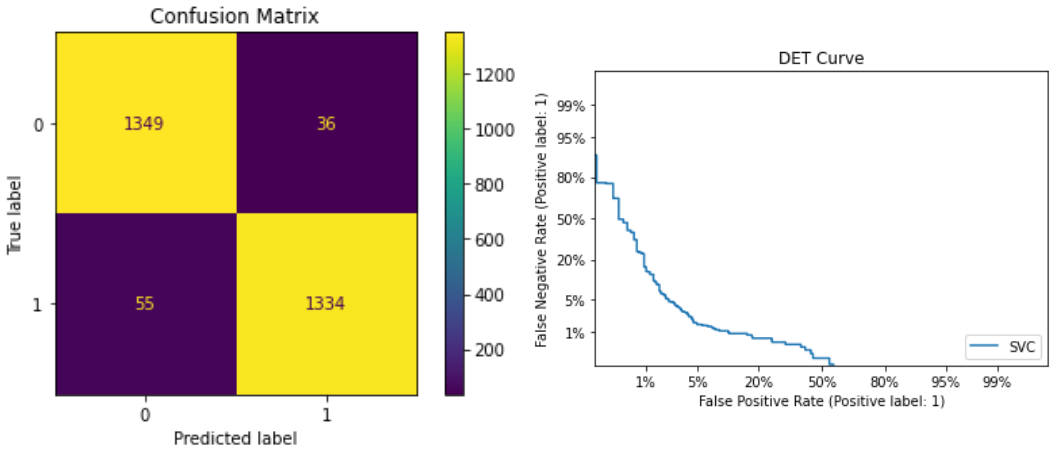
גרעין Linear

קיבלנו כ-95% אחוזי הצלחה בסיווג, כאשר השגיאות הינן מאוזנות.



גרעין Polynomial (order 3)

קיבלנו כ-96.7% אחוזי הצלחה בסיווג, כאשר השגיאות הינן מאוזנות.



ניתן לראות כי גרעין ה-RBF סיפק את התוצאות הטובות ביותר בניסוי.

ניסוי 3 – Grid Search

מתוך ניסויים 1,2 קיבלנו כי על מנת לקבל את התוצאות הטובות ביותר, נשתמש בכ-40 מסננים, ובגרעין RBF. כעת, נרצה לכייל את שאר הפרמטרים של המכונה – C ו-γ.

פרמטרים אלו מאפיינים את ה-hyperplane ואת ה-support vectors שנבחרים בעת אימון המכונה, לכן לאחר מציאת הפרמטרים הטובים ביותר, נבצע אימון חוזר למכונה על אותו הסט.

על מנת למצוא את הפרמטרים האידיאלים, אנו משתמשים בשיטת ה-Grid Search. בשיטה זו אנו מייצרים רשת של ערכי C ו-γ, ובודקים עבור כל זוג אפשרי, מהי רמת דיוק הסיווג המתקבלת. מתוך כלל התוצאות שהתקבלו, נבחר את התוצאה הטובה ביותר, ונאמן את המכונה בעזרתה.

כיוון שתהליך זה צורך משאבי זמן רבים, שכן יש לבדוק מקרים רבים, נהוג לפצל את הבדיקה לאיטרציות. בכל איטרציה ניתן לבחור את ה"אזור" הטוב ביותר ב-Grid, לצמצם את טווח הפרמטרים האפשרי לאזור זה, ולבצע Grid Search פעם נוספת באזור זה בלבד.

בנוסף, על מנת למנוע over-fitting על סט האימון, משתמשים ב-Cross Validation, כלומר חלוקה של סט האימון ל-n סטים קטנים ושווים בגודלם, ובדיקה של כל סט בעזרת הפרמטרים שהתקבלו מה-Grid Search על שאר הסטים.

בחרנו לבצע בדיקה עקרונית על סט ערכים בין 10^{-3} ל-10^{1}, כך שיש 5 ערכים אפשריים לכל פרמטרים, וכ-25 זוגות אפשריים לבדיקה, וב-Cross Validation עם n=2, כך שבפועל נבצע 50 בדיקות.

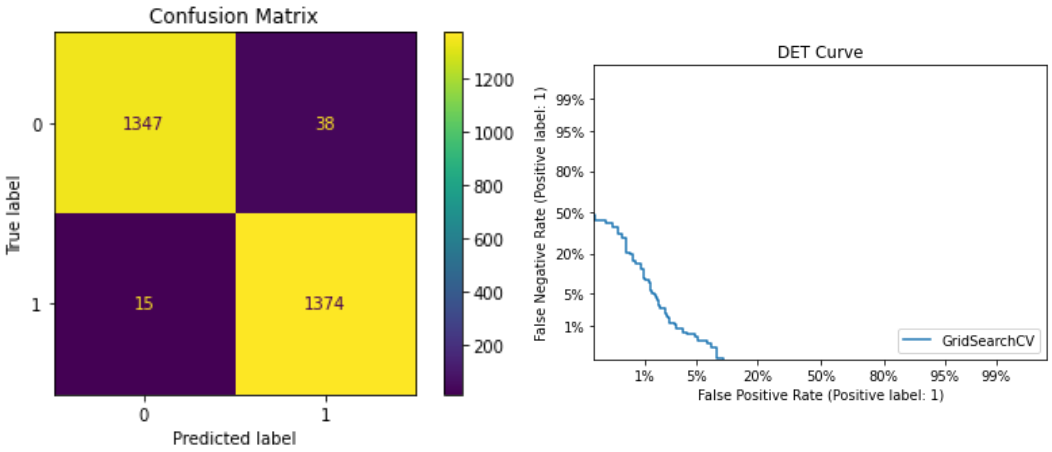
```python
# Find the best model
print(lucas.best_params_)
```

```
{'C': 10, 'gamma': 0.01}
```

איור 11 – הפרמטרים שהתקבלו בהרצת Grid Search על המודל

כעת, נבצע אימון למכונה עם פרמטרים אלו, ונבדוק מחדש את התוצאות שהתקבלו על סט ה-Dev:

קיבלנו כ-**98.1%** אחוזי הצלחה בסיווג.
זוהי אכן התוצאה הגבוהה ביותר שהתקבלה עד כה, כלומר הניסויים וכיול הפרמטרים הועילו.

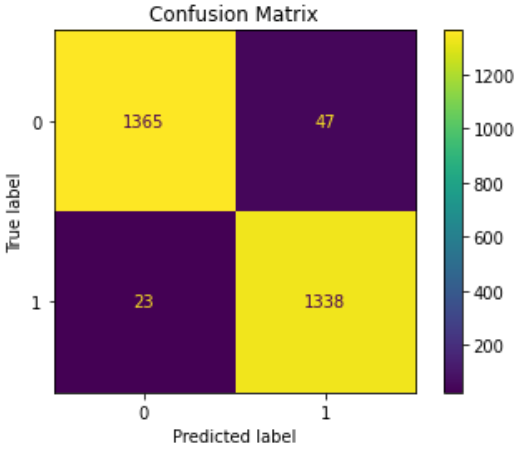


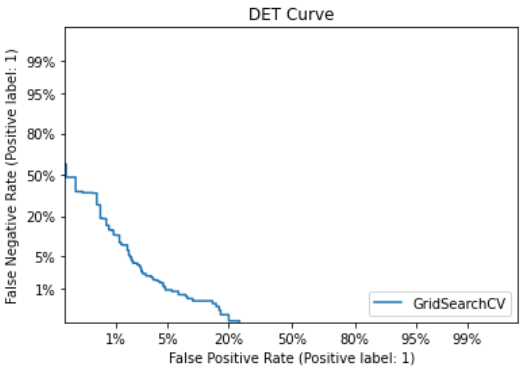
סיכום

בדיקת המודל האופטימלי – סט ה-Test

לאחר כלל הניסויים וכיול המכונה, אנו רוצים לבדוק כיצד המכונה מתמודדת עם מידע חדש, בו לא נתקלה מעולם, ועליו לא בוצע אף ניסוי או כיול.
הפרמטרים שהתקבלו: 40 פילטרים, גרעין RBG, C=10, γ=0.01

קיבלנו כ-**97.5%** אחוזי הצלחה בסיווג.





סיכום התהליך ומסקנות

ראשית, למדנו על תהליך עיבוד אותות דיבור דיגיטליים, על מאפייניהם, ועל הפרמטרים הטובים ביותר לסיווג בין מין הדובר באות שכזה. בעזרת שימוש מתקדם בטכניקות בסיסיות מעולם עיבוד האותות, כגון התמרת פוריה, מסננים פשוטים, ועוד, אנו בונים כלי מתקדם (MFCC) לעיבוד אותות דיבור, ומקבלים בעזרתו מקדמים איכותיים לתיאור אות דיבור.

בעזרת כלים אלו ניגשנו להתמודד עם סט נתונים אמיתי, המכיל קטעי קול אנושיים מרחבי העולם. בתהליך העיבוד, נתקלנו בקשיים המבהירים את הפער בין התיאוריה המתמטית, לבין החלק המעשי של עיבוד אותות דיבור, ולמדנו מהספרות כיצד פותרים קשיים אלו בעולם הדיגיטלי.

לאחר מכן, בעזרת SVM, אנו מצליחים לסווג בין הקלטות שאותן שהמודל אינו מכיר (test), ולדעת ברמת דיוק גבוהה מאוד האם מדובר בגבר או באישה. בעזרת מספר ניסויים הצלחנו אף לשפר את תוצאות אלו, ולקבל את המודל האופטימלי עבורנו.

ככלל, ראינו כי המודל נמצא בנקודה מאוזנת בסיווג גברים/נשים. אנו משתמשים בסיווג בינארי, כלומר סיווג בין '1' או '0', על מנת לסווג בין 2 קבוצות בעלות משמעות זהה, ולכן אין לנו תעדוף לשגיאות מסוג false positive/negative. כלומר - מספיק לנו להמצא בנקודת עבודה בה השגיאות קרובות בגודלן וביחסן, ולנסות להקטין אותן ככל הניתן, ואין לנו צורך לשפר את אחד המדדים הנ"ל.

כמו כן, למדנו כי החישובים המתמטיים הנדרשים לעיבוד המידע ולאימון המודל דורשים משאבי מחשוב גדולים מאוד, שבהעדרם יש להמתין זמן רב לקבלת תוצאות. המתנה זו מחדדת את הצורך בדיוק החישובים וההחלטות בשלב המקדים לפעולות אלו.

לסיכום, ניתן לראות כי ישנו משקל כבד לעיבוד המידע ולמשמעותו, וכי יש לעבד אותו באופן המתאר את מאפייניו בצורה הטובה ביותר. בעזרת עיבוד שכזה, ומודלי הלמידה ההולכים ומשתפרים, אנו יכולים לקבל רמת דיוק גבוהה מאוד בסיווג של מידע חיצוני.

ביבליוגרפיה

Articles:

- Fokoue, Ernest and Ma, Zichen, "Speaker Gender Recognition via MFCCs and SVMs" (2013).
- Chadawan Ittichaichareon, Siwat Suksri and Thaweesak Yingthawornsuk, "Speech Recognition using MFCC" (2012)
- Bhanu Priya, Sukhvinder Kaur, " Comparative study of male and female voices using MFCC and DTW algorithm in speaker recognition" (2014)
- A. Martin*, G. Doddington#, T. Kamm+, M. Ordowski+, M. Przybocki*, "The DET curve in assessment of detection task performance" (1997)
- Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, "A Practical Guide to Support Vector Classification" (2003, Updated 2016)
- Laxmi Narayana M, Sunil Kumar Kopparapu, "Choice of Mel Filter Bank in Computing MFCC of a Resampled Speech" (2014)

Projects:

- sharansankar, GitHub, "Gender recognition SVM" (2017)
- Abdou Rockikz, PythonCode, "How to Perform Voice Gender Recognition using TensorFlow in Python" (2021)
- ANMOUR, Kaggle, "SVM using MFCC features" (2018)
- Om Rastogi, Medium, " Using Machine Learning to classify Instrument Sounds (2020)

Other Sources:

- "Scikit learn" library documentation
- "Python speech features" library documentation
- "Librosa" library documentation
- Kaggle, "Common Voice" dataset
- Leland Roberts, Analytics Vidhya, "Understanding the Mel Spectrogram" (2020)
- Practical Cryptography, Mel Frequency Cepstral Coefficient (MFCC) tutorial
- Mic University, "Facts about speech intelligibility" (2021)