תאריך הגשה: ‏07/05/2024

מגישים: רתם קשאני 209073352 ודויד קופלב 208870279

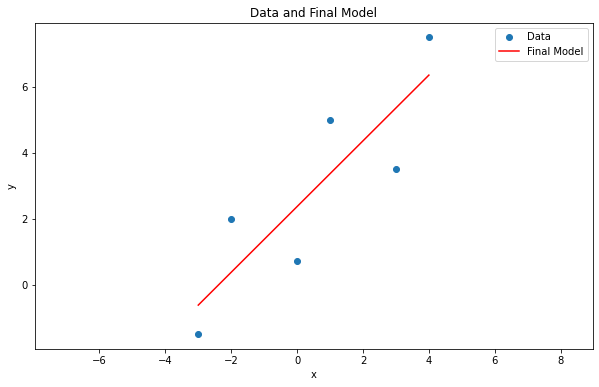
פרויקט מסכם 2024

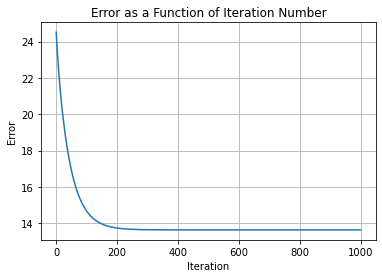
**חלק ראשון:**

**שאלה 1**

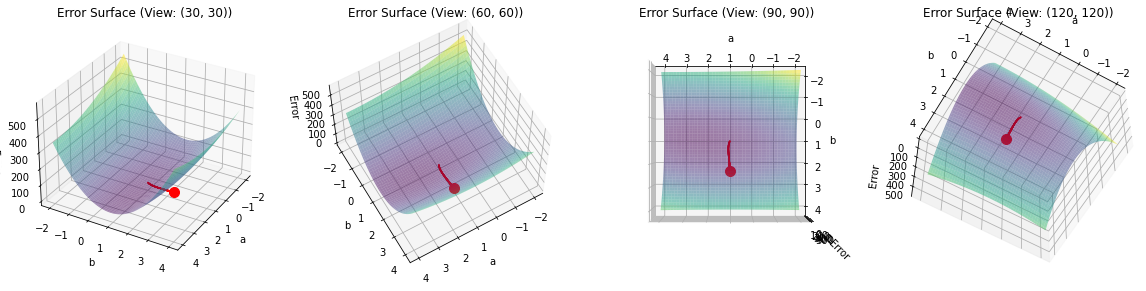
**סעיף א' 1**   
ערכי הפרמטרים הסופיים שנמצאו

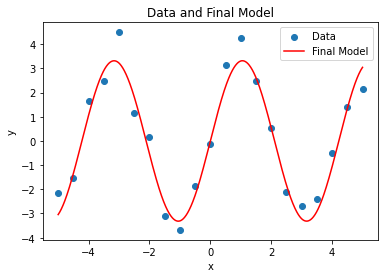
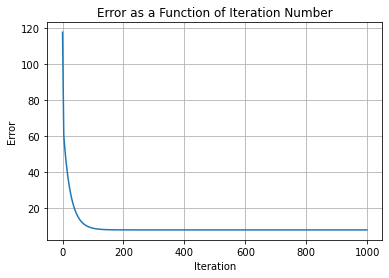
a = 0.9973345422672457  
 b = 2.3679865927174606

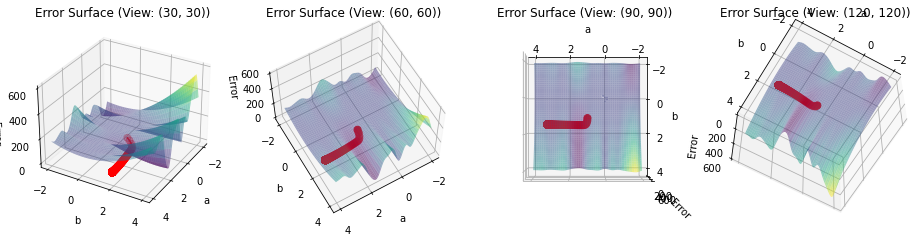
**סעיף א' 2**   
הצגת הגרף של המודל עם הפרמטריים הסופיים (קו ישר) על גבי הנתונים להמחשה של ההתאמה

**סעיף א' 3**   
הצגת הגרף של ערך השגיאה כפונקציה של מספר האיטרציה

**סעיף א' 4**   
הצגת הגרפים של משטח השגיאה מעל הפרמטרים



**סעיף ג'**  
הצגת שלושת הגרפים



הפרמטרים שמצאנו:

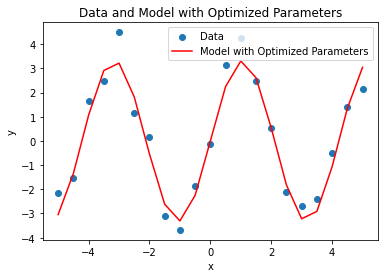
a = 3.3181459209149744

b = 1.4889775873532645  
קצב הלמידה ומספר האיטרציות בהם השתמשנו:

learning\_rate = 0.001

max\_iterations = 1000

**סעיף ד'**

תצוגה של המודל עם הפרמטרים שנמצאו על ידי השיטה כשהם מוצגים על גבי הנתונים, כך שאפשר יהיה לבחון בעין את טיב ההתאמה

הפרמטרים שמצאנו בשיטה זו

a = 3.318145436025057

b = 1.488977981749276

בחינה של התוצאות והסבר

ערכי הפרמטרים יצאו מאוד דומים לסעיף ג' מכיוון שהרעיון של curve\_fit שהוא נותן תוצאות מדויקות יותר בהשוואה לשיטת ירידה בשיפוע. אלגוריתם האופטימיזציה בשימוש curve\_fit ממזער את השאריות בין המודל לנתונים כדי למצוא את הפרמטרים המתאימים ביותר. בנוסף, הגרפים של שני הסעיפים יצאו מאוד דומים כי המודל שנמצא על ידי שני השיטות מתאים באופן דומה לנתונים. אלגוריתם האופטימיזציה בשימוש ב־curve\_fit מזער את השאריות בין המודל לנתונים כדי למצוא את הפרמטרים המתאימים ביותר. בניגוד לכך, בשיטת ירידת השיפוע, אנו מנסים למזער את השגיאה ישירות, מה שעשוי לגרום להבדלים קטנים בתוצאות.

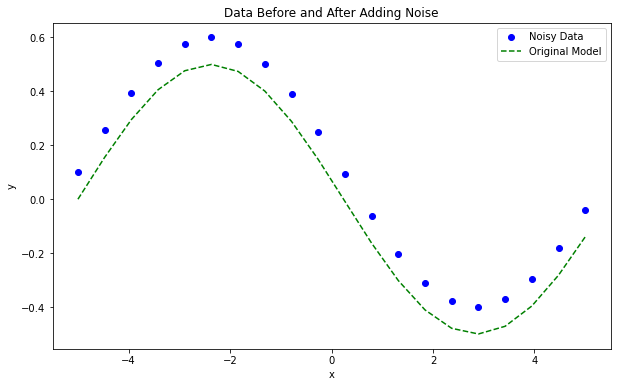
**סעיף ה'**

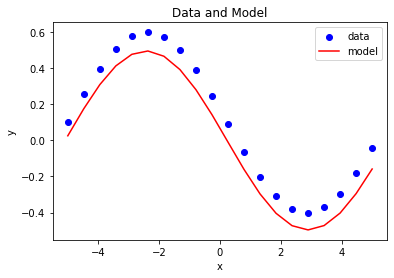
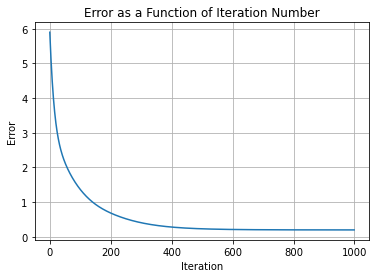
הצגת המודל

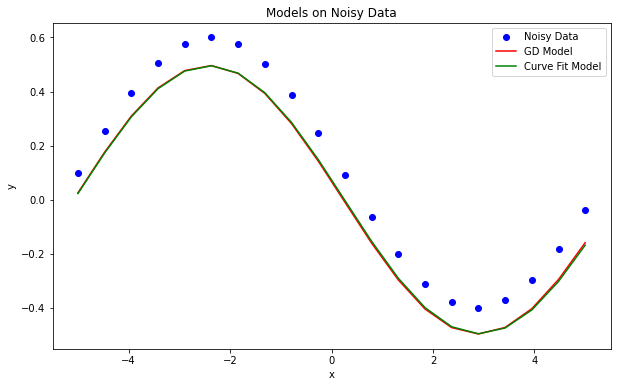
כאשר ערכי הפרמטרים המודל שבחרנו לעבוד איתו הוא

הראשוניים הם:   
a = 0.5

b = 0.6

c = 3  
  
גרף ובו הנתונים שייצרנו לפני ואחרי הוספת הרעש

הגרפים בדומה לסעיף א' של שאלה זו מלבד ההצגה של משטח השגיאה

הצגת גרף ובו שני המודלים (המודל עם הפרמטרים של GD והמודל עם הפרמטרים של curve\_fit) על גבי הנתונים – הנקודות המורעשות

הצגת הפרמטרים שמצאנו בעזרת gradient descent לאלה שמחזירה curve\_fit

gradient descent parameters:

a = -0.49635546672914405

b = 0.590631671889475

c = -0.136665897761742

curve\_fit parameters:

a = -0.495689616647054

b = 0.5891602810562054

c = -0.14956789399766265

הצגת הזמן שלוקח למצוא את הפרמטרים בכל שיטה

Gradient Descent time: 0.04168558120727539

Curve Fit time: 0.0015106201171875

תיאור ההבדלים בין שתי השיטות על פי מה שהצגנו בשלושת הסעיפים הקודמים והסבר מדוע לדעתנו יש הבדלים(אם אכן יש)

שיטת ירידת השיפוע (Gradient Descent) ושיטת האופטימיזציה curve\_fit הן שתי שיטות שימושיות לאיתור פרמטרים מיטביים עבור מודלים. הנה הבדלים עיקריים ביניהן:

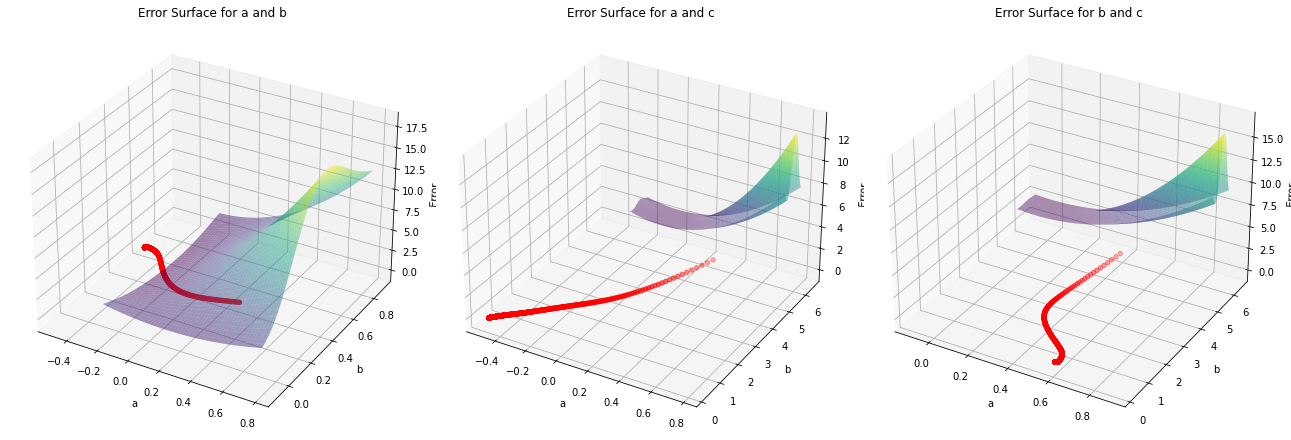
1. **זמן חישוב:** שיטת ירידת השיפוע דורשת מחשבון רב יותר מהפונקציה curve\_fit. זאת מכיוון שהיא מבצעת איטרציות על פי שיעורי השיפוע כדי לעדכן את הפרמטרים, בעוד שcurve\_fit מציעה פתרון תיאורטי מדויק מהיר יותר.

2. **דיוק בתוצאות:** כמו שניתן לראות מהתוצאות שהוצגו, תוצאות הפרמטרים שנמצאו על ידי curve\_fit נראות להיות יותר מדויקות מאלו שנמצאו על ידי ירידת השיפוע. זאת מכיוון שcurve\_fit משתמשת באלגוריתם אופטימיזציה מתקדם המנסה למזער את השגיאה בין המודל לנתונים בצורה מיטבית, בעוד שירידת השיפוע עשויה להיתקע בנקודת קיצון מקומית או להיקלע לקצב של קפיצות גדולות במרחק נמוך מנקודת המינימום.

3. **יעילות:** שיטת curve\_fit נחשבת ליעילה יותר בהיבט של מציאת הפרמטרים המיטביים במודל. זאת מכיוון שהיא משתמשת באלגוריתם מתקדם המתאים באופן אופטימלי את המודל לנתונים בצורה מיטבית. לעומת זאת, שיטת ירידת השיפוע עשויה לדרוש מספר רב יותר של איטרציות על מנת להתקרב לפרמטרים המיטביים, ולכן היא יכולה להיות פחות יעילה מבחינת זמן החישוב והמשאבים הנדרשים.

4. **רגישות להתחלות שונות:** שיטת curve\_fit עשויה להיות רגישה יותר לנקודת ההתחלה שבה מתחילה את תהליך האופטימיזציה. אם התחלת המודל אינה מקרובה מספיק לערכים המיטביים, יתכן והתוצאות יהיו פחות מדויקות או שהאלגוריתם יכול להיתקל בקושי להגיע לפתרון אופטימלי. לעומת זאת, שיטת ירידת השיפוע יכולה להיות פחות רגישה לנקודת ההתחלה ולקפיצות גדולות יותר במרחק נמוך מהמינימום, תלוי בהתאמת הפרמטרים learning rate ומספר האיטרציות.

בסיכום, שיטת curve\_fit מציעה פתרון מדויק ויעיל יותר לאיתור פרמטרים מיטביים במודלים מאשר שיטת ירידת השיפוע, אך יכולה להיות רגישה יותר לנקודת ההתחלה של האופטימיזציה. לעומת זאת, שיטת ירידת השיפוע עשויה להיות פחות יעילה ופחות מדויקת, אך יכולה להיות פחות רגישה לנקודת ההתחלה ולקפיצות גדולות יותר במרחק נמוך מהמינימום.

בונוס מימוש דרך להצגת משטח השגיאה, ועליו את הפרמטרים שנמצאו בכל האיטרציות

הסבר: המשטח עצמו הוא רשת דו-ממדית במרחב המורכבת מנקודות a, b, ו-c. בכל נקודה במשטח, חישוב פונקציית השגיאה E מאפשר ליצור מפה דו-ממדית של משטח השגיאה, המתאר את רמת השגיאה בקשר לערכי הפרמטרים. הנקודות האדומות על קו המתאר מייצגות את הפרמטרים האופטימליים שנמצאו בכל איטרציה של ירידה בשיפוע:   
a\_list, b\_list, c0. כך, באמצעות התצפית במשטח השגיאה, אפשר לבחון את הקשר בין הפרמטרים לרמת השגיאה, ולנתח את השינויים בערכים שלהם במהלך האיטרציות.

**חלק שני:**

**שאלה 4**

**סעיף א'**   
הסבר מלא של השיטה שלנו לביצוע הסיווג, כולל ההחלטות על בניית הסטים השונים, ובחירת המאפיינים

1. ייצור תכונות:

השלב הראשון הוא ליצור תכונות מהאותות השמע. בקוד זה, התכונות הנפלטות כוללות:

* Mel-frequency cepstral coefficients (MFCCs)
* Spectral centroid
* Spectral bandwidth
* Zero-crossing rate
* Root mean square (RMS) energy

אלה הן תכונות נפוצות בעיבוד אותות שמע ויכולות ללכוד מגוון רחב של מאפיינים של האות, כגון מאפיינים ספקטרליים, טימברליים ודינמיקה זמנית.

2. חלוקה לסגמנטים:  
לפני יצירת התכונות, כל האות המקורי מחולק לסגמנטים קטנים יותר באורך קבוע (0.5 שניות במקרה זה). החלוקה הזו נעשית כדי לקבל מספר רב של תצפיות לכל קובץ שמע, מה שיכול לשפר את העמידות וההכללה של המחלק.  
האורך של כל חלק (hop length) מקביל למרווח החופף בין חלקים סמוכים. בקוד זה, הוא מוגדר כך שחלקים סמוכים חופפים מעט.

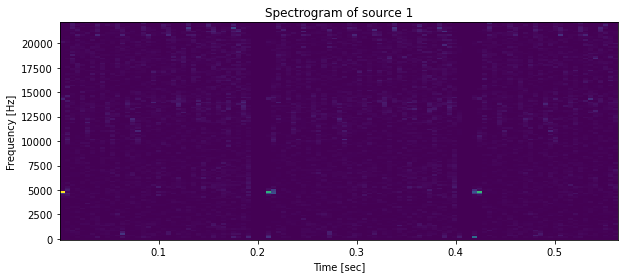
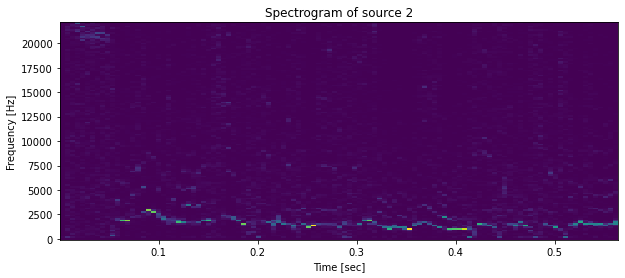
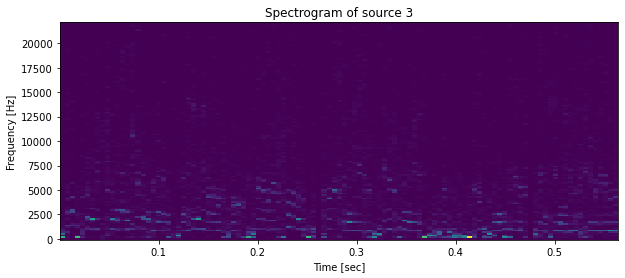
3. בניית קבוצות תכונות:  
לאחר יצירת התכונות והחלוקה לסגמנטים, התכונות שנקלטו מכל החלקים של כל האותות המקוריים מאוחסנות במטריצה תכונות יחידה (`all\_features`). כל שורה במטריצה זו מייצגת את הווקטור התכונות של חלק.  
תווי התגובה המתאימים המציינים את מקור החלק (כלומר, לאיזה אות מקורי הוא שייך) גם הם נאספים למטריצת תווי יחידה (`all\_labels`).  
מטריצת התכונות והווקטור של התוויות מחולקים לקבוצות אימון ובדיקה באמצעות הפונקציה `train\_test\_split` מהספרייה scikit-learn. החלוקה נעשית עם שימור שכיחות המחלקות בשני הקבוצות.

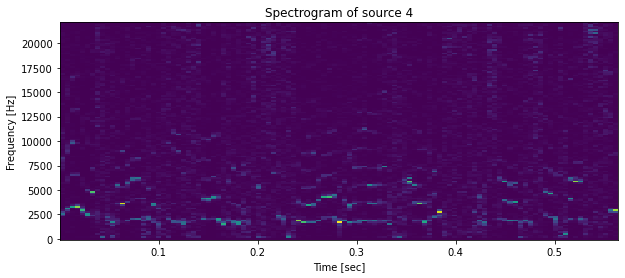
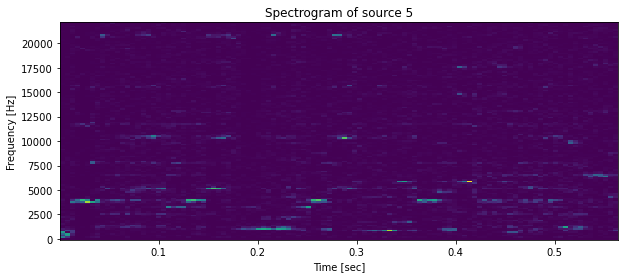
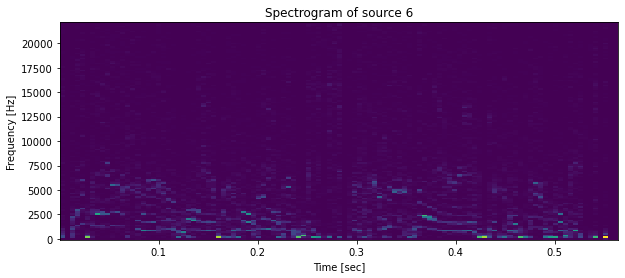
4. השטחה של מטריצת תכונה:  
 לפני העברת התכונות למחלק, מטריצת התכונות משוטחת. זה אומר שכל שורה של מטריצת התכונות, שבהתחלה מייצגת חלק, עוצבה מחדש למערך חד-ממדי. ההשטחה הזו מקלה על פורמט הקלט למחלק.

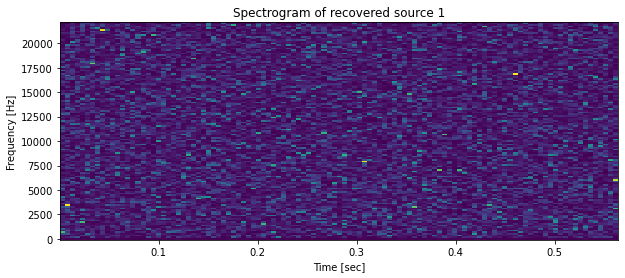
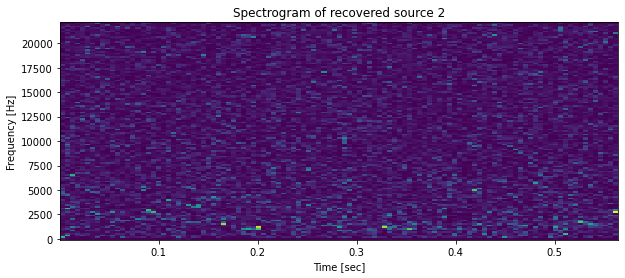
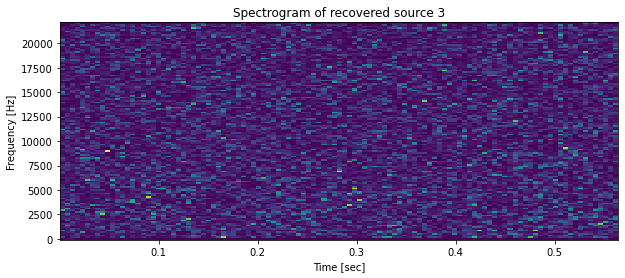
5. בחירת המחלק והאימון:  
בקוד זה, Random Forest classifier נבחר למשימת הסיווג. Random Forest הוא שיטת למידת אנסמבל שבונה מספר רב של עצי החלטה ומשלבת את הניבויים שלהם כדי לשפר את הדיוק ולהפחית את ההתאמת יתר.  
המחלק מאתחל עם היפרפרמטרים כמו מספר העצים (`n\_estimators`) והעומק המרבי של כל עץ (`max\_depth`).  
המחלק מאומן על מטריצת התכונות המשוטחת (`X\_train\_flattened`) והתוויות המתאימות (`y\_train`).

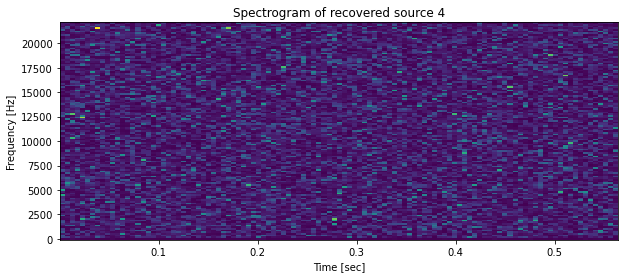
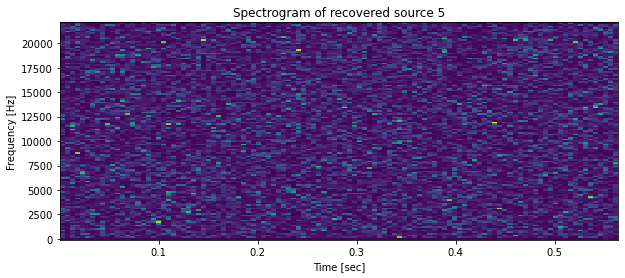
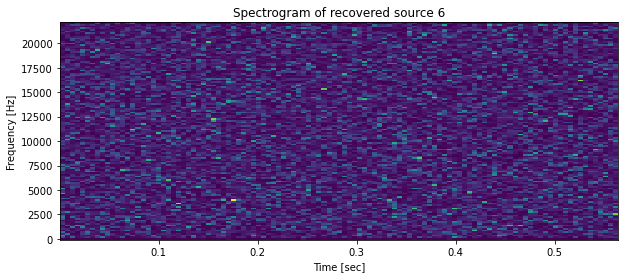
6. הערכה:  
לאחר האימון, המחלק משמש לחזות תוויות עבור קבוצת הבדיקה (`X\_test\_flattened`). התוויות החזויות משווה לתוויות האמיתיות (`y\_test`), והדיוק של המחלק מחושב באמצעות הפונקציה `accuracy\_score`.

**סעיף ב'**   
הצגת הספקטרוגרמות:

ספקטוגרמות של כל אחד מהאותות לפני שיחזור ונרמול



ספקטוגרמות של כל אחד מהאותות המשוחזרים המנורמלים



**סעיף ג'**   
הצגת תוצאות ההפרדה ודיון ביקורתי בתוצאות

ניתן לראות שבאותות המקוריים יש Frequency נמוכה לאורך הזמן לעומת זאת, באותות לאחר השחזור והנרמול יש Frequency בכל הטווח([0-20,000]) לאורך כל הזמן. לדוגמא, במקור השני לפני הנרמול והשחזור היה Frequency בטווח [0-3,000] ולאחר הנרמול והשחזור היה Frequency בכל הטווח. כך גם בשאר האותות שבהם המקוריים עם Frequency רק בחלק מהטווח ואילו לאחר הנרמול והשחזור היה Frequency בכל הטווח. בנוסף, בשמע של האותות המשוחזרים ניתן לשמוע רעש סטטי ולא את האות המקורי ולכן ניתן להסיק כי ICA אינו מצליח להפריד את 6 האותות כראוי.

**סעיף ד'**   
הצגת איכות הקלסיפיקציה על האותות המקוריים ודיון ביקורתי בתוצאות

הדיוק של מודל הסיווג על האותות המקוריים הוא כ-16.67%. משמעות הדבר היא שהמודל מציג ביצועים גרועים בהבחנה בין המקורות השונים של אותות אודיו. נדון בסיבות האפשריות לדיוק הנמוך הזה ולשיפורים הפוטנציאליים:

1. חילוץ תכונה: ייתכן שהתכונות שחולצו מאותות האודיו לא יקבלו מספיק מידע מפליל כדי להבחין ביעילות בין המקורות. בעוד ש-MFCCs (מקדמי ספסל בתדר Mel) משמשים בדרך כלל לניתוח אודיו, ייתכן שהם לא יספיקו למשימה זו. ניסוי עם טכניקות שונות לחילוץ תכונות או שילוב תכונות נוספות עשוי לשפר את הביצועים.

2. חלוקה לסגמנטים: ייתכן שהחלוקה לסגמנטים של האותות למקטעים קצרים יותר (0.5 שניות במקרה זה) לא יתפוס מספיק הקשר או מידע רלוונטי לסיווג. התאמת אורך המקטע או החפיפה עשויה להוביל לתוצאות טובות יותר.

3. מורכבות המודל: ייתכן שהמודל הנבחר (Random Forest) לא יוכל ללכוד את היחסים המורכבים הקיימים באותות האודיו. חקר מודלים מתוחכמים יותר כגון רשתות עצביות עמוקות עשוי להניב ביצועים טובים יותר.

4. הגדלת נתונים: הגדלת נתוני האימון עם וריאציות כגון שינוי גובה, מתיחת זמן או הוספת רעשי רקע עשויה לעזור למודל להכליל טוב יותר לנתונים בלתי נראים.

5. חוסר איזון במחלקה: אם מערך הנתונים מכיל חלוקה לא שווה של דגימות על פני מחלקות שונות, הוא עלול להטות את המודל למחלקת הרוב. טכניקות כמו שקלול כיתות או דגימת יתר של כיתות המיעוטים יכולות לעזור לטפל בבעיה זו.

6. כיוונון היפרפרמטר: הפרמטרים ההיפרפרמטרים של מודל הסיווג (למשל, מספר אומדנים, עומק מרבי) נבחרו באופן שרירותי. ביצוע חיפוש שיטתי אחר היפרפרמטרים אופטימליים באמצעות טכניקות כמו חיפוש רשת או חיפוש אקראי עשוי לשפר את הביצועים.

לסיכום, שיפור דיוק הסיווג על האותות המקוריים דורש שילוב של ייצוג תכונות טוב יותר, בחירת מודל, עיבוד מוקדם של נתונים ואסטרטגיות הערכה. ניסוי עם היבטים אלה באופן איטרטיבי תוך ניטור קפדני של ביצועים יכול להוביל לתוצאות טובות יותר.

**סעיף ה'**   
הצגת איכות הקלסיפיקציה על האותות המשוחזרים ודיון ביקורתי בתוצאות

הדיוק של מודל הסיווג על האותות המשוחזרים הוא מדד חשוב להערכת היעילות של ניתוח הרכיבים העצמאיים (ICA) בהפרדת מקורות השמע המעורבים. נבחן את התוצאות ונדון בשיפורים אפשריים:

1. חילוץ תכונות מאותות משוחזרים: בדומה לאותות המקוריים, תהליך חילוץ התכונות ממלא תפקיד מכריע בלכידת מידע מפליל מהאותות המשוחזרים. אמנם נעשה שימוש באותן תכונות (למשל, MFCCs, מרכז ספקטרלי), המאפיינים של האותות המשוחזרים עשויים להיות שונים מאלה המקוריים. ניסוי בטכניקות שונות לחילוץ תכונות המותאמות לאותות המשוחזרים עשוי להוביל לביצועי סיווג טובים יותר.

2. חלוקה לסגמנטים של אותות משוחזרים: פרמטרי החלוקה לסגמנטים המשמשים עבור האותות המקוריים עשויים שלא להיות אופטימליים עבור האותות המשוחזרים. מכיוון שאופי האותות השתנה לאחר ICA, התאמת אורך המקטע והחפיפה במיוחד עבור האותות המשוחזרים עשויה לשפר את דיוק הסיווג.

3. בחירת מודלים: בעוד שמסווג Random Forest שימש עבור האותות המקוריים, ייתכן שהוא לא הבחירה הטובה ביותר עבור האותות המשוחזרים. ייתכן שהמאפיינים של הנתונים השתנו לאחר ICA, מה שהצריך גישת מודלים שונה. חקירת סיווגים חלופיים או אפילו ארכיטקטורות למידה עמוקה עשויה להועיל.

4. עיבוד מוקדם של נתונים: שלבי עיבוד מקדים כגון נורמליזציה או שיפור עשויים להיות נחוצים עבור האותות המשוחזרים כדי לשפר את איכותם ולהקל על מיצוי תכונות טוב יותר.

5. כיוונון היפרפרמטר: ייתכן שיהיה צורך לכוון את הפרמטרים ההיפרפרמטרים של מודל הסיווג באופן שונה עבור האותות המשוחזרים בהשוואה לאותות המקוריים. ביצוע חיפוש שיטתי אחר היפרפרמטרים אופטימליים באמצעות טכניקות כמו אימות צולב עשוי לשפר את ביצועי המודל.

6. שיטות אנסמבל: טכניקות למידת אנסמבל, כגון שילוב של מספר מסווגים או מודלים, עשויות להיות מועילות לשיפור דיוק הסיווג על האותות המשוחזרים. שיטות אנסמבל מניבות לרוב ביצועים טובים יותר על ידי מינוף החוזקות של מודלים שונים.

לסיכום, הערכת איכות הסיווג על האותות המשוחזרים דורשת התייחסות מדוקדקת של השינויים שהוכנסו על ידי תהליך ה-ICA והתאמת אסטרטגיות החילוץ, המידול וההערכה בהתאם. על ידי חידוד איטרטיבי של היבטים אלה והתנסות בגישות שונות, ניתן להשיג דיוק סיווג טוב יותר ולהפריד ביעילות בין מקורות האודיו המעורבים.

**סעיף ו'**   
הצעת שיפורים להמשך פיתוח או מחקר

הצעות של השיפורים מתוארים בסעיף ד' ובסעיף ה'. לאחר הבחינה של התוצאות דנו בהצעות לשיפורים אפשריים.