



מכללת כנפי רוח, קריית נוער ירושלים (סמל מוסד 140129)

**ספר פרויקט לעובדות גמר  
שאלון 714918 - י"ד הנדסאי תוכנה**

**בנושא:**

**Genre Finder**



**שם הסטודנט: נועם אבוטבול  
ת"ז הסטודנט: #####  
שם המנחה: מר אילן פרץ**

**תשפ"ג 2023**

# הצהרת הסטודנט ואישור הגשה

## הצהרת הסטודנט

**שם הסטודנט:** נועם אבוטבול **ת"ז:** #####

אני החתום מטה מצהיר כי עבדות הגמר המכ"ב, שכוללת ספר וקוד פרויקט, נכתבו על ידי בלבד. כתיבת עבודות הגמר נעשתה בלובי המנהה האישי, והתבססה על נושאים שלמדתי לאורך לימודי התואר, ועל נושאים נוספים שלמדתי באופן מיוחד למימוש עבודות גמר זו. פירוט רישימת המקורות (ביבליוגרפיה) שלמדתי מהם באופן אישי, וועליהם השתמכתי, נמצא בסוף ספר זה. בחתימה על הצהרה זו, אני מתחייב שככל מה שנכתב בעבודת הגמר המוגשת, הוא פרי יצירתי האישית.

\_\_\_\_\_**חתימת הסטודנט:** \_\_\_\_\_**תאריך:** \_\_\_\_\_

## אישור המנהה

הריני מאשר שעבודת הגמר בוצעה בהנחייתי, בדקתי את קוד הפרויקט וקרأت את ספר הפרויקט, ומוצאתי כי הוא מוכן להגשה לבחינת ההגנה של משרד החינוך.

\_\_\_\_\_**שם המנהה:** \_\_\_\_\_**תאריך:** \_\_\_\_\_**חתימה:** \_\_\_\_\_

## אישור מנהל המכללה

הריני מאשר הגשת עבודת גמר זו, לבחינת ההגנה של משרד החינוך.

\_\_\_\_\_**שם מנהל המכללה:** \_\_\_\_\_**תאריך:** \_\_\_\_\_**חתימה:** \_\_\_\_\_

**תוכן עניינים**

5.....	הצעה פרויקט לעבודת גמר בהנדסת תוכנה – כתבה יד – שאלון 714918
9.....	<b> מבוא</b>
9.....	הרקע לפרויקט
11.....	תהליכי חומרה
12.....	סקירות ספרות
14.....	אתגרים מרכזים
15.....	בעיות שה提מודדתי איתם
15.....	הסיבות לבחירת הנושא
16.....	מוטיבציה לעבודה
16.....	הצורך של הפרויקט
16.....	הצגת פתרונות לבעה
17.....	<b> מטרות / יעדים</b>
18.....	מידי הצלחה למערכת
19.....	<b>תיאור מבכזב קיימן</b>
20.....	ניתוח חלופות מערכתי
21.....	תיאור החלופה הנבחרת
22.....	<b> אפיון המערכת</b>
22.....	ניתוח דרישות המערכת
22.....	מודול המערכת
23.....	אפיון פונקציונלי
23.....	ביצועים עיקריים
23.....	אילוצים
24.....	<b> תיאור הארכיטקטורה</b>
24.....	הארכיטקטורה של הפתרון המוצע ( <i>Top-Down Level Design</i> )
24.....	תיאור הרכיבים בפתרון
25.....	תיאור תהליכי של מערכת הפעלה המתבצעים בפרויקט
25.....	תיאור פרוטוקולי התקשרות
26.....	שרף-לקוח
27.....	<b> תהליכי אבטחת מידע במערכת</b>
27.....	תיאור ההגנה
28.....	<b> למידת המכונה</b>
28.....	תהליכי איסוף הנתונים
35.....	סוג הלמידה הנבחר
36.....	נוסחאות ורשומות ללמידה
40.....	<b> ניתוח ותרשימים המערכת</b>

40.....	תיאור אלגוריתמים עיקריים
43.....	הציג מקרה ( <i>Use Cases</i> ) עבור כל הפונקציה העיקרית בפרויקט
43.....	
44.....	מבנה הננתונים בהם היה שימוש
45.....	חישוב ייעילות האלגוריתם
46.....	הקשרים בין היחידות השונות
46.....	ען מודולים
46.....	<i>Use Case Diagram</i>
47.....	<i>Use Cases</i>
48.....	תרשים <i>UML</i>
49.....	
50.....	תיאור המחלקות המוצעת
52.....	תיאור התוכנה
52.....	סביבת עבודה
52.....	שפות תכנות
53.....	ממשק משתמש
53.....	תרשים מסכימים ( <i>Screen Flow Diagram</i> )
54.....	תיאור מסך הפתיחה
54.....	תיאור מסכימים
62.....	קוד התוכנית
62.....	פונקציות עיקריות
70.....	תיאור מסד הנתונים
70.....	תרשים <i>ERD</i>
70.....	סכמה כללית
71.....	מדריך למשתמש
72.....	בדיקות והערכתה
74.....	מסקנות
75.....	פיתוחים עתידיים
76.....	ביבליוגרפיה

בג"ד  
תשפ"ג

דצמבר 2022

## הצעה פרויקט לעבודת גמר בהנדסת תוכנה – כיתה יד – שאלון 714918

### פרטי המגיש והמנחה

שם הסטודנט :	נועם אבוטבול
ת"ז הסטודנט :	#####
שם המכללה :	כנפי רוח, קריית נוער ירושלים.
סמל מוסד :	140129
שם המנחה :	מר אילן פרץ

### פרטי הפרויקט

**שם הפרויקט:** תוכנה לזיהוי ז'אנר של שיר.

### **תיאור הפרויקט**

מערכת המקבלת שיר ונותנת אינדיקציה באחוזים לאיזה ז'אנר שיר השיר.

### **הגדרת הבעה האלגוריתמית**

הבעיה היא בעיית סיווג שיר לז'אנר. כלומר, המערכת תקבל שיר שלא "שמעה מעולם" ותיהיה מסוגלת לסווג בדיק טוב ככל שניתן, את ז'אנר השיר כגון: רוק, פופ, קללאס ... ועוד שסביר בהמשך.

כדי לממש פתרון לבעה, השתמש באחד מאלגוריתמי למידת מכונה שפותרים בעיות סיווג.

### **רקע תיאורטי בתחום הפרויקט**

למידת מכונה היא מענה לפתרון בעיות המאפשרות למחשב ללמידה מתוך דוגמאות הניתנות מראש. קיימים כיוום שלושה מודלים עיקריים של למידת מכונה: למידה מונחית (supervised learning), למידה בלתי מונחית (unsupervised learning) ולמידה חיזוק (reinforcement learning). לבעה זו מודל הלמידה המונחית מתאים (learning).

למידה מונחתית (supervised learning) היא מענה לפתרון בעיות סיווג, כאשר נתונים לה בעיה שלא ראתה מעולם. אופן הלמידה מתבצע בעזרת אוסף גדול של נתונים מסווגים מראש (Dataset), המחשב-Amor ללמידה מאפיינים (features) דומים בעיה חדשה למ Lager הנתונים שלו, ועל פי מידע זה לפתור את בעיית הסיווג.

בענף הלמידה המונחתית ישנו הרבה אלגוריתמים קיימים כגון: רשתות נירוניים (Artificial Neural Network), אלגוריתם שכן קרוב (k-Nearest Neighbors algorithm), עצי החלטה וישנים עוד הרבה. בפרויקט שלי החלטתי למשתמש אלגוריתם רשתות נירוניים (Artificial Neural Network או בקיצור ANN).

כiamo ישנו הרבה תוכנות שמצוות שם של שיר על ידי השמעה לתוכנה מה השיר, לדוגמה: Shazam וכו', בנוסף יש אפליקציה אחת שמצוות ז'אנר של שיר ששמה: AI - AI Identifier, אך היא לא עובדת בצורה טובת.

על מנת למשתמש את הפרויקט היה צריך ללמוד מה זה למידת מכונה, מודלים שונים של למידת מכונה ובחירה איזה מודל יתאים לעביה שלי, איך לפתור בעיות סיווג על ידי אלגוריתמי למידת מכונה הקיימים כיום, איך לבנות Dataset לאימון המודל, התעסקות עם מסד נתונים לא רלוונטי בענן, התעסקות עם קבצי שמע בשפת Java, בניית שרת HTTP והכנות בקשות HTTP מהלקוות.

### **הלייצים עיקריים בפרויקט**

- איסוף וניתוח נתונים – בניית Dataset מתאים לעביה הסיווג
- אימון המודל – בדיקת הנתונים הקיימים עם האלגוריתם הפותר את הבעיה ועדכונו
- זיהוי ז'אנר – הסיווג הסופי באחזois שהמחשב נותן לשיר שלא שמע מעולם

### **מרכיבי הפרויקט - תיאור טכנולוגיה והנדסה**

#### **• צד שרת**

שרת HTTP שיתחבר למסד נתונים בענן ועליו ישב האלגוריתם הראשי, השרת יძין בקשות HTTP מהלקוות ויתן מענה לעביה סיווג ז'אנר השיר. צד השרת יבוצע במודול MVC, ויכתב בשפת Java.

#### **• צד לקוח**

אפליקציה skycity desktop בשפת Java שתוכל ליצור בקשות HTTP לתקשורת עם צד השרת, ויתן אפשרות של העלאת שיר על מנת לקבל אינדיקציה על ז'אנר השיר באחזois. צד לקוח יבנה בתכורת "לקוח רזה" ובמודול MVC.

• **פרוטוקולי תקשורת**

1. פרוטוקול HTTP על מנת לתקשר בין הלקוח לשרת ובנוסף לקבל נתונים לשרת מהען באמצעות Rest API.
2. יכול להיות שההמשר משתמש בעוד פרוטוקולי צד שלישי לטעינת קבצים וכו'.

• **asad נתונים**

asad נתונים של חברת MongoDB שישמור שירים ומידע עליהם לצורך איסוף וניתוח נתונים.

**פיתוחים עתידיים**

1. פיתוח התוכנה עבור עוד פלטפורמות, כגון: IOS, Android.
2. מתן אפשרות ללקוח לבחור זאנר אהוב ועל פיו לתת לו המלצות לפלייליסטים שיימצאו חן בעינו.

**לוחות זמנים**

לשיט עד לתאריך	שלבי עבודה
1.12.22	בחירת פרויקט, חקירה ולמידה לעומק של נושא הפרויקט
18.12.22	כתיבה והגשת הצעת הפרויקט לאישור משרד החינוך
15.1.23	בנייה אלגוריתם ראשי וביצוע בדיקות ושיפורים
5.2.23	בנייה צד שרת
19.2.23	בנייה מסד הנתונים ושלובו
5.3.23	בנייה צד לקוח
26.3.23	כתיבת ספר הפרויקט
27.3.23	הגשת הפרויקט כולו (ספר + קוד) לבוחני משרד החינוך

**חתימת סטודנט:**

**חתימת רכז המגמה:**

**אישור משרד החינוך:**

## מבוא

### הרקע לפרויקט

במשך לימודיו בתיכון למדתי בмагמת הנדסת תוכנה שכוללת 5 יחל עיוני ו 5 יחל מעשי בשפט #C, לאחר מכן התחלתי מסלול הנדסאי תוכנה ובמשך הזמן למדתי המון דבירים בתחום והכנתי הרבה פרויקטים מעוניינים בשפות שונות, כתע זהה פרויקט הגמר שמסכם את לימודיו.

למיידת מכונה תופסת חלק גדול היום בחיננו והטכנולוגיות העתידיות פונות לשם, כיצד לגרום למחשב ללמידה בכוחות עצמו. ישנו שני סוגים למידה עיקריים: למידה מונחית ולמידה בלתי מונחית.

בלמידה מונחיתאפשר למחשב ללמידה מתרך אוסף גדול של דוגמאות פתורות, לדוגמא אם רצים שהמחשב יזהה אם יש חתול בתמונה, נתונים לו כמות גדולה מאוד של תמונות שיש בהן חתול ותמונות שאין בהן חתול ומזהירים לו מה יש חתול או מה אין חתול. לאחר האימון הזה נתונים לו תמונה שהוא לא ראה ושאלים אותו אם יש חתול או אין.

בלמידה בלתי מונחית נתונים למchèב אוסף גדול של מידע לא מတיג והמחשב צריך להבין איפה המכונה המשותף וכייז את הנתונים לפי כל מיני שיקולים. לדוגמא, מבאים למchèב איסוף דוגמאות מדרדים רפואיים של אנשים, אבל לא אומרם לו מי חוליה ומיברא, המchèב אולי לא יוכל לדעת בוודאות מי חוליה או מי בריא אך הוא יוכל למצוא מכונה משותף בין המדרדים לפי שיקולו.

בפרויקט שלי אנו צריכים לזהות ז'אנר של שיר, וכך אנו משתמשים בלמידה מונחית ונטרכה בה.

ישנו שלושה שלבים עיקריים בלמידה מונחית:

- ניתוח ואייסוף הנתונים
- אימון המודל
- חשיפת המכונה למידע חדש ובדיקה התקינות

בניתוח ואייסוף הנתונים אנו מחפשים מידע רלוונטי שאפשר להוציא משיר מסוים. לדוגמא: משך זמן השיר, עצמת המוזיקה וכו'. למידע הזה קוראים "פיצרים", בנוסף לפיצרים אנו נאוסף את הז'אנרים של השירים שאוותם נenna "תגיות".

לאחר שאספנו אוסף גדול של נתונים (dataset) נפצל אותו לשניים, חלק של אימון וחלק של בדיקה (בדרכ כל היחס הוא לאימון 70% ובדיקה 30% אך יכול להשתנות).

באיםו המודל נדרש להחליט באיזה אלגוריתם למידה להשתמש, בلمידה מונחתית לפתורן הבעיה שלנו עומדים לפניו כמה אופציות:

- אלגוריתם השכן הקרוב (*k*-Nearest Neighbors algorithm)
- עץ החלטה (Decision Tree)
- רשת נירונית (Artificial Neural Network)

בפרויקט בחרתי למשר רשת נירונית באlgorigthm הלומד.

לאחר בחירת האלגוריתם צריך לאמן את המודל, כלומר לתת לו את חלק האימון של המידע שאספנו, בחלק זהה האלגוריתם צריך לפרק את הפיצרים החשובים לו בשיר ולראות לאיזה ז'אנר הם שייכים.

לאחר האימון אנו נתונים לו את חלק המידע לבדיקה שהכנו מראש נתונים לו לחזות בלבד לאיזה ז'אנר שיר השיר. במידה והוא טועה אנו לוקחים את ההפרש בין החיזוי לבין התיאוג האמיתי וمعدכנים באlgorigthm את המשתנים הנחוצים.

לבסוף אפשר להכניס שיר ללא המידע עליו לתוכנה, התוכנה צריכה להוציא מהשיר את הפיצרים שהמודל יודע לעבוד איתם (במה שअרט על הפיצרים שבחרתי) ולשלוח למcona על מנת לקבל את הז'אנר כפלט.

### תהליכי המחק

מן שלא למדתי מעולם לפני הפרויקט על למידת מכונה היתי צריך לחקור רבות את הנושא התיאורטי על מנת להביא תוצר ראוי.

בהתחלת חקרתי על סוגים של למידות מכונה (מונייח ובלתי מונייח) ובתוכם חקרתי עוד על איך סוג של למידה אני צריך בבעיה שלי. לאחר שהבנתי שמדובר בלמידה מונייח היתי צריך לחקור על אלגוריתמים קיימים בתחום.

חקרתי את האלגוריתמים הקיימים והאלגוריתם שהכי עניין אותי היה רשת הנוירונים, מפני שבשונה מאלגוריתמים אחרים היא מחקה את פעילות המוח שלנו בהעברת המידע והלמידה.

לאחר בחירת האלגוריתם התחלתי לנסות למשו אותו אך לא היה לי מספיק ידע על מנת לעשות זאת כראוי, لكن חזרתי ללמידה כיצד בונים רשת נוירונים מאפס. עם הרבה מאמצ ותיאוריה מתמטית על כיצד הרשת באמצעות פונקציית הפעלה, הփסוד והחיזוי ולאחר שילוב הידע מקורס אלגברה ליניארית שמתעסק רבות בפעולות מתמטיות על מטריצות, מה שרשת הנוירונים עשויה רבות, הצלחת לייצר רשת נוירונים עם מחלוקת עצル פועלות על מטריצות.

לאחר כל העבודה הזאת, הבנתי שחסרים לי המון חלקים בلمמש את הרשת בעצמי, ולכן חיפשתי framework ב - java שנוטן לי את הכלים לבנות רשת ולأمان אותה. השתמשתי ב [Encog](#).

לאחר כל זאת היתי צריך לחקור הרבה על איך מידע אפשר להוציא משיר, איך שיר שמור במחשב, איך אפשר לקרוא שיר במחשב. עם הרבה מחקר הגעתי להבנה שגם נתונים היקלטה של אותו השיר אך עם כל נגינה שונים, המחשב רואה את זה כשיר שונה ולכן הסכנה היא שהציגו שנקבל יהיה שונה לגמרי.

עם הרבה מחקר בתיאורית המוזיקה על שמות האלמנטים שיש למוזיקה (גובה, קצב, מלודיה וכו') הבנתי שאצטרך להשתמש בפיצרים האלו, אך הם לא באמת מספקים בשבייל ליצג שיר במחשב, לכן יש תוכנות נוספות שמייצגות מוזיקה באמצעות דיגיטליים: ספקטוגרם (Mel-frequency cepstrum) וכרכומה (chroma features).

הספקטוגרם מתייחס להספק של הצליל ש商量וס על חישוב מתמטי בסולם מל של תדר והcrcומה לעומת זאת מתייחס לשינוע גובה הצלילים בצורה משמעותית.

לאחר שיקול רב, הגעתי להבנה שבעזרת כרומה אפשר ליצג צלילים בצורה חד משמעות במחשב (ולכן זה פותר את בעיית נתינת אותו השיר בכלים אחרים – אלו אותם התווים ולכך הcrcומה תישאר אותו הדבר) אבל בפרויקט אני מיצא את הcrcומה של השירים ומקטלgas לפיהן (כמובן שגם עוד נתונים אבל זה העיקרי).

## סקירה ספרות

במאמר של (2019, Editor) כחלק מחקר שלי על תיאורית המוזיקה ועל האלמנטים שאפשר לקבל מכל יצירה מוזיקלית, מדובר על מפתחות מוזיקליים שכימיים, כלומר לכל יצירה יש מפתח מוזיקלי שהוא מנוגן בה, לדוגמה: בפסקטת המפתח לניגון התווים הגבוהים בדרך כלל יהיה מפתח סול ולניגון התווים הנמוכים יהיה מפתח רה. המפתח הוא בעצם מצין את גובה הצליליםшибואו לאחריו. סך הכל המפתח יכול לומר לנו באיזה כל נגינה נעשתה היצירה, דבר שיכל להשפיע על הדיאנה.

במאמר של (2005, Meinard Muller) מדובר על הchroma feature (chroma feature) ואיך למצוא תיאום בין שני קובצי אודיו במחשב בעזרת הchroma שלהם, במאמר הם מסבירים על וקטור הגבהים של הצלילים של אותו קובץ אודיו וכייד הוא מחושב, כמו כן מסבירים כיצד ניתן לבדוק תיאום בין שני קובצי אודיו שנגנים את אותה היצירה עם כלים שונים, כך שוקטור הגובה של הצלילים נשאר אותו הווקטור.

במאמר של (Shah, 2019) מדובר גם כיצד הchroma עבדת מייצגת, הפעם מדובר בכלליות יותר ולא כיצד מתאימים בין שני קובצי אודיו בעזרת הchroma שלהם. המאמר עזר לי להבין יותר לעומק בכלליות כיצד הchroma עבדת עם ישומים שונים ודוגמאות לזרחי אקורדים. בסרטון של (Muller, 2013) ישנה הרצאה שמסבירה לעומק כיצד הchroma מייצגת במחשב ואיך היא בעצם נראית עם התווים המנוגנים ביצירה בפועל, הסרטון נותן לי הבנה הרבה יותר עמוקה כיצד הדבר עובד בפועל.

בסרטון של (Velardo, 2021) מוסבר על ספקטוגרמה בסולם מל (mel-frequency cepstrum), מוסבר התיאוריה הכללית ואיך זה עובד בפועל. עזר לי בעיקר להחליט במה כדאי לי להשתמש בchroma או בספקטוגרמה.

במאמר

במאמר של (2022, Huei-Huei Cheng) מוסבר כיצד מסווגים ז'אנרים בעזרת הספקטוגרמה, מראים שלב מה צריך לעשות. מפני שבחרתי לעבוד עם הchroma ולא עם הספקטוגרמה המאמר פחות עזר לי, אך תמיד טוב ללמידה עוד בנושא זהה.

במאמר של (1999, BHADESH) מדובר לעומק על רשת ניירונים, כיצד היא פועלת, מהם שכבות הקלט והפלט והשכבות החבויות. המאמר מדבר יותר בקטע התיאורטי של העניין אך מראה חישובים מתמטיים וקירובים של פונקציות עם דוגמאות.

בסרטון של (2018, Blue1Brown) מוסבר איך רשת עבדת עם זרימה קדימה, אילו חישובים צריך לעשות בפועל והכפלות סקלריות של הערכים שהובאו מהניירונים הקודמים. הסרטון עזר לי להבין לעומק את רעיון הרשת וכייד אפשר לגרום לה לפעול בפועל.

בסרטון של (Coding, 2021) מוסבר כיצד בונים רשת ניירונים עם זרימה קדימה למשזה בשפת Java, הסרטון עם הסברים לכל הפונקציות והמחלקות, והוא גם מסביר את

התיאוריה המדוקיקת לכל עניין ולכל חישוב. הסרטון עזר לי להבין כיצד אני צריך לבנות את הרשות שלי, אך הסרטון הזה לא הספיק עד הסוף לבניית הרשות.

במאמר של (Sonawane, 2020) מוסבר מימוש של רשות נוירונים עם זרימה קדימה ופעוף אחורה בשפת Java, לאחר כל קטע קוד הוא מסביר מה בדיק הוא עושה עם כל החישובים שצורך לעשות. המאמר עזר לי מאוד ולפי ידעתו כיצד לבנות את הרשות שלי בפרויקט ולהתאים אותה אליו.

מאמר של (projectpro, 2023) מציג כיצד פרויקט עם הסברים שמאפשר לסוג צ'אנרים, בדיק הבעיה שנייה לפטור. במאמר הוא מציג את התהילה עם דוגמאות כיצד לפתור את הבעיה בעזרת ספריות בפייתון, כמו כן הוא לא באמת בונה כשלעצמם את כל הדברים, אך הוא נותן כיוון רעיון.

### אתגרים מרכזיים

מכיוון שלעולם לא התעסקתי בנושא למדת המכונה היה לי נורא קשה ללמידה במיקוד טוב את החומר, כי יש הרבה מידע חומר ולכך הילכתי לאיבוד. עם עזרה מכוונת מהמנחה התמקדתי בנושא שלי והצלחתי ללמידה את התיאוריה ששיכת לפROYיקט.

בנוסף לכך ההתעסקות עם ספירותי חיצניות כמו MP3, WAV, Chroma לקחו זמן ועל אף הדוקומנטציה היה מתガר להבין איך להשתמש בהם בצורה הילכתי בטובה בפרויקט שלי. ההתעסקות עם קבצי H5 של ה – dataset הייתה מאוד מתוגרת, מפני שצריך ספירותי מיוחדות לקרוא קבצים כאלה, שבדרך כלל יש אותם בעיות תאימות. ולבסוף לאחר הרבה הרים נסילים נאלצתי להחליף dataset לפורמט שהוא יותר ידידותי.

האתגר הילכתי מרכדי לדעתי היה לבנות ממש את רשת הנוירונים מפני שהיא קשה מאוד למצוא הדריכה איך עושים את זה נכון עם הסברים באמצעות מה כל פונקציה עשוה ואייפה המתמטיקה משתמשת. ולאחר הבנה זו למצוא framework שבאמת יעוז לך, כי לבנות הכל נכון זו דרישת לא אפשרית לפרק זמן מוגבל בצורה זו.

בנוסף לכך, "יצוא הפיצרים" (Feature Extraction) היהאתגר קשה מפני שהספרייה שעושה את הפעולה זו היא כתובה בשפת Python ואני כתבתי את הפרויקט שלי בשפת Java, ובשפת Java לא באמצעות הילכתי ספרייה מסודרת שעשתה את העבודה זו. לכן החלטתי לנוט לכתוב בעצמי את כל הדברים, אך לאחר שהבנתי שאני לא יודע במדוקן באילו אלגוריתמים משתמש הספרייה Librosa (של Python) הבנתי שאצטרך למצוא דרך לשלב קבצים של Python בפרויקט.

בעיות שהתמודדתי איתם בתחילת, ניסיתי למשר רשת נוירונים מאפס, דבר שהתרברר כלל אפשרי מפאת הזמן הקצר לעובדה. لكن החלטתי להשתמש ב - Encog framework (לאחר ניסיון כושל להשתמש ב - DeepLearning4, מפני שהפונקציונליות שלו לא עובדת באופן תקין) ולאמן את הרשת בעצמי (לאחר הגדרת ארכיטקטורת הרשת).

בעיה נוספת שהתגלתה היא בעית Overfitting שיפורסמת בקרב מכונה למידה. בעיה זו מואפיינית כבעיה שבה המודל (רשת הנוירונים במקורו שלנו) מותאם יתר על המידה לאוסף הנתונים שעליו אומן (dataset). הסיבה העיקרית לבעיה הייתה שאוסף הנתונים שלי היה דל מדי בשבייל לאמן עלי אמת הרשת לביצוע תחזיות טובות, פתרתי את הבעיה בכך שהגדלתិי באופן מלאכותי את הנתונים. למשל, ה – dataset שהשתמשתי בו סיפק לי אלפי שירים לעשרה ז'אנרים שונים (100 שירים לכל ז'אנר), אני בחרתי לסוג רק כ ארבעה שירים וכן נשארתי עם סך הכל 400 שירים לאימון, אך 400 שירים לא באמת מספיקים לאמן بصورة טובה מודל של למידת מכונה. لكن החלטתי לשירים עצם שסיפקו לי, כל שיר נמשך 30 שניות, שכן החלטתי לחלק כל שיר לעשרה חלקים של 3 שניות לכל אחד ולבנות קובץ של dataset חדש על פי השירים האלה (בעזרת קוד Python שכתבתי בשימוש הספרייה Librosa).

עוד סיבה שיכלה לגרום להטאמת יתר (Overfitting) היא מספר הנוירונים בכל שכבה חבוייה בראשת, למשל אם נגדיר פחות מדי נוירונים בכל שכבה – הרשת שלנו לא תצליח ללמוד נכון, אך אם נגדיר מספר רב מדי של נוירונים בכל שכבה המודל עלול להגיד למצב של התאמה יתר (Overfitting).

#### הסיבות לבחירת הנושא

תמיד היה לי חיבור למוזיקה, אני מנגן כבר חמיש שנים על גיטרה ופסנתר, ותמיד רציתי לחקור לעומק יותר את נושא המוזיקה. אף על פי שלמדתי את תיאורית המוזיקה, רציתי יותר לגשת לפרקטייה של התיאוריה זו, והפרויקט זהה הוא הzdמנות מצוינת לביצוע המשימה.

התלבטתי רבות בין שני פרויקטים שרציתי לעשות: הפרויקט הנוכחי ופרויקט של דיזיין תווים משיר שמנוגן. שני הפרויקטים מאד עניינו אותי, אך הפרויקט של דיזיין התווים הוא פרויקט הרבה יותר פשוט שלא עוסק בלמידת מכונה, ומאוד רציתי לגשת לעולם הזה של למידת המכונה בשבייל להכיר אותו יותר ולצבור בו יותר ניסיון.

мотיבציה לעבודה

פרויקט זה הוא פרויקט שמסכם את שנות לימודיו לתעודה ההנדסאי ולא רק, הוא אמר לחת ל' ניסיון לקראת השירות הצבאי וההגעה לאזרחות. لكن אהבתו להתעסק הרבה עם הפרויקט ולהשקייע, הוא נתן לי יותר משמעות כי הרגשתו שבעל פעם שהוא עוזב עליו הוא נותן לי' עוד ניסיון לבניית פרויקטים מסווגים.

כמובן עניין החקיר המוזיקלי נתן לי עוד מוטיבציה לעבוד על הפרויקט כי כמו שהזכירתי לעללה אני מאוד אוהב להתעסק במוזיקה.

הצורף של הפרויקט

ח'יפשתי רבות kali טוב שמצויה ז'אנר ברוחבי האינטרנט, מצאתי כמה כלים אך הם לא עושים את העבודה באופן טוב מופיע, لكن החלטתי להרים את הception ולבנות kali כזה בעצמי, ואולי יומן אחד להעלות את הפרויקט הזה באמת לציבור הרחב באינטרנט.

הציגות הדרומית לבעיה

לאורך תהליך מחקר למדידת המוכנה נתקלתי בכמה סוגים של אלגוריתמים לביעית סיווג בلمידה מונחית:

**אלגוריתם K השכן הקרוב (k-Nearest Neighbors algorithm):** זהו אלגוריתם המשמש לסייע או לרגסיה, הוא מסוג דוגמה חדשה באמצעות השוואה שלה לדוגמאות הקרובות לה ביותר ממרחב האימון. אלגוריתם זה הוא פשוט יחסית וכאן לא בחרתי להסתמש בו.

**עץ החלטה (Decision Tree):** עץ בינארי שמשמש לשאלות על הקלט הנקנס ובסוף העץ  
לאט לआט מתקרב לסיווג, כאשר הענפים בעץ הם הסיווגים האפשריים. לא בחרתי  
באלגוריתם זה לפROYיקט שלי מפני שכבר עשית פROYיקט שמשתמש באלגוריתם זהה ורציתי  
ללמוד משהו חדש.

מן שרצית אלגוריתם מסווג יותר בחרתי להשתמש בראש עצבית מלאכותית (Artificial Neural Network – ANN) או בבחירה ראש נוירונים.

## מטרות / יעדים

- ללמידה דברים חדשים בכוחות עצמו ללא מסגרת שיעורית לימודית
- להתנסות בכתיבת פרויקט רחוב היקף
- להתנסות בפיתוח קוד לקראת השירות הצבאי והאזורות
- לסכם את החומר הנלמד מהຕיכון ועד כיתה י"ד
- להתעסוק בטכנולוגיה העתידית (בינה מלאכותית)
- לבנות אתר בפעם הראשונה
- ללמידה עוד דברים במוזיקה

## מדדי הצלחה למערכת

המערכת צריכה לתת מענה למספר דברים:

- סיווג כמה שייתר נכון
- אתר נוח וידידותי למשתמש
- המערכת לא תכלול שגיאות ובאגים

מבחינת הסיווג המערכת צריכה לתת קירוב טוב לז'אנר, ככל שהקירוב יהיה יותר חזק המערכת תואFINEן כמצויה יוטר. על מנת להגיע לקירוב טוב צריך שככל אבני היסוד של המודל יבנו על הצד הטוב ביותר, ככלمر איסוף נתונים בצורה מעולה ואימון המודל על הצד הכי טוב שאפשר.

איסוף הנתונים הדרוש מחייב הרבה ומענה על שאלות כמו:

- אילו פיצרים מתאימים בכך לסוג ז'אנר?
- האם נוכל לקחת את כל הפיצרים שמספקים לנו בdataset?
- נוכל לסיווג יותר מארבעה ז'אנרים שבחרנו?

ולאחר הרבה מחקר מצאנו את הפיצרים שמספקים לנו בשבייל לסיווג, והבנו שגם ניקח את כל הפיצרים שנדרשו לנו תחילת הלמידה ייקח יותר מדי זמן ואוותה תשובה גם לשאלת השלישית, מפאת הזמן נבחר לסיווג רק ארבעה ז'אנרים.  
באיומון המודל התלבטתי עם איזה framework לעבוד, והיכן אוכל לאמן את המודל כך שלא ייקח יותר מדי זמן לאמן אותו, ואיך להשתדל להימנע מההתאמת יתר.

スク הכל במידה ובאני היסוד של למידת המכונה געשו כראוי, המכונה תביא את הקירוב הכי טוב שיכולה ושמה נמדד אם היא מוצלחת או לא.  
הערה: בפרק **בדיקות והערכת מושבר** בהרחבת הערכת המודל.

## היאור מצב קיים

קיימים קיימים כלים לזיהוי שיר (על ידי זיהוי הדפוס בספקטוגרמה שלהם) כמו shazam ועוד. אך בעית סיווג ז'אנר זו בעיה רצינית יותר שמצויכה התעסקות עם למידה והפקת נתונים מהמחשב, וכך קיימים כמה כלים שיוכולים לסיווג ז'אנר, אך לא מצאתי כל שבאמת עשו את העבודה בצורה הטובה ביותר.

בנוסף ישנה אפליקציה لأنדרואיד שנקראה AI Music Genre Identifier או לאחר בדיקתה היא לא עבדת בצורה מספיק טוב. וכך לקחתי על עצמי את האתגר לנסות להצליח לבנות כל שיעשה זאת זה בצורה הטובה ככל אפשר.

## ניתוח חלופות מערכתי

חלופה	יתרונות	חרוגנות
SpringBoot framework with maven	עללה שרת tomcat בצורה מהירה, עם ניהול sessions בצורה אמינה וטובה, והתמודדות עם ניתוקים	מערכת סגורה יותר, Ai אפשר למש用工 עצמן כל מיני דברים
בנייה שרת מאפס	IMPLEMENTATION_OF_CLOUD_IN_SELF_STANDING_MODE	רבה זמן ומאמץ שפותה שיר לlibat הפרויקט
Vaadin framework	IMPLEMENTATION_OF_CLOUD_IN_SELF_STANDING_MODE	מערכת סגורה יותר, Ai אפשר לעשות איתה הכל
HTML	IMPLEMENTATION_OF_CLOUD_IN_SELF_STANDING_MODE	לוקח יותר זמן ללמידה ולהתעסק אליו, מה שפותה קשור לlibat הפרויקט
Dataset in H5 format	IMPLEMENTATION_OF_CLOUD_IN_SELF_STANDING_MODE	התמיכה בשפת java מוגבלת מאוד
Dataset in CVS format	IMPLEMENTATION_OF_CLOUD_IN_SELF_STANDING_MODE	בשיתוף של הרבה מידע יכול להימצא כפחות עיל
Deeplearning4j framework	IMPLEMENTATION_OF_CLOUD_IN_SELF_STANDING_MODE	בספריות של maven חסודות פעולות חשובות
Encog framework	IMPLEMENTATION_OF_CLOUD_IN_SELF_STANDING_MODE	מערכת סגורה, ישנים כמה דברים שלא קיימים בה
IMPLEMENTATION_OF_CLOUD_IN_SELF_STANDING_MODE	הבנייה הרבה יותר עמוקה	לוקח הרבה זמן ומחקר למש用工 עצמן
SQL	IMPLEMENTATION_OF_CLOUD_IN_SELF_STANDING_MODE	נשמר בצורה לוקאלית
MongoDB	IMPLEMENTATION_OF_CLOUD_IN_SELF_STANDING_MODE	לא כולל כמה פעולות של איחוד טבלאות בקלות

## תיאור החלופה הנבחרת

בחرتني להשתמש בשפת Java בפרויקט, מפני שהוא שפה נוחה שהכרתי לעומק בשנות הלימודים שלי.

SpringBoot: מכיוון וליבת הפרויקט שלי היא לא להקים שרת, אלא לוסף זינאר, משתמש בכלים שעושה את זה למען ותעסוק בלבד יוטר.

Vaadin framework: במקום לבזבז הרבה זמן יקר, זו חלופה הנדרת לבניית צד הלקוח מוביל לכתוב בשפת HTML אלא הכל כתוב בשפת Java וצריך רק ללמוד איך להשתמש. Dataset in CSV format: לאחר ניסיון רב להשתמש בפורמט H5 וביעילות תאימות רבות החלטתי לעבור לפורמט אחר.

Encog framework: לבניית הרשות בצורה נוחה ואמינה. MongoDB: לשמירה הנתונים בענן מכיוון שלא רציתי שמירה לוקאלית של נתונים.

## אפיון המערכת

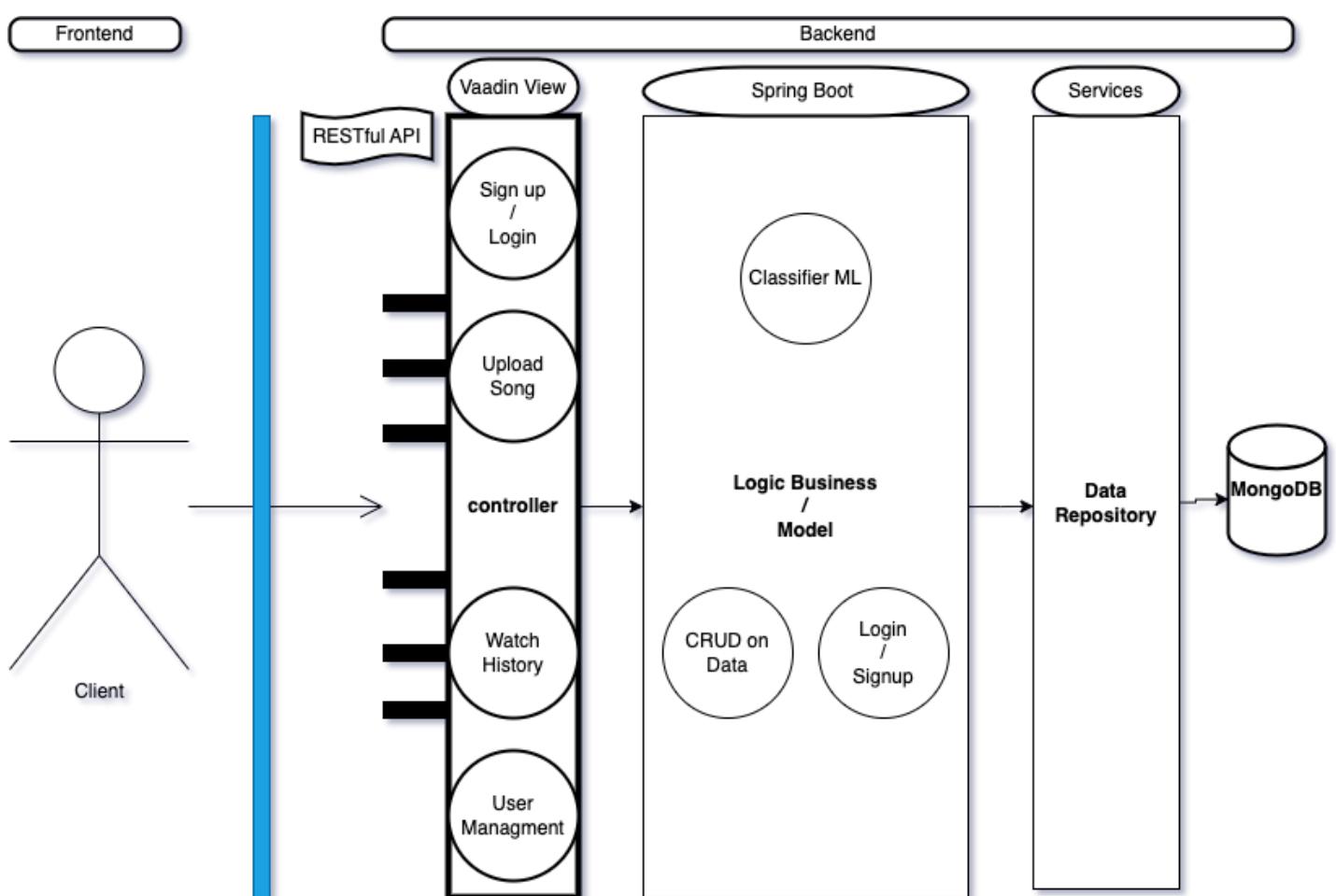
### נитוח דרישות המערכת

המערכת צריכה לספק תמיכה לדרישות הבאות:

- רשות מאומנת על מידע קיימ
- חיזוי ז'אנר לשיר שלא קיים במאגר המידע
- התחברות / הרשמה לאתר
- צפיה בהיסטורית הסיווגים של אותו משתמש
- דף ניהול משתמשים
- העלאת קובץ שיר

### מודול המערכת

להלן תרשימים מודול המערכת:



### אפיון פונקציונלי

- Login : (Authentication) זיהוי המשתמש באמצעות טכנולוגיית הסשנים.
- Sign up : במידה ואין חשבון קיים למשתמש, יכול באמצעות שירות הרשמה ליצור חשבון חדש ולהשתמש בשירותי האתר.
- Sessions , (Sessions), נותן ללקוח אפשרות להתחבר על מנת להשתמש בשירותי האתר וצפיה בהיסטוריה וכו'.
- Upload File : מתן העלתת קובץ של שיר על מנת לקבל את הדיאןר שאליו שיר ולהכנסו לרשימת ההיסטוריה.
- Predict Genre : חיזוי דיאןר השיר שהמשתמש העלה, זהו השימוש באлогוריתם למידת המכונה, לאחר שהמודל מאמין כבר על מידע קיים.
- User Management : אפשרות ניהול משתמשים בקלות (מבצע על ידי משתמש מסווג מנהל) תוך ביצוע פעולות על - DB (פעולות יצירה, עדכון, מחיקה ושליפה).

### ביצועים עיקריים

ביצועי המערכת נבדקו באמצעות העלתת קובץ לשרת וקבלת החיזוי ממנו, דבר שלא לוקח הרבה זמן. על מנת לקבל את החיזוי המערכת מעבדת את המידע בשיר על ידי ייצוא הפיצרים (Feature Extraction) והכנסתם לתוך המודל. תהליך אימון המודל זו פעולה שדורשת זמן רב כדי ליצור תוצאות טובות של חיזוי המודל. קבלת תשובות מהשרת על ידי שליחת בקשות מניבה תוצאות טובות, והשרת מתמודד איתם וכך.

### אילוצים

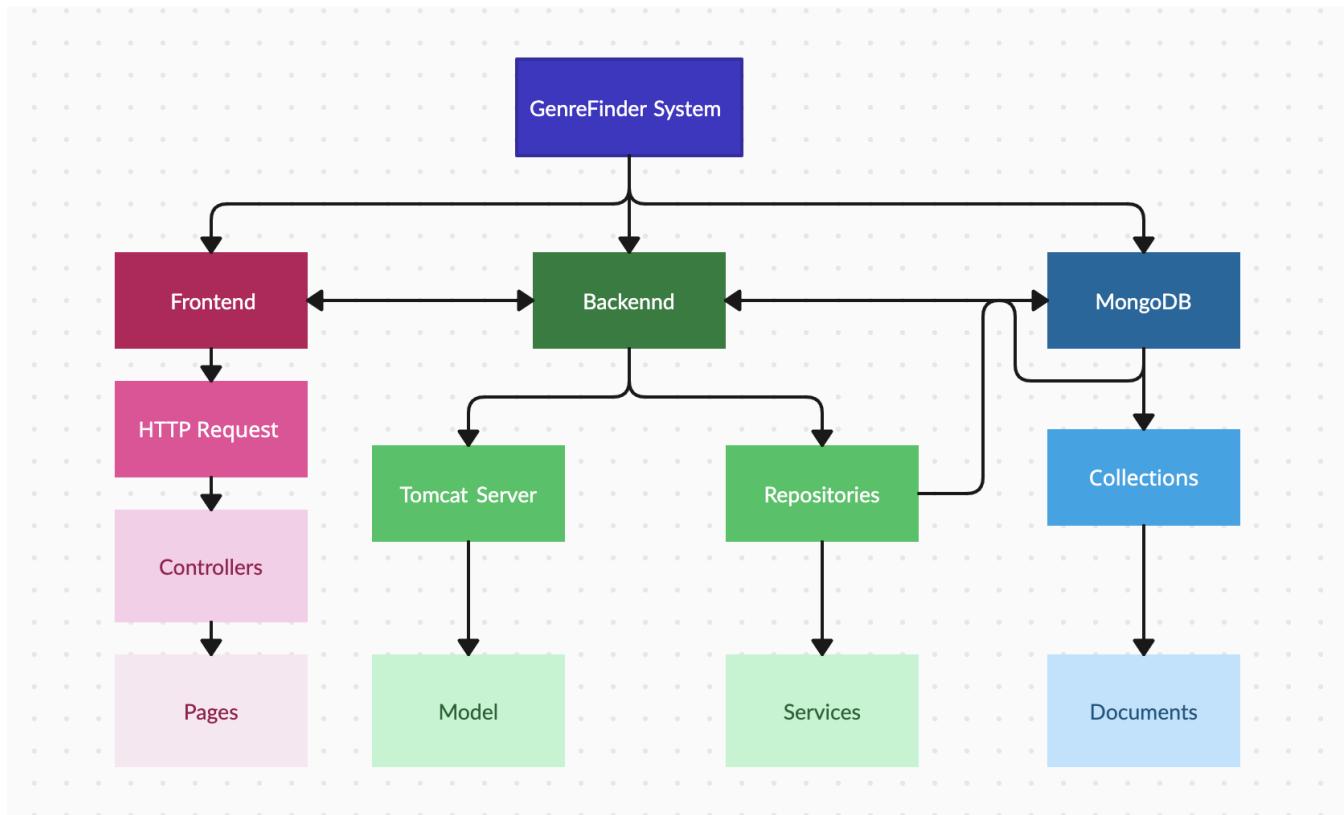
האילוץ העיקרי הוא הזמן, מכיוון שנכנסתי לעולם שלא הייתי בו אף פעם הייתה צריכה ללמוד הרבה מאוד חומר תיאורתי בעצמי, ולאחר מכן לנסות לנסות לעבור לפרויקטיה של קבוצה זמן רב גם היא.

התקינה של סדריות שעוסקות בקבצי שמע שנדרדק אליהן. שימוש ב - dataset מוקן מהאינטרנט מפתח הזמן.

## תיאור הארכיטקטורה

הארQUITקטורה של הפתרון המוצע (Top-Down Level Design)

להלן תרשימים ארכיטקטורה בסיסי של המערכת בפורמט



### תיאור הרכיבים בפתרון

#### הרכיבים העיקריים בצד השירות:

- **שרת MongoDB:** שרת לשימור נתונים ב – Database בענן בתצורה "לא רלציוני" (NoSQL). כלומר הנתונים נשמרים באוספים (Collection) ובכל אוסף נשמרים מסמכים (Documents).
- **שרת Tomcat:** בעזרת Tomcat Framework אנו מעלים שרת HTTP שמאזין לבקשת וידע לטפל בהם בעזרת בקרים (Controllers) שהם מנהליים בעזרת Springboot. בתוך השירות יושבים המודל, שהוא קורים כל החישובים הלוגיים (אלגוריתם הליבת – מודל המערכת) ומאגרי נתונים שמנוהלים באמצעות שירותים (Services) שמספק לנו השירות ה – DB שלנו, וכמוון עם תוספות משלנו.

### הרכיבים העיקריים לצד הלוקו:

- על מנת להתחבר לשרת שלנו אנו שולחים בבקשת HTTP ומקבלים תשובה. הבקשה נשלחת לבקרים (Controllers) דרך השרת והבקרים יודעים איפה לדוח לשלוח ללקוח על ידי בקשה ה – HTTP (נתיב הכתובת, כמו למשל איזה שירות אנו מבקשים מהשרת. לדוגמה: הדף הראשי "Main").
- הדים נכתבו בעזרת Vaadin Framework שהוא כתיבה של HTML באמצעות מחלקות ב – Java, בדים יש קומponentes HTML שימושו באמצעות Java.

### תיאור תהליכי של מערכת הפעלה המתבצעים בפרויקט

אחר ואנו מעלה שרת Tomcat באמצעות Springboot, השרת מקבל socket ופורט פנו' במחשב המערכת הפעלה. בנוסף לכך, השרת מסוגל לטפל בכמותLKochות בו זמן, זאת אומרת מאחורי הקלעים מתנהל ניהול של Threads (תהליכונים) במערכת הפעלה. פונקציית חילוץ הפיצרים בפרויקט מוגדרת כ – "קטע קרייטי" מפני שהיא מರיצה שפת Python שכותבת את הפיצרים לתוך קובץ על מנת לקרוא ממנו לאחר מכן ולהכניס את הנתונים לתוך מערכת. קטע קרייטי במחשב מנוהל על ידי כל' בرمת מערכת הפעלה שנקרא Semaphore".

### תיאור פרוטוקולי התקשרות

פרוטוקול HTTP: הוא פרוטוקול תקשורת שנועד בעיקר להעברת דפי HTML ללוקו', נמצא בשכבה האפליקציה במודל שבע השכבות (מודול ה – OSI). הפרוטוקול מאופיין כחסרי מצבים (Stateless) כלומר ללקוח שולח בקשה, השרת עונה תשובה והחיבור נסגר. בפרויקט שלי נשלחות בקשות HTTP לשרת על ידי תוכנת צד לקוח (דף אינטרנט, או כל כל' שידוע לבנות בבקשת HTTP) והשרת מחזיר תשובה ומנטק את הקשר (על מנת לשמור את הקשר אנו משתמשים בטכנולוגיית עוגיות Cookies) ששומרת פיסת מידע קטנה אצל לקוח שלשרת יש גישה אליה על מנת לzechot את הלוקו' (או כל מיני שימושים אחרים).

ישנם כמה סוגי של בקשות HTTP (אני לא משתמש בכלם וכן לא אסביר על כלם):

- בקשת GET: בקשה שמיועדת לקבל מידע שנמצא על השרת
- בקשת POST: בקשה המכילה גוף הודעה, כמו למשל הלקוח שולח לשרת מידע גדול

### שרת-לקוח

בחרתי להשתמש בפרויקט זה בארכיטקטורת שרת-לקוח (Client-Server) ולא בעמידת עמית (Peer to Peer). מפני שאני רוצה שיהיה לי שרת אחד שמספק את כל השירותים לכל לקוח אפשרי, שלא כמו בעמידת עמית שכולם משמשים גם כשרתים בפני עצם, מפני שאצל הלקוח יוזם את החיבור לשרת ולא ההפן.

בנוסף, בחרתי להשתמש בטכנולוגיית "לקוח רזה" על מנת שלא להכביר על צד הלקוח בדברים מיוחדים, אלא במידה והוא רוצה לפנות לאחד השירותים הוא שולח בקשה HTTP לשרת ומקבל תשובה.

## תהליכי אבטחה מידע במערכת

### תיאור ההגנה

מן שאני מנהל משתמשים בפרויקט, הייתה צורך לוודא שימוש יכול להגיע רק לדפים שתהייה לו גישה אליהם (תהליך זה נקרא אונטנטיקציה - Authentication).  
בניתי את התצורה ככזה:

- במידה והמשתמש לא רשום, אני לא נותן לו אף אחד משירות המערכת עד שהוא לא פותח חשבון.
- במידה והמשתמש כן רשום והוא ברמת משתמש רגיל, תהיה לו גישה לשירותים הבסיסיים (חיזוי ז'אנר, היסטוריות חיזויים).
- במידה והמשתמש כן רשום והוא ברמת משתמש מנהל, תהיה לו גישה לבנוסף לשירותי המשתמש הרגיל, גישה לדף ניהול המשתמשים (User Management).

על מנת לוודא את זהות המשתמש השתמשתי בטכנולוגיית העוגיות (Cookies) שזהה שמיירת מידע אצל הלוקו שהליך מאפשר להיכנס לשירותים מסוימים באתר. בנוסף לכך ישנו אוסף (Collection) בשרת ה – DB למשתמשים ומידע עליהם, על מנת להשתמש בפונקציית ה – Login וה – Signup עם שמירת הנתונים ב – DB.

## למידת המכונה

### תהליכי איסוף הנתונים

אנו עומדים לפני בעית למידה מונחית לסיווג ז'אנר של שיר, ולכן הנתונים שלנו יהיו אוסף של דוגמאות פתרונות, כלומר שירים עם המידע הנוחז עליהם והז'אנר שלהם. התלבטתי רבות איזה פיצרים אני יכול להוציא משיר ומה באמת נכון לי לז'אנר, כלומר כנראה שעוצמה הצליל תהיה פיצר טוב, אך שם המבצע לא יהיה רלוונטיפה. לאחר מחקר רב בתיאורית המוזיקה הבנתי שהפיצרים המתאים לי יהיו: טמפו, מושך השיר, הכרומה של השיר.

בתוך הכרומה מוכל וקטור של 12 ערכים עיקריים שונים עוד הרבה מידע נחוץ כגון גובה, גוון וכו'.

לאחר שהבנתי בדיק מה אני צריך מכל דבר תחילה לעבוד על בניית dataset מתאים, לאחר זמן מה הבנתי שלבנות dataset בעצמי זה לא יעיל ויקח לי את כל הזמן העבודה על הפרויקט וכן החלטתי להשתמש בתחלופה קיימת, dataset מוקן מה האינטרנט.

ישנו dataset מפורסם מאד לשירים שנקרא: MilionSongDataset שאפליקציות גדולות כמו Spotify משתמשות בו לקבל מידע על שירים מרחבי העולם. ה - dataset שלהם מכיל את כל הפיצרים שהיתי צריך, יותר אף אסתפק בפיצרים שכינתי. בנוסף, ה - dataset שלהם מכיל המון מידע על המון שירים וכן שוקל כ – 280GB ואני לא אוכל לצרף אותו לפרויקט שלי, שכן הם מציעים חלופה של תת dataset ששוקל כ – 2GB וכן העבודה איתו. לאחר מאיץ רב לעבוד עם ה – dataset הזה נשמר בפורמט H5, היתי צריך להוציא ספריה לפרויקט שלי שiodעת לעבוד עם הפורמט הזה, מצאת לבסוף ספריה שiodעת לעבוד אליו אך זה היה מסובבל מדי וכן החלטתי לחפש dataset אחר.

ה – dataset השני שמצאתו הוא נקרא "GTZAN Dataset - Music Genre Classification", זה dataset שנבנה במיוחד לביעות סיווג ז'אנרים. ה – dataset מספק עשרה אלפיים שירים מעשרה ז'אנרים שונים (100 שירים לכל ז'אנר) והוא מגיע בפורטט 'דידוטי' יותר, CSV. אמנם מפני שהוא dataset מספק רק 100 שירים לכל ז'אנר (כל שיר הוא 30 שניות), אך לקחת את כל השירים לפי התקיות של הז'אנרים שלהם ובאזור שפת Python חילקתי כל שיר לעשרה חלקים וחילצתי את הפיצרים מכל קטע אחד כזה על מנת להכניס לקובץ ה – dataset החדש שייצור, שמכיל 1000 דוגמאות מכל ז'אנר (5% הכל 4000 דוגמאות), וכל זה על מנת שתהיליך האימון יבוצע בצורה טובה יותר וכך להימנע ממצב של Overfitting (התאמת יתר – שהמודל לומד את הנתונים שהביאו לו בצורה מותאמת מדי למידע, דבר שיגרום להרבה שיבושים בפונקציית החיזוי).

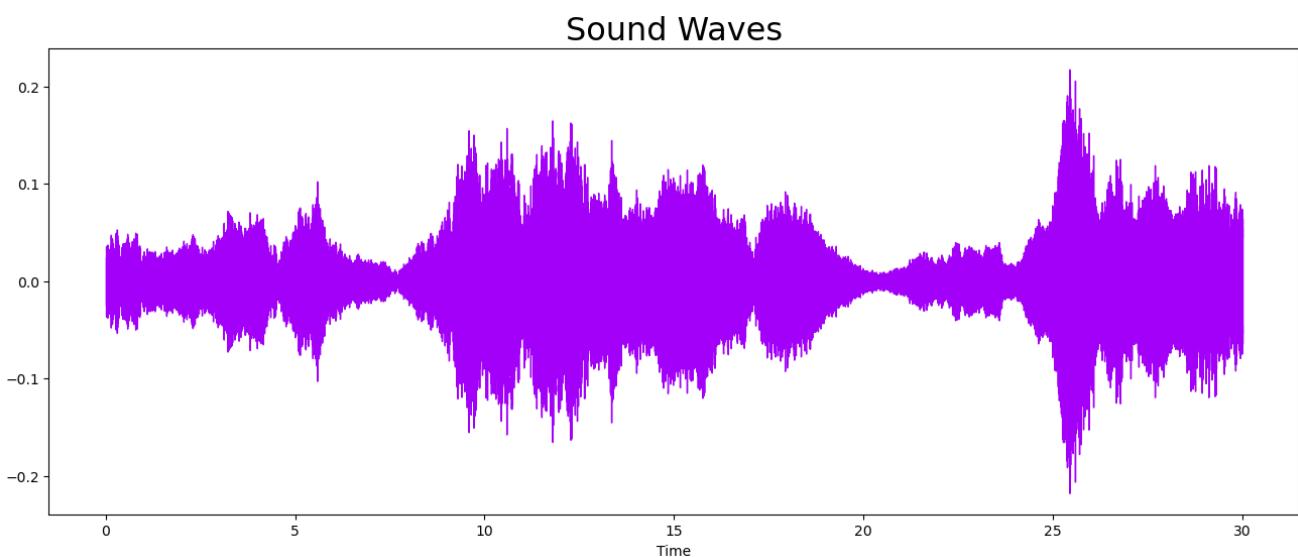
לאחר שציירפתי את ה – dataset לפרויקט היתי צריך לדלוות ממנו רק את הנתונים שבאמת ארצها לעובד איתם. במקום לקחת את כל עשרת היז'אנרים, לקחתי רק ארבעה שאיתם העבודה: רוק, פופ, קלאסי ובלוז. لكن כתבתתי פונקציה שמוחקת מהקובץ את כל היז'אנריםyla לא לולוונטיים בשביבי. לאחר מכן חקרתי אילו פיצרים מסוימים חוץ מהכרומה לשיווג ז'אנר, הגיעתי למסקנה שהיא אכן 17 פיצרים:

- chroma\_stft\_mean: מייצג את הערך הממוצע של מקדמי טרנספורמציה פורייה בזמן קצר (STFT). תכונותroma לוכדות את התפלגות האנרגיה בשיעורי הגובה המוזיקליים השונים ומספרות מידע על התוכן ההרמוני של אותן האודיו.
- chroma\_stft\_var: מציין את השונות של מקדמי STFT של chroma. תכונה זו משקפת את השונות או ההתפשטות של מידע הכרומה על פני אותן האודיו.
- rms\_mean: מدد לאmplיטודה הכלולה של אותן שמע, מייצג את הערך הממוצע של שורש ממוצע הריבועים (RMS).
- rms\_var: מייצג את השונות של ערכי ה – RMS, לוכד את השונות באmplיטודה על פני אותן השמע.
- spectral\_centroid\_mean: מייצג את הערך הממוצע של המרכז הספקטראלי על פני אותן.
- spectral\_centroid\_var: מציין את השונות של ערכי המרכז הספקטראלי, מספק מידע על השונות של התפלגות התדרים באות האודיו.
- spectral\_bandwidth\_mean: רוחב פס ספקטראלי מודד את טווח התדרים באות שמע, מייצג את הערך הממוצע של רוחב הפס הספקטראלי על פני אותן.
- spectral\_bandwidth\_var: הוא מייצג את השונות של ערכי רוחב הפס הספקטראלי, לוכד את השונות של טווח התדרים על פני אותן השמע.
- rolloff\_mean: מייצג את הערך הממוצע של תדר ה-rolloff (מדד לצורת התפלגות התדר) על פני אותן.
- rolloff\_var: מציין את השונות בצורה הצורה התפלגות התדרים.
- zcr\_mean: מציין ממוצע מדידת הקצב שבו אותן משתנה הסימן שלו.
- zcr\_var: מייצג את השונות בקצב שינוי הסימנים באות האודיו.
- harmony\_mean: מייצג את הערך הממוצע של תכונת ההרמונייה (תפיסת היציבות והעיצור באות האודיו) על פני אותן.
- harmony\_var: מייצג את השונות של היציבות והעיצורים באות האודיו.
- perceptr\_mean: מייצג את הערך הממוצע של תכונת התפיסה או הפרצפייה (העוצמה והדומיננטיות של גבהים מסוימים באות שמע) על פני אותן.

- `perceptr_var`: מצין את השונות בעוצמה ובדומיננטיות של גובה האצלילים באות האודיו.
- `tempo`: מתייחס ל מהירות או הקצב שבו המוזיקה מושמעת. הוא מייצג את הקצב המשוער של אות השמע בפעימות לדקה (BPM).

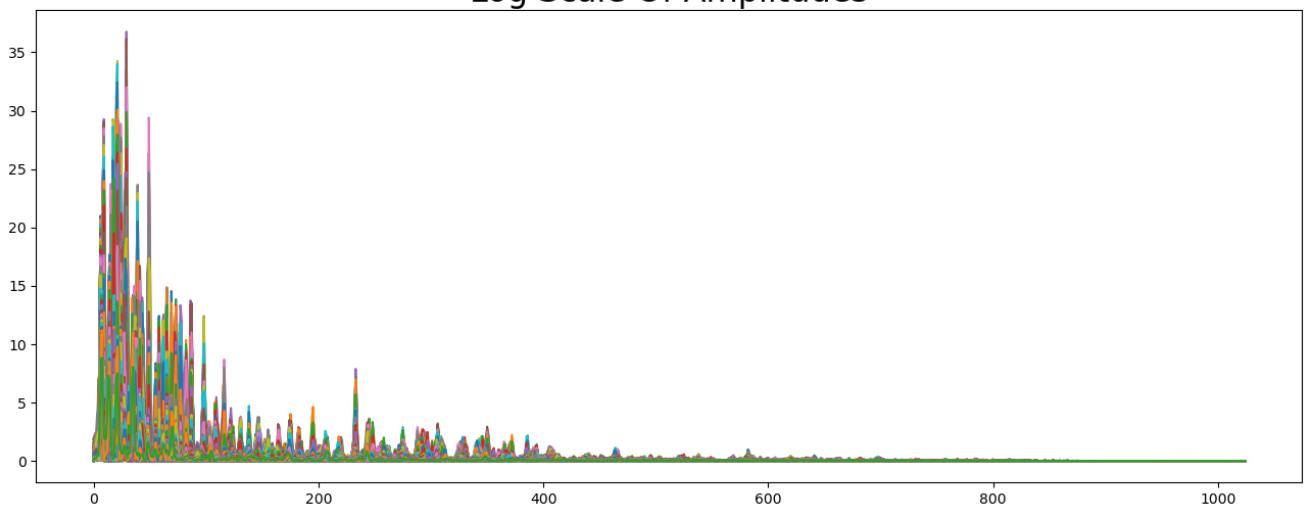
דוגמה ליצוא הפיצ'רים (Features Extraction) וניתוח המידע באמצעות שימוש בספריית Python Librosa של שיר שמו: "classical.00009" שהובא מתוך ה dataset

גלי השמע בשיר:

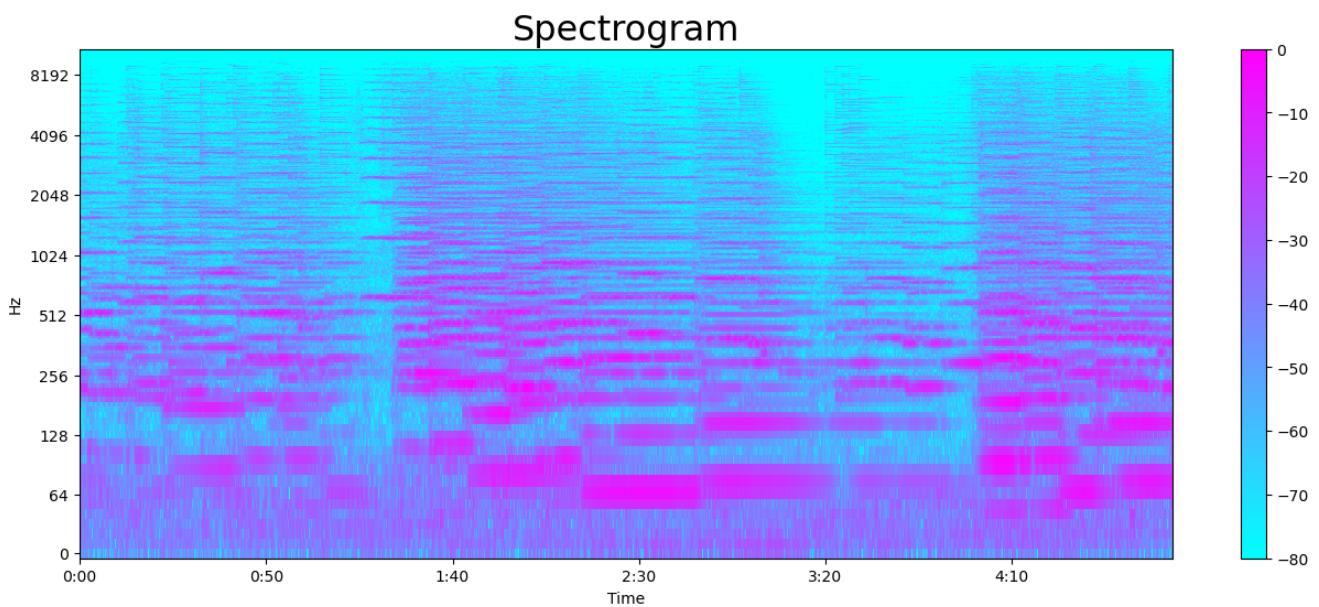


סולם היומן של האמפליטודות (באמצעות Fourier Transform):

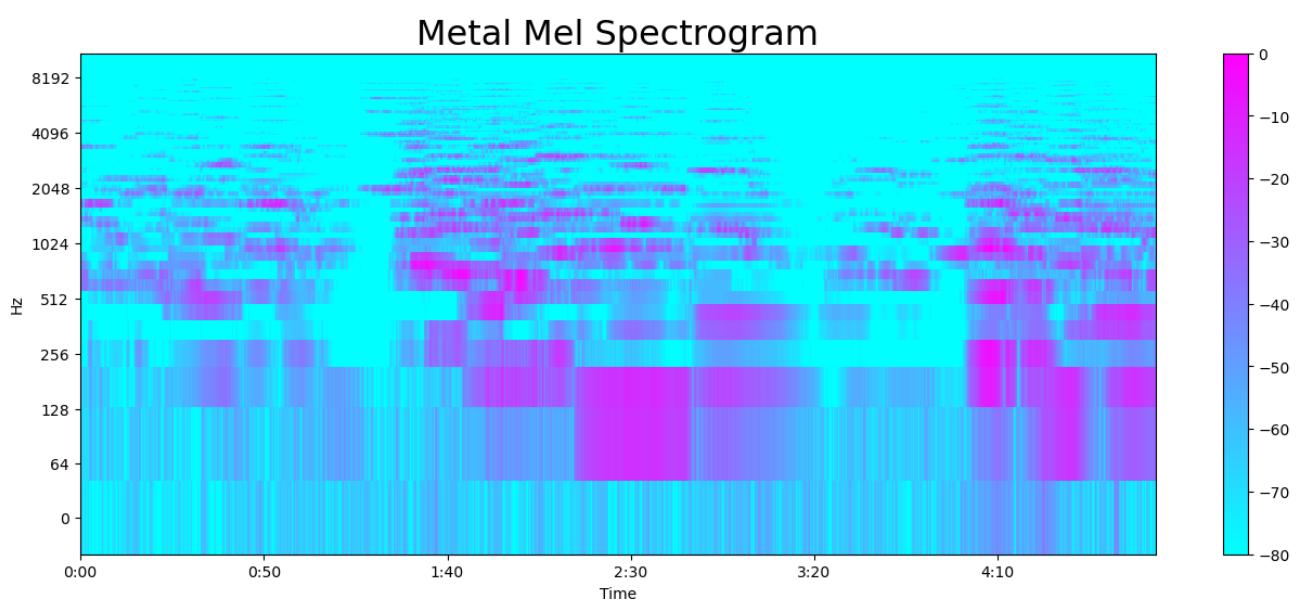
Log Scale Of Amplitudes



הספקטוגרמה של השיר:

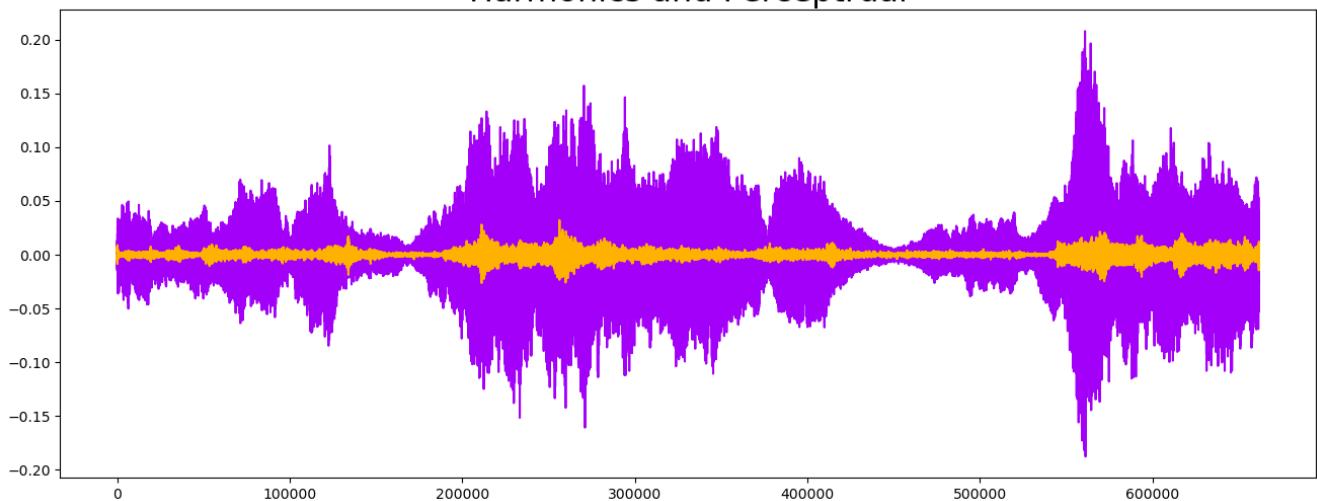


ספקטוגרמת מל (Mel Spectrogram) של השיר:



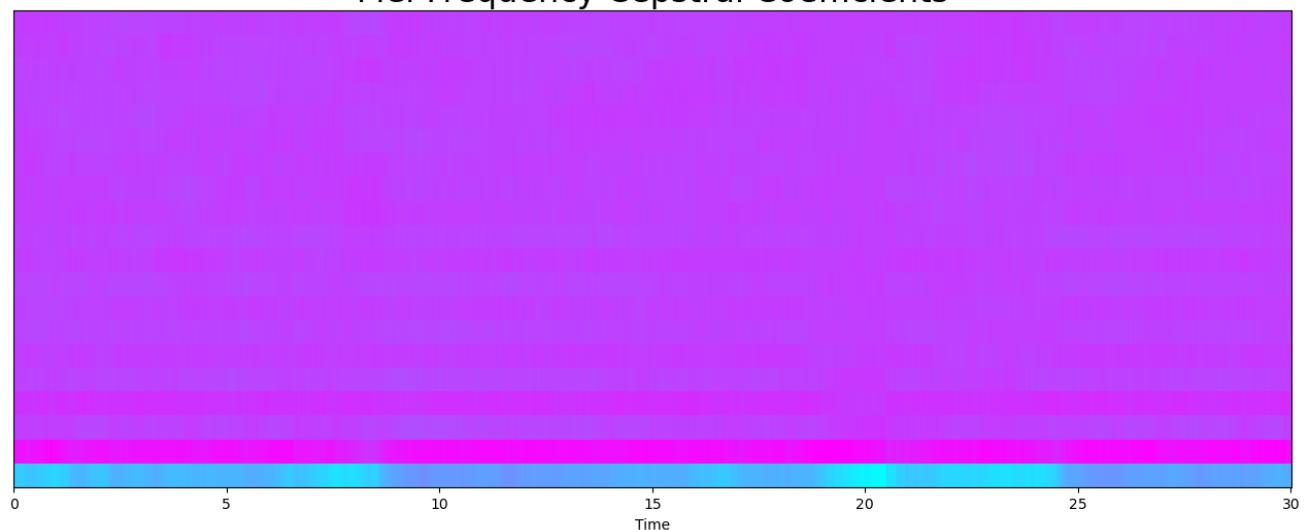
ההרמונייה והפרצפציה של השיר:

Harmonics and Perceptual



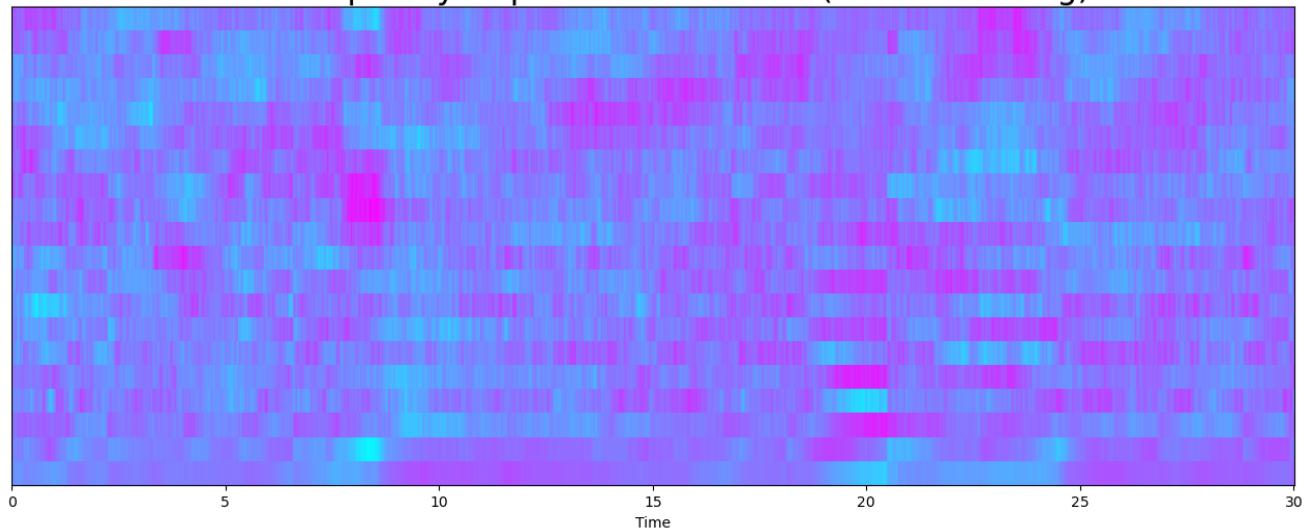
: של השיר) MFCCs (Mel-Frequency Cepstral Coefficients)

Mel-Frequency Cepstral Coefficients



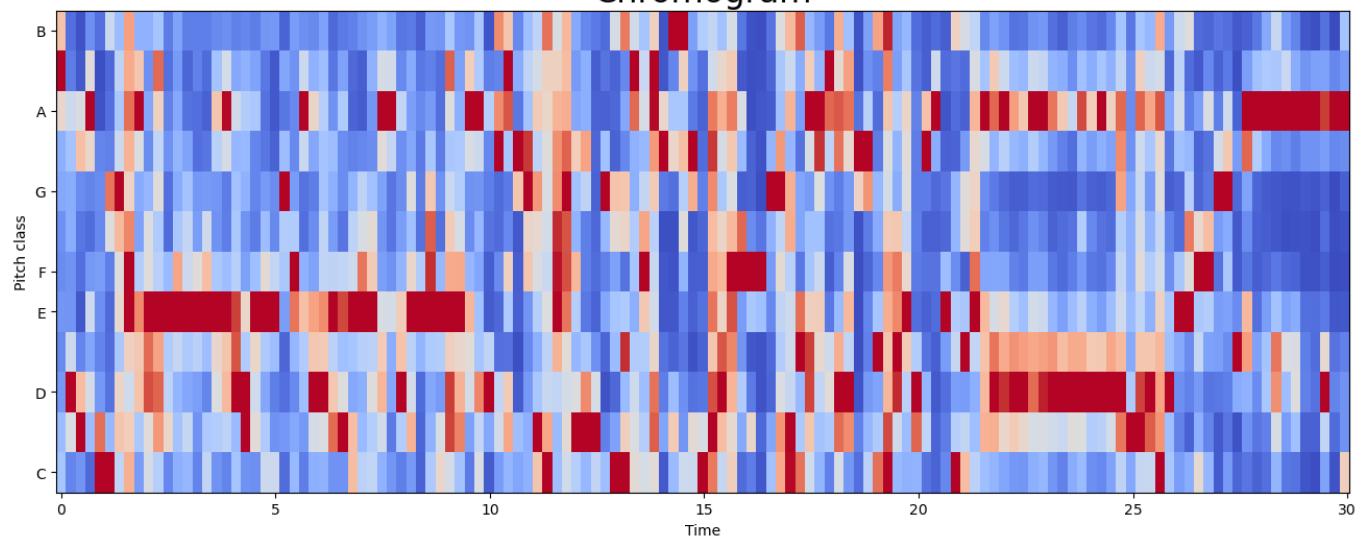
לאחר התאמת קנה המידה:

Mel-Frequency Cepstral Coefficients (after matching)



כרכומוגרפם השיר:

Chromogram



סיכון כל ערכי הפיזרים:

Feature	Mean	Variance
chroma_stft	0.38740283	0.08593862
rms	0.023791904	0.00019983428
spectral_centroids	1216.3713862548866	121036.90483760233
spectral_bandwidth	1582.8796259425067	72844.33023891032
rolloff	2560.7691505945477	691703.4337447325
zero_crossings	0.056318129206369354	0.05314639752906405
harmonic	-3.4384062e-05	0.000713687
perceptual	-7.0125345e-05	7.14925e-06
tempo	234.90767045454547	N/A

לאחר שניקיתי את ה – dataset מידע שלא רלוונטי אליו הייתה צריכה לפצל אותו לשניים, אוסף נתונים לאימון ואוסף נתונים לבדיקה.  
אוסף הנתונים לאימון (70% מהמידע) משומש לאימון המודל, כלומר אנו מבאים את כל הנתונים האלה למחשב בתור "דוגמאות פתרונות" על מנת שיבין את המשקל של כל פיזר בעניין סיווג הציגן.

אוסף הנתונים לבדיקה (30% מהמידע) משומש לבדיקת חיזוי המודל. כלומר, לאחר שהעבכנו למודל את אוסף האימון ניתן לו את אוסף הבדיקה ונשלח אותו לחיזוי המודל ונוכל לבדוק אם החיזוי שקיבלנו הוא הדינאר הנכון, במידה ולא האלגוריתם יעדכן בהתאם את המשקלים הנחוצים לפיצרים על ידי השימוש בהפרש בין המידע שהזינו לבין המידע האמיתי שבקובץ הבדיקה.

### סוג הלמידה הנבחר

על מנת לבחור את סוג הלמידה הרاءו לנו הינו צריכים לחקור אילו סוגים של מידה יש, ויש שניים עיקריים: למידה מונחית ולמידה בלתי מונחית.

בלמידה מונחית אפשריים למחשב ללמידה מתוך אוסף גדול של דוגמאות פתרות, לדוגמה אם רוצים שהמחשב יזהה אם יש חתול בתמונה, נתונים לו כמות גדולה מאוד של תמונות שיש בהן חתול ותמונות שאין בהן חתול ומזהירים לו מה יש חתול או מה אין חתול. לאחר האימון הזה נתונים לו תמונה שהוא לא ראה ושאלים אותו אם יש חתול או אין.

בלמידה בלתי מונחית נתונים למחשב אוסף גדול של מידע לא מတיג והמחשב צריך להבין איפה המכנה המשותף וכייד לתציג את הנתונים לפי כל מיני שיקולים. לדוגמה, מבאים למחשב איסוף דוגמאות מודדים רפואיים של אנשים, אבל לא אומרם לו מי חוליה ומילריא, המחשב אולי לא יוכל לדעת בוודאות מי חוליה או מי בריא אך הוא יוכל למצוא מכנה משותף בין המודדים לפי שיקולו.

סוג הלמידה הנבחר הינו למידה מונחית עם מודל דיסקרטיניטיבי, עובד בצורה הבאה: נתונים לemachine אוסף גדול של דוגמאות "פתרות" (בעניין שלנו שירים עם מידע עליהם ו蒂יג של הדגמאות משפיע על הפתרון (הסיווג הסופי).

קיים מגוון אלגוריתמים בלמידה מונחית, והעיקריים שבהם:

- אלגוריתם השכן הקרוב (*k*-Nearest Neighbors algorithm)
- עצי החלטה (Decision Tree)
- רשת נירוניים (Artificial Neural Network)

אלגוריתם השכן הקרוב (*k*-Nearest Neighbors algorithm) בעבודת סיווג מקבל שיר חדש שלא מכיר ומחפש תכונות דומות לשירים שהוא מכיר, הוא יחפש את השכן הכי קרוב מבחינת המידע על שאר השירים ויסווג אותו לאותה הקבוצה.

עץ החלטה (Decision Tree) עץ ביניاري גדול עם שאלה בכל צומת, התשובה לכל שאלה תנוט את השיר עד לעלה המסוימת. כל עלה הוא סיווג אפשרי, במקרה שלנו עלה יהיה ז'אנר.

רשת נירוניים (Artificial Neural Network), כמו שבמוה של בני האדם יש רשתות נירוניים אשר כל נירון אחראי לחישוב אחד קטן והעבירתו להלאה האלגוריתם הזה פועל, הוא בעצם מחקה את פועלות המוח שלנו על מנת לגרום למידה. בעזרתו אימון נוכן של המודול

ופונקציות הפעלה והפסד מתאימות נוכל ליצור למידה חזקה עם קירוב טוב מאוד. בפרויקט בחרתי למש את רשת הנוירונים כי היא נראה לי הכי מעניינת.

### נוסחאות ורשנותות למידה

לאחר ההחלה על אלגוריתם הלמידה רשת נוירונים (Neural Network), היה צריך לחקור את האלגוריתם לעומק וכייז הוא עובד.  
האלגוריתם עובד באופן זהה:

יש לנו רשת נוירונים, וכך נסביר קודם מהו נוירון אחד. במוח שלנו לנו נוירון (נקרא גם תא עצב) הוא תא שנמצא במערכת העצבים ומשמש כיחידת התפקוד הבסיסי של המוח. תפקידו הוא קלולות, לעבד ולהעביר מידע.

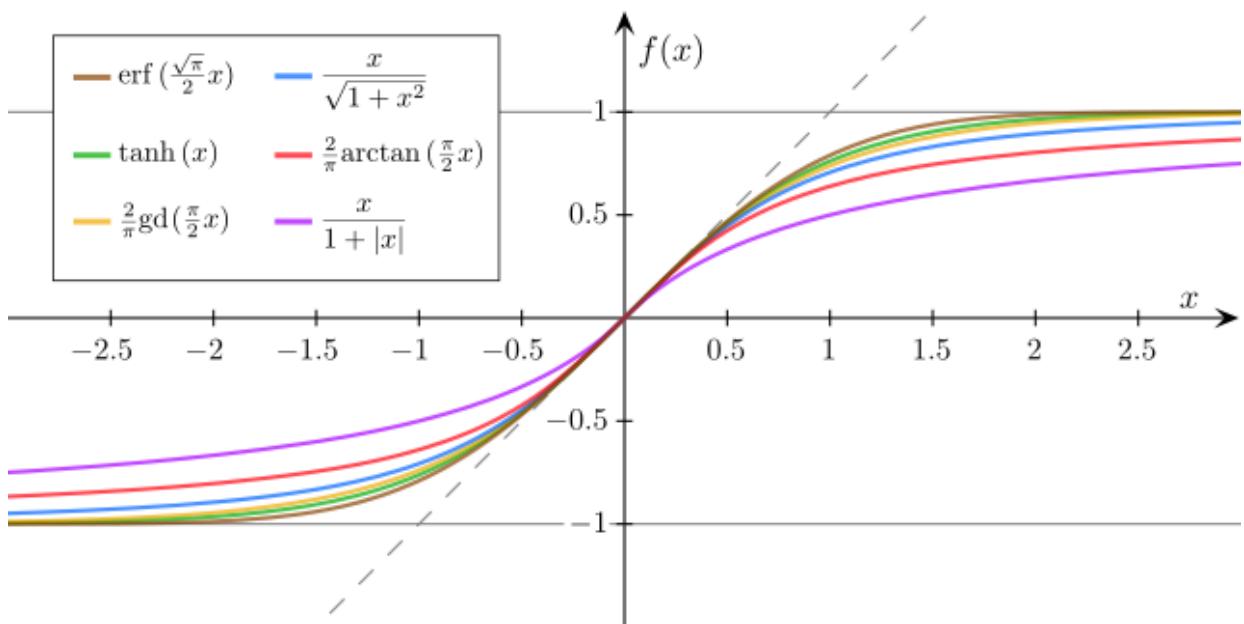
ככה גם אצלנו כל נוירון הוא קולט מידע כלשהו, מעבד אותו ועביר אותו לנוירון הבא. ככלומר, יש לו קלט, חישוב מסוים ופלט.

הчисוב שהנוירון עשה תלויות בבעיה שהוא מנסה לפתור, החישוב נקרא "פונקציית הפעלה" (activation function). ישנו כמה פונקציות אפשריות לשימוש. אני בחירת להשתמש בפונקציית הסיגמודיד (sigmoid).

פונקציית הסיגמודיד (sigmoid) יוצרת בעצם עקומה בצורה האות S ומוגדרת בין 1 ל -1. ההגדרה הפורמלית:

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

השוואה בין "עקומות סיגמודיד" בגרף:

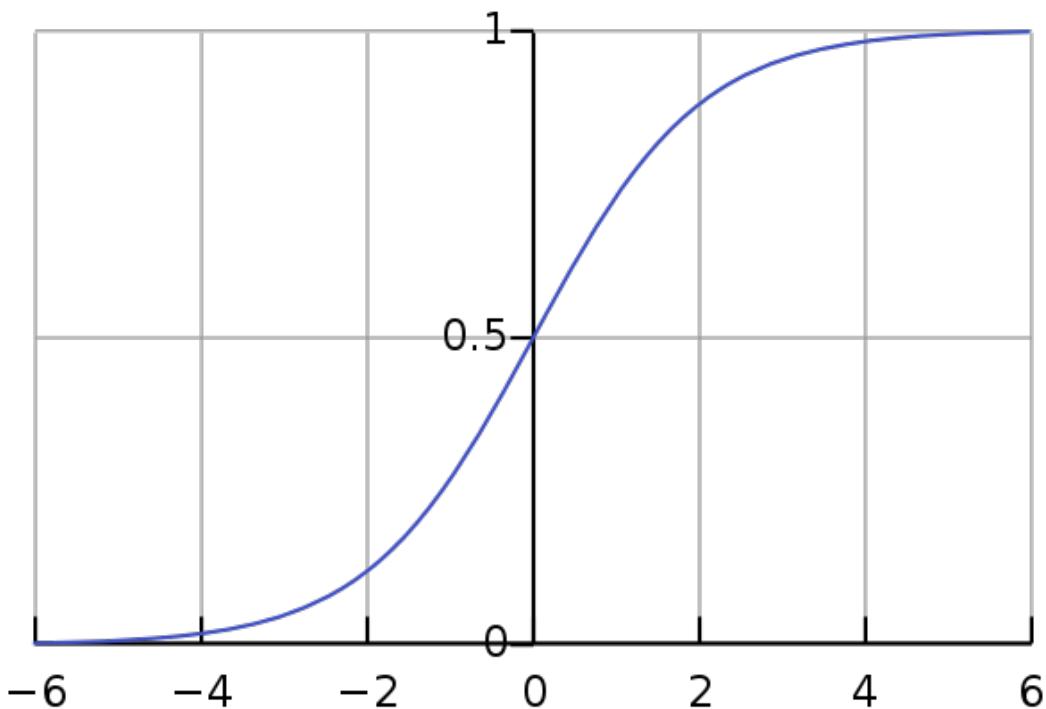


ישן כמה שימושים לפונקציה זו, וכך אצלנו היא נקראת דואק "הfonקציה הלוגיסטיבית" (logistic function) ומוגדרת על ידי הנוסחה הבאה:

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}$$

כאשר  $x_0$  הוא ערך ה- $x$ -של נקודת המרכז של הסיגמודיד.

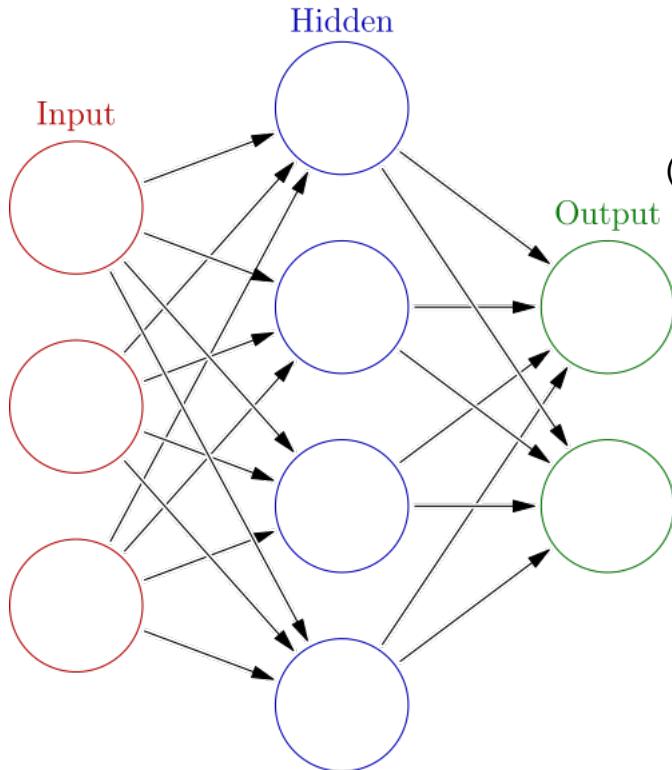
graf הפונקציה הלוגיסטיבית:



לכל ניירון ישנו ערך נוסף: משקלות (weight), הניירון מכניס את המשקלות לפונקציית הפעלה שלו, המשקלות היא זו שבאים שולטת על ערך הפלט שיוצא מהניירון.

לאחר החישוב הזה על הפלט שהניירון מקבל הוא מעביר אותו לנירון הבא.

ארQUITקטורת הרשת נראית כך:



יש לנו שכבות (layers) של נוירונים:

- שכבת הקלט (input layer)
- שכבות חבויות (hidden layers)
- שכבת הפלט (output layer)

מה שנכנס במקורה שלנו אל שכבת הקלט הוא השיר לסיאוג, בשכבות החבויות המכונה תצטרך לבדוק מכנים מסווגים למידע שעלייה אומנה ולהעביר לשכבת הפלט, ובשכבת הפלט יהיו לנו 4 ז'אנרים אפשריים: רוק, פופ, קלסי ובלוז.

מן שאנו מסוויגים 4 ז'אנרים, והמכונה היא נותנת רק קיורוב לז'אנר שאותו חוזים אך הפונקציה בשכבת הפלט צריכה להיות פונקציה הסתברותית, ולכן אנו משתמש בפונקציית softmax.

פונקציית ה – softmax מקבלת וקטור של מספרים ומנרמלת אותם להתפלגות של  $k$  סיווגים אפשריים (במקרה שלנו 4). הפונקציה מוגדרת כך:

The standard (unit) softmax function  $\sigma : \mathbb{R}^K \rightarrow (0, 1)^K$  is defined when  $K \geq 1$  by the formula

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad \text{for } i = 1, \dots, K \text{ and } \mathbf{z} = (z_1, \dots, z_K) \in \mathbb{R}^K.$$

לאחר שהבנו מה כל ניירון ושכבה ניירונים עשו, נסביר כיצד הלמידה באמצעות מתבצעת. לכל ניירון ישנו את המשקלות שלו שבעצם שליטה על הערך שיוצא ממנו, זאת אומרת המשחקן עם המשקלות יכול לתת לנו ערך פלט שונים. ולכן זה בעצם המשחקן, אנו צריכים לכוון את המשקלות האלו עד הפלט הרצוי שלנו, וכך בעצם מתבצעת הלמידה, ישנים כמה אלגוריתמי כיוון המשקלות, המפורטים שביהם הוא אלגוריתם Gradient Descent. תהליך עדכון המשקלות נקרא "תהליך האימון" (training).

**אלגוריתם Gradient Descent:**  
אלגוריתם זה הוא שיטת אופטימיזציה מסדר ראשון למציאת מינימום מקומי של פונקציה. ככלומר הוא יכול לתת לנו קירוב היכן נקודת המינימום שלנו, ולכן הוא יכול לעזור את השגיאה שלנו.

פונקציית השגיאה (error function) נקראת גם פונקציית ההפסד (loss function) היא הפונקציה המתארת את ההפרש בין מה שהמכונה חצתה בפועל לבין מה שבאמת הייתה אמורה לחצות, וכן האימון שלנו עוסק במצוור פונקציית השגיאה, ופה נכנס האלגוריתם שלנו - Gradient Descent. האלגוריתם יקח את פונקציית השגיאה שמוגדרת על ידי הנוסחה הבאה:

$$E(\vec{w}) = \frac{1}{2} \sum_d (t_d - o_d)^2$$

המטרה שלנו היא למזער את השגיאה שלנו, על ידי עדכון המשקלות, ככלומר נתחיל עם קטור משקלות רנדומליים ולאט לאט נמזער את השגיאה על ידי הפעלת האלגוריתם על הפונקציה ועדכון המשקלות בהתאם.

**הנוסחה לעדכון המשקלות של תא מסוים תהיה (בשימוש ה-Gradient Descent):**

$$w_i \leftarrow w_i + \eta \sum_d (t_d - o_d) \cdot o_d \cdot (1 - o_d) x_{i,d}$$

לאחר שיש לנו את המשוואה לעדכון המשקלות של תא מסוים, נדרש לעדכן בכל פעם את המשקלות של השכבות החבויות וכן קיימים אלגוריתם: "פעוף לאחרר" (-BackPropagation) שבעצם עובר כל פעם לבין מה קיבלנו ומה רצינו לקבל וمعدכן את המשקלות אחרת בהתאם.

## ביתוח ותרשים המערה

### תיאור אלגוריתמים עיקריים

למודל השימושי באלגוריתם רשת נוירונים (Neural Network) שהוא מודל חישובי מתකדם המורכב מיחידות קטנות שהן בעצם הנוירונים. הנוירונים בעצם מחוברים אחד לשני על ידי חלוקה לשכבות ובהיבור בין אחד לשני מוגדרים המשקלות (weights), כל נוירון מקבל קלט, מעביר אותו דרך פונקציית הפעלה שלו (activation function) ומעביר אותו לנוירון הבא אחריו ברשף.

בשכבת הקלט והשכבות החבויות ברשות השימושי בפונקציה הלוגיסטיבית של סיגמוואיד (sigmoid) בתור פונקציית הפעלה, פונקציה זו מקבלת קלט ומחשבת באמצעות הנוסחה:

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}$$

(מוסבר בהרחבה בפרק הקודם).

בשכבת הקלט השימושי בפונקציית softmax בתור פונקציית הפעלה, מפני שפונקציה זו הופכת את הקלט שקיבלה לנตอน הסתברותי של חיזוי המודל.

על מנת להשתמש ברשת נוירונים צריך להגדיר את ארכיטקטורת הרשת של אותה המשימה שאותה מנסים לבצע, ולכן ארכיטקטורת הרשת שליה היא מוגדרת כר' :

- 17 נוירונים בשכבה הקלט (מספר הפיצרים באוסף הנתונים)
- 256 נוירונים בשכבה החבوية הראשונה (על ידי ניסוי וטעיה)
- 128 נוירונים בשכבה החבوية השנייה (על ידי ניסוי וטעיה)
- 40 נוירונים בשכבה החבوية השלישית (על ידי ניסוי וטעיה)
- 10 נוירונים בשכבה החבوية הרביעית (על ידי ניסוי וטעיה)
- 4 בשכבת הפלט (מספר הסיווגים האפשריים)

סוג הרשת: רשת רגילה, יש כמה סוג רשתות נוירונים כמו (CNN ו – RNN) אך אני השתמשתי ברשת כללית שספקה על ידי Encog Framework.

תהליך הלמידה הוא תהליך חשוב מאוד שבו מאמנים את הרשות והיא לומדת מואסף הנתונים איך הפיצרים משפיעים על הסיווגים השונים, וכן זהו תהליך מסובך שמשלב כמה אלגוריתמים, בפרויקט החלטתי להשתמש באלגוריתמים הבאים:

- Gradient Descent
- Back Propagation

Gradient Descent: הוא אלגוריתם אופטימייזציה למציאת מינימום של פונקציה, במקרה שלנו אנו רוצים למצער את פונקציית השגיאה (شمוגדרת כהפרש בין החיזוי של המודל המקורי לבין מה שרצינו שהוא יזכה), ופה האלגוריתם מփש את המינימום בפונקציית השגיאה בהתאם לכך משנה מעט את המשקלים ושוב מփש את המינימום לאחר מכן, עד שהוא מוצא את המינימום.

Back Propagation: (פעוף אחרה) בראשו שלנו הנתונים זורמים קדימה על מנת למצוא חיזוי, ככלمر נכנס וקטור של פיצרים בתור הקלט אל הרשות ואנו מצפים שי יצא צ'אנר מהפלט. זהוי זרימה קדימה (Forward Propagation), אך פה אנו רוצים לכת לכיוון הנגדי, ככלומר לאחר שאלגוריתם ה - Gradient Descent מספק לנו כיצד לשנות את המשקל המתאים של הרשות נctrar לכת אחרת באמצעות האלגוריתם ולעדיין אותו.

זהו אלגוריתם הלמידה שלנו, אך ישנו עוד כמה פרמטרים (נקראים גם היפרפרמטרים (Hyperparameter) להגדיר את הלמידה):

- Learning rate: מגדר את גודל הצעד שבו מתעדכנים המשקלים של הרשות בתהליך מצער השגיאה.
- Batch size: מגדר את מספר דוגמאות האימון המשמשות במעבר קדימה או אחרת של הרשות. ככלמר, במקרים לעדכן את משקלי המודל לאחר כל דוגמא בודדת, אנו נלמד קבוצה של דוגמאות בבתול ולאחר מכן נעדכן את משקלי המודל.
- Momentum: טכניקה המשמשת להאצת תהליכי האופטימייזציה בראשת. עוזר למודל להתגבר על המינימום המקומי ולהתכנס מהר יותר על ידי הוספת חלק מהעדכון הקודם לשלב העדכון הנוכחי. ה – Momentum צובר את שינוי השיפוע לאורך זמן, ומאפשר למודל להמשיך לנوع בכיוון עקי, גם כאשר כיוון השיפוע המיידי משתנה. פרמטר המומנטום, מסוון לרובה בסמל  $\beta$  (ביטה), שולט בתרומות העדכון הקודם לעדכון הנוכחי. ערך טיפוסי ל- $\beta$  הוא בין 0.9 ל-0.99, מה שמצויב על תרומה גבוהה מהעדכון הקודם. ערך גדול יותר של  $\beta$  הופך את תהליכי האופטימייזציה לחלק יותר ומוסיע למודל לנוט באזורי שטוחים יותר של משטח האובדן.

על ידי שילוב של הגרדיאנט הנוכחי עם מונח ה - Momentum, שלב העדכון הופך לסכום משקלל של הגרדיאנט הנוכחי ושל מעברי העבר המוצטברים. זה מאפשר למודל להמשיך לנوع לאורך היכיונים הרלוונטיים, מה שmobiel להתכנסות מהירה יותר ואופטימיזציה טוביה יותר.

- Epoch: מוגדר כמספר הפעמים (איטרציות) שכל אוסף מידע האימון עבר באלגוריתם הלמידה (ניתן להגביל את איטרציות האימון).

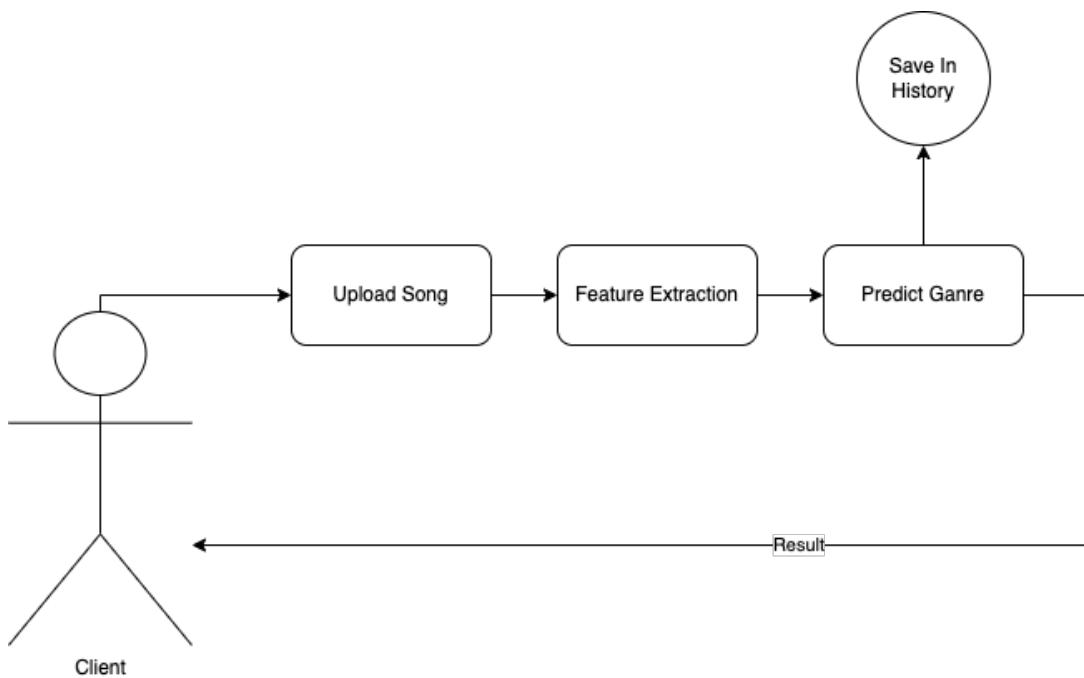
חשוב לומר שככל שהוא בפרמטרים האלו יכול לגרום בזמן אימון קצר יותר או ארוך יותר אף עם זאת חשוב לקחת בחשבון מניעה מהגעה למצב של התאמת יתר (Overfitting) וגם על מנת לקבל מודל יותר חכם, האימון יצרך לקחת יותר זמן.

אתחול הפרמטרים של תהליכי האימון שלו: (נקבעו על ידי ניסוי וטעיה)

- 0.001 :Learning rate •
- 32 :Batch size •
- 0.8 :Momentum •
- 70000 :Max epochs •

## הציגת מקרה (Use Cases) עבור כל הפונקציה העיקרית בפרויקט

דיהוי צ'אנר:



- Upload Song: הולקו מULA קובץ שמע של שיר לאתר באמצעות קומפוננט Vaadin על מנת לקבל את חיזוי הצלינר ממודל המערכת.
- Feature Extraction: השיר שהועלה הולקו עבור תהליך שנקרא "חילוץ פיצ'רים" ובו מופעל על קובץ האודיו חישובים מתמטיים בעזרת הספרייה Librosa של Python על מנת להוציא את המידע הנחוץ למודל המידע.
- Predict Genre: הפיצ'רים שחולצו מהשיר עוברים לתהליך שנקרא נרמול המידע, תהליך זה הופך את וקטור הפיצ'רים לערכים שלמודל יהיה קל ומהר לעבוד איתם (בין 0 ל - 1), מחושב באמצעות נוסחה פשוטה על ערכי המקסימום והמינימום מסט האימון. לאחר הנרמול הוקטור מוכנס למוכנה, המכונה מזרימה קידימה את השיר בראשת הנוירונים והשכבה של הפלט מחזירה לנו את חיזוי הצלינר. לבסוף הצלינר החזוי מוצג בחזרה להולקה.

**מבנה הנתונים בהם היה שימוש**  
**מערך:** מבנה נתוני פשוט, אוסף של איברים מאותו הטיפוס שמסודרים אחד אחרי השני. והגישה אל כל איבר היא באמצעות ציון האינדקס שלו. השתמשתי במערך על מנת לקרוא נתונים מקובץ csv.

**מטריצה:** מוגדרת כמערך של מערכים, מעין "טבלה" של אותו טיפוס נתונים. הגישה אל כל איבר היא באמצעות ציון אינדקס השורה שלו וציון אינדקס העמודה שלו. השתמשתי במטריצה על מנת להציג את הנתונים שבקובץ ה – dataset לאימון.

**MLDataSet:** מבנה נתונים של Encog Framework, הוא משמש לטיענת dataset והפעלת פעולות עלייו, והוא מתממשק לרשת הנירונית. השתמשתי ב - MLDataSet על מנת לטען לתוכו את הנתונים zusätzlich המידעד.

**רשימה:** רשימה היא מעין מערך אך ללא הגבלת מקום, קלומר ניתן לאחסן בה מספר לא ידוע מראש של פריטים. הגישה אל האיברים יכולה להתבצע באמצעות אינדקסים או לפי הסדר שהוכנו. השתמשתי ברשימה בהרבה מקומות, ובעיקר בשימוש שוטף מול ה – DB.

### חישוב יעילות האלגוריתם

אלגוריתם החיזוי הוא אלגוריתם פשוט יחסית: מחלצים פיצ'רים לשיר המתקבל כקלט, עוברים פעם אחת קדימה ברשות וublisherים את החיזוי, וכך הוא לא לוקח הרבה ממאנץ. סיבוכיות מעבר קדימה אחד ברשות היא:

$$T(n) = O(N)$$

כאשר  $N$  הוא גודל הקובץ שממנו מחלצים את הפיצ'רים.

אך מפני שתהלייך חילוץ הפיצ'רים משתמש באלגוריתמים מסובכים לחישוב הפיצ'רים, אך לא נוכל לדעת בוודאות מה הסיבוכיות (הספריה לא חושפת את הקוד שלו) אך נוכל להניח כי הסיבוכיות לא עוברת במקרה הגורע את הערך זהה:

$$T(n) = O(N^2)$$

ולכן נסכם שיעילות פועלות החיזוי היא:

$$O(N^2)$$

בנוסף, תהליך הלימוד הוא תהליך שלוקח הרבה מאוד זמן, מפני שעוברים המון בתוך הרשות גם קדימה וגם אחרת וمعدכנים את המשקولات כשצריך. זמן תהליך הלימוד מושפע מכמה פרמטרים:

- גודל הרשות.
- אלגוריתם האופטימיזציה.
- הפרמטריםuaiים מכוונים את אופי הלימוד (מוסבר בהרחבה בתיאור אלגוריתמים עיקריים).
- גודל אוסף הנתונים.

הчисוב העיקרי באימון הוא מספר איטרציות האופטימיזציה שהיא בעצם מספר האיטרציות של ניסיון למצוור פונקציית השגיאה עד אשר היא קטנה מה – learning rate, נגדיר את מספר האיטרציות ב –  $k$ . בנוסף נגדיר באות  $N$  את מספר הנתונים בסט האימון. מפני שבכל ניסיון למצוור פונקציית השגיאה אנו עוברים על  $N$  נתונים / batch size. לכן הסיבוכיות הכוללת היא:

$$T(n) = O(k * N)$$

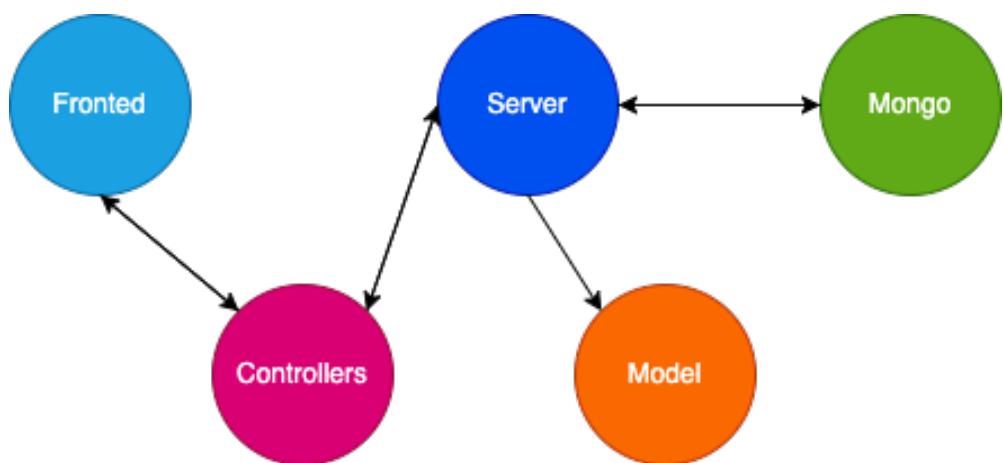
כאשר  $N$  הוא גודל הנתונים של האימון, ו –  $k$  הוא מספר האיטרציות למצוור פונקציית השגיאה.

## הקשרים בין היחידות השונות

- צד השירותamazon לבקשת HTTP מצד הלוקו וمتקשר עם שרת ה – DB במקרה הצורך, ומנהל את תהליך החיזוי וכל מה הקשור בו (אימון המודול וכו').
- צד הלוקו מתקשר עם השירות על ידי שליחת בקשות HTTP, ומתקבל תשובה מהשרת.

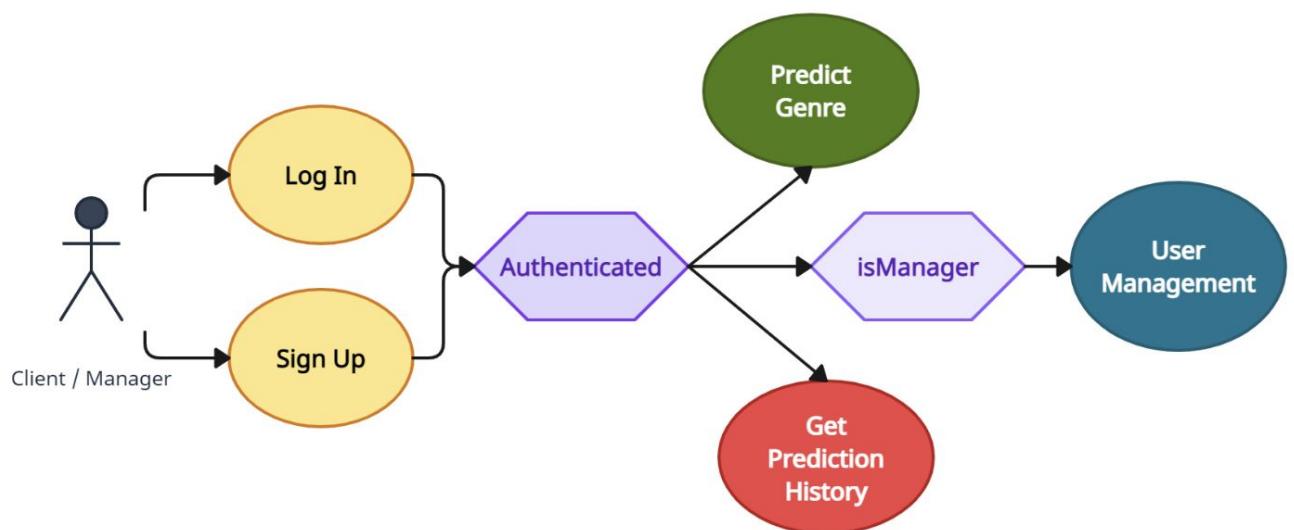
## עיצמודולים

להלן תרשימים בסיסי של עיצמודולים במערכת:



## Use Case Diagram

להלן תרשימים בסיסי של מקרים המערכת:



### רישימת Use Cases

תהליכי זיהוי המשתמש:

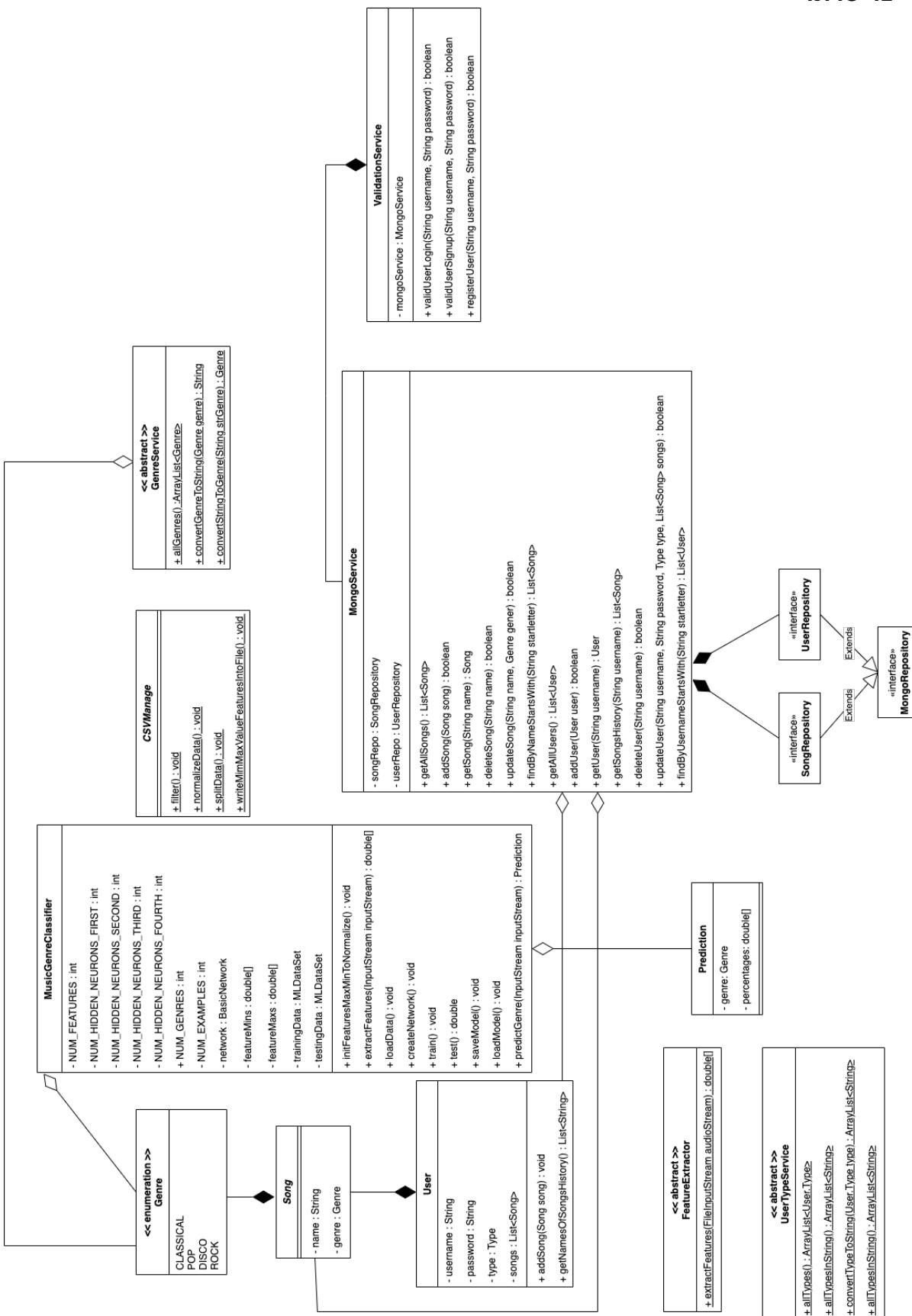
- Log In: התחברות למערכת, הלקוח מזין שם משתמש וסיסמה במידה ורשום למערכת.
- Sign Up: במידה והמשתמש עדיין לא רשום במערכת, נרשם כאן על ידי סיפוק שם משתמש חדש וסיסמה חדשה על מנת להתחבר לשירותי המערכת.

סיפוק שירותים המערכת:

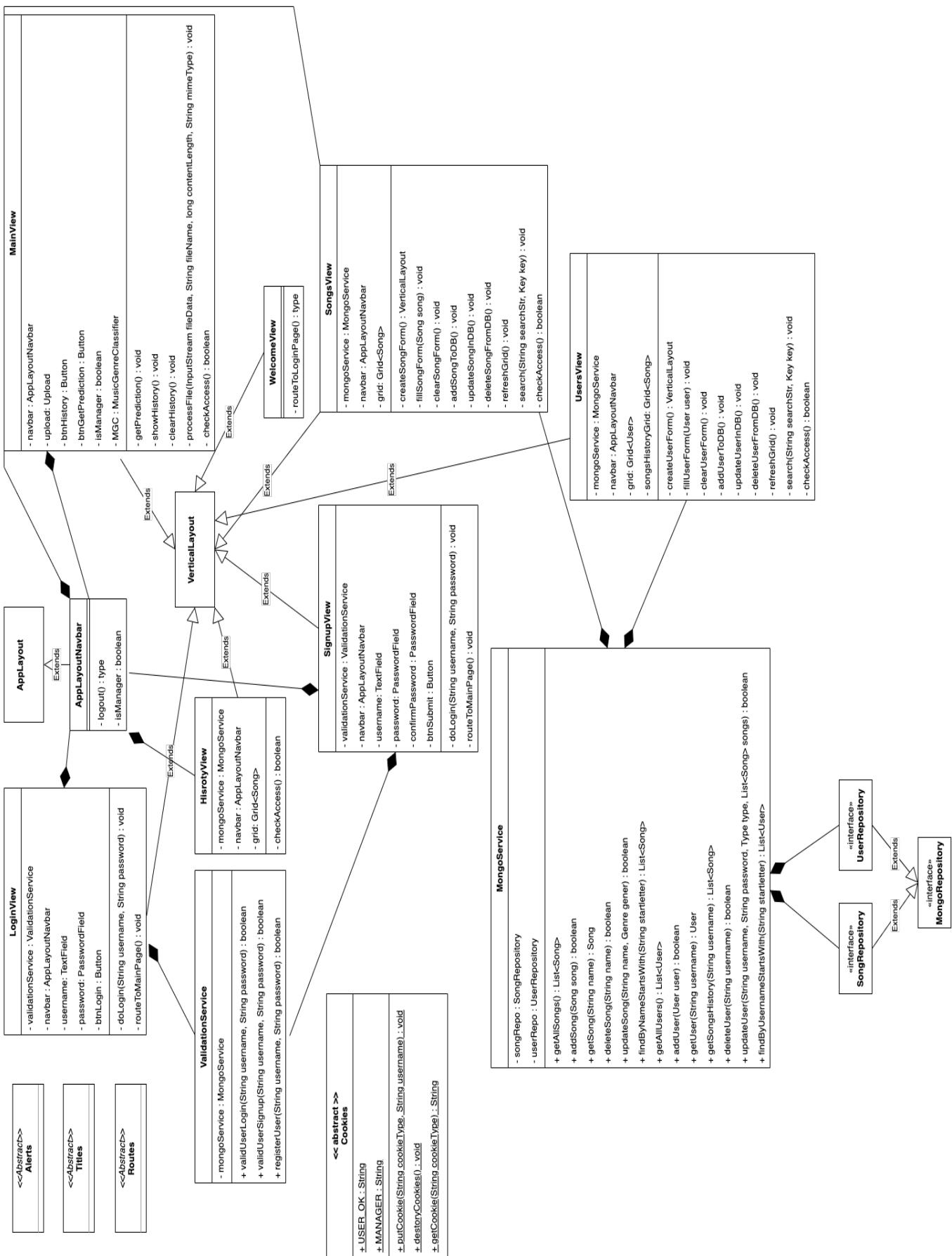
- Predict Genre: למשתמש אפשרות להעלות קובץ שמע למערכת על מנת לקבל חיזוי מהבינה המלאכותית לאיזה ז'אנר משתיך השיר.
- Get Prediction History: למשתמש אפשרות לצפות בהיסטורית חיזוי השירים שלו מה עבר.

במידה והמשתמש המחבר הוא מסוג מנהל (Manager) יסופק גם השירות:

- User Management: יכולת לנוהל את המידע של המשתמשים במערכת.



צד לקוח:



## תיאור המחלקות המוצעות

## צד שרת:

- **Genre**: תפקido להגדיר 4 ז'אנרים לסיווג: קלאסי, פופ, דיסקו ורוק.
- **GenreService**: מחלקת שירות אבסטרקטית (@Service) שמספקת שירותים לעובדה עם הז'אנרים.
- **Song**: מחלקת המייצגת שיר (לעובדה נוחה יותר עם ה – DB). מכילה: שם השיר מסוג מהרוזת ודינאר השיר מסוג Genre.
- **UserTypeService**: מחלקת שירות אבסטרקטית (@Service) שמספקת שירותים לעובדה עם סוגי המשתמשים (גיל ומנהל).
- **User**: מחלקת המייצגת משתמש (לעובדה נוחה יותר עם ה – DB). מכילה: שם המשתמש מסוג מהרוזת, סיסמה מסוג מהרוזת, סוג משתמש מסוג enum Type (NORMAL\_USER ומשתמש פנימי להגדרת סוג משתמשים: משתמש רגיל (NORMAL\_USER) וממשתמש מנהל (MANAGER\_USER)) ורשימה של שירים לייצג את היסטוריית השירים.
- **ValidationService**: מחלקת שירות (@Service) שמספקת שירותים לכל מה שנוגע לבדיקה משתמשים מול ה – DB.
- **SongRepository**: ממשק שיורש מממשק שmagiu מספק שרת ה – DB (MongoDB) לפועלות על האוסף "שירים" (songs) שנמצא ב – DB, מוגדר על מנת להשתמש בפעולות המובאות לנו ועוד פעולות שאנו נרצה להוסיף.
- **UserRepository**: ממשק שיורש מממשק שmagiu מספק שרת ה – DB (MongoDB) לפועלות על האוסף "משתמשים" (users) שנמצא ב – DB, מוגדר על מנת להשתמש בפעולות המובאות לנו ועוד פעולות שאנו נרצה להוסיף.
- **MongoService**: מחלקת שירות (@Service) שמספקת שירותים לכל מה שנוגע לפעולות מול שרת ה – DB.
- **CSVManage**: מחלקת אבסטרקטית שנועדת לספק פעולה על קובץ אוסף המידע (שמור בפורמט csv), תומכת בфиוקי (filter) המידע מידע שלא נכון, פיצול המידע לקובץ אימון ובדיקה וכטיבת ערכי מקסימום ומינימום של הפיצרים לתוך קובץ לצורך הנרמול.
- **MusicGenreClassifier**: מחלקת שתפקידה לעשות את כל התהליכי שקשורים למודל הלמידה. מכילה את רשות הנוירונים, אוסף המידע לאימון ובדיקה וערכי מקסימום ומינימום לפיצרים. תומכת בטעינת המידע, יצירה רשות הנוירונים, אימון המכונה, בדיקת המכונה, טעינת המודל והחיזוי.

- מחלקת FeatureExtractor מבצעת את הפעולה על ידי הרצת סקריפט של Python שמשתמש בספריה Librosa על מנת להשלים את הפעולה.
- מחלקת Prediction מגדירה את החיזוי, שומרת את הדיאנර החזוי ואת ערכיו של כל הדיאנרים באחוזים.

**צד לקוח:**

- AppLayoutNavbar: מחלקת המגדירה את הפאנל העליון לשאר הדפים, כולל: כותרת, כפתורי ניוט וקיטור התנטקות.
  - WelcomeView: מחלקת המגדירה את דף הבית, כולל: כותרת, הסבר על האתר ושירותיו, כפתור התחברות ולינק להרשמה.
  - MainView: מחלקת שמנגדירה את הדף הראשי, הדף אחראי על מתן העלאת קובץ שמע ללקוח על מנת לקבל חיזוי לדיאניר השיר. כולל: פאנל עליון, כותרת, כפתור העלאת שיר, כפתור צפיה בהיסטורית החיזויים וכלי למתן העלאת קובץ שמע.
  - HistoryView: מחלקת המגדירה את דף ההיסטוריה החיזויים, בדף מוצג ההיסטוריה החיזויים של המשתמש שמחובר למערכת.
  - LoginView: מחלקת המייצגת את דף ההתחברות לאתר. כוללת: כותרת, שדה לשם משתמש, שדה לסיסמה, כפתור כניסה ולינק להרשמה לאתר.
  - SignupView: מחלקת המייצגת את דף ההרשה לאתר. כוללת: כותרת, שדה לשם משתמש, שדה לסיסמה, שדה לאישור הסיסמה, כפתור הרשמה.
  - UsersView: מחלקת המייצגת את דף ניהול המשתמשים במערכת (מונגש למנחים בלבד). כוללת: פאנל עליון, כותרת, גרייד להציג המשתמשים, גרייד להציג ההיסטוריה חיפושי השירים של המשתמשים, שדה חיפוש, כפתור הוספה, כפתור עדכון, כפתור מחיקת ההיסטוריה החיפושים, כפתור מחיקה של משתמש, כפתור ניקוי הטופס, שדה שם משתמש, שדה סיסמה, comboBox לבחירת סוג המשתמש.
  - Cookies: מחלקת אבסטרקטית ששומרת את סוגי העוגיות שהמערכת משתמש בהם, וכן כל פונקציות ניהול עוגיות הלקוח.
  - Titles: מחלקת אבסטרקטית שמספקת את כל הכותרות לבניית התצוגה.
  - Alerts: מחלקת אבסטרקטית שמספקת את כל ההודעות לצד הלקוח.
- הערה: כל המחלקות שמייצגות דפים משמשים גם Controllers.

## תיאור התוכנה

### סביבה העבודה

עבדתי על מערכת הפעלה MacOS בסביבת NetBeans של חברת Oracle (גרסה 12.6), זהה סביבת פיתוח נוחה מאוד למגוון פלטפורמות, כגון: Desktop, Web ועוד. הסביבה תומכת במגוון רחב של שפות ומאפשרת כל עיצוב (GUI) בצורה נוחה, בנוסף מאפשרת מעקב על היזכרון ועל מצב התהילכנים באמצעות תרשימים נוחים.

בנוסף לכך, עבדתי בסביבת Jupyter, Project, זהה סביבת פיתוח שמספקת בניית מחרבת בשפת Python, העבודה עם הסביבה זו היא מאוד נוחה, ועזרה לי מאוד בתהילך AISOF הנתונים ויצוא הפיצרים.

כמו כן, עבדתי בסביבת MongoDB Compass לצורך התחברות למסד הנתונים בצורה נוחה, צפיה במידע ופעולות נוספות.

בכל postman לבניית בקשות HTTP מורכבות והבנה יותר עמוקה של הנושא.

### שפות תכנות

השפה העיקרית שהשתמשתי בה הייתה שפת Java (גרסה 17), בה כתבתי את צד השרת (בעזרת SpringBoot שמעלה שרת Tomcat בעצמו) ואת האלגוריתמים הראשיים (בעזרת צירוף Encog Framework למידת מכונה בשפת Java). ניהול הפרויקט התרחש בעזרת Maven.

בנוסף לכך גם את צד הלקוח כתבתי בשפת Java בעזרת Vaadin Framework שהוא בעצם עוזר לכתיבה דפי HTML ללא שימוש בשפה זו, אלא הוא כתוב עם אותן הקומponentות אך בשפת Java.

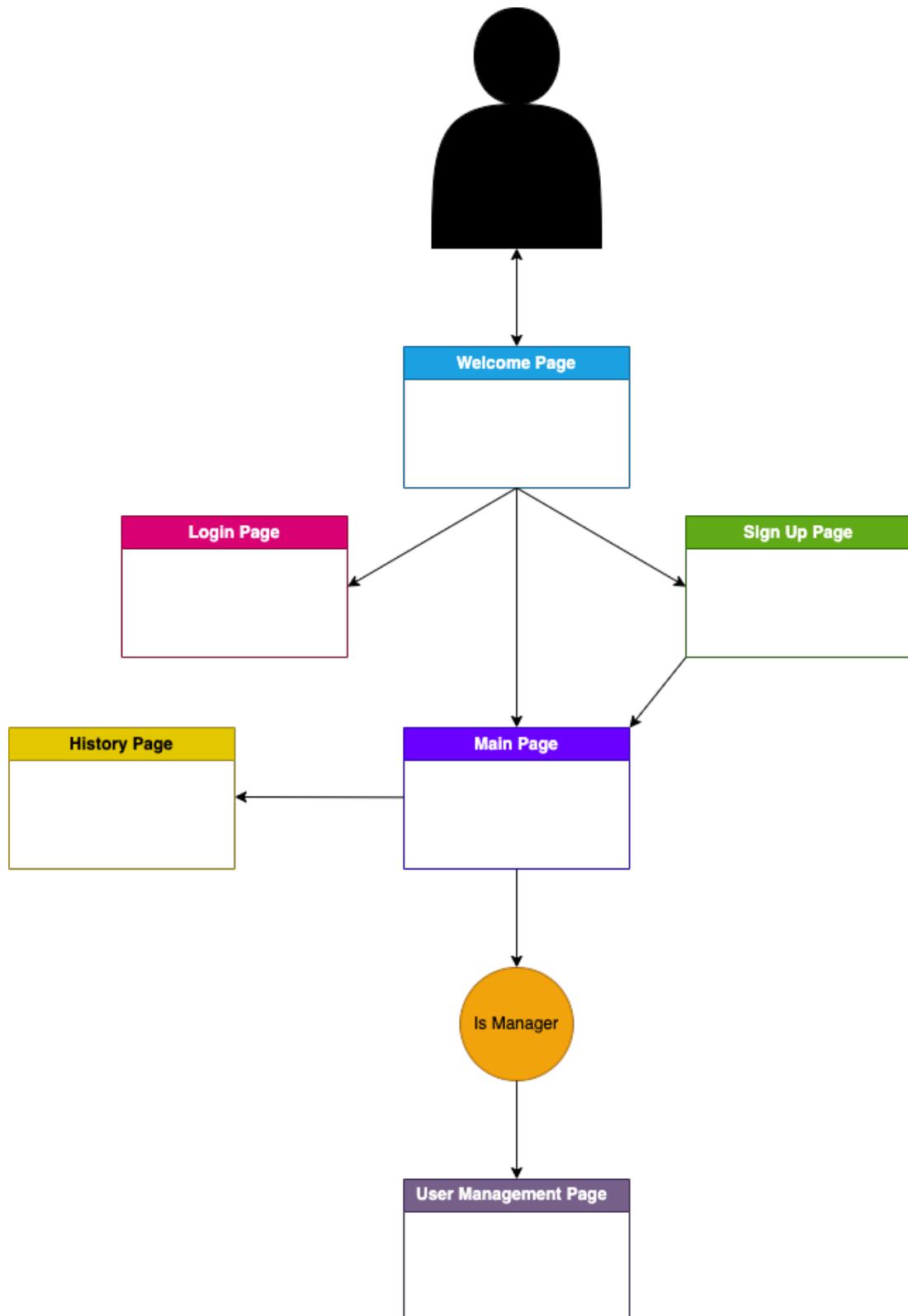
על מנת ליצא פיצרים משיר (Feature Extraction) כדי להפעיל את פונקציית החיזוי של המודל, הייתה צריכה להתעסק עם ספרייה של Python שנקראת Librosa ועם עוד ספריות של Python על מנת לראות ויזואלית את הגרפים של הפיצרים השונים של השיר ולהבין עד הסוף.

בנוסף לכך, לאחר ניסוי וטעיה הבנתי שספר הדוגמאות ב – dataset שלו קטן, אז החלמתי לתקן את השירים המספקים מה – dataset ולחבל אותם לשירים נוספים יותר, ובכך הכפלתי בעשר את מספר הדוגמאות ב – dataset.

## ממשק משתמש

(Screen Flow Diagram)

להלן תרשימים היררכית המרכיבים במערכת:

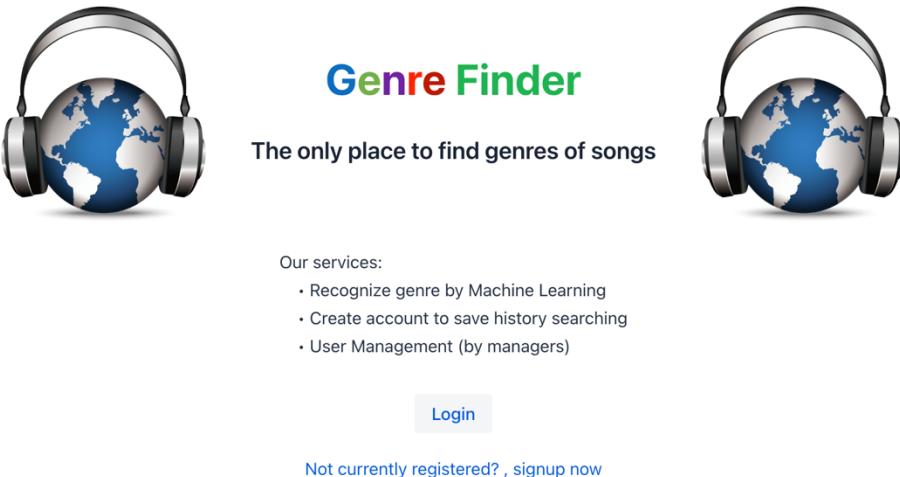


**תיאור מסך הפתיחה**

מסך הפתיחה הוא המסך הראשון שמשתמש רואה כשהוא מתחבר למערכת, זאת אומרת שהוא מסך ממשועטי ביותר. המסך מספק הסבר על שירותים האתר ומכל כפטור ניוט להתחברות למערכת ולינק להרשמה למשתמש מידת והמשתמש עדין לא רשום.

**תיאור מסכים**

**Welcome Page:** מסך הפתיחה, המסך הראשון שעולה כשמתחברים על השרת, תפקידו להסביר על השירותים ועל אפשרות כניסה או הרשמה לאתר.



במסך זה יש לנו את הפיצרים הבאים:

- כותרת ראשית.
- כותרת משנה.
- הסבר על שירותים האתר.
- כפטור ניוט למסך ההתחברות למערכת.
- לינק לניווט לדף ההרשמה מידת והמשתמש עדין לא רשום.

מסמך כניסה משתמש רשומים, תפקידו הוא לבצע את זהה המשתמש על מנת שיוכל להשתמש בשירותי המערכת.

# Login



Username:

Password:

[Login](#)

[Not currently registered? , signup now](#)

במסמך זה יש לנו את הפיצ'רים הבאים:

- כותרת ראשית.
- שדה לכתיבה שם משתמש.
- שדה לכתיבה סיסמה (עם אופציה לצפות בה אם לוחצים על כפתור העין בצדיה).
- כפתור התחברות – מחזיר חיווי האם שם משתמש וסיסמה נכונים, במידה וכן מכניס את המשתמש לדף הראשי. במידה ולא מעלה הודעה ללקוח שהפרטים אינם נכונים.
- Link לדף ההרשמה למערת במידה והמשתמש עדין לא רשום.

Sign Up Page: מסך הרשמה משתמשים חדשים במידה ולא רשומים, תפקידו הוא להכניס את המשתמש למסד הנתונים של המערכת לצורכי שימוש שוטף במערכת.

# Sign Up



username:

password:

confirm password:

Submit

[Already have an account? , login now](#)

במסך זה יש לנו את הפיצרים הבאים:

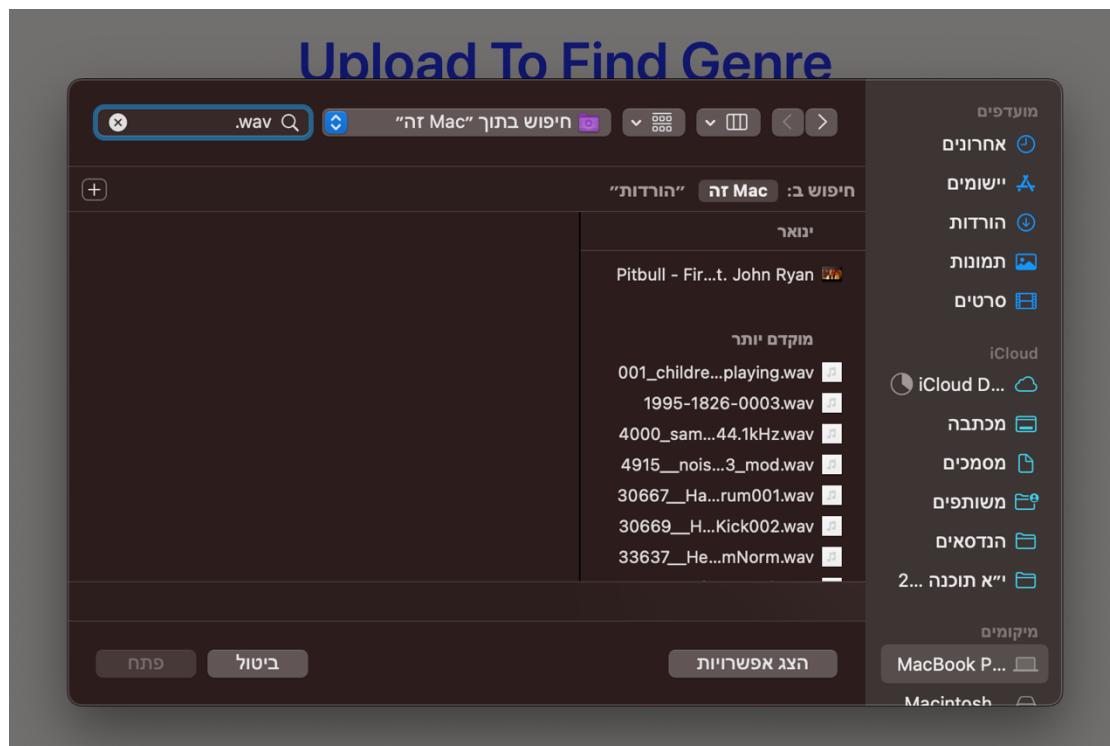
- כוורת ראשית.
- שדה לכתיבה שם משתמש.
- שדה לכתיבה סיסמה (עם אופציה לצפות בה אם לוחצים על כפתור העין בצדיה).
- שדה לאימות הסיסמה (עם אופציה לצפות בה אם לוחצים על כפתור העין בצדיה).
- כפתור הגשת הנתונים למערכת – מחזיר חיווי האם הסיסמה ואימות הסיסמה תואמים, במידה וכן רושם את הלוקו למערכת ו מעביר אותו לדף הראשי. במידה ולא מעלה הודעה שהסיסמות אינן תואמות אחת לשנייה.
- לינק לדף ההתחברות למערכת במידה והמשתמש בן רשום.

Main Page: המסר העיקרי במערכת, תפקידו הוא לספק את שירות חיזוי המודל של המערכת. כלומר, הלקוח יכול להעלות שיר ולקבל את חיזוי הדיאנר מהמודל. בנוסף מספק צפיה בהיסטורית החיזויים של המשתמש.

במסך זה יש לנו את הפיצרים הבאים:

- פאנל עליון לניווט במערכת שמכיל:
  - ◆ שם האתר (במידה ונחץ עליו נוכל להגיע לדף הראשי).
  - ◆ תוכית בשם "Find Genre" זהו כפתור שיוביל אותנו לדף הראשי.
  - ◆ תוכית בשם "Users Management" (אפשר רק למנהלים) זהו כפתור שיוביל אותנו לדף ניהול המשתמשים במערכת.
  - ◆ כפתור "Logout" להתנתקות המשתמש מהמערכת.
- כותרת ראשית.
- כפתור לצפיה בהיסטורית החיזויים – יעביר אותנו לדף היסטורית החיזויים של המשתמש הנוכחי.
- כפתור למחיקת ההיסטוריה – מוחק את היסטורית החיזויים של המשתמש הנוכחי, מחזיר חיוי האם המחיקה הצליחה או לא.
- הסבר על תמיינט קבצי השמע להעלאה.
- כלי להעלאת קבצים.
- כפתור לקבלת חיזוי לדיאן השיר הנוכחי.

במידה ונלכד על הכפתור "...Upload File" נקבל את חלון חיווי מערכת הפעלה להעלאת קבצים, לדוגמה:



לאחר שנבחר שיר ונעלה אותו למערכת נראה את שם הקובץ שהעלו ונקבל חיווי אם עלה בהצלחה:

כעת נוכל ללחוץ על הכפתור "Get Genre Prediction" ונקבל את חיזוי הז'אנר באחוזים, לדוגמה:

Genre Finder

Find Genre

Users Management

Logout

## Upload To Find Genre

Show History Clear History

Maximum file size: 1 GB | Accepted file formats: WAV (.wav)

Upload file... Drop file here

✓ Fireball.wav

Get Genre Prediction

Classical: 0%

Pop: 11%

Disco: 89%

Rock: 0%

בנוסף נוכל ללחוץ על הכפתור "Show History" על מנת לצפות בהיסטוריית החיזויים, לאחר הלחיצה נוכל לראות את הדף הבא:

Genre Finder

Find Genre

Users Management

Logout

## History Grid

Name	Genre
classical.00000.wav	CLASSICAL
Fireball.wav	DISCO
hold the line.wav	DISCO
pop example.wav	POP
Fireball.wav	DISCO

במידה ונלחץ על אחד השירים בטבלה הנגן שבצד ימין למעלה יוכל לפעול עם אותו השיר הנבחר.

כמו כן נוכל ללחוץ בדף הראשי על כפתור "Clear History" על מנת למחוק את היסטוריית החיזויים.

User Management Page: מסך ניהול המשתמשים במערכת, מונגשת למשתמשים מנהליים בלבד, בו ניתן לצפות מיידע המשתמשים ששמור בסיס הנתונים ולבצע פעולות על המידע.

Username	Password	Type User
Noam	1234	Manager
YinonTsadik	Yinon321	Normal
Nati	3333	Normal
Nety	5555	Manager

במסך זה יש לנו את הפיצ'רים הבאים:

- פאנל עליון לניווט במערכת שמכיל:
  - ♦ שם האתר (במידה ונחוץ עליו נוכל להגיע לדף הראשי).
  - ♦ תווית בשם "Find Genre" זהו כפתור שיוביל אותנו לדף הראשי.
  - ♦ תווית בשם "Users Management" (אפשר רק למנהלים) זהו כפתור שיוביל אותנו לדף ניהול המשתמשים במערכת.
  - ♦ כפתור "Logout" להתנתקות המשתמש מהמערכת.
  - ♦ כותרת ראשית.
- שדה לחיפוש משתמש במערכת (לחיצה על מקש Enter במקלדת לחיפוש).
- כפתור להוספה משתמש למערכת (מהמידע המסופק על ידי המשתמש בטופס שבצדיו הימין של המסך). מחייב חישוי האם הפעולה הצלילה או לא.
- כפתור לעדכון משתמש במערכת (יהיה זמין רק לאחר בחירת משתמש שמוצג בטבלה). מחייב חישוי האם הפעולה הצלילה או לא.
- כפתור למחיקת משתמש מהמערכת (יהיה זמין רק לאחר בחירת משתמש שמוצג בטבלה). מחייב חישוי האם הפעולה הצלילה או לא.
- טבלת (grid) נתוני המשתמשים במערכת.

- טבלת (grid) היסטורית השירותים שהלכה לעלייה לחיזוי ד'אנר (המידע יתעדכן לאחר בחירת משתמש שמצוג בטבלה).

במידה ונלץ על אחד המשתמשים יוכל לראות את הדבר הבא:

The screenshot displays two main sections of the application:

**User Management Section:**

- Header: Genre Finder, Find Genre, Users Management, Logout.
- Buttons: Filter by name, + Add User, Update User, Clear History, Delete User.
- Table: Shows a list of users with columns: Username, Password, Type User.

Username	Password	Type User
Noam	1234	Manager
YinonTsadik	Yinon321	Normal
Nati	3333	Normal
Nety	5555	Manager

**user form:**

- Username: Noam
- Password: 1234
- User Type: Manager
- Clear Form button.

**File Genre Mapping Section:**

Name	Genre
classical.00000.wav	CLASSICAL
Fireball.wav	DISCO
hold the line.wav	DISCO
pop example.wav	POP

## קוד התוכנית

### פונקציות עיקריות

בפרויקט ישנו כמה פונקציות עיקריות שעזרות למודל לפתור את בעיית הסיווג:

- פונקציית נרמול הנתונים בסט האימון
- פונקציית בניית ארכיטקטורת הרשת
- פונקציית אימון המודל
- פונקציית חילוץ הפיצ'רים
- פונקציית חיזוי המודל
- פונקציית הערכת ביצועי המודל

### פונקציית נרמול הנתונים בסט האימון:

```
/*
 * פונקציה לנרמול הנתונים בקובץ נתונים
 * הפונקציה שומרה על ליניאריזציה של מערך נתונים
 * מערך מינימום ומספר מינימום ומספר מקסימום
 * לאחר מכן טבורה שוב ליניאריזציה וטבורה ליניאריזציה מחדש
 * על ידי חישוב פשטוט עם תעריך המקסימלי והמינימלי שנשלל כלכל
 * בכל פיצ'ר
 *
 * פעולה
 * O(N)
 * N: גודל קובץ הנתונים
 */
public static void normalizeData()
{
    double[] featureMins = new double[NUM_FEATURES];
    double[] featureMaxs = new double[NUM_FEATURES];

    BufferedReader br;
    BufferedWriter bw;
    StringBuilder sb;

    String line = "";

    try
    {
        br = new BufferedReader(new FileReader(TRAIN_FILE));

        // Read header line
        br.readLine();

        // Initialize feature mins/maxs
        for (int i = 0; i < NUM_FEATURES; i++)
        {
            featureMins[i] = Double.MAX_VALUE;
            featureMaxs[i] = Double.MIN_VALUE;
        }

        // Process each data row
        while ((line = br.readLine()) != null)
        {
            String[] values = line.split(CSV_SPLIT_SIGN);

            // Update min/max values for each feature
            for (int i = 0; i < NUM_FEATURES; i++)
            {
                double value = Double.parseDouble(values[i]);
                if (value < featureMins[i])
                    featureMins[i] = value;

                if (value > featureMaxs[i])
                    featureMaxs[i] = value;
            }
        }
        br.close();
    }
    catch (IOException e)
    {
        e.printStackTrace();
        return;
    }
}
```

```

// Normalize data and write to output file
try
{
    br = new BufferedReader(new FileReader(TRAIN_FILE));
    bw = new BufferedWriter(new FileWriter(CSV_DATA_OUTPUT));
    sb = new StringBuilder();

    // Write header line
    bw.write(br.readLine());
    bw.newLine();

    // Process each data row
    while ((line = br.readLine()) != null)
    {
        String[] values = line.split(CSV_SPLIT_SIGN);

        // Normalize each feature value
        for (int i = 0; i < NUM_FEATURES; i++)
        {
            double value = Double.parseDouble(values[i]);
            double scaledValue;
            if (featureMins[i] == featureMaxs[i])
            {
                System.out.println("Skipping normalization for feature index: " + i);
                scaledValue = value; // Keep the original value if min == max
            }
            else
                scaledValue = (value - featureMins[i]) / (featureMaxs[i] - featureMins[i]);

            if (Double.isNaN(scaledValue))
                System.out.println("NaN value encountered for feature index: " + i);

            values[i] = String.format("%.6f", scaledValue); // Round to 6 decimal places
        }

        // Concatenate processed data row to output string
        sb.append(String.join(",", values));
        sb.append("\n");
    }

    bw.write(sb.toString());
    bw.close();
}
catch (IOException e)
{
    e.printStackTrace();
    return;
}

// Delete original file
File oldFile = new File(TRAIN_FILE);
oldFile.delete();

// Rename output file
File newFile = new File(CSV_DATA_OUTPUT);
newFile.renameTo(oldFile);
}

```

**פונקציית בניית ארכיטקטורת הרשת:**

```
/**  
 * פונקציה המדרירה את ארכיטקטורת הרשת  
 */  
public void createNetwork()  
{  
    // Create network  
    ActivationFunction inputLayerActivation = new ActivationSigmoid();  
    ActivationFunction hiddenLayerActivation = new ActivationSigmoid();  
    ActivationFunction outputLayerActivation = new ActivationSoftMax();  
    network = new BasicNetwork();  
  
    // input layer  
    network.addLayer(new BasicLayer(inputLayerActivation, true, NUM_FEATURES));  
  
    // hidden layer  
    network.addLayer(new BasicLayer(hiddenLayerActivation, true, NUM_HIDDEN_NEURONS_FIRST));  
    network.addLayer(new BasicLayer(hiddenLayerActivation, true, NUM_HIDDEN_NEURONS_SECOND));  
    network.addLayer(new BasicLayer(hiddenLayerActivation, true, NUM_HIDDEN_NEURONS_THIRD));  
    network.addLayer(new BasicLayer(hiddenLayerActivation, true, NUM_HIDDEN_NEURONS_FOURTH));  
  
    // output layer  
    network.addLayer(new BasicLayer(outputLayerActivation, false, NUM_GENRES));  
    network.getStructure().finalizeStructure(); // מסים את הבנייה של השכבות  
    network.reset();  
}
```

**פונקציית אימון המודל:**

```
/*
 * פונקציה אימון המודל
 * לוקחת את המידע מסט האימון ומחילה לאמן את המודל על ידי אלגוריתם רדיאנט דנסט
 * מגדירה בונוסף את הפרמטרים של האימון
 *
 * בסיסם הפעולה שומרת את נתוני המודל בקובץ
 *
 * יילול הפעולה
 * O(k * N)
 * k: מספר האיטרציות של אלגוריתם הגרדיאנט
 * N: מסנן את גודל סט נתונים האימון
 *
 * הישולות תלויה בשוד פרמטרים אבל אלו העיקריים
 */
public void train()
{
    // Train network
    System.out.println("I am Starting training now, Data = \n" + trainingData.toString());

    final StochasticGradientDescent trainer = new StochasticGradientDescent(network, trainingData);

    final int batchSize = 32;
    final double learningRate = 0.001;
    final double momentum = 0.8;

    trainer.setBatchSize(batchSize);
    trainer.setLearningRate(learningRate);
    trainer.setMomentum(momentum);

    ErrorCalculation error = new ErrorCalculation();

    int epoch = 1;
    final int MAX_EPOCHS = 70000;
    do
    {
        trainer.iteration();
        error.reset();
        for (MLDataPair pair : trainingData)
        {
            MLDATA output = network.compute(pair.getInput());
            error.updateError(output.getData(), pair.getIdeal().getData(), 1.0);
        }
        System.out.println("Epoch #" + epoch + " -> Training error: " + error.calculate());
        epoch++;

        if (epoch >= MAX_EPOCHS)
            break;
    } while (error.calculate() > learningRate);
    trainer.finishTraining();
}

// Save trained network
saveModel();
}
```

## פונקציית חילוץ הפיצרים:

הקוד ב – Java

```
/*
 * פועלות חילוץ הפיצרים
 * מפעילה סקריפט בפייתון שבודק קובץ את הפיצרים
 * @param audioStream טרום קובץ השיר להילוץ הפיצרים
 * @return מערך הפיצרים המחלצים מהשיר
 *
 * עיולות הפעולה
 * O(N^2)
 * N: מסמן את גודל הקובץ השיר
 *
 * הספירה שיעשה את החילוץ משתמש באלגוריתם
 * ואנו מניחים שבקרה הגורע היא לא עוברת את ריגולות הוו
 */
public synchronized static double[] extractFeatures(FileInputStream audioStream)
{
    try
    {
        // Create temporary input and output files
        File inputFile = File.createTempFile("input", ".bin");
        File outputFile = new
File("src/main/java/com/noama/GenreIdentificationServer/outputFromPython.txt");

        FileOutputStream inputFileStream = new FileOutputStream(inputFile);
        // Write audio data to the input file
        byte[] buffer = new byte[4096];
        int bytesRead;
        while ((bytesRead = audioStream.read(buffer)) != -1)
        {
            inputFileStream.write(buffer, 0, bytesRead);
        }

        // Close the input file stream
        inputFileStream.close();

        // Build the command
        ProcessBuilder processBuilder = new ProcessBuilder(PYTHON_INTERPRETER_PATH, PATH_PY,
inputFile.getAbsolutePath(), outputFile.getAbsolutePath());

        // Run the command
        Process process = processBuilder.start();

        // Wait for the process to finish
        int exitCode = process.waitFor();
        System.out.println("Python script execution completed with exit code: " + exitCode);

        // Read the output file
        BufferedReader reader = new BufferedReader(new InputStreamReader(new
FileInputStream(outputFile)));
        String line;
        while ((line = reader.readLine()) != null)
        {
            System.out.println(line);
        }

        // Delete the temporary file
        inputFile.delete();

        return readOutputFile();
    }
    catch (Exception ex)
    {
        System.out.println("Error during feature extraction " + ex);
    }

    return null;
}
```

## הקליד ב Python –

```

import sys
import traceback
import librosa

def process_audio(input_file, output_file):
    try:
        # Read the audio data from the input file
        y, sr = librosa.load(input_file)

        # Perform audio processing and feature extraction
        chroma_stft = librosa.feature.chroma_stft(y=y, sr=sr)
        spectral_centroids = librosa.feature.spectral_centroid(y=y, sr=sr)[0]
        spectral_bandwidth = librosa.feature.spectral_bandwidth(y=y, sr=sr)[0]
        y_harm, y_perc = librosa.effects.hpss(y)
        zero_crossings = librosa.zero_crossings(y=y, pad=False)
        rms = librosa.feature.rms(y=y)
        spectral_rolloff = librosa.feature.spectral_rolloff(y=y, sr=sr)[0]
        tempo, _ = librosa.beat.beat_track(y=y)

        # Generate output data or perform additional processing
        featuresExtraction = [chroma_stft.mean(), chroma_stft.var(), rms.mean(), rms.var(),
        spectral_centroids.mean(), spectral_centroids.var(), spectral_bandwidth.mean(),
        spectral_bandwidth.var(), spectral_rolloff.mean(), spectral_rolloff.var(), zero_crossings.mean(),
        zero_crossings.var(), y_harm.mean(), y_harm.var(), y_perc.mean(), y_perc.var(), tempo]

        # Open the output file and write the output
        with open(output_file, 'w') as f:
            # Write the extracted features to the output file
            for feature in [chroma_stft.mean(), chroma_stft.var(), rms.mean(), rms.var(),
            spectral_centroids.mean(), spectral_centroids.var(), spectral_bandwidth.mean(),
            spectral_bandwidth.var(), spectral_rolloff.mean(), spectral_rolloff.var(), zero_crossings.mean(),
            zero_crossings.var(), y_harm.mean(), y_harm.var(), y_perc.mean(), y_perc.var(),
            tempo]:
                f.write(str(feature) + '\n')

    except Exception as e:
        # Print the traceback of the exception to standard error
        traceback.print_exc(file=sys.stderr)

if __name__ == '__main__':
    # Get the input and output file paths from command-line arguments
    input_file = sys.argv[1]
    output_file = sys.argv[2]

    # Process the audio and generate output
    process_audio(input_file, output_file)

```

## פונקציית חיזוי המודל:

```
/*
 * פונקציית חיזוי הויינר
 *
 * @param inputStream הסתרים של הקובץ לחיזוי
 * @return עצם מסוג חיזוי השומר בתוכו את חיזוי המודל לשיר ואת האוזנים המתאימים להיזוי
 * מהוירה ערך ריק אם הייתה שגיאה במהלך החיזוי
 *
 * عملות הפעולה
 * O(N^2)
 * N: מסמן את גודל הקובץ
 *
 * הפעולה קוראת לפעולות ניהול הפיצרים שאנו
 * מנהים שיעילתה במקרה הגרוע היא
 * O(N^2)
 *
 */
public Prediction predictGenre(InputStream inputStream)
{
    try
    {
        double[] features = extractFeatures(inputStream);

        // Normalize features using same parameters used during training
        double[] normalizedFeatures = normalizeFeatures(features);

        MLData input = new BasicMLData(normalizedFeatures);
        MLData output = network.compute(input);

        // Get predicted genre
        int predictedGenreIndex = getMaxIndex(output.getData());
        Genre predictedGenre = Genre.values()[predictedGenreIndex];

        Prediction prediction = new Prediction(predictedGenre);
        double[] percentages = output.getData();
        for (int i = 0; i < percentages.length; i++)
        {
            double percentage = percentages[i] * 100;
            prediction.setPercentages(percentage, i);
        }

        System.out.println(prediction);
        return prediction;
    } catch (Exception e)
    {
        System.out.println("Error during prediction");
    }
    return null;
}
```

## פונקציית הערכת ביצועי המודל:

```
/*
 * פוליה המאריצה את ביצועי המודל
 * על ידי מעבר על סט הבדיקה וניהוח ביצועי המודל
 * פרמטרים הקטטיים של ביצועי המודל
 *
 * יישולות הפעלה
 * O(N^2)
 * N: מסמן את גודל המדריך בסט הבדיקה
 *
 * הושתת הוו נובעת מיעילות פונקציית החוויה
 */
public void evaluateModel()
{
    int correctPredictions = 0;
    int totalPredictions = testingData.size();

    int[] yTrue = new int[totalPredictions];
    int[] yPred = new int[totalPredictions];

    for (int i = 0; i < totalPredictions; i++)
    {
        MLDataPair pair = testingData.get(i);

        // Get the input features and true label
        MLData input = pair.getInput();
        MLData target = pair.getIdeal();

        Genre predictedGenre = null;
        try
        {
            // Use the model to predict the genre
            predictedGenre = predictGenre(input.getData());
        }
        catch (Exception e)
        {
            System.out.println("Error during evaluate model");
        }

        if (predictedGenre != null)
        {
            // Compare the predicted genre with the true label
            if (predictedGenre == Genre.values()[getMaxIndex(target.getData())])
                correctPredictions++;

            // Store true label and predicted label for later evaluation
            yTrue[i] = getMaxIndex(target.getData());
            yPred[i] = predictedGenre.ordinal();
        }
    }

    // Calculate accuracy
    double accuracy = (double) correctPredictions / totalPredictions;

    // Calculate precision, recall, and F1 score
    int tp = 0; // True positives
    int fp = 0; // False positives
    int fn = 0; // False negatives

    for (int i = 0; i < totalPredictions; i++)
    {
        if (yTrue[i] == 1 && yPred[i] == 1)
        {
            tp++;
        }
        else if (yTrue[i] == 0 && yPred[i] == 1)
        {
            fp++;
        }
        else if (yTrue[i] == 1 && yPred[i] == 0)
        {
            fn++;
        }
    }

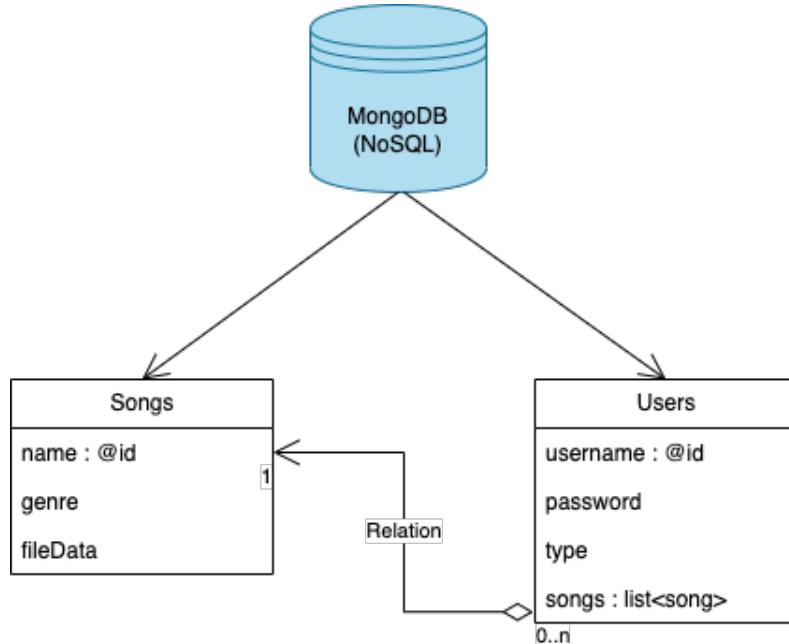
    double precision = tp / (double) (tp + fp);
    double recall = tp / (double) (tp + fn);
    double f1Score = 2 * (precision * recall) / (precision + recall);

    System.out.println("The Summary Of The Model Evaluation:\n-----");
    System.out.println("Accuracy: " + accuracy);
    System.out.println("Precision: " + precision);
    System.out.println("Recall: " + recall);
    System.out.println("F1 Score: " + f1Score);
}
```

## תיאור מסלול הנתונים

מסד נתונים לא רלציוני (NoSQL) שMASPFOK באמצעות MongoDB. בתוך מסד הנתונים נשמר מידע על המשתמשים במערכת על מנת לספק שירות ברמה טובה יותר.

### תרשים ERD



### סקמה כללית

מאוחסנים שני אוסףים (Collections) בתחום מסד הנתונים:

- **שירים (songs):** בהם נשמרים שלושה נתונים:
  1. שם השיר מסווג מחרוזת המשמש כמפתח (לא ניתן להכניס ערך null).
  2. ז'אנר השיר מסווג Genre (ניתן להכניס ערך null).
  3. המידע של הקובץ מסווג מערך של ביטים (ניתן להכניס ערך null).
- **משתמשים (users):** בהם נשמרים ארבעה נתונים:
  1. שם המשתמש מסווג מחרוזת המשמש כמפתח (לא ניתן להכניס ערך null).
  2. סיסמה מסווג מחרוזת (לא ניתן להכניס ערך null).
  3. סוג המשתמש מסווג Type (לא ניתן להכניס ערך null).
  4. רשימת היסטורית החזויים מסווג songs [משמש כמפתחZR לאוסף השירים] (ניתן להכניס ערך null).

## מדריך למשתמש

על מנת להריץ את השירות (רץ על כתובת ה - IP הנוכחי של המחשב ועל פורט 8080):  
1. חלץ את הקובץ של הפרויקט.

2. הכנס את התיקייה לתוכה סביבת הפיתוח **.NetBeans**.

3. הרץ את הקובץ **java Application.java**.

על מנת להריץ את צד הלפקות:

1. פתח דפדפן שתומך בבקשות HTTP.

2. הקלד בשורת הכתובת "localhost:8080".

על מנת להשתמש בשירותי המערכת:

1.התחבר / הירשם לאתר.

2. העלה קובץ שמע של שיר (בפורמט wav).

3. לחץ על כפתור החיזוי.

4. המתן לקבלת החיזוי מהמודול.

5. צפה בהיסטוריה החיזויים.

במידה והמשתמש הוא מנהל:

6. צפה בניהול המשתמשים של המערכת.

## בדיקות והערכתה

על מנת לבדוק ולהעריך את איכות המודל, הייתה צריך להכין אימון טוב למודל. המודל מאומן מראש על ידי מערך נתונים לאימון המודל, אך הערכת המודל תלויה רבות באימון המודל. על מנת ליצור אימון טוב (זהו דבר שנערך על ידי ניסוי וטעיה להגדרת הפרמטרים) הייתה צריכה לאמן את המודל שלו כמות רובה של פעמים, ולאחר כל אימון לבדוק את המודל על ידי סט הבדיקה, וכך לתרת לו את סט הבדיקה (שאנו יודעים מה הפלט צריך להיות מהמכונה) ולבדוק מה המכונה שהוציא ומה היא באמת שווה בפועל.

על מנת להעריך את ביצועי המודל השתמשתי בסט הבדיקה שהכנתי מראש (פיצלתי את סט המידע הכלול לשני סטים: אימון ובדיקה (training set & testing set), כאשר האימון משמש לתהיליך אימון המודל וסת הבדיקה משמש להערכת המודל).

על מנת לחשב ולהעריך באמת כמה המודל טוב צריך לתרת לו לנסות לחזות את כל המידע שנמצא בסט הבדיקה ולבדוק את מדי סטטיסטיות ההצלחה שלו. על מנת לחשב את המדים נסביר באילו מדים השתמש:

- נוכנות (Accuracy)
- דיוק (Precision)
- רגישות (Recall) ("היזכרות")
- ציון F1 (F1 Score)

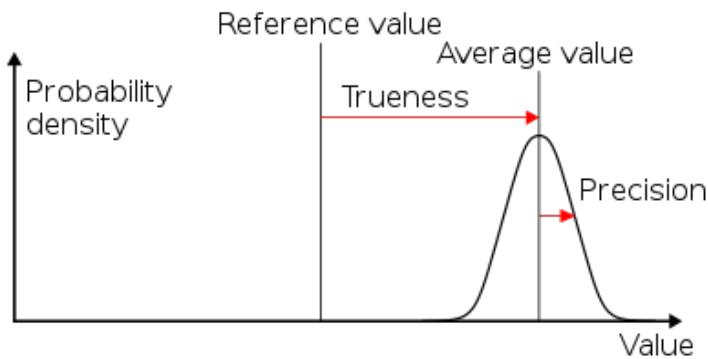
nocnoot (Accuracy): היחס בין התוצאות הנכונות למספר הכלול של התוצאות. מחושב כך:

$$\text{Accuracy} = \frac{\text{correct classifications}}{\text{all classifications}}$$

דיוק (Precision): מדד לכמה מהמקרים החוביים החזויים הם באמת חיוביים, כלומר כיצד מתאר תוצאות המדידה מתפזרות סביב הממוצע שלהם.

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

המחשת דיוק ונכונות:



رجישות (Recall) – “היזכרות”: החיצויים הרלוונטיים שאוחזו (mbosso על רלוונטיות שמצוינת באיזו מידת מסמך או קבוצת מסמכים מסוימים עוניים על צורכי המידע של המשתמש), כלומר ממד לכמה המודל חזה באמת את היז'אנר שהוא אמור לחזוז.

$$\text{sensitivity} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false negatives}}$$

ציון F1 (F1 score): מצין את הממוצע ההרמוני של הדיוק (Precision) והרגישות (Recall), ולכן מייצג באופן סימטרי את שני הממדדים במדד אחד.

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = 2 \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2\text{tp}}{2\text{tp} + \text{fp} + \text{fn}}.$$

לאחר הרצת פונקציית הערכת המודל למודל הנוכחי קיבלתי את הערכים הבאים:

#### The Summary of The Model Evaluation:

---

**Accuracy:** 0.7191666666666666

**Precision:** 0.9903225806451613

**Recall:** 0.9935275080906149

**F1 Score:** 0.9919224555735057

זאת אומרת שהערכת המודל היא טובה ויש לנו מודל חזק שנutan דיוק של 71 אחוזים לחיזוי נכון.

## מסקנות

פרויקט זה הוא הפרויקט המסכם את שנות לימודי כהנדסאי תוכנה, ובכלל התנסות בלעסוק בפרויקט כזה רחוב היקף הייתה נורא מأتגרת. אך עם כל הקשיים לדעתו החלטתי לעמוד במשימה על הצד הטוב ביותר שיכלתי עם מגבלת הזמן.

מגבלת הזמן הקשתה הרבה במהלך השנה על אופי העבודה ועם שילוב הלוח הלימודי היה עוד יותר קשה, שכן המסקנה העיקרית שלי היא שצריך יותר זמן בתכנון זמן העבודה מראש, ולעבד הרבה על איך לא להتفسר ולהתעסק בדברים שפחות רלוונטיים למטרות והיעדים שהוקצבו (בפרויקט גדול קל מאד להتفسר ולאבד את הדרך), אך על ידי עזרה וליפוי מהמנחה מפסיקים להتفسר ומתכוונים להגיע ליעדים.

ישנו הרבה חומר חדש שצריך ללמוד בלבד מנת להגיע לפרויקט כזה, וכן מסקנה נוספת היא לפני שניגשים לעבודה לעשות יותר עבודה תיאורטית, לבוא יותר מוכנים מבחינת הידע לפתרת הבעיה, אחרת זה יותר מסובך להגיע ליעדים ויותר קל להتفسר.

אך סך הכל אני חושב שעם עבודה קשה בסוף אפשר להגיע לתוצאות טובות ואפשר לקצור את הפירות, בכל דבר בחיים וזה רק עוד הוכחה לכך שהכל אפשרי.

## פיתוחים עתידיים

- להוסיף פידבק על החיזוי. כלומר, משתמש שקיבל חיזוי יהיה לו אפשרות לתת פידבק אם החיזוי היה נכון או לא, ובשילוב המידע זהה לאמן שוב את המודל בכך ליצור מודל יותר חזק. במלחים אחרות המשתמשים יעזרו לאמן את המודל.
- פיתוח לעוד פלטפורמות כגון: Android ו – iOS.
- הגדלתה – dataset, הוספה עוד דוגמאות על מנת ליצור מודל יותר חזק.
- יצירת מכונת המלצות, כמו למשל למשתמש המלצהות לשירים שאולן יאהב על פי הז'אנר שהוא בחר.
- לתת גם למשתמשים אורחים באתר להשתמש בשירותי האתר.
- להוסיף תמיינה לקובץ mp3.
- להוסיף פיצ'ר לניהול שירים במערכת

## ביבליוגרפיה

- Britannica: .clef .(2019 11 21) .Editor  
<https://www.britannica.com/art/clef>
- AUDIO MATCHING VIA CHROMA- .(2005) .'Meinard Muller, F' K  
 ismir2005: .אוחזר מתוך .BASED STATISTICAL FEATURES  
<https://ismir2005.ismir.net/proceedings/1019.pdf>
- . Chroma Feature Extraction .(2019 1) .'Shah, A' K  
 researchgate:  
[https://www.researchgate.net/publication/330796993\\_Chroma\\_Feature\\_Extraction](https://www.researchgate.net/publication/330796993_Chroma_Feature_Extraction)
- Music Processing using Chroma Features .(2013) .'Muller, M [סרט].
- Mel-Frequency Cepstral Coefficients .(2021) .'Velardo, V [סרט] Explained Easily
- Machine Learning for Music .(2022 11 24) .'Yu-Huei Cheng, C'-N' K  
 .Genre Classification Using Visual Mel Spectrum  
<https://www.mdpi.com/2227-7390/10/23/4427/pdf>
- .Neural Networks in Materials Science .(1999 6 12) .'BHADESH, H' K  
 אוחזר מתוך phase-trans: <https://www.phase-trans.msm.cam.ac.uk/abstracts/neural.review.pdf>
- But what is a neural network? | Chapter .(2018) .Blue1Brown3 [סרט] 1, Deep learning
- Build an AI from Scratch | Neural Network .(2021) .'Coding, A [סרט] Java
- Understanding and Implementing Neural .(2020 1 21) .'Sonawane, S [סרט] Networks in Java from Scratch  
 towardsdatascience: .אוחזר מתוך . Networks in Java from Scratch  
<https://towardsdatascience.com/understanding-and-implementing-neural-networks-in-java-from-scratch-61421bb6352c>
- Solved Music Genre Classification Project .(2023 4 26) .projectpro [סרט] using Deep Learning  
 projectpro: .אוחזר מתוך .using Deep Learning  
<https://www.projectpro.io/article/music-genre-classification-project-python-code/566>