**ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – ВАРНА**

**Катедра “Компютърни Науки и Технологии”**

# ЗАДАНИЕ

# за курсов проект по дисциплината

**„Изкуствен интелект - проект“**

на студента Калоян Енев

фак.№ 21621330

Тема на проекта: Приложение на методите за машинно обучение за разпознаване на 10 човека по гласа им

Преподавател:……………..……………..

/проф. д-р инж. Тодор Ганчев/

Студент:…………………….

/ име, подпис/

Съдържание

[ЗАДАНИЕ 1](#_Toc186922238)

[1. Въведение 3](#_Toc186922240)

[2. Анализ на съществуващи технологии 4](#_Toc186922241)

[3. Използвани технологии 6](#_Toc186922242)

[4. Проектиране и реализация 7](#_Toc186922243)

[5. Тестване и резултати 11](#_Toc186922244)

[6. Изводи и бъдещо развитие 14](#_Toc186922245)

[Източници 15](#_Toc186922246)

[Приложение 16](#_Toc186922247)

# Въведение

В последните десетилетия машинното обучение и изкуственият интелект революционизираха начина, по който взаимодействаме с технологиите. Един от най-значимите и широко използвани аспекти на тези технологии е автоматичното разпознаване на глас, което е в основата на много иновации – от гласови асистенти като Siri и Alexa до биометрични системи за сигурност.

Гласът е уникална характеристика на всеки човек, която може да бъде използвана за идентификация. Това прави технологиите за разпознаване на глас важен инструмент в съвременното общество, както за улеснение на ежедневния живот, така и за повишаване на сигурността. Настоящият проект има за цел разработването на система, която използва методи на машинното обучение, за да разпознае гласа на 10 различни човека.

Проблемът има голяма социална значимост и множество практически приложения:

* Сигурност и автентикация: Въведението на гласови биометрични технологии предлага ново ниво на защита за системи, които изискват идентификация – например банкови транзакции, достъп до устройства или дори лични данни. Гласът на потребителя става своеобразна парола, която е трудна за подправяне.
* Персонализация: В Smart домове и устройства, разпознаването на глас може да позволи персонализиране на взаимодействието – например автоматично настройване на музика, осветление или друго оборудване според предпочитанията на човек.
* Достъпност: За хора с физически увреждания, които не могат да използват стандартни методи за въвеждане на данни, разпознаването на глас предоставя нов начин за комуникация и управление на технологии.
* Здравеопазване: Технологията може да бъде интегрирана в системи за мониторинг и предоставяне на помощ на пациенти от разстояние, както и за ранна диагностика на здравословни проблеми чрез анализ на гласа.

Настоящият проект демонстрира как аудио данни могат да бъдат анализирани, обработени и класифицирани с помощта на съвременни методи като екстракция на характеристиките чрез MFCC (Mel Frequency Cepstral Coefficients) и използване на алгоритми за класификация като поддържащи векторни машини (SVM).

# Анализ на съществуващи технологии

Разпознаването на глас е област, която се развива активно от средата на 20-ти век. С навлизането на съвременните технологии за изчисление и изкуствения интелект, методите за разпознаване на глас стават все по-усъвършенствани. Съществуват два основни аспекта на проблема: разпознаване на реч и разпознаване на говорител.

Настоящият проект се фокусира върху втория аспект – разпознаване на говорител. Системите за разпознаване на говорител са класифицирани в две основни категории: идентификация (определяне на самоличността на говорителя сред група от известни лица) и верификация (потвърждаване на идентичността на говорителя спрямо предоставена претенция).

Първите системи за разпознаване на говорител разчитат на класически подходи, базирани на статистически модели:

* MFCC (Mel-Frequency Cepstral Coefficients): Един от най-популярните методи за извличане на характеристики от аудио сигнали. MFCC анализира звуковия спектър, за да идентифицира уникалните особености на гласа.
* Линейна дискриминантна функция (LDA): Използва се за намаляване на размерността на данните, като същевременно запазва важните характеристики, необходими за класификация.
* GMM (Gaussian Mixture Models): Използвани за моделиране на разпределението на гласовите характеристики.
* HMM (Hidden Markov Models): Популярни за обработка на времеви последователности, този подход се използва за моделиране на времевата структура на речта. HMM често се комбинира с MFCC за постигане на по-добра точност.

Тези подходи са ефективни за малки и контролирани среди, но се затрудняват при наличие на шум или големи вариации в данните. Те са основа за разработването на модерни алгоритми, но имат ограничена способност да се справят с големи и разнообразни набори от данни.

С навлизането на Deep Learning се появяват нови подходи, които значително подобряват точността и надеждността на системите за разпознаване на говорител:

* Дълбоки невронни мрежи (DNN): Тези мрежи използват многослойни архитектури за извличане на по-сложни характеристики от аудио данните.
* Конволюционни невронни мрежи (CNN): Особено подходящи за анализ на спектрограми – визуални представяния на звука, които улавят честотните характеристики на гласа.
* Поддържащи векторни машини (SVM): Често използвани за задачи, свързани с класификация и верификация.
* Рекурентни невронни мрежи (RNN): използвани за обработка на времеви зависимости, като например промените в гласа във времето.
* Speaker Embeddings: Методи като x-vectors и d-vectors са модерни подходи за представяне на характеристиките на гласа като вектор в многомерно пространство.
* Linear Predictive Coding (LPC): Използва линейни модели за прогнозиране на следващия сигнал на базата на предишните.

Въпреки значителния напредък в областта, съществуват редица предизвикателства, които ограничават практическото приложение на системите за разпознаване на говорител:

* Шум в средата: Наличие на фонов шум може да затрудни извличането на точни характеристики.
* Вариации в гласа: Гласът на един и същ човек може да варира в зависимост от здравословното състояние, емоционалното състояние или възрастта.
* Ограничения на данните: Липсата на достатъчно разнообразни и качествени аудиозаписи за обучение може да доведе до по-ниска точност.
* Сигурност: Възможността за имитация на глас или използване на синтезиран глас представлява заплаха за системите за верификация.
* Обработка в реално време: Много приложения изискват системите да работят в реално време, което увеличава изчислителната сложност.

Системите за разпознаване на говорител продължават да се развиват с акцент върху:

* Намаляване на чувствителността към шум и фонови смущения.
* Разработване на методи за обучение с по-малки набори от данни.
* Интеграция на мултимодални технологии, които използват не само звук, но и други биометрични данни за идентификация.

Настоящият проект се позиционира в контекста на съвременните изисквания и технологии, като се основава на утвърдени методи като MFCC за извличане на характеристики и поддържащи векторни машини (SVM) за класификация. Въпреки че не използва най-новите подходи като дълбоки невронни мрежи, проектът има за цел да демонстрира ефективността на класически, но все още приложими методи в малък мащаб. Тази система може да бъде основа за надграждане с по-усъвършенствани технологии в бъдеще.

# Използвани технологии

Избраният език за разработка е **Python**, който се отличава с широк набор от библиотеки за машинно обучение и обработка на аудио. Python е универсален и широко разпространен, с добра документация, което го прави предпочитан избор за задачи от подобен характер.

За извличане на характеристиките на гласа е използвана библиотеката **Librosa**. Тя предоставя готови инструменти за обработка на аудио сигнали, включително изчисляване на Mel-frequency cepstral coefficients (MFCC), които са доказали своята ефективност като основни характеристики при разпознаване на говорители.

Избраният алгоритъм за разпознаване е **поддържаща векторна машина (Support Vector Machine, SVM).** Основните причини за използването на SVM са ефективността му при малък набор от данни, добра интерпретация лесно управление и способността за класификация.

Библиотеката Scikit-learn е избрана за реализация на SVM модела. Scikit-learn предлага интуитивен интерфейс за създаване, обучение и тестване на модели, както и готови функции за разделяне на данни и оценка на производителността на алгоритмите.

Средата за разработка е Visual Studio Code, която е лека, лесна за конфигуриране и предлага интеграция с Python и виртуални среди. Използването на виртуална среда (venv) гарантира изолиране на зависимостите, което предотвратява конфликти между различни библиотеки.

Предварително намерените аудио файлове са в MP3 формат. За да бъдат обработени от Librosa, файловете са преобразувани в WAV формат с помощта на инструмента FFmpeg. Това решение осигурява съвместимост с библиотеките за обработка на аудио и висока точност при извличане на характеристики.

# Проектиране и реализация

Общата архитектура на проектът е базирана на последователност от обработващи стъпки, които извършват следните основни действия:

1. Зареждане на данни: Данни, съхранявани в поддиректории, всяка съдържаща аудио записи на различен човек, се зареждат и подготвят за обработка.
2. Извличане на характеристики: За всяка аудио проба се извличат характеристики (MFCC) чрез библиотеката librosa, които се използват за обучение на модела.
3. Обучение на модел: Извлечените характеристики се използват за обучение на класификатор, в този случай — SVC (Support Vector Classificator).
4. Оценка на модела: След обучението моделът се тестват с тестова част от данните, за да се оцени неговата точност.
5. Предсказване на нови записи: Моделът се използва за предсказване на идентичността на нови аудио записи.

Първият модул представлява алгоритъм, който има цел да подготви суровите аудио данни във формат, подходящ за извличане на характеристики. Това включва преобразуване на файловете в унифициран формат (WAV), стандартна честота на дискретизация и отстраняване на потенциални проблеми със звуковите данни.

Алгоритъмът започва с четене на входните файлове. Входният файл се зарежда чрез библиотеката librosa, но за да бъде прочетен правилно е необходимо да бъде конвертиран чрез инструментът FFmpeg ако е в неподходящ формат. Намерените аудио записи бяха във формат MP3, но за по-точно извличане на характеристики е нужно да бъдат преобразувани към WAV формат, който позволява работа с librosa. Извършва се и проверка на настройка на честотата на дискретизация, която е необходимо да бъде 16 kHz. Намерени са поне по 20 аудио файла на човек, разделени в 10 поддиректории, като някои от тях не са с добро качество и включват гласове на мъже и жени от различни възрастови групи.

Модулът за извличане на характеристики (extract mfcc) включва зареждане на аудиото от входните файлове с помощта на librosa.load(), като се задава честота 16 kHz. След това се извличат MFCC характеристиките от даден аудио файл с помощта на librosa.feature.mfcc().

MFCC представлява Mel-frequency cepstral coefficients. Това са характеристики, използвани за представяне на звуковите сигнали, които се използват основно в разпознаването на говор и аудио класификация. Те се изчисляват чрез преобразуване на аудиофайловете в честотния спектър, като се използва мела-скала, която имитира начина, по който човешкото ухо възприема звуците. MFCC предоставят компактни и представителни характеристики на звука, които се използват в машинно обучение и невронни мрежи за задачи като разпознаване на говор и идентификация на личности по глас.

Поради различни дължини на аудио пробите, ако дължината на MFCC е по-малка от предварително зададена стойност, то тя се запълва с нули, или се скъсява, за да стане еднаква за всички проби. Изходните данни са масив, който съдържа извлечените характеристики.

Модулът за зареждане на данни(load\_dataset) има като функция да зареди всички аудио файлове от зададената директория , като за всяка категория (човек) присвоява уникален етикет. Като входни данни получава път към главната директория, съдържаща папките с аудио записи на различни хора, а като изходни данни изкарва масиви с характеристики (x) и етиките (y), както и карта на етикетите към имената на хората( label\_map). За целта се преглеждат всички поддиректории и всеки аудио файл се обработва чрез extract\_mfcc, за да се получат характеристиките и пробите се добавят в списъците x и y.

Модулът за разделяне на данни (split\_dataset) има за цел да раздели данните на тренировъчни и тестови набори. Като вход получава масиви с характеристики и етикети. Използва функцията train\_test\_split от библиотека sklearn, за да раздели данните на тренировъчни и тестови, като използва 35% от данните за тест. След тестване с различни стойности, най-добра точност се получава ако тестовите данни са 35%. Изходните данни са тренировъчни и тестови набори от характеристики и етикети.

Модулът за обучение на модел (train\_model) има функцията да обучи класификатор на основата на SVM. Тренировъчните характеристики и етикети са входните данни и се използва класификатор SVC(Support Vector Classificator) с ядро от тип линейно, което е ефективно при работа с малък брой класове както е случаят с настоящата програма. Следва обучение на моделът върху тренировъчните данни. Като изход излиза обучения модел.

Следващ е модулът за оценка на модела (evaluate\_model). Този модул извършва оценяване на производителността на получения модел. Като входни данни получава тестовите данни и етикети, както и етикетната карта. Моделът прави предсказания върху тестовите данни. Оценката се извършва чрез изчисляване на точността и генериране на класификационен отчет с помощта на accuracy\_score и classification\_report от библиотеката sklearn. Classification report включва няколко категории оценки.

Първата от тях е precision (прецизност). Това е процентът от предсказаните примери за даден клас, които наистина принадлежат към този клас. Измерва колко точен е моделът, когато предсказва даден клас.

Recall (пълнота) е процентът от истинските примери за даден клас, които са правилно класифицирани от модела. Измерва колко добре моделът разпознава всички примери от даден клас.

F1-score е зависимост между recall и precision. Измерва общото представяне на модела за даден клас, комбинирайки точността и чувствителността.

Accuracy (точност) е процентът от правилно класифицирани примери спрямо всички примери. Измерва общото представяне на модела върху целия тестови набор.

Support показва броя на аудиофайловете, които са в тестовото множество за даден клас. Важно е то да е достатъчно и да е равномерно за всички класове, защото при небалансиран брой, има възможност да се получат грешки при измерването на останалите оценки.

Модулът за предсказване на нови записи извършва предположение за идентичността на даден нов аудио файл. Като входни данни получава път към новия аудио файл, обучен модел и етикетна карта. Извлича MFCC характеристиките от новия аудио файл моделът прави предположение за етикета на новия аудио файл на база на данните, които са минали през него. Свързва характеристиките на новия аудио файл с най-близките характеристики от тренировъчните данни. Изходни данни са етикета на новия аудио файл, съответстващ на директорията с най-близки аудио характеристики.

В този проект е реализирана система за разпознаване на гласове на 10 различни души, като е използван метод за машинно обучение. Процесът на реализация включва редица решения, които имат за цел да осигурят ефективност, точност и оптимизация на цялостния процес.

MFCC (Mel Frequency Cepstral Coefficients) е избран като основен метод за извличане на характеристики от аудиофайловете. Това решение е мотивирано от факта, че MFCC представляват един от най-ефективните и широко използвани методи за обработка на звукови сигнали, особено в областта на разпознаването на реч.

Използването на поне 40 MFCC характеристики, които се извличат от всяка аудио извадка, позволява на системата да улови основните акустични особености на гласа на различните хора.

Важен момент в реализацията е подходът към обработката на аудиофайловете, които могат да имат различна дължина. В тази програма се извършва допълване или изрязване, за да се постигне еднаквост на на MFCC характеристиките, така че да се осигури еднакъв размер на входните данни за машинно обучение, независимо от различията в дължината на аудиофайловете. Ако размерът на MFCC характеристиките е по-малък от максимално зададения (максимално 174 времеви кадъра), тогава се добавят "нулеви" стойности, докато не се постигне нужния размер. Това решение гарантира, че размерността на входа за модела остава постоянна.

Използван е Support Vector Machine (SVM) с линейно ядро за класификация на гласовете. Това решение е мотивирано от факта, че SVM е мощен и широко използван метод за класификация, особено когато се работи с многоизмерни данни, каквито са MFCC характеристиките.

Използването на линейно ядро за SVM е избрано с цел простота и ефективност. Линейното ядро често се представя като ефективно за задачи с малко на брой класове и данни, като разпознаване на гласа на 10 души в този случай.

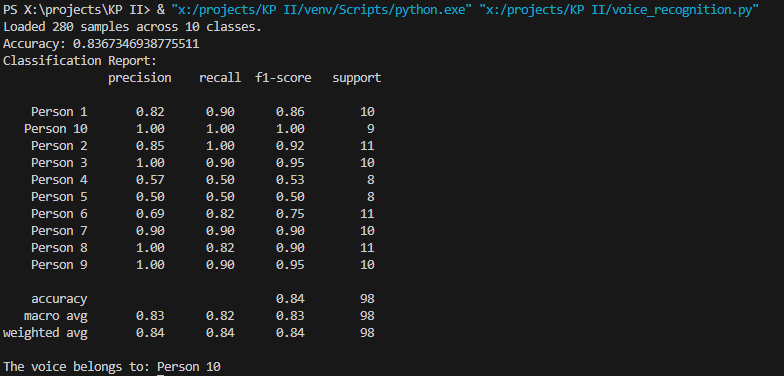
За разделянето на данните на тренировъчни и тестови е използвана стандартната техника train\_test\_split от библиотеката sklearn.model\_selection. Избрано е разделение от 80% за обучение и 20% за тест, което е стандартно за малки бази от данни и осигурява балансиран подход за обучение и оценка на модела.

Оценката на производителността на модела е извършена чрез два основни метрики: точност и репорти за класификация, включващи прецизност, обхват и F1 мярка. Това решение е взето с цел детайлно измерване на представянето на модела, като се вземат предвид не само точността на класификацията, но и възможните грешки в класификацията на различните класове.

След обучението на модела, е реализирана функция за прогнозиране на нови аудиофайлове. Чрез извличане на MFCC характеристики от ново аудио и подаването им на обучената SVM класификаторна система, системата е способна да предскаже кой човек от 10-те класове принадлежи на аудиото.

# Тестване и резултати

При тестване на програмата е подаден, като вход, път към нов аудио файл, който не присъства в набора от аудио записи, които са използвани за обучение на модела. Новия запис съдържа гласът на един от десетте класа(хора). Съотношението обучителни/тестови файлове е 65/35.



Първо се потвърждава броя на заредените проби – 290 и броя на класовете, на които са разделени – 10. После се извежда точността на модела (Accuracy). Точността на модела е 78.43%, което означава, че моделът правилно класифицира около 3/4 от тестовите примери.

Macro Avg обобщава средните стойности на precision, recall и f1-score. Стойностите са съответно 0.78, 0.78 и 0.77. Това означава, че показват сравнително добра производителност на модела, но има пропуски при отделните класове.

Weighted Avg взема предвид размера на всеки клас и показва сходни резултати като точността (78%), което потвърждава балансирана производителност.

Наблюдения за отделните класове:

Person 1:

* Precision: 0.82  
  Висока стойност, което показва, че моделът рядко класифицира други класове като "Person 1".
* Recall: 0.90  
  Моделът разпознава голяма част от истинските примери за "Person 1".
* F1-score: 0.86  
  Високо общо представяне

Person 2:

* Precision: 0.85  
  Висока стойност, което означава, че моделът рядко сбърква, когато класифицира "Person 2".
* Recall: 1.00  
  Перфектно разпознаване на истинските примери за "Person 2".
* F1-score: 0.92  
  Отлично общо представяне.

Person 3:

* Precision: 1.00  
  Перфектна стойност, което означава висока увереност при класификация на този клас.
* Recall: 0.90  
  Моделът разпознава почти всички примери.
* F1-score: 0.95  
  Отлично представяне.

Person 4:

* Precision: 0.57  
  Средна стойност, което показва умерена точност при този клас.
* Recall: 0.50  
  Моделът разпознава половината от истинските примери.
* F1-score: 0.53  
  Средно общо представяне

Person 5:

* Precision: 0.50  
  Средна стойност; моделът има умерена точност за този клас.
* Recall: 0.50  
  Само половината от истинските примери са правилно разпознати.
* F1-score: 0.50  
  Най-ниско общо представяне.

Person 6:

* Precision: 0.69  
  Умерена точност, но не е впечатляваща.
* Recall: 0.82  
  Голяма част от истинските примери са правилно разпознати.
* F1-score: 0.75  
  Задоволително общо представяне.

Person 7:

* Precision: 0.90  
  Отлична стойност, моделът е много уверен в класификациите за този клас.
* Recall: 0.90  
  Не разпознава малка част от истинските примери.
* F1-score: 0.90  
  Отлично общо представяне.

Person 8:

* Precision: 1.00  
  Перфектна точност.
* Recall: 0.82  
  Висока стойност; моделът разпознава повечето примери.
* F1-score: 0.90  
  Отлично общо представяне.

Person 9:

* Precision: 1.00  
  Перфектна точност; моделът не допуска грешки при класификацията на "Person 9".
* Recall: 0.90  
  Почти всички примери са правилно разпознати.
* F1-score: 0.95  
  Отлично общо представяне.

Person 10:

* Precision: 1.00  
  Перфектна стойност, което означава, че моделът правилно разпознава "Person 10" във всички случаи.
* Recall: 1.00  
  Моделът разпознава всички от истинските примери за "Person 10".
* F1-score: 1.00  
  Перфектно представяне.

Въпреки някои по-ниски стойности за някои класове, програмата успява да разпознае гласовете, принадлежащи на всеки от класовете.





# Изводи и бъдещо развитие

Моделът постига стабилни резултати с високи стойности на precision, recall и F1-score, което показва, че правилно класифицира аудио пробите на различни класове. Показва адаптация към различни категории гласове(мъже, жени, различни възрастови групи, с или без заден шум). Разпознава нови аудио файлове от всеки от класовете без грешка. Моделът е **ефективен и надежден** при разпознаване на говорители в контролирани условия. Използването на MFCC за извличане на аудио характеристики се оказва ефективно за разпознаване на различни говорители, като се постигат високи оценки за почти всички класове.

Някои от недостатъците му са неравномерно представяне между класовете. Въпреки, че повечето от класовете дават отлични резултати, някои от тях показват средни стойности. Това може да се дължи на липса на характерни гласови особености или недостатъчно представителни данни в тренировъчния набор. Въпреки, че MFCC е лесен за упортеба и сравнително прост начин да се постигне задачата, неговите възможности за различаване на гласове, които имат близки, но различни характеристики и особености е ограничена. Това може да доведе до объркване между класове.

Моделът може да бъде чувствителен на вариации в качеството на аудио пробите и да изисква по-голямо количество данни за различни условия, за да се избегнат грешки в класификацията при нови ситуации.

Макар моделът да е успешен за текущия набор от данни, все още има място за подобрения, особено когато се работи с шумни или неравномерно записани аудио файлове.

Възможна е допълнителна обработка на тренировъчния набор от файлове. Приложение на техника за изолиране на гласовете от страничния шум или премахването на този шум би увеличило устойчивостта на модела към реални условия. При добавяне на повече проби към тестовия и тренировъчния набор и работата с класове, които съдържат по-голямо количество проби, би подобрило точността на модела и би постигнало по-добра производителност. Също от помощ за генерализацията на модела би било разширяването включването на аудио записи съдържащи различни акценти, езици и вариации в тоналността.

Преход към по-напреднали алгоритми като дълбоки невронни мрежи (Deep Neural Networks), Convolutional Neural Networks (CNN) или Recurrent Neural Networks (RNN), които са специално проектирани за работа със звукови данни може да предложи по-добро представяне при задачи с голям набор от данни. Също включването на други аудио характеристики като Chroma features или Spectral contrast, би предоставило по-богата информация за гласовите модели.

# Източници

 *Librosa Documentation*. Официалната документация за библиотеката Librosa. https://librosa.org/doc/latest/index.html

 *Scikit-learn Documentation*. Документация за библиотеката scikit-learn, използвана за SVM. https://scikit-learn.org/stable/

 *GitHub*. Ресурси и проекти с отворен код за разпознаване на глас. <https://github.com/topics/voice-recognition>

 *Librispeech ASR Corpus*. Оттук са взети аудиофайловете. https://www.openslr.org/12

# Приложение

import os import numpy as np import librosa from sklearn.model\_selection import train\_test\_split

from sklearn.svm import SVC from sklearn.metrics import accuracy\_score, classification\_report

# Извеждане на MFCC характеристики

def extract\_mfcc(file\_path, n\_mfcc=40, max\_len=174):

audio, sr = librosa.load(file\_path, sr=16000) # Зарежда аудио с честота 16kHz

mfccs = librosa.feature.mfcc(y=audio, sr=sr, n\_mfcc=n\_mfcc)

# Допълва се с нули или се изрязва част от файла, за еднаква дължина на данните

if mfccs.shape[1] < max\_len:

pad\_width = max\_len - mfccs.shape[1]

mfccs = np.pad(mfccs, pad\_width=((0, 0), (0, pad\_width)), mode='constant')

else:

mfccs = mfccs[:, :max\_len]

return mfccs.flatten() # преобразува се до едноизмерен масив

# Зареждане на набор от данни от директория

def load\_dataset(dataset\_path):

X, y = [], []

label\_map = {}

for label, person\_folder in enumerate(sorted(os.listdir(dataset\_path))):

label\_map[label] = person\_folder

person\_path = os.path.join(dataset\_path, person\_folder)

# Преминава през всички файлове на поддиректориите

for file in os.listdir(person\_path):

if file.endswith('.mp3'): # Работа с mp3 файлове

file\_path = os.path.join(person\_path, file)

features = extract\_mfcc(file\_path)

X.append(features)

y.append(label)

return np.array(X), np.array(y), label\_map

# Трениране на SVM

def train\_model(X\_train, y\_train):

model = SVC(kernel='linear', probability=True) # Support Vector Classifier с линейно ядро

model.fit(X\_train, y\_train)

return model

# Оценка на модела

def evaluate\_model(model, X\_test, y\_test, label\_map):

y\_pred = model.predict(X\_test)

print("Accuracy:", accuracy\_score(y\_test, y\_pred))

print("Classification Report:\n", classification\_report(y\_test, y\_pred, target\_names=label\_map.values()))

# Предположение чии е новия аудио файл

def predict\_person(file\_path, model, label\_map):

features = extract\_mfcc(file\_path).reshape(1, -1)

prediction = model.predict(features)

return label\_map[prediction[0]]

dataset\_path = "X:\\projects\\KP II\\Dataset"

# Зареждане на набор от данни

X, y, label\_map = load\_dataset(dataset\_path)

print(f"Loaded {X.shape[0]} samples across {len(label\_map)} classes.")

# Разделяне на данните

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.35, random\_state=42, stratify=y)

# Трениране на SVM модел

model = train\_model(X\_train, y\_train)

# Оценка на модела

evaluate\_model(model, X\_test, y\_test, label\_map)

# Тестване на нов аудио файл

new\_sample = "X:\\projects\\KP II\\test\\Person 10 test.mp3"

predicted\_person = predict\_person(new\_sample, model, label\_map)

print(f"The voice belongs to: {predicted\_person}")