КУРСОВА РОБОТА
з дисципліни “Машинне Навчання”

Передбачення чисел з аудіо-файлів

Виконали студенти групи СА-33
Бичков І.О.
Процюк Ю.Р.

Мета

Розробка системи класифікації аудіофайлів, що містять вимовлені цифри (0-9),
з використанням методів машинного навчання та обробки сигналів для подальшого використання у системах голосового введення, біометричних системах та підвищення доступності технологій.

Задачі

1. **Збір та підготовка даних:**
   - Завантаження аудіофайлів з датасету, що містить 30000 зразків вимовлених цифр (0-9) від 60 різних мовців.
   - Використання бібліотеки Librosa для отримання MFCC (Мел-частотних кепстральних коефіцієнтів) з аудіофайлів.
   - Формування даних для навчання та тестування шляхом обчислення середніх значень MFCC для кожного аудіофайлу та додавання цифрової мітки, що відповідає вимовленій цифрі.
2. **Обробка даних:**
   - Стандартизація фіч за допомогою методу StandardScaler для зменшення впливу масштабу та забезпечення кращої продуктивності моделі.
   - Розподіл даних на навчальну та тестову вибірки для оцінки ефективності моделі.
3. **Навчання моделі:**
   - Використання алгоритму Random Forest для класифікації аудіофайлів з вимовленими цифрами.
   - Навчання моделі на навчальній вибірці та перевірка її точності на тестовій вибірці.
   - Використання метрик класифікації, таких як точність, відновлення та F1-оцінка, для оцінки якості моделі.
4. **Тестування:**
   - Перевірка точності та ефективності розробленої системи на тестових даних.

Що таке MFCC (Mel-Frequency Cepstral Coefficients)

Мел-частотні кепстральні коефіцієнти (MFCC) - це метод для представлення звукових сигналів у задачах обробки мови та звуків.

Основні концепції:

1. **Частотна шкала Мела**:
   - Шкала Мела є перцептивною шкалою частот, яка більш точно відповідає людському сприйняттю звуку. Вона нелінійно масштабована таким чином, що різниці в нижчих частотах є більш виразними для людського вуха, ніж у вищих частотах.
2. **Кепстральний аналіз**:
   - Кепстральний аналіз перетворює сигнал із частотного простору в кепстральний, що дозволяє більш ефективно відокремлювати різні джерела звуку та оцінювати їхні характеристики.

Етапи обчислення MFCC

1. Завантаження аудіофайлу:

```
y, sr = librosa.load(audio_path)
```

- ***y*** - це масив значень амплітуд сигналу, ***sr*** - частота дискретизації (sample rate).

2. Обчислення MFCC:

```
mfccs = librosa.feature.mfcc(y=y, sr=sr)
```

- mfccs - матриця MFCC, де кожен рядок відповідає одному з коефіцієнтів MFCC, а кожен стовпець відповідає окремому часовому вікну.

2.1. **Розбивка сигналу на фрейми:**
- Аудіосигнал ***y*** розбивається на короткі інтервали часу (фрейми) з перекриттям.

2.2 **Застосування віконної функції:**
- До кожного фрейму застосовується віконна функція (зазвичай, функція Хеннінга або Хаммінга).

2.3 **Перетворення Фур'є (FFT):**
- Виконання швидкого перетворення Фур'є (FFT) для кожного фрейму. Перетворює сигнал з часового простору у частотний.

2.4 **Розрахунок спектральної потужності:**
- Спектральна потужність обчислюється як квадрат амплітуд результатів FFT. Отримуємо енергетичної інформації у частотному просторі.

2.5 **Застосування мел-фільтрів:**
- Спектральна потужність проходить через набір мел-фільтрів, що моделюють людське сприйняття звуку.
- Перетворення частотної шкали на мел-шкалу, яка ближче до сприйняття людини.

2.6 **Логарифмування:**
- Логарифмування енергій, отриманих після мел-фільтрації. Перетворює мультиплікативних відносин в адитивні.

2.7 **Дискретне косинусне перетворення (DCT):**
- Застосування дискретного косинусного перетворення (DCT) до логарифмованих енергій для отримання кепстральних коефіцієнтів. Зменшує кореляції між фічами та забезпечує компактне представлення звукових характеристик.

MFCC

Також після обчислень MFCC, її усереднюють по осі часу.

```
mfccs_mean = np.mean(mfccs, axis=1)
```

Це додатковий крок, який роблять для зменшення розмірності даних та отримання єдиної характеристики для кожного аудіофайлу.

Розбір спектограм для аудіофайлів

Ось X/Time:

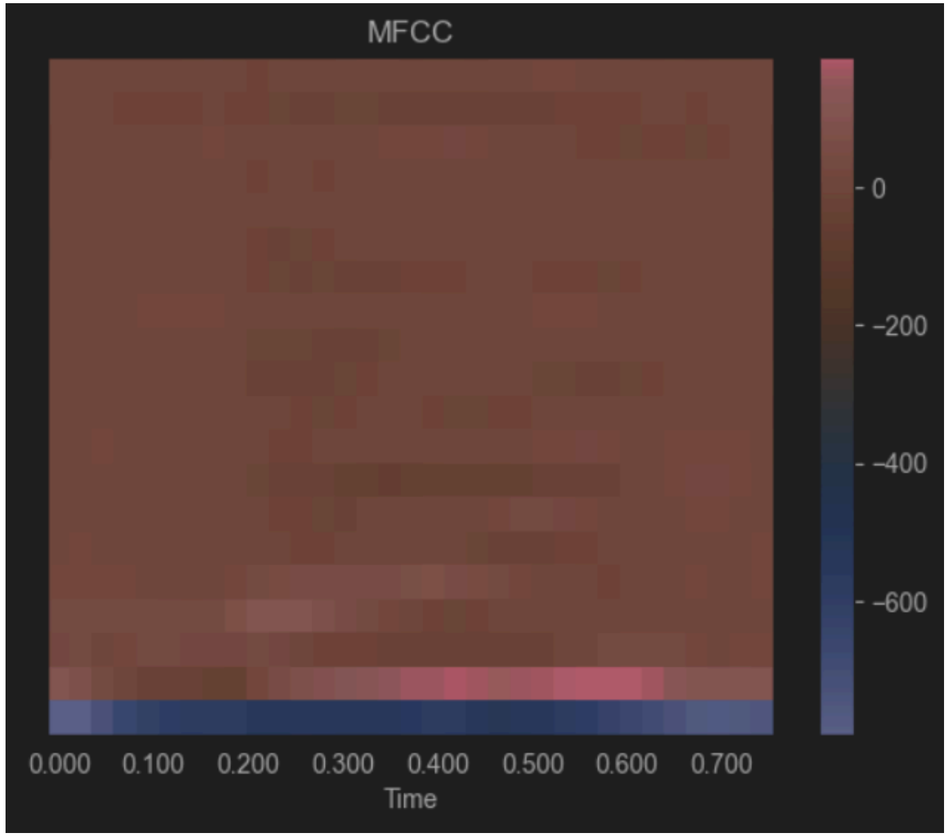
- Відображає час. Виміряється у секундах. На зображенні час варіюється від 0 до приблизно 0.7 секунди.

Ось Y/(Вертикальна вісь):

- Відображає різні MFCC коефіцієнти. На зображенні зазвичай представлені від 1 до 13 або від 1 до 20 MFCC коефіцієнтів (у даному випадку, можливо, 13).

Колірна шкала:

- Відображає амплітуду або інтенсивність кожного MFCC коефіцієнта в даний момент часу. Вона показує, наскільки інтенсивно певний коефіцієнт присутній у сигналі.
- Темніші відтінки (ближче до синього) показують нижчі значення, а світліші відтінки (ближче до червоного) показують вищі значення.



Препроцесинг

З усередненних значень створили датафрейм:

```
cols = ['mean_' + str(x+1) for x in range(20)] + ['Digit']
df = pd.DataFrame(data, columns=cols)
```

Отримали такий фрейм:

де mean - це фічи, що показують усереднене значення по тому коефіцієнту.

Digit - таргет змінна, класи наших цифр.

mean_12	mean_13	mean_14	mean_15	mean_16	mean_17	mean_18	mean_19	mean_20	Digit
-7.153427	1.122348	-4.694363	-7.230122	5.245928	-9.295115	2.525946	-9.663425	-0.382305	1.0
0.357722	2.423094	6.758825	10.924612	8.935596	-5.561665	11.661551	0.743707	3.665671	6.0
-8.513130	-5.621245	4.379300	-4.371952	11.276179	-1.743025	0.263442	1.419058	5.511829	3.0
6.326621	-4.907639	-12.400064	0.059062	-2.710019	-7.463569	-0.181983	-8.825382	2.653709	1.0
4.504506	-2.152862	1.037812	-5.616712	1.134958	-5.944080	-5.113520	-2.000279	4.643931	1.0
1.187116	-5.577405	-5.053595	7.706206	4.347718	-3.134221	4.279936	-4.799533	4.889969	5.0
20.422298	-8.380698	9.559462	-6.420155	9.432133	-5.571363	-3.591152	6.536330	-5.477557	5.0
5.323957	-3.537404	6.591034	3.265106	5.971255	-4.742702	1.260968	0.822067	3.118725	3.0
7.142104	7.442900	-2.340296	-2.449526	-0.696164	-6.394629	-3.647335	1.069006	1.198559	1.0
7.175766	-7.501061	-2.808023	11.165538	-4.489010	-11.327753	7.221345	-1.874481	-5.737396	7.0
6.904252	9.968782	5.764201	4.534809	2.399212	-0.534864	3.714726	-4.519302	6.855125	6.0
-7.127539	7.460071	7.262164	-4.544582	1.183448	-6.967472	-1.830624	-4.983619	4.384624	0.0
-2.841932	4.625839	-0.486699	-11.900468	9.911427	-7.281892	-7.238706	-0.222137	-8.787226	9.0
0.869518	-3.086722	2.629725	-5.001273	-2.433813	2.148644	7.341896	1.327934	15.432953	2.0
2.797651	1.043978	5.939509	1.260175	-0.080332	-0.172183	1.850710	-5.184192	5.710159	1.0
2.751803	6.332938	6.815901	-4.451080	8.476392	0.141473	7.763018	4.576303	9.824545	6.0
3.013999	-0.836679	7.815318	-0.325137	12.986958	2.064237	-0.089098	2.217944	6.293891	6.0
-2.122406	-0.114312	4.668982	6.162301	7.505690	-0.715171	0.509390	-4.249457	3.085228	0.0
1.042399	-1.897561	-2.322818	1.740340	8.655058	2.634704	3.713929	-3.287013	2.409018	3.0
0.617276	-3.465986	9.304767	1.486391	4.802594	-3.202544	1.478946	0.220384	0.881627	7.0

Препроцесинг

Відокремлення фіч і міток:

```
X=df.drop("Digit", axis=1)
y=df["Digit"]
```

X - це DataFrame, що містить всі фічі (усереднені MFCC коефіцієнти),
y - це Series, що містить таргет змінну(цифри).

Стандартизація фіч:

```
scaler = StandardScaler()
features_scaled = scaler.fit_transform(X)
```

Стандартизація фіч шляхом приведення їх до нульового середнього значення та одиничного стандартного відхилення. Це покращує продуктивність моделі машинного навчання, оскільки масштаби різних фіч стають однаковими.

	0	1	2	3	4	5	6	7	8	9	10
0	-0.254593	0.390670	-0.404940	-0.268522	0.657139	-0.223622	0.676759	-1.603547	1.551632	-0.471218	-0.364442
1	-0.641431	0.481332	0.176165	0.145117	1.195123	0.108171	0.191884	-1.046942	1.462123	-1.116777	-0.051634
2	0.102769	0.368093	-0.830204	-0.644284	0.761936	-0.132086	0.691258	-1.568576	0.941512	-0.855915	-0.149890
3	0.298382	0.662655	-0.859082	-0.624719	0.664230	-0.419256	-0.605071	-2.098284	0.950574	-0.113923	0.047448
4	-0.294699	0.277463	0.242865	-0.238114	0.203283	0.146152	-0.243516	-1.501019	1.165621	-0.817397	-0.313948
5	-0.424054	0.172451	0.594831	0.176651	0.846653	0.166679	-0.094420	-0.718564	1.464959	-1.087072	0.242448
6	-0.604037	0.120010	0.247996	0.125773	0.364335	-0.041590	-0.307134	-0.963908	1.341966	-0.924079	0.029999
7	0.069265	0.164125	-0.314200	-0.334573	0.677948	-0.257700	0.029175	-2.108904	1.251391	-0.367185	0.104956
8	0.331073	-0.040469	0.258229	-0.262135	-0.006601	0.221996	-0.789034	-1.661118	1.083862	-0.208105	0.020538
9	-0.593473	0.427248	-0.074984	-0.147014	0.736604	0.133717	1.013430	-1.613986	0.885742	-0.391284	-0.332527

Препроцесинг

Зменшення вимірності за допомогою PCA:

```
pca = PCA()
features_pca = pca.fit_transform(features_scaled)
explained_variance = pca.explained_variance_ratio_
```

pca - це екземпляр PCA.
features_pca - це фічі, зменшені за розмірністю за допомогою PCA.
explained_variance - це відношення дисперсії, яку пояснює кожен з компонентів PCA.

Зменшення вимірності даних, зберігаючи найбільш важливу інформацію. PCA дозволяє зменшити кількість фіч, що може прискорити навчання нашої моделі та зменшити ризик перенавчання.

Розподіл даних на навчальну та тестову вибірки:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

Розподіл даних на навчальну (70%) і тестову (30%) вибірки для навчання та оцінки моделі. Навчальна вибірка використовується для навчання моделі, а тестова - для оцінки її продуктивності на невідомих даних.

Навчання моделі (Random Forest Classifier)

RandomForestClassifier будує багато дерев рішень під час навчання. Кожне дерево рішень є окремою моделлю, яка може прогнозувати результат на основі навчальних даних.

Чому RandomForestClassifier ?

Висока точність та стабільність:

- Random Forest складається з великої кількості дерев рішень (decision trees), які працюють разом як ансамбль. Кожне дерево створюється з випадкової вибірки даних та випадкової підмножини фіч. Це зменшує ризик перенавчання (overfitting) і робить модель стійкою до шуму в даних.

Можливість обробки великої кількості фіч:

- Random Forest добре працює з даними, які мають багато фіч, як у випадку з MFCC, де кожен аудіофайл представлений 20 коефіцієнтами.

Мала потреба в налаштуванні параметрів:

- На відміну від інших моделей, таких як SVM або нейронні мережі, Random Forest часто потребує меншого налаштування параметрів і може добре працювати "з коробки".

Результат передбачення (Random Forest Classifier)

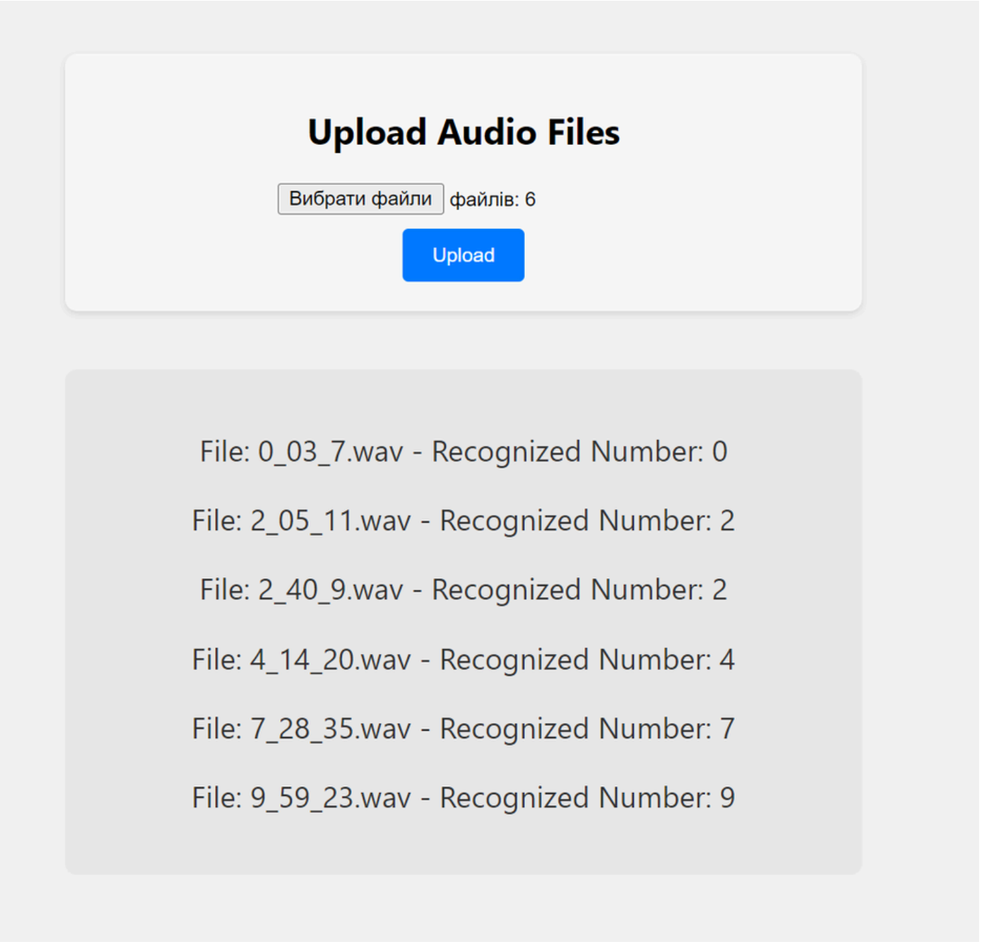
```
y_pred = best_model.predict(X_test)
print(classification_report(y_test, y_pred))
```

Результат передбаченння становить близько 0.97 до кожного класу цифри, що означає що модель дуже добре передбачує сказані в аудіо числа

	precision	recall	f1-score	support
0.0	0.96	0.95	0.95	907
1.0	0.97	0.96	0.96	919
2.0	0.97	0.94	0.95	879
3.0	0.94	0.96	0.95	873
4.0	0.98	0.99	0.99	927
5.0	0.97	0.99	0.98	902
6.0	0.99	0.99	0.99	917
7.0	0.96	0.98	0.97	878
8.0	0.97	0.97	0.97	909
9.0	0.96	0.96	0.96	889
accuracy			0.97	9000
macro avg	0.97	0.97	0.97	9000
weighted avg	0.97	0.97	0.97	9000

Результат передбачення (Random Forest Classifier)

Було створено сайт, на який можна встановити аудіо-файли і подивитись передбачення самостійно



Висновок

У курсовій роботі розроблено систему класифікації аудіофайлів з вимовленими цифрами (0-9) за допомогою методів машинного навчання, зокрема алгоритму Random Forest.
Основні етапи роботи включали:
1. **Збір та підготовка даних**: Використання 30000 зразків аудіофайлів та отримання MFCC фіч для кожного аудіофайлу.
2. **Обробка даних**: Стандартизація фіч і зменшення вимірності за допомогою PCA.
3. **Навчання моделі**: Алгоритм Random Forest показав високу точність передбачення (~97%) на тестовій вибірці. 4. Тестування: Модель успішно передбачує вимовлені цифри в аудіофайлах.