Tonanalyse Code verstehen

Status	In Progress
Project	Praxis: Sichtung, Verstehen & Integration des bisherigen Programmcodes
Tags	

Zweck	Modelle/ Verfahren	Output
Segmentierung	pyannotate	Umwadelt eine Audiodatei in verschiedene Segmente, wobei jedem Segment ein Zeitabschnitt entspricht, in dem ein bestimmter Sprecher spricht
Geschlechtererkennungsprozess	TensorFlow 2	binäre Klassifikation als männlich oder weiblich sowie entsprechende Wahrscheinlichkeit
Audiotranskription	KI-Modells WhisperAI (Version „Large V2“) + Modell „Distilbert-base-uncased-emotion“	Text als Output: Transkript von Human voice
Sentimentanalyse	Modell „Distilbert-base-uncased-emotion“	„love“, „joy“, „surprise“, „sadness“, „anger“ und „fear“
Stimmungswechsel	Rollierendes Verfahren	Verschiedene Intervalle analysiert (z.B. 1-10, 5-15 Wörter), um Änderungen in der Stimmung besser zu erkennen; Intervall von 20 Wörtern zeigte beste Ergebnisse
Tonkomplexität	Python-Bibliothek acoustic_indices → Acoustic Complexity Index (ACI)	- Messung der Tonkomplexität für Korrelation zur Werbewirkung - Der ACI misst, wie stark die Intensität des Signals über verschiedene Frequenzbereiche variiert. Ein hoher ACI-Wert deutet auf ein komplexes Klangbild hin, das oft bei dynamischen, abwechslungsreichen Werbungen mit vielen Soundeffekten oder musikalischen Elementen zu finden ist.
Messung der Energie bzw. Stärke eines Audiosignals	Root-Mean-Square (RMS)	- Output: RMS-Wert - Amplitude und Energie des Signals werden dargestellt
Vergleich der Tonspuren bzgl. Dateigröße und Komplexität	Komprimierungsrate	Komprimierungsrate je Sekunde - Bestimmung der Originalgröße der WAV-Datei, anschließende Komprimierung in MP3 und Berechnung der Komprimierungsrate pro Sekunde, um Komplexität der Tonspur zu bewerten

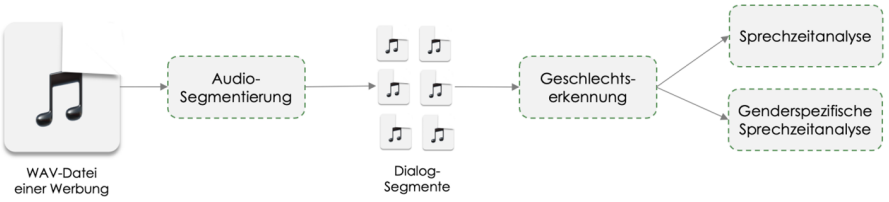

Abbildung 3.17: Prozess der Audioanalyse vom Segmentieren bis zur Analyse.

Aufbau von Ordnern

1. Acoustic_Indicies: Quellcode und manuelle Evaluation.
2. Geschlechtserkennung: Erstellte Audiosegmente (nicht zu finden) und manuelle Evaluation.
3. Stimmungsanalyse: WhisperAI-Dateien und manuelle Evaluation.
4. main_sound_recognition_FINAL: Umfassender Python-Code für die Tonanalyse.

1. Ordner: Acoustic Indices

> 📖 Hier werden Tonkomplexität untersucht, um herauszufinden, wie die Tonkomplexität die Wahrnehmung und Bewertung von Superbowl-Werbespots beeinflussen kann.
> Ziel: Kennzahlen wie der Acoustic Complexity Index (ACI), RMS-Energy und andere Indizes zur Bewertung der Tonkomplexität und zur Kategorisierung von Werbespots hinsichtlich akustischer Eigenschaften wie Lautstärke, Dynamik und Energie genutzt.

Datei *00 Manuelle Prüfung der Indices.ipynb* produziert *01 Ausgewählte Acoustic Indices.pdf* und *02 Maxima und Minima Acoustic Indices.pdf*

0. ***Manuelle Prüfung der Indices.ipynb***: Die Berechnung oder Auswertung verschiedener Indizes wie Acoustic Complexity, Diversity, Evenness, etc.
    a. Ziel: **relevantesten Kennzahlen** für die spätere Analyse der Werbespots zu identifizieren, indem es diese Kennzahlen mit dem Spektrogramm vergleicht.
1. ***Ausgewählte Acoustic Indices.pdf****:* Eine Sammlung von Audioanalysen für Superbowl-Werbespots dar. Es enthält detaillierte Werte für mehrere akustische Indizes pro Werbespot, darunter:
    - **Acoustic Complexity Index**
    - **Acoustic Diversity Index**
    - **Acoustic Evenness Index**
    - **RMS energy mean**
    - **Spectral centroid mean**
    - **Spectral Entropy**
    - **Temporal Entropy**
    - **Zero-Crossing Rate (ZCR) mean**
2. ***Maxima und Minima Acoustic Indices.pdf****:* Sie zeigt die Werbespots mit den höchsten und niedrigsten Werten für die jeweiligen Indizes, wie z. B. der höchste Acoustic Complexity Index bei AD0661 und der niedrigste bei AD0576. (nicht relevant für uns)
3. ***outputs.xlsx***: Diese Excel-Datei scheint die aggregierten Analyseergebnisse oder Tabellen der berechneten Werte aus den Audioprüfungen zu enthalten. Sie dürfte Daten für mehrere Werbespots enthalten, einschließlich der berechneten Indizes wie Complexity und Diversity, und dient als kompaktes Format für die Auswertung und den Vergleich der Ergebnisse.

2. Ordner: Geschlechtserkennung

1. *Audio_Gender_Prediction_Evaluation.xlsx* : Berechnung von Recall, Accuracy, Precision und F1-Score der Geschlechtsidentifikation für die Segmente der jeweiligen Werbungen (Manuelle Überprüfung).

3. Ordner: Stimmungsanalyse

1. *Modellevaluation_Stimmungsanalyse:* Modellevaluation_Stimmungsanalyse: Hier werden drei Excel-Dateien erstellt, die die Transkription in unterschiedliche Intervalllängen (15, 20, 25) unterteilen, um zu bestimmen, welche Intervalllänge die Stimmungswechsel am besten vorhersagt. Die Genauigkeit (Accuracy) ist bei einer Intervalllänge von 20 am höchsten. (nicht relevant für uns, da dieser nur eine Vorbereitung für Modellaufbau ist)
2. *Modellevaluation WhisperAI.xlsx :* Erstellte Transkription für jeden Werbungen werden hier gezeigt, diese werden geprüft ob die Inhalte richtig erkannte wurde → Korrektheit in % und jeweilige Begründung angegeben

---

4. main_sound_recognition_FINAL.ipynb

This script includes 3 different models which analyze different parts of audio in super bowl ads.

1. *Gender specific speaking time* (and durations of speaking parts)
2. *Emotion recognition from Transcription* (uses only transcription from WhisperAI for analysis)
3. *Acoustic Indizes* (many different indicators like min/max_energy, db and tempo)

1. Gender specific speaking time

Note: Um **pyannote** verwenden zu können, muss man sich einloggen und die Nutzungsbedingungen akzeptieren, um Zugang zu den erforderlichen Dateien und Inhalten zu erhalten.

- accept the user conditions on hf.co/pyannote/speaker-diarization-3.1
- accept the user conditions on hf.co/pyannote/segmentation-3.0
- login using `notebook_login` below: # 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

Functions definition at beginnnig

Utils: load data, split data and created simple neuron model with 5 hidden layers from 256 units to 64 units

- `is_silent(snd_data)` : Diese Funktion prüft, ob die Lautstärke der Audiodaten (snd_data) unterhalb des festgelegten Schwellenwerts (THRESHOLD) liegt.
- `normalize(snd_data)` : **normalisiert** die Lautstärke der Audiodaten, sodass die maximale Amplitude auf MAXIMUM (16384) gesetzt wird.
- `trim(snd_data):` entfernt **stille Bereiche** am Anfang und Ende der Audiodaten
- add_silence(snd_data, seconds): Diese Funktion fügt den Audiodaten **Stille** am Anfang und Ende hinzu. Die Länge der hinzugefügten Stille wird durch “seconds” in Sekunden angegeben.
- `extract_feature(file_name, **kwargs):` Extract feature from audio file
  - MFCC (mfcc)
  - Chroma (chroma)
  - MEL Spectrogram Frequency (mel)
  - Contrast (contrast)
  - Tonnetz (tonnetz)
- `audio_splitter(input_folder, output_base_folder, output_base_excel_folder):`

  Prozess:
  - Laden des Modells: Erstelle und lade ein vortrainiertes Modell zur Sprecheranalyse.
  - Iteriere durch Jahresordner (2013-2022):
    - Lade jede `.wav` Datei aus den Unterordnern.
    - Segmentiere Audio: Teile die Audiodateien in Segmente basierend auf Sprecher- und Geschlechtsanalyse (mittels Pipeline und Modell).
    - Speichere die Segmenteigenschaften (Start/Ende, Sprecher, Geschlecht) in einer Liste.

  Ausgabe:

- Erstellung von Excel-Dateien:
  - Speichern der Segmentdaten: Schreibe Ergebnisse wie `Start`, `Ende`, `Sprecher`, `Geschlecht` in Excel-Dateien.
  - Excel-Verwaltung: Entweder bestehende Datei erweitern oder eine neue Datei für jedes Jahr und jede Datei anlegen.

Durchführung → Total speaking time (Conclusion):

Prozess:

- Iteration durch Jahresordner (2013-2022):
  - Verarbeitung jeder `.wav` Datei: Lade die Datei und berechne die Gesamtdauer des Audios.
  - Excel-Laden oder Erstellen: Überprüfe, ob ein Excel-Sheet ('Gender_speaking_time') existiert:
    - Falls vorhanden, lade die Daten.
    - Falls nicht, erstelle ein neues leeres DataFrame.
  - Berechnungen durchführen:
    - Gesamte Sprechzeit im Audio und den Anteil der Zeit berechnen.
    - Prozentuale Sprechzeiten für männliche und weibliche Sprecher berechnen.

Ausgabe:

- Excel-Verwaltung:
  - Kombiniere bestehende Daten mit neuen Berechnungen.
  - Schreibe die kombinierten Daten in das Excel-Sheet 'Gender_speaking_time'. Speichern der geänderten Datei.

2. Emotion Recognition from Transcription

1. **Whisper AI für Transkription**
   - Verwende das Whisper-Modell, um die Audiodateien zu **transkribieren**.
   - Speichere die Transkription in einer **.txt-Datei**.
2. **Emotion recognition**
   a. `classify_emotion_from_file(file_path):` Diese Funktion dient zur Analyse eines gesamten Textdokuments, das aus einer Datei eingelesen wird, und zur Klassifikation von Emotionen für den gesamten Text sowie für kleinere Segmente.
   - Prozess:
     - Die Funktion liest eine Textdatei (file_path) ein.
     - Es wird überprüft, ob der Text länger als 320 Wörter ist:
       - Wenn ja, wird der Text in zwei Teile aufgeteilt und die Emotionen werden separat klassifiziert.
     - Anschließend wird der Text in kleinere 20-Wort-Segmente unterteilt, und die Emotionen werden für jedes Segment klassifiziert.
     - Für die verbleibenden Wörter (falls vorhanden) wird ebenfalls eine Emotionserkennung durchgeführt.
   - Output:
     - Eine Liste von Ergebnissen für jedes Segment, die Folgendes enthält:

       'AD-Number', 'Transcription', 'Word range', 'Emotion', 'Probability'

   b. `extract_emotions_and_scores(text, predictions, ad_number):` Diese Funktion dient zur Feinabstimmung der Emotionserkennung für einen gegebenen Text. Sie arbeitet mit vorher erstellten Emotionsergebnissen (predictions) und führt eine weitergehende Segmentanalyse durch.
   - Prozess:

    - Der Text wird zunächst in 20-Wort-Segmente aufgeteilt.
      Für jedes Segment werden Emotionen erneut klassifiziert, aber die Funktion prüft zuerst, ob das ursprüngliche Ergebnis (predictions) eine hohe Wahrscheinlichkeit für eine bestimmte Emotion (> 0,8) hat.
    - Zusätzlich speichert die Funktion die Ergebnisse der ersten Emotionserkennung und geht dann zur Segmentierung über.
  - Output:
    - Eine Liste von Emotionsergebnissen für jedes Segment, die Folgendes enthält:

      'AD-Number', 'Transcription', 'Word range', 'Emotion', 'Probability'

📖
- "Obwohl der Text auch in den Intervallen in den meisten Fällen zwar richtig klassifiziert wird, werden die Stimmungswechsel der Werbung nicht immer korrekt wiedergegeben. Grund hierfür ist vorwiegend der Unterschied zwischen der Audiotranskription und der eigentlichen Werbung. Trotz der guten durchschnittlichen „Umwandlung" von 77,67%, lässt sich sagen, dass Aussagen über Stimmungswechsel nur schwer zu treffen sind, da in den meisten Werbungen bzw. Analysen Fehler enthalten sind. Die Zahl lässt sich nämlich so interpretieren, dass durchschnittlich 77,67% einer Werbung korrekt klassifiziert werden. Insgesamt wurden nur 6 der 30 Werbungen vollständig korrekt erkannt, was einer Genauigkeit von 20% entspricht."
- Wenn Musik im Werbung auftaucht, hat das Modell eine bessere Ergebnis
  (Quelle: Projektbericht)

2.1 Combination of emotion from image and audio

1. `classify_and_check(predictions)`

- **Überprüfung der Emotionsergebnisse** auf eine Mindestwahrscheinlichkeit und eine Zuordnung von "neutral", falls keine Emotion eine hohe Wahrscheinlichkeit hat.

2. `classify_emotion()`

- nimmt einen **Text** als Eingabe und führt die tatsächliche **Emotionserkennung** durch.

3. `process_json_file(json_file_path, excel_file_path)`:

   Prozess:

   - Initialisierung:
     - Lade eine vorhandene Excel-Tabelle (`Transcription_and_Mood`) als DataFrame.
     - Lade die JSON-Datei (`json_file_path`) mit allen Segmenten und Textdaten.
   - Datenextraktion:
     - Extrahiere Metadaten: Name der Datei (`ad_number`) und der gesamte Text (`full_text`).
     - Iteriere durch Segmente im JSON:
       - Für jedes Segment werden Informationen wie ID, Start, Ende, und Transkription extrahiert.
       - Emotionserkennung wird auf die Transkription angewendet, und die Ergebnisse werden gespeichert.
   - Emotionserkennung für gesamten Text:
     - Füge eine erste Zeile hinzu, die den gesamten Text (`full_text`) darstellt.
     - Wenn der Text länger als 320 Wörter ist, wird die Emotion aus der vorhandenen Tabelle kopiert.
     - Ansonsten wird eine neue Emotionserkennung durchgeführt und das Ergebnis hinzugefügt.
   - Zusammenführung mit vorhandenen Daten:
     - Ein neuer DataFrame (`new_data_df`) wird mit den neu extrahierten Daten erstellt.

    - Ein leerer Spaltenbereich wird in den vorhandenen DataFrame eingefügt, und die neuen Daten werden daran angehängt.

Ausgabe:

- Excel-Aktualisierung: Schreibe die aktualisierten Daten (inklusive der neuen Segmente und deren Emotionen) in das "Transcription_and_Mood"-Sheet der Excel-Datei.

- Die Funktion kombiniert **JSON-Daten** und **Excel-Daten**.
- Sie führt eine **Emotionserkennung** durch und speichert die **aktualisierten Ergebnisse** in einer Excel-Tabelle.
- Die Funktion kombiniert **JSON-Daten** und **Excel-Daten**.
- Sie führt eine **Emotionserkennung** durch und speichert die **aktualisierten Ergebnisse** in einer Excel-Tabelle.
- Die Funktion kombiniert **JSON-Daten** und **Excel-Daten**.
- Sie führt eine **Emotionserkennung** durch und speichert die **aktualisierten Ergebnisse** in einer Excel-Tabelle.

2.2 Analysis Emotion Image & Audio

`comparison_emotions_image_audio()` : vergleicht Emotionen aus der Bildanalyse mit Emotionen aus der Audioanalyse für jedes Segment eines Werbevideos. Sie bestimmt die **Übereinstimmung zwischen den Emotionen**, speichert die Ergebnisse und fügt eine **statistische Zusammenfassung** hinzu, die zeigt, wie ähnlich die Emotionserkennung aus den beiden Modalitäten (Bild und Audio) ist.

- Stimmungswechsel in der Audiotranskription ist effektiv
- Stimmungswechsel (Abgleich von Audio und Bilder, ob sie im gleichen Stimmung ist) ist schwer mit diesem Modell zu erkennen:
    - “Generell lässt sich feststellen, dass die Stimmung in der Werbung durch eine vorherige Audi otranskription effektiv erkannt werden kann. Eventuelle Stimmungswechsel sind jedoch bei den verschiedenen Verfahren deutlich fehleranfälliger, da die Abweichung von Audiotranskription zu tatsächlicher Werbung in solchen Fällen zu groß ist. Durch die isolierte Betrachtung einzelner Intervalle fehlt der Kontext zu anderen Zeitabschnitten sowie zu weiteren Parametern wie Stimmlage, Umgebung, Musik usw. Die Bestimmung der einzelnen Intervalle ist daher zwar häufig korrekt, gibt allerdings nicht immer die richtigen Stimmungswechsel der Werbung wie der.”

3. Acoustic Indices

`calculate_additional_values(y)` **:** Gibt die Werte `duration`, `tempo`, `avg_db`, `min_db`, `max_db`, `max_db_value` zurück, um zusätzliche akustische Merkmale zu charakterisieren.

`compress_wav_to_mp3(input_wav_path, output_mp3_path, bitrate='192k'):` Komprimiert eine **.wav**Datei in eine **.mp3** Datei

Main:
Der Code führt eine umfassende
Analyse von Audiodateien durch, wobei **akustische Indizes** berechnet, Audiodateien **vorverarbeitet** und die Ergebnisse in **Excel** gespeichert werden.

- Der gesamte Prozess wird entweder in einer einzigen Datei (`single_output_file`) →wenn `single_output_file` = true
- oder in mehreren individuellen Dateien (`not single_output_file`) ausgegeben. → wenn `single_output_file` = false
- Prozess

1. **Initialisierung und Argumente**
   - Einige der Argumente (`config_file`, `audio_dir`, `output_csv_file`) werden definiert, aber in Kommentaren hinterlegt.
   - Variablen wie `single_output_file` werden verwendet, um zu bestimmen, ob die Ausgabe in einer oder mehreren Dateien erfolgt.
2. **Iterieren über Audiodateien**
   - Iteriere über eine Liste von Audiodateipfaden (`all_audio_file_path`).
   - Für jede Datei:
     - **Signal einlesen**: Lade die Audiodatei und extrahiere ihre Daten.
3. **Vorverarbeitung**
   - Führt **High-Pass-Filter** auf dem Audiosignal aus.
     - Unterstützt zwei Filtertypen: **Butterworth** und **Fenster-Sinc**.
     - Jeder Filtertyp wird entsprechend konfiguriert, z. B. durch die Berechnung der Filterkoeffizienten und das Anwenden des Filters auf das Signal.
4. **Berechnung von akustischen Indizes**
   - Für jede Audiodatei werden verschiedene **akustische Indizes** berechnet, darunter:
     - **Acoustic Complexity Index**, **Bio-acoustic Index**, **Spectral Entropy**, **RMS energy**, etc.
     - Einige Indizes werden auf **rauschreduzierte Spektrogramme** angewendet, z. B. **Bio_acoustic_Index_NR**.
     - Die Berechnungen werden durch Methoden durchgeführt, die basierend auf der Konfiguration (`ci`) dynamisch aufgerufen werden.
5. **Berechnung von zusätzlichen akustischen Eigenschaften**
   - **Dauer**, **Tempo**, und verschiedene **dB-Werte** (Durchschnitt, Minimum, Maximum) werden mit der Funktion `calculate_additional_values()` berechnet.
6. **Komprimierung der Audiodatei**
   - Jede `.wav` Datei wird in eine **.mp3**Datei komprimiert.
   - Das **Kompressionsverhältnis** und das **Kompressionsverhältnis pro Sekunde** werden berechnet.
   - Die komprimierte Datei wird anschließend gelöscht, um Speicherplatz freizugeben.
7. **Erstellen und Speichern der Ergebnisse**
   - Die berechneten Indizes und zusätzlichen Werte werden in ein **Dictionary** (`file_data`) gespeichert.
   - Diese Ergebnisse werden entweder in:
     - Eine **einzelne Excel-Datei** geschrieben (`single_output_file = True`).
     - **Individuelle Excel-Dateien** für jede Datei geschrieben (`not single_output_file`).
   - Die Excel-Dateien enthalten ein **Sheet namens "Acoustic_Indices"**.
   - Bestehende Dateien werden entweder **überschrieben** oder **neu erstellt**, wenn sie beschädigt oder nicht vorhanden sind.