Dokumentation der Portfolioprüfung

im Fach

Data Exploration Project

Wirtschaftsinformatik – Data Science

Gruppenname:	AirTelligence
Studenten:	Philipp Becht (9443009)
	Yannik Hubrich (2249266)
	Simon Wrigg (5874903)
Kurs:	WWI19DSB
Dozent:	Simon Poll

Inhalt

1. Thema und Motivation

Im Rahmen der Vorlesung Data Exploration Project hat sich die Gruppe AirTelligence das Thema „Speech Emotion Recognition“ ausgesucht. Ziel des Projektes soll es sein, interessante Erkenntnisse aus den vorliegenden Audio-Dateien abzuleiten und die zugrundeliegenden Emotionen aus den Daten zu klassifizieren.

Der US-amerikanische Forscher Albert Mehrabian fand durch ein Experiment heraus, dass die zwischenmenschliche Kommunikation nur kaum von dem gesprochenen Wort, sondern viel mehr von der Art und Weise wie wir miteinander kommunizieren beeinflusst wird. Aus seinem Experiment leitete Mehrabian die 7-38-55 Regel ab, welche besagt, dass der tatsächlich gesprochene Inhalt nur 7% der wörtlichen Rede ausmacht, und dem Tonfall mit 38%, sowie der Körpersprache mit 55% eine deutlich höhere Bedeutung zukommt.

Diese Erkenntnis war ausschlaggebend für die Themenwahl der Gruppe. Mit der Analyse der zugrundeliegenden Emotionen soll ein tieferes Verständnis unserer menschlichen Kommunikation erlangt werden. Heutzutage finden Emotions-Detektoren beispielsweise bei Vorstellungsgesprächen, Verhören, Lügendetektoren oder auch bei jeglichen Mensch-Maschine-Schnittstellen wie zum Beispiel bei Robotern eine Verwendung.

2. Related Work

Der Forschungsbereich der Sprach-Emotionserkennung ist seit mehreren Jahren sehr aktiv. So werden jedes Jahr mehrere Duzend Artikel in Fachzeitschriften und Konferenzbänden veröffentlicht.[1] Diese Artikel setzen meist einen der drei folgenden Schwerpunkte: Das Aufbauen von Datenbanken mit werthaltigen und nützlichen Audio-Dateien, die Suche nach geeigneten Sprachmerkmalen in den Audio-Dateien oder verschiedene Klassifizierungstechniken, die zur Maximierung der Erkennungsgenauigkeit bei der Spracherkennung dienen.[2]

Die Forschungen in diesem Gebiet haben bereits in Bezug auf die Themen Merkmalsextraktion und Klassifikationsalgorithmen einige „Best Practices“ hervorgebracht, die bei der Projektdurchführung ebenfalls berücksichtigt worden sind.[3]

---

[1] Springer, Speech emotion recognition research: an analysis of research focus.
[2] Springer, Speech emotion recognition research: an analysis of research focus.
[3] Abbaschian, B., Deep Learning Techniques for Speech Emotion Recognition from Databases to Models.

3. Verwendete Technologien und Bibliotheken

Im Folgenden sollen die wichtigsten verwendeten Bibliotheken vorgestellt werden, die bei der Analyse der Daten und dem Training benutzt worden sind.

Die Analyse von Audiodateien ist vor allem durch die Python-Packages „wave“ und „librosa“ vorgenommen worden, zumal das „wave“-Modul eine komfortable Schnittstelle zum *.wav-Audioformat bietet. Dabei lassen sich Soundfiles analysieren und auslesen und mithilfe von Funktionen in numerische Werte umwandeln.[4] Das Python-Package „librosa“ kann ebenfalls zur Musik- und Audioanalyse verwendet werden und bietet die notwendigen Bausteine, um Speech Emotion Recogniton Systeme zu bauen.[5]

Neben den sehr spezifischen Bibliotheken für die Analyse von Audiodateien, sind allerdings auch „klassische“ Python-Bibliotheken herangezogen worden.

So zum Beispiel sind für die Vorbereitung der Daten die Module „glob“ und „os“ verwendet worden. Mithilfe von „glob“ lassen sich alle zutreffenden Pfadnamen finden, die einem bestimmten Muster entsprechen. Das Modul „os“ stellt verschiedene betriebssystemabhängige Funktionalitäten zur Verfügung, wie beispielsweise das Lesen und Schreiben von Dateien, oder das Verändern von Pfadnamen.[6]

Die Werkzeuge der Bibliothek „pandas“ sind benutzt worden, um die eigens erstellte *emotion_dataset.csv* – Datei zu verarbeiten, welche numerische Werte für verschiedene Features der Soundfiles enthält. Für die Erstellung der Plots ist die Bibliothek „matplotlib“ herangezogen worden.

Die Anwendung von Lernalgorithmen auf den Daten ist hauptsächlich durch „Scikit-Learn“ und „Keras“ realisiert worden. „Scikit-Learn“ ist eine Bibliothek für maschinelles Lernen in der Programmiersprache Python. Sie enthält Algorithmen für supervised und unsupervised Learning und stellt hierfür einfache und effektive Tools zur Verfügung. Keras ist eine Bibliothek für neuronale Netze die Tensorflow als Backend verwendet.

[4] Wave, Python Documentation.
[5] Librosa, Python Documentation.
[6] Os, Python Documentation.

4. Präsentation der Ergebnisse

Die Herangehensweise der Gruppe lässt sich prinzipiell in zwei Phasen unterteilen. In der ersten Phase ging es primär darum, die vorliegenden Daten zu verstehen und einen Überblick über mögliche Merkmale zu erhalten, die Aufschluss über den vorliegenden Gefühlszustand geben. Die zweite Phase beschäftigte sich anschließend mit der Anwendung von passenden Lernalgorithmen, welche eine bestmögliche Klassifizierung auf den Testdaten erreichen sollen.

Zu Beginn ist eine Correlation Matrix erstellt worden, mit der untersucht werden sollte, ob bestimmte Features direkt mit dem Label zusammen korrelieren. Leider musste festgestellt werden, dass keine aussagekräftigen Beziehungen bestehen und somit die Features nicht direkt eingegrenzt werden konnten. In der Konsequenz wurden viele verschiedene Features im weiteren Projektvorgehen genauer untersucht.

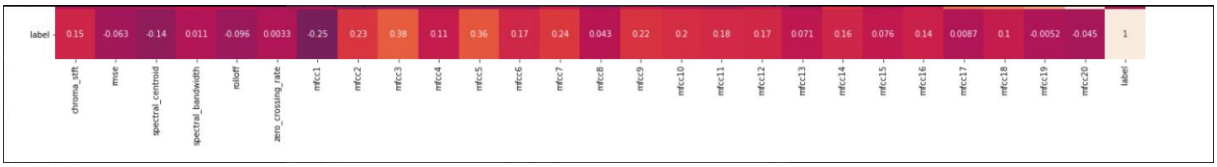

Abb. 1 – Correlation Matrix Ausschnitt – Features in Bezug auf Label

Schließlich konnte mit der Visualisierung der Daten begonnen werden. Die verwendeten Python-Bibliotheken stellen nützliche Funktionen bereit, um beispielsweise Spektrogramme und Soundwaves auf den Daten zu plotten. Spektrogramme sind visuelle Darstellungen des Spektrums der Frequenzen von Audiosignalen und zeigen, wie sie sich mit der Zeit verändern. Darüber hinaus geben sie Auskunft über die Emotionslage der SprecherInnen. Soundwave Visualisierungen zeigen hingegen die Lautstärke der Audiodatei über einen bestimmten Zeitraum.

Die Unterschiede der Wellenform von besonders verschiedenen Emotionen waren direkt ersichtlich. Bei genauerer Analyse der akustischen Energie in den Spektrogrammen kann zusätzlich festgestellt werden, dass der Frequenzbereich oberhalb von 4000hz bei der Emotion Angry stärker ausgeprägt ist als bei der Emotion Sadness.

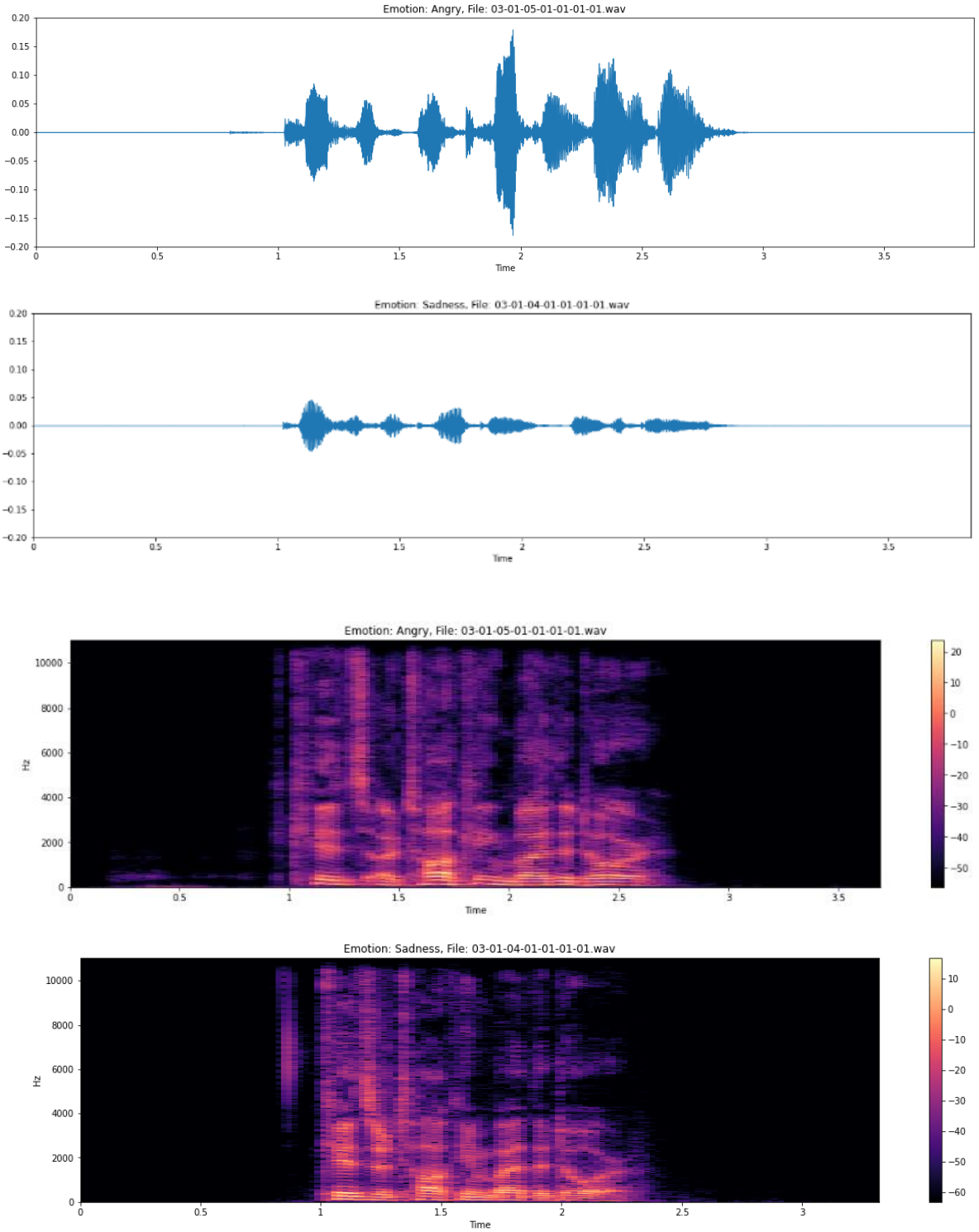

Abb.2-5: Soundwave- & Spektrogramm-Plots für die Emotionen Angry und Sadness

Darüber hinaus sind weitere Audio-Features analysiert und durch entsprechende Plots visualisiert worden. Für die weiteren Analysen sind beispielsweise die Features „Zero Crossing Rate“, also die Anzahl der Vorzeichenwechsel entlang eines Signals, und die „Spectral Centroids“, die angeben, wo sich der Massenschwerpunkt eines Klangs befindet, herangezogen worden.

Ebenfalls ist eine Analyse des MFCC Features (Mel-Frequency Cepstral Coefficients) durchgeführt worden. Es beschreibt das Frequenzspektrum zusammen mit der wahrgenommenen Tonhöhe und eignet sich somit ebenfalls für die Untersuchungen der Stimmdaten.

Weitere Visualisierungen der Daten sind der beigefügten Abschlusspräsentation beziehungsweise des Outputs unseres Quellcodes zu entnehmen.

Um letztendlich geeignete Lernalgorithmen auf den Daten anwenden zu können, ist es zunächst notwendig gewesen, die Audio-Dateien in numerische Daten zu transformieren. Teilweise handelt es sich bei den Werten um Mittelwerte oder andernfalls um die tatsächlichen Werte zu einem bestimmten Zeitraum respektive Zeitpunkt. Aus diesen Attributen ist eine .csv-Datei generiert worden, die insgesamt 26 Features, sowie ein entsprechendes Label für jede Instanz enthält. Eine Instanz stellt in diesem Fall eine Audio-Datei dar. Diese .csv-Datei wurde mithilfe einer scikit-learn Funktion in einen Trainings- und Testdatensatz aufgesplittet.

Für das Training ist ein Multi Layer Perceptron Classifier gewählt worden. Hierfür sind die Initialisierungsparameter auf den Use-Case angepasst worden. Somit ist eine logistische Aktivierungsfunktion und „Adam“ als Gradient Weights Optimizer gewählt worden. Durch die Angabe der Toleranz wird überprüft, ob sich der Loss innerhalb der letzten 1000 Iterationen nicht mehr um einen gewissen (sehr kleinen) Wert verändert hat. In diesem Fall, würde das Training abbrechen, da man davon ausgehen könne, dass sich die Qualität des Modells nicht mehr signifikant verbessert.

Nach dem ersten Trainingsdurchlauf ist folgendes Modell trainiert worden:

```
Training loss did not improve more than tol=0.000000 for 1000 consecutive epochs. Stopping.
Training finished after: 1123.4 seconds
Number of iterations: 74326
Number of iterations no change: 1000
Model accuracy: 60.26%
              precision    recall  f1-score   support

           0       0.76      0.72      0.74        65
           1       0.59      0.57      0.58        46
           2       0.67      0.59      0.62        51
           3       0.48      0.60      0.53        47
           4       0.62      0.53      0.57        47
           5       0.49      0.57      0.53        46

    accuracy                           0.60       302
   macro avg       0.60      0.59      0.60       302
weighted avg       0.61      0.60      0.61       302
```

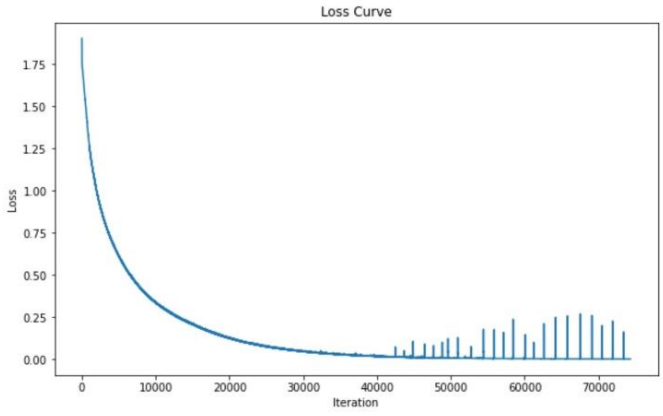

Abb. 6: Auswertung des ersten Trainings (Kennzahlen und Loss-Curve)

Die durch das Modell erreichte Accuracy nach dem initialen Training beträgt 60,3%. Um das Modell zu verbessern, sind über mehrere Anläufe hinweg die Hyperparameter immer wieder angepasst worden. Das finale Training konnte eine Accuracy von 65% vorweisen.

```
Training loss did not improve more than tol=0.000000 for 1000 consecutive epochs. Stopping.
Training finished after: 10.2 minutes
Number of iterations: 29137
Number of iterations no change: 1000
Model accuracy: 65.23%
              precision    recall  f1-score   support

           0       0.81      0.71      0.75        65
           1       0.61      0.74      0.67        46
           2       0.65      0.55      0.60        51
           3       0.49      0.55      0.52        47
           4       0.76      0.68      0.72        47
           5       0.61      0.67      0.64        46

    accuracy                           0.65       302
   macro avg       0.65      0.65      0.65       302
weighted avg       0.66      0.65      0.65       302
```

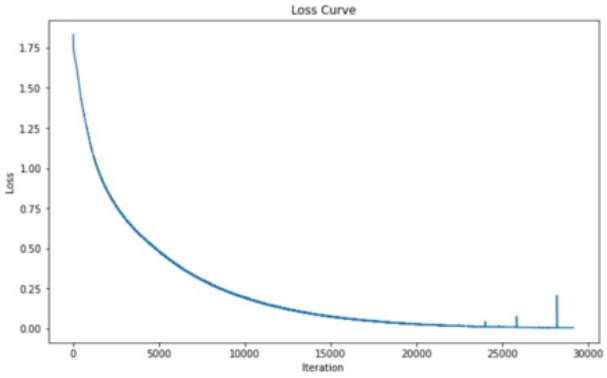

Abb.7: Auswertung des finalen Trainings (Kennzahlen & Loss-Curve)

Mithilfe der Library Keras ist ein weiterer MLP-Classifier initialisiert worden, der nicht nur eine deutlich verkürzte Trainingszeit vorweisen konnte, sondern auch weniger Interationen benötigte, um eine letztendliche Accuracy von 69% zu erreichen.

Neben den zwei MLP Classifiern ist auch ein K-Nearest-Neighbors Algorithmus trainiert worden. Bei diesem Modell ist allerdings anzunehmen, dass die Klassifikation einer willkürlichen Klassifikation entspricht. Somit wurde dieses Modell verworfen.

5. Kritische Bewertung der Ergebnisse

Die anfänglichen Befürchtungen der Gruppe, dass die Datenqualität und -quantität nicht ausreichend seien, sind nicht eingetreten. Grund für diese Annahme sind die aussagekräftigen Visualisierungen, in denen die Unterschiede der Emotionen mit bloßem Auge erkennbar waren. Zwar muss festgestellt werden, dass unsere Lernalgorithmen eine maximale Accuracy von 69% aufweisen, was auf den ersten Blick als nicht ausreichend gewertet werden müsste, jedoch ist Recherchen nach zur Folge dieser Wert durchaus zufriedenstellend. Selbstverständlich müsste abhängig vom Use Case entschieden werden, ob die Güte des Modells ausreicht.

Als Herausforderung kann durchaus die Wahl des richtigen Lernalgorithmus und die jeweils richtige Einstellung der benötigten Hyperparameter angesehen werden. Zweiteres hat nämlich entscheidenden Einfluss auf die Güte des trainierten Modells. Es hat mehrere Anläufe mit unterschiedlichsten Konfigurationen der Hyperparameter gebraucht, bis ein zufriedenstellendes Modell trainiert wurde.

In den ersten Versuchen sind einige Klassen komplett unterrepräsentiert gewesen, während andere Klassen eine Precision von 95% vorweisen konnten. Grund hierfür war wohl eine unvorteilhafte Konfiguration des MLP Classifiers beziehungsweise eine zu geringe Anzahl an Iterationen. Nach weiteren Versuchen und einer günstigeren Initialisierung des MLP Classifiers sind bereits deutlich bessere Ergebnisse erreicht worden. Von diesem Punkt an ging es eher um das „Finetuning“ des Modells, um eine bestmögliche Klassifizierung auf den Daten zu erhalten.

Angesichts der erreichten Accuracy hätte man davon ausgehen können, dass die trainierten Modelle unter Underfitting leiden. Trotz mehrerer Versuche, die Accuracy und weitere Metriken, die Aufschluss über die Güte des Modells geben, zu steigern, ist eine signifikante Verbesserung der Modelle nicht gelungen.

Ein nicht zu vernachlässigender Punkt sei ebenfalls die kritische Auseinandersetzung mit dem Datensatz. Aufgrund der Entscheidung, zwei unterschiedliche Datensätze in zwei unterschiedlichen Sprachen zu verwenden, muss auf der einen Seite davon ausgegangen werden, dass das Modell robuster wird, aber auf der anderen Seite könnte dies auch mit einer Verschlechterung der Accuracy einher geht. Dies liegt zum einen an den unterschiedlichen Sprachen und zum anderen an dem divergenten Aufbau der Audiodateien selbst.

Anzumerken ist ebenfalls, dass die Trainingsdurchläufe mit teilweise sehr langen Laufzeiten verbunden waren und sehr ressourcenaufwändig waren.

6. Anmerkungen zum Quellcode im Anhang

Der hergestellte Code ist in der Entwicklungsumgebung „Google Colab“ geschrieben. https://colab.research.google.com/drive/15Hw7jYDc8pCHqjqqdPHJQ1ZnQxTsNHy7?hl=de#scrollTo=MnJBno6N7X_c

Um den Code auszuführen, müssen alle Zellen einzeln ausgeführt werden. Dies kann durch den „Play“-Button in der linken, oberen Ecke einer jeden Codezelle durchgeführt werden. Alle benötigten Dependencies und Packages sind im Code integriert.

Der Datensatz ist lokal in der Entwicklungsumgebung in dem Ordner „Content“ gespeichert.

7. Quellen

Abbaschian, B	Deep Learning Techniques for Speech Emotion Recognition from Databases to Models, Februar 2021
Python Documentation	https://docs.python.org/3/library/os.html https://docs.python.org/3/library/wave.html https://librosa.org/doc/latest/index.html
Springer	Speech emotion recognition research: an analysis of research focus, Januar 2018