



Wirtschaftsinformatik - Data Science

Fallstudie 3. Semester

Duale Hochschule Baden-Württemberg Mannheim

Analgebank WBHD - Face Bank

Verfasser/in & Matrikelnummer:	Florian Frey	7199749
	Frederick Neugebauer	4521985
	Oliver Hinkel	8636405
	Antonino Pilorio	1035820
	Mya-Melissa Jahic	2774085
Kurs:	WWI 20 DSB	
Dozent:	Benjamin Jung	
Studiengangsleiter:	Bernhard Drabant	
Bearbeitungszeitraum:	10.12.2021 – 28.02.2022	

Inhaltsverzeichnis

Einführung	3
Aufgabenstellung - Entwicklerseite	3
Aufgabenstellung - Unternehmensseite	3
Vorstellung fiktives Unternehmen	3
Produktportfolio	5
Wissenschaftliche Analyse	5
SWOT-Analyse	5
Projektablauf	7
Verwaltung	7
Meetings	8
Systementwürfe	9
Ereignis-Reaktionsmodell	9
Entscheidungstabellen	10
Aktivitätsdiagramm	11
Use-Case Schablone	12
Marketing	13
Konzept	13
Face Recognition	15
Funktionsweise	15
Deepface	15
Testing mit Datensätzen	18
Probleme	20
Lösungsversuche	21
Analysevorgang in Microsoft	22
Azure Face API	22
Power Automate Flow	22
Power App	23
Fazit und kritische Reflektion	25

Einführung

Aufgabenstellung - Entwicklerseite

Die Aufgabe war es eine App zu entwickeln, die den Kunden einer Bank Produkte vorschlagen soll. Dabei sollen die Produkte an den Emotionen des Nutzers abgestimmt sein, indem der Kunde mithilfe der App ein Bild von seinem Gesicht aufnimmt. Die App soll dabei die Emotionen Freude, Ekel, Wut, Trauer, Langeweile und Angst unterstützen sowie auch zwischen den Altersgruppen alt, Erwachsener und jugendlich unterscheiden.

Neben den Funktionen der App, sollte sie natürlich ein ansprechendes User Interface haben. Somit kann die Aufmerksamkeit geweckt werden und neue Zielgruppen lassen sich erschließen.

Aufgabenstellung - Unternehmensseite

Aus Sicht des Unternehmens, welches fiktiv erstellt werden sollte, erfolgten verschiedene Systementwürfe wie beispielsweise ein Ereignis-Reaktionsmodell, eine Entscheidungstabelle oder auch eine Use-Case-Schablone. Somit kann die Funktion einzelner Bestandteile hergeleitet werden.

Das Projektvorgehen innerhalb des Teams musste ebenso festgelegt und dokumentiert werden. Dazu gehören die Arbeitsweise, die Projektrollen, aber auch ein Ausgearbeitetes SLA.

Damit die Lage innerhalb und außerhalb des Unternehmens klar gemacht werden konnte, wurden wirtschaftliche Analysen erstellt.

Vorstellung fiktives Unternehmen

Um das Projekt umzusetzen, sollte sich die Gruppe ein Unternehmen ausdenken, mit dem sie die App entwickelt. Es wurde entschieden kein Software-Unternehmen zu sein, welches Anwendungen für Kunden entwickelt. Stattdessen ist man selbst eine Bank, die Produktinnovationen in Eigenregie entwickelt. Das bietet auch im Bereich Marketing deutlich mehr Möglichkeiten.

Das Unternehmen sollte auch schon lange am Markt tätig sein, um Seriosität und Erfahrung darzustellen. Gleichzeitig sollten neben der damit verbunden Tradition, die auch gerne hervorgehoben wird, auch Innovationen und moderne Zukunftstechnologien einen Platz in der Unternehmensstrategie finden. Dazu wurde das unten zu sehende Logo für die Bank entwickelt.



Die Anlagebank WBHD stellt dementsprechend eine alteingesessene Anlagebank dar, die seit 1864 in Deutschland tätig ist. Sie zählt sowohl Geschäft- als auch Privatkunden in ihrem Kundenstamm und kann sich guter Reputation erfreuen. Jedoch macht das digitale Zeitalter auch nicht vor einer Bank halt und so wurde vom Management entschieden, dass neueste Technologien genutzt werden sollen, um dem Umsatz als auch die Servicequalität zu verbessern, indem den Kunden möglichst individuelle Finanzprodukte angeboten werden sollen. Die IT-Abteilung der Anlagebank WBHD wurde aus diesem Grund beauftragt, innovative Lösungsansätze zu entwickeln und diese gegebenenfalls auch umzusetzen. Nach Beratungen und Diskussionen kam die IT-Abteilung auf die Idee eine neue Funktion in die bereits vorhandene Banking App einzubauen. Bei dieser Funktion sollen Kunden in der App ein Selfie machen können. Anhand dieses Selfies erkennt die App das Alter, sowie die aktuelle Emotion des Kunden und schlägt aufgrund dieser beiden Parameter ein Finanzprodukt vor. Die Idee der IT-Abteilung kam im Management gut an, was zur Entwicklung der App führte. Zur Unterstützung kamen neben einem Finanzprodukt Experten auch eine Marketingexpertin ins Projektteam dazu. Während die IT-Abteilung vorrangig für die App-Entwicklung und das System zuständig waren, entwickelte der

Finanzprodukte Experte das Konzept auf dem die Zuweisung der jeweiligen Produkte zu Altersklassen und Emotionen erfolgen soll. Die Marketingexpertin sorgte für eine ausgereifte Marketing Kampagne die sowohl in sozialen Medien unterwegs war als auch auf offener Straße Interessenten ansprechen sollte.

Produktportfolio

Das Produktportfolio der Anlagebank WBHD ist breit gefächert. Um ein möglichst breites Angebot zu gewährleisten, wurde sich für folgende Finanzprodukte entschieden:

Festgeld:	Geld wird bis ausgewählten Termin für eine gewisse Rendite angelegt
Kreditkarte:	Eine Karte um bargeldlos Waren und Dienstleistungen zu bezahlen, vor allem beim Online-Shopping beliebt
Rahmenkredit:	Es wird eigens ein Konto für einen Kredit mit variabler Verzinsung bereitgestellt
Sparbrief:	Eine Einmalanlage, mit der ein Geldbetrag für eine Laufzeit von 1-10 Jahren zu einem Zinssatz angelegt wird
Sparkonto:	Eine Geldanlage ohne feste Laufzeit. Das Guthaben wird i.d.R. variabel verzinst
ETFs:	Eine Möglichkeit Vermögen aufzubauen mit niedrigen Gebühren: Exchange Traded Funds
Bausparvertrag:	Ein Sparvertrag, um eine Immobilie zu finanzieren, bei Kauf oder Umbau von Wohnung und Haus
Aktien:	Anteile an einem Unternehmen kaufen und dabei Dividenden und Rendite abkassieren

Es wird dabei gleichzeitig beachtet, dass die Kunden die richtigen Produkte zugewiesen bekommen und dabei werden das Alter und die Emotion des Kunden eine Rolle spielen.

Wissenschaftliche Analyse

SWOT-Analyse

Um einen Überblick über die wirtschaftliche Situation zu erhalten, soll eine SWOT-Analyse Informationen liefern. Zu den Stärken des Unternehmens gehört

die gute Beratung durch Emotionserkennung durch die FaceBank App und das kompetente Team aus ausgezeichneten Experten und Expertinnen das dahintersteckt. Außerdem gibt es einen großen Vorsprung im Markt und viele Interessensgruppen.

Die Neuheit am Markt gilt zunächst als Schwäche, auch deshalb, weil dadurch bisher wenige historische Daten gesammelt konnten. Außerdem wirft die neuartige Technik einige Fragen auf, vor allem die Präzision der Software muss sich noch bewähren.

Zu unseren Chancen zählt, dass sobald die Technik beliebt wird ein starker Anstieg unseres Marktanteils zu erwarten ist, unterstützt dadurch, dass unsere Technik auf dem Stand der aktuellsten Trends ist.

Ebenfalls ist bemerkbar, dass die Menschen lockerer im Umgang mit Datenschutz werden, was auch hilft die gesammelten Daten an andere Banken mit hohem Profit zu verkaufen. Allgemein sind höhere Erlös Zahlen durch die optimale Beratung zu erwarten.

Natürlich hat jede Änderung auch einige Risiken. Die Angst vor Datenmissbrauch könnte dem Geschäftsmodell schaden. Außerdem kann das Geschäftsmodell infrage gestellt werden, wenn die gesamte Konkurrenz nicht nachzieht.

Die hohen Kosten die durch Fremdkapital gedeckt wurden könnten sich auch auf die Erlöse negativ auswirken und in der breiten Masse konnte die Software bisher nicht getestet werden.

Außerdem könnte die App schlicht und einfach unbeliebt sein, wenn die Kunden den Nutzen nicht sehen.

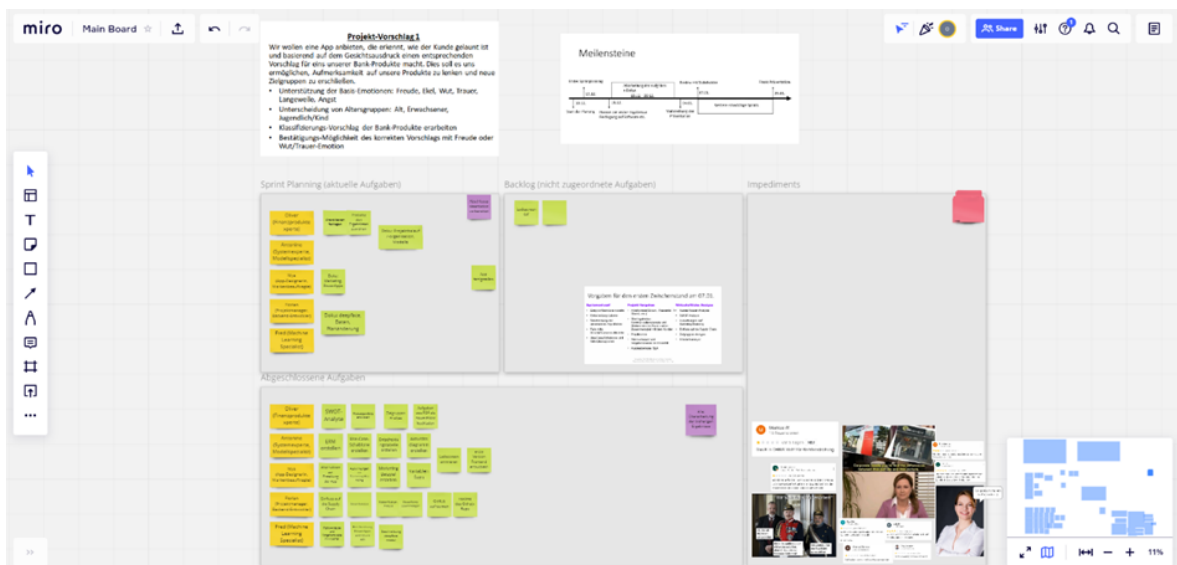
Das Ergebnis dieser Analyse ist, dass durch Tests der Software das Misstrauen gesenkt wird, die alleinige Position im Markt absichern, Ausfall der Dividendenausschüttung aufgrund der Investitionen erklären und dabei auch in Datenschutzbeauftragte investieren damit das Geschäftsmodell seriös und vertrauenswürdig aufgenommen wird.

Projektablauf

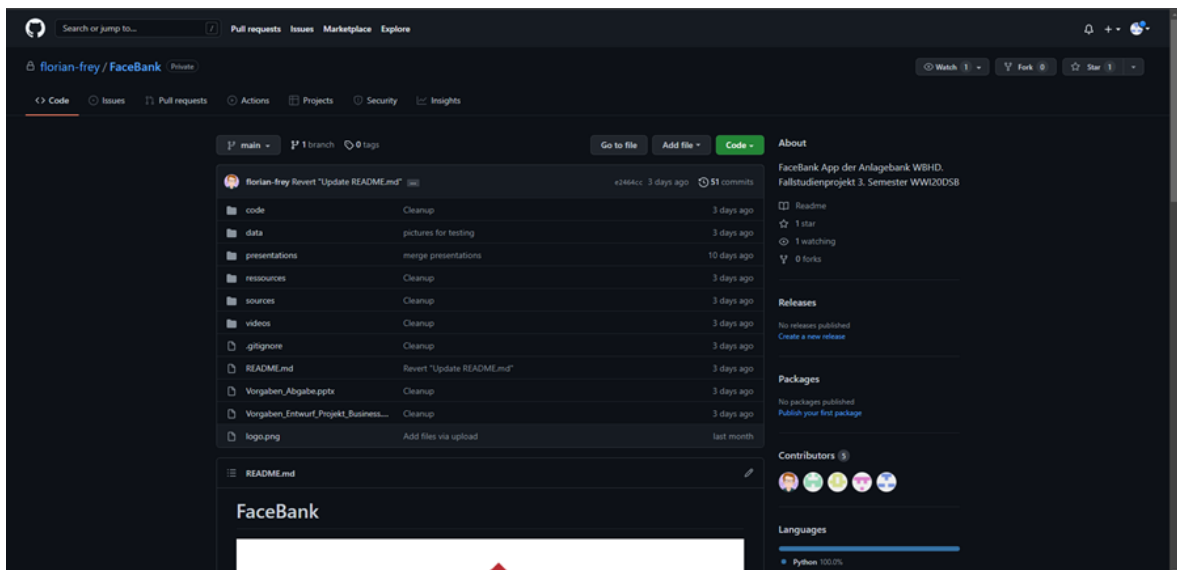
Die Teamarbeit innerhalb der Fallstudie wurde auf verschiedene Arten durchgeführt. Für die Auswahl der Methoden und Werkzeuge fand ein erstes Treffen nach der Vorlesung statt.

Verwaltung

Der erste Ansatz um das Projekt zu planen war, dass die Software „Jira“ das Projekt verwaltet. Die Auswahl ist jedoch schnell auf „Miro“ gefallen. Miro ist ein kostenloses digitales Whiteboard, auf welchem man in Echtzeit miteinander arbeiten kann. Es kam das Whiteboard für die Aufgabenverteilung zum Einsatz, indem Aufgaben in „Sticky Notes“ verfasst und dann auf einer Tafel auf die Projektmitglieder verteilt wurden.



Außerdem wurden Screenshots der Meilensteine und des Projektvorschlags angeheftet um die wichtigsten Termine und Informationen immer griffbereit zu haben. Ebenso wurde ein Github angelegt in dem die Aufgaben abgeben und sowohl Code als auch Bilder und Videos abspeichern und mit den Gruppenmitgliedern werden.



Meetings

Häufig gab es Vereinbarungen über ein Treffen in einer WhatsApp Gruppe in der festgelegt wurde bis zu welcher Zeit die nächsten abgeschlossenen Aufgaben in GitHub hochgeladen werden sollten. Die WhatsApp Gruppe wurde erstellt um sich schnell über Fragen und Informationen auszutauschen. Ähnlich zu dieser Gruppe gab es Treffen auf einem Discord Server, der als Text Channel und Sprach Channel dient um dort die Meetings abzuhalten. Discord ist eine kostenlose Anwendung auf der man Server erstellen kann und zu unserem Vorteil auch seinen Bildschirm teilen kann, es erinnert an Skype. Die Meetings folgten normalerweise nach den Fallstudie-Vorlesungen und an jedem Samstag oder Sonntag. Bis dahin gab es eine Art Hausaufgaben System in dem jeder zu seiner gewünschten Zeit seine Aufgaben abarbeiten kann.

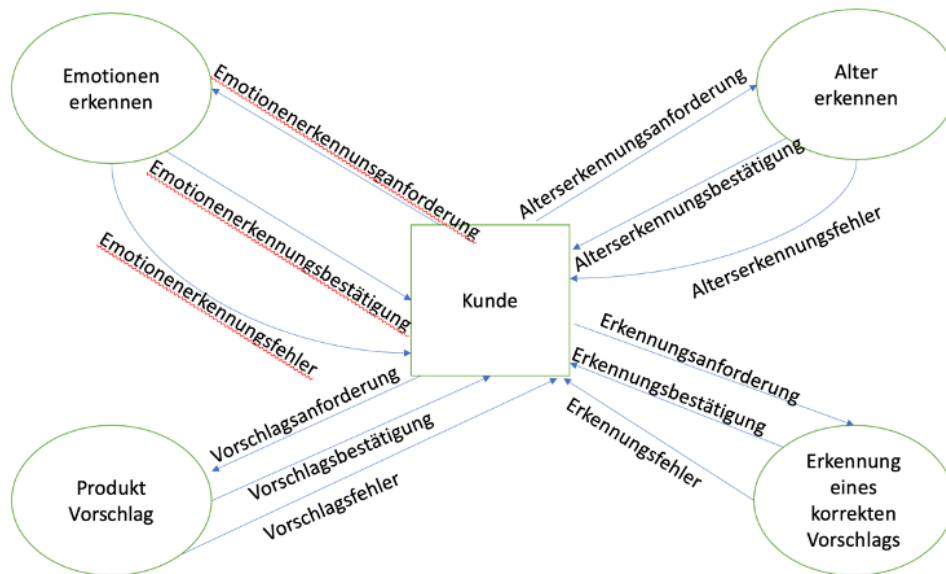
Zu den Aufgaben gehörten anfangs die von Herr Jung gestellten Vorgaben, darunter auch die Entwicklung der verschiedenen Modelle. Hierbei wurde aufgeteilt in Systementwurf, Projektvorgehen und wirtschaftliche Analyse. Bei der wirtschaftlichen Analyse sind anfangs Fragen aufgekommen, wie man diese auf einem fiktiven Unternehmen durchführt. Die Analysen und Modelle sind dann auf fiktiver Grundlage erstellt worden.

Systementwürfe

Um in die Aufgabe einzusteigen und systematisch vorzugehen war es nötig, einen Systementwurf fertigzustellen. Dafür wurden ein Ereignis-Reaktionsmodell (ERM), Entscheidungstabellen, ein Aktivitätsdiagramm und Use-Case-Schablonen gefordert.

Ereignis-Reaktionsmodell

Als erstes wurde das ERM entworfen. Als Nachbarsystem wurde der Kunde gewählt, welches vier Funktionen zur Verfügung hat. Durch drei Datenflüsse wird das Nachbarsystem mit den Funktionen verbunden. Die erste Funktion ist die „Erkennung der Emotionen“. Vom Kunden zur Funktion wird die Emotionenerkennungsanforderung gesendet, welches die Funktion mit einer Emotionenerkennungsbestätigung oder einem Emotionenerkennungsfehler antwortet. Die zweite Funktion „Erkennung des Alters“. Der Kunde fordert die Alterserkennung und es wird mit einer Alterserkennungsbestätigung oder einem Fehler reagiert. Die dritte Funktion gibt einen „Produkt Vorschlag“. Hier sind die reaktionauslösenden Datenflüsse ebenfalls eine Anforderung, eine Bestätigung oder ein Fehler. Die vierte Funktion des ERM ist die „Erkennung eines korrekten Vorschlags“. Wie alle anderen Funktion wird vom Kunde eine Erkennung angefordert, die mit einer Bestätigung oder einem Fehler beantwortet wird. Das Modell soll zeigen welche Funktionen, Ereignisse und Reaktionen das Projekt beinhalten soll.



Entscheidungstabellen

Um Ordnung zwischen den Funktionen wurden Entscheidungstabellen erstellt. Es wurden acht Regeln, drei Bedingungen und 3 Aktionen aufgestellt. Die drei Bedingungen sind: Natürliche Person + registriert, Emotionen sind bekannt, Alter ist bekannt. Die drei Aktionen: Emotionen in DB (Datenbank), Alter in DB, Produktvorschlag.

Um eine einen Produktvorschlag zu bekommen ist es wichtig, dass alle Bedingungen erfüllt sind. Denn nur so können alle Aktionen ausgeführt werden. Einen Vorschlag kann das System nur geben, wenn es beide Informationen, das Alter und die Emotion, besitzt. Das Impliziert, dass der Kunde immer eine Natürliche Person und registriert sein muss. Ein Bot kann beispielhaft keinen Produktvorschlag bekommen.

Bedingungen Aktionen	ET	R1	R2	R3	R4	R5	R6	R7	R8
	Natürliche Person + registriert	J	J	J	J	N	N	N	N
	Emotionen sind bekannt	J	J	N	N	J	J	N	N
	Alter ist bekannt	J	N	J	N	J	N	J	N
	Emotion in DB	X	X	-	-	-	-	-	-
	Alter in DB	X	-	X	-	-	-	-	-
	Produkt Vorschlag	X	-	-	-	-	-	-	-

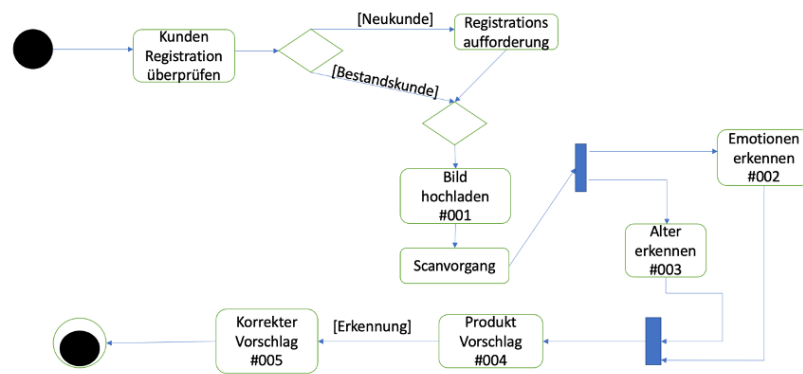
Da mehrere Regeln nicht nutzbar sind, aufgrund des nicht Erfüllen einiger oder aller Bedingung wurde die Tabelle konsolidiert. Somit sind nur fünf Regel wesentlich und wichtig für das gesamte System.

ET	R1	R2	R3	R4	R5
Natürliche Person + registriert	J	J	J	-	N
Emotionen sind bekannt	J	J	N	N	-
Alter ist bekannt	J	N	J	N	-
Emotion in DB	X	X	-	-	-
Alter in DB	X	-	X	-	-
Produkt Vorschlag	X	-	-	-	-

Aktivitätsdiagramm

Einen ganzen Durchlauf des Systems wird im Aktivitätsdiagramm ersichtlich. Das Diagramm beginnt mit einem Startknoten, welches zum ersten Aktivitätsschritt führt. Es wird überprüft ob der Kunde bereits registriert ist oder Neukunde ist. Es folgt ein Entscheidungsknoten welches sich in Bestandskunde und Neukunde aufteilt, das folglich auffordert sich zu registrieren. Danach wird ein Bild des Kunden hochgeladen, welches kurz darauf gescannt wird. Wurde das Bild erkannt, wird im Diagramm ein aufteilendes UND notiert. Das ist ein sogenannter Parallelisierungsknoten, denn die weiteren Schritte erfolgen gleichzeitig. Die

Erkennung der Emotionen und des Alters verlaufen parallel. Diese Schritte werden in einem zusammenführendes UND beendet, damit beide evaluiert werden und auf Basis dessen einen Produktvorschlag erlangt wird. Der nächste und letzte Schritt ist die Erkennung ob es sich um einen korrekten Vorschlag gehandelt hat. Das Aktivitätsdiagramm wird mit einem Schlussknoten beendet.



Use-Case Schablone

Auf Basis aller Modelle wurden 5 Use-Case Schablonen entworfen. Hier ein Beispiel des Use-Cases „Emotionen erkennen“. Ziel ist es, dass die Emotionen erkannt werden und verarbeitet werden. Als Vorbedingung gilt, dass ein Bild hochgeladen wird und verarbeitet werden kann. Das Resultat ist eine erfolgreiche Erkennung oder einen Fehler. In diesem Use-Case, sind die Akteure der Kunde und die App mit dem System. Das Ereignis löst aus, dass das Alter gescannt wird. Dazu wurden einige Erweiterungen und Alternativen hinzugefügt.

1. **Use Case:** Emotionen erkennen #002
2. **Ziel:** Emotionen des Kunden werden erkannt und können verarbeitet werden
3. **Vorbedingung:** Bild wurde erfolgreich hochgeladen
4. **Nachbedingung Erfolg:** Emotionen werden erkannt
5. **Nachbedingung Fehlschlag:** Fehlerausgabe da System Emotionen nicht erkennen kann
6. **Akteure:** Kunde, App(-system)
7. **Auslösendes Ereignis:** Alter wird als nächstes gescannt
8. **Beschreibung:**
1. Emotionen werden erkannt; 2. Aufforderung zur nächsten Erkennungsfunktion
9. **Erweiterung**
1a. Emotionen werden nicht erkannt; 2a. Kunde bricht App ab
10. **Alternativen**
1a. Datenbank für Emotionen muss erweitert werden

Marketing

Das wichtigste der Bank ist das Vertrauen, die Transparenz, die Verfügbarkeit und die Nähe zu ihren Kunden. Um das gewährleisten zu können, muss das passende Marketingkonzept herausgearbeitet werden.

Damit unterschiedliche Demografien bzw. Zielgruppen anvisiert werden können, sollten verschiedene Marketingkanäle genutzt werden. Bei Nichtbeachtung kann es durch die Kanalbeschränkung zum Zielgruppenverlust führen. Somit fiel die Entscheidung auf Social Media und direktes Marketing. Die richtige Kampagne in den Marketingkanälen steigert das Vertrauen der Kunden und kann auch mit bestimmten Produkten zu einem individuellen Kundenerlebnis führen. Des Weiteren sollten Wettbewerber nicht in den Hintergrund rücken, da es zu Kundenverlust führen kann. Exklusive Produkte und die optimale Kundenbetreuung sind somit sehr wichtig.

Konzept

Nachdem die Kriterien festgelegt sind, kann die Durchführung der Konzepte beginnen. Als Social Media Kanal wurde Instagram mit einer Finanzinfluencer Kampagne gewählt. Der Account ist unter dem Namen "facebankwbhd" zu finden. Unter dieser Kampagne versteht man eine Person, welche sich sehr gut in Finanzen auskennt und Informationen in seinen Social Media Kanälen teilt. In diesem Fall ist es Markus, der viele Erfahrungen in dem Segment gesammelt hat.

Es werden Tipps veröffentlicht, die einer Form von Beratung dienen. Durch diese Kampagne kann das Vertrauen der Kunden gewonnen werden.

Des Weiteren wird Living Marketing geführt, heißt der Kunde kann durch das Produkt FaceBank ein für sich spezifisches Produkt bekommen. Dies fördert ebenso das Erlebnis der Kunden und somit auch die Zufriedenheit.

Wie schon erwähnt, ist eine weitere Methode das direkte Marketing. Hierzu wird in Fußgängerzonen oder in Einkaufszentren ein Bildschirm aufgestellt, welcher eine Kamera enthält. So können potenzielle Kunden live das Produkt FaceBank testen und sich ein spezifisches Produkt vorschlagen lassen. Dies ist eine gute Möglichkeit verschiedenen Altersgruppen das Produkt zu zeigen.



Face Recognition

Funktionsweise

Face Recognition (deutsch: Gesichtserkennung) ist die Technologie, automatisiert Gesichter in Bild-Dateien zu identifizieren. Hierfür wird versucht, verschiedenste biometrische Muster in Trainingsdaten zu identifizieren, um diese später in Echtzeit auf neuen Bildern anwenden zu können.¹ Um diese Gesichtserkennung durch ein System durchführen lassen zu können, müssen zunächst einige Vorbereitungen getroffen werden. Da die Bilder mit Gesichtern, die in unserer App mit verschiedenen Geräten aufgenommen werden in allen möglichen Größen und Auflösungen auftreten können, müssen diese zunächst standardisiert werden. Dazu werden alle auf eine einheitliche Größe skaliert und zugeschnitten, ohne dabei das Gesicht zu entfernen. Zudem ist es für das Modell, welches später das Alter und die Emotion klassifizieren soll, von Vorteil, wenn es mit besonders einheitlichen Daten arbeiten kann. Dazu ist neben der Standardisierung der Bilder an sich, auch die Positionierung und Neigung der Gesichter im Bild wichtig. Das Modell wird die sogenannte "face-region of interest" feststellen. Also der Bereich des Bildes in dem das Gesicht und somit die zur Klassifizierung wichtigen Merkmale enthalten sind. Dazu können Ankerpunkte im Gesicht, wie beispielsweise die Augen, die Nasenspitze oder der Mund gewählt werden, um alle Gesichter einheitlich auszurichten. Alleine diese Ausrichtung kann die Genauigkeit von Modellen wohl um knapp 1% erhöhen.²

Da auch Menschen das genaue Alter nicht nur anhand des Gesichtes einschätzen können - schließlich hängt die optische Alterung von vielen Faktoren ab - ist es auch für das Modell nur realistisch, die Gesichter in verschiedene Altersgruppen zu klassifizieren.

Deepface

Bei der Recherche nach Möglichkeiten zur Face Recognition mit Python sind wir schnell auf das Modul deepface gestoßen. Neben dem einfachen Erkennen von

¹ <https://www.electronicid.eu/en/blog/post/face-recognition/en>

² <https://github.com/serengil/deepface>

Gesichtern in Bildern, welches immer den ersten Schritt darstellt, bietet das Modul auch Funktionen zur Verifikation von Gesichtern, indem es zwei oder mehr Bilder vergleicht. Für unsere App wichtiger ist allerdings die Funktion zur Analyse von Gesichtsattributen. Hiermit können auf erkannten Gesichtern die Attribute „age“, „gender“, „facial expression“, also die Emotion, und „race“ bestimmt werden. Nach Aussagen der Dokumentation des Moduls³, hat die Altersbestimmung im mittleren absoluten Fehler eine Abweichung von nur $\pm 4,65$ Jahren. Das entspricht in etwa dem, was man auch von einem Menschen erwarten könnte, der das Alter anderer Menschen anhand ihres Gesichts einschätzt. Natürlich kann weder Mensch noch Maschine anhand des Gesichtes ein exaktes Alter bestimmen, weshalb die Klassifikation durch das Modell für unsere App in Altersklassen geschehen soll. Hierfür ist eine Aufteilung in junge, erwachsene und alte Menschen vorgesehen.

Das deepface Modul bietet verschiedene ‚state-of-the-art‘-Modelle zur Gesichtserkennung. Standardmäßig wird das VGG-face model⁴ genutzt. Es kann aber auch zwischen Google FaceNet, Facebook DeepFace und weiteren Modellen gewählt werden. Jedes Modell bringt seine eigenen Vor- und Nachteile mit sich. Allerdings ist für die Analyse von Gesichtsattributen, welche für uns relevant ist, nur eine Standardeinstellung möglich. Anpassbar ist jedoch das Backend zur Face-Detection. OpenCV ist hier der Standard. Da andere Modelle laut Dokumentation overperformen oder deutlich langsamer sind, haben wir uns dazu entschieden mit der Standardeinstellung fortzufahren.

Um das Modul auszuprobieren, haben wir kurzen Code geschrieben, in dem einzelne Bilder geladen und analysiert werden können. Das Skript hierzu soll kurz beschrieben werden. Auch das Ergebnis der Analyse anhand eines beispielhaften Bildes ist folgend dargestellt:



Age:	31
Emotion:	neutral
Gender:	Man

³ <https://github.com/serengil/deepface>

⁴ <https://sefiks.com/2018/08/06/deep-face-recognition-with-keras/>

Race: white

Facebank/data/seb.jpg

Datei: Facebank/code/facerec.py

```
1  import cv2
2  from deepface import DeepFace
3
4  # define path to image and read image
5  imgpath = r'./data/seb.jpg'
6  image = cv2.imread(imgpath)
7
8  # analyze image, save attributes and print them
9  analyze = DeepFace.analyze(image, actions=['age',
'emotion', 'gender', 'race'])
10
11  age = analyze['age']
12  emotion = analyze['dominant_emotion']
13  gender = analyze['gender']
14  race = analyze['dominant_race']
15
16  print(age, emotion, gender, race)
```

Die Implementierung des Moduls im Code ist sehr einfach. Wir importieren lediglich das Modul selbst sowie ein Modul namens cv2 zum Einlesen von Bilddateien. In den Zeilen 4-6 wird der Pfad des zu analysierenden Bildes angegeben und dieses anschließend von cv2 in einer Variable gespeichert. In Zeile 9 geschieht die eigentliche Analyse des Bildes. Mit dem Aufruf der Funktion ‚analyze‘ des deepface Moduls und der Übergabe des Bildes sowie der auszuführenden Aktionen als Parameter wird die Analyse durchgeführt. Die Ergebnisse werden anschließend in Variablen gespeichert und in der Konsole ausgegeben.

Testing mit Datensätzen

Um die Qualität der FaceBank App sicherzustellen wollen wir die Gesichtsanalyse nun natürlich an einem großen Datensatz, und nicht nur an einzelnen Bildern Testen. Der erste Schritt dafür ist es, einen guten und ausreichend großen Datensatz zu finden. Dieser muss natürlich annotiert sein, damit die vorhergesagten Ergebnisse des Modells mit den echten Ergebnissen verglichen werden kann. Um das Deepface-Modul zu testen wurden auf Kaggle Datensätze mit annotierten Gesichtern gesucht. Dabei wurden zwei Datensätze⁵ mit Bildern von Gesichtern, verschiedener Altersklassen benutzt. Diese waren teilweise nach genauem Alter gelabelt, während ein zweiter nach Altersgruppen gelabelt war.

Um das Modell von Deepface zu testen wurde wie oben beschrieben Daten in ein extra dafür geschriebenes Pythons-Programm mit Hilfe des cv2-Moduls eingelesen und die Deepface Alterserkennung wurde anschließend auf die Bilder angewandt. Dabei wurde dem Programm eine gewisse Abweichung erlaubt, da in der Praxis nur grobe Altersgruppen und nicht das exakte Alter relevant ist. Bei der Toleranz handelt es sich um jeweils fünf Jahre jünger und fünf Jahre älter um eine richtige Erkennung auszugeben. Lag das vom Modell erkannte Alter innerhalb dieser Grenzen des echten, annotierten Alters des Gesichts, wurde dem Gesicht "true", andernfalls "false" zugewiesen.

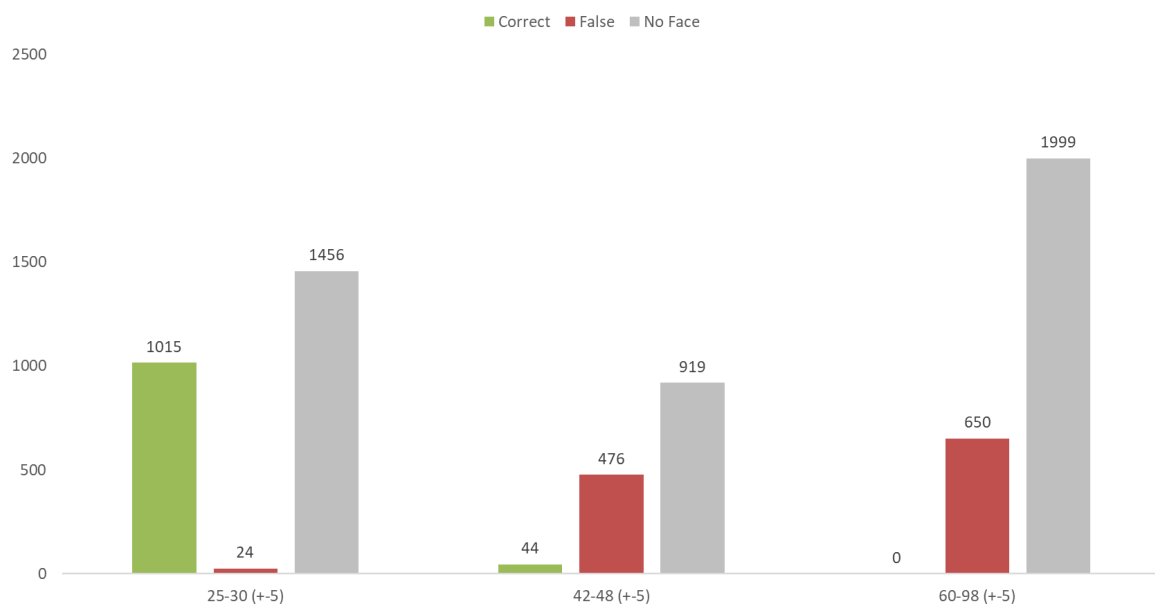
Die Testergebnisse waren allerdings nur teilweise zufriedenstellend. Bei den ersten Versuchen wurde eine große Anzahl an Gesichtern als "false" gewertet. Es stellte sich jedoch heraus, dass dies an einer hohen Anzahl nicht erkannter Gesichter lag, weshalb eine dritte Kategorie "no face" eingefügt wurde. Die neuen Testergebnisse zeigten dann, dass es zwar mehr "true" als "false" Klassifizierungen gab, ein Großteil der Gesichter jedoch gar nicht erkannt wurde.

Dies lag vermutlich daran, dass die Qualität der Gesichter in den vorhandenen Datensätzen, wie in den folgenden Beispielen zu sehen, durchweg schlechter ist, als die einer aktuellen Smartphone-Innenkamera.

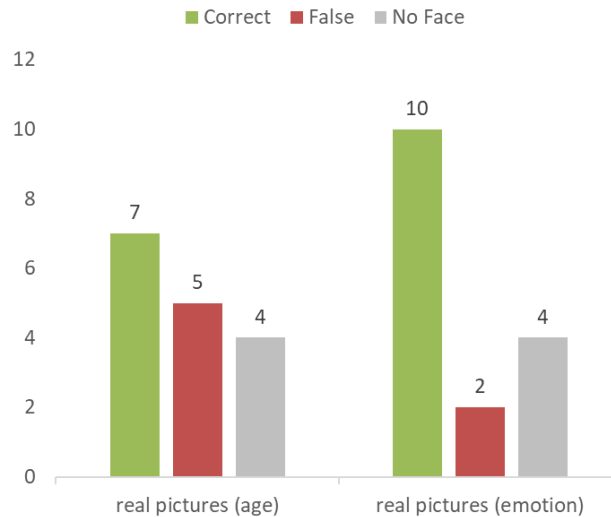
⁵ <https://www.kaggle.com/frabbisw/facial-age> und <https://www.kaggle.com/ttungal/adiance-benchmark-gender-and-age-classification>



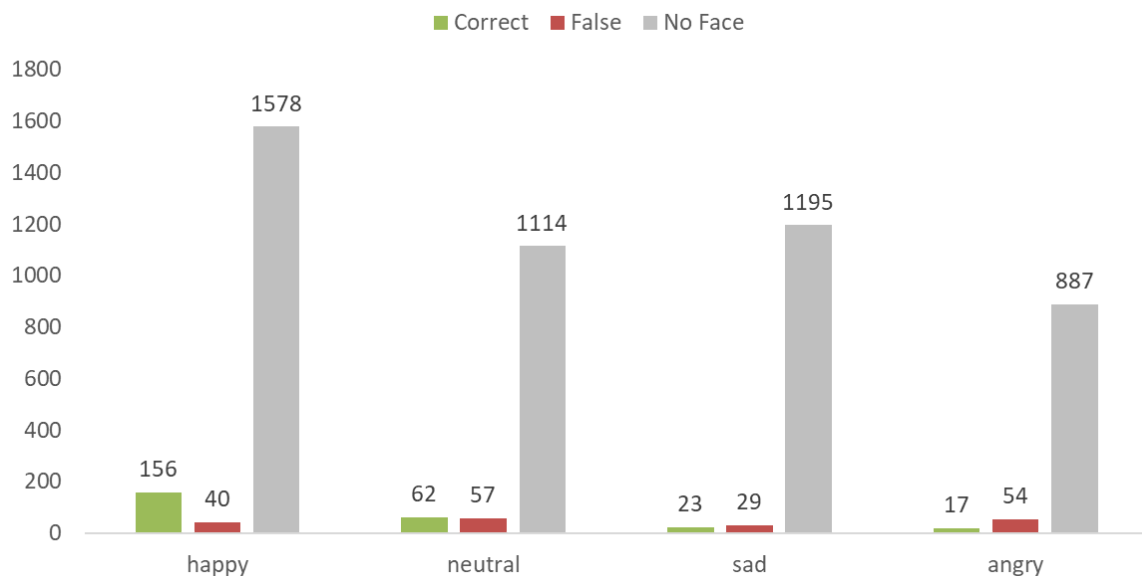
Außerdem war auffällig, dass das Modell hauptsächlich Gesichter in jungen Altersklassen von circa 20 bis 30 Jahren richtig erkannt hat. Ältere Menschen wurden oft jünger klassifiziert, als sie tatsächlich sind. Die Testergebnisse hierzu sind in folgendem Diagramm aufgezeigt:



Diese Umstände wurde vom Team sofort als Risiko betrachtet, da diese Testbedingungen die Realität schlecht widerspiegeln. Daher wurden neben diesem Test mit einer großen Anzahl an Daten auch manuell mit selbst aufgenommenen beziehungsweise qualitativ hochwertigeren Bildern von Prominenten in kleiner Stückzahl getestet. Die Testergebnisse hierbei waren deutlich besser, jedoch bei einer so kleinen Anzahl an Testdaten nicht signifikant.



Ebenfalls wies das Deepface-Modul nicht nur die schwächen in der Alterserkennung bestimmter Altersgruppen auf, sondern auch in der Emotionserkennung der meisten geforderten Emotionen. Während “neutral” und “happy” gut erkannt wurden, tat sich das Modell bei den anderen Emotionen wie zum Beispiel “sad” oder “angry” schwer.



Probleme

Wie bereits beschrieben lagen die Probleme des Deepface-Moduls darin, dass zum einen bei vielen der Bilder aus den Testdatensätzen keine Gesichter erkannt wurden. Zwar lag die Quote bei den händisch getesteten Bildern höher, jedoch

können so nicht genug Bilder getestet werden, um eine aussagekräftige Schlussfolgerung über die Qualität des Deepface-Modells zu ziehen. Außerdem wurde ersichtlich, dass die Erkennung des Alters in einigen Bereichen gut funktionierte, während in anderen Altersgruppen nur sehr ungenaue Zuweisungen erfolgt sind. Dies könnte auf eine Schwäche des Modells hinweisen, dass auf manche Altersgruppen unzureichend trainiert ist. Durch die enge Zeitplanung war es auch nicht möglich, selbst einen Datensatz zu erstellen und zu labeln, da dies die Fertigstellung des Programms zu weit hinausgezögert hätte und zum Verpassen der Deadline geführt hätte.

Lösungsversuche

Um das Problem der scheinbar schlechten Testdatensätze zu beheben, wurde auf Hinweis eines Stakeholders beim Fraunhofer-Institut angefragt, ob es eine Möglichkeit gibt, von diesem geeignete Testdaten zu erhalten. Trotz mehrfacher Nachfrage per E-Mail erhielten wir keine Antwort. Da auch andere Versuche an einen qualitativ hochwertigen Datensatz zu kommen erfolglos blieben, wurde über einen generell neuen Ansatz für die Lösung des Problems diskutiert. Dabei wurde das Team auf die Azure Face API⁶ aufmerksam. Da das Frontend ohnehin mit Microsoft Power Apps geplant wurde, zu gestalten und die Azure Face API ebenfalls von Microsoft geliefert wird, ist diese einfach in die App integrierbar. Daher wurde schnell beschlossen, die Azure Face API zu testen.

Nach dem Testen der Azure Face API und der Einbindung in Microsoft Powerapps funktioniert wurde schlussendlich beschlossen, diesen Weg im Projekt zu gehen. Die Schnittstelle lieferte konstant gute Ergebnisse sowohl in der Alters- als auch Emotionsklassifizierung, ließ sich mit deutschem Datenschutz vereinbaren und war zudem schneller, als das Deepface-Modul, was die spätere Benutzerfreundlichkeit positiv beeinflussen kann. Alternativ hätte man die Bilder aus der App in ein externes Python-Skript exportieren müssen, um dort die Analyse mit einem qualitativ fraglichen Modell durchführen zu können und das Ergebnis anschließend wieder in die App-Oberfläche zurückzuführen.

⁶ <https://azure.microsoft.com/de-de/services/cognitive-services/face/#overview>

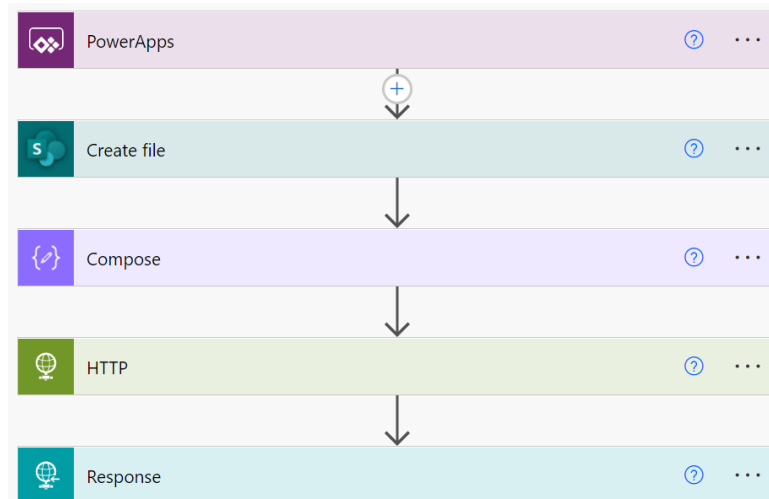
Analysevorgang in Microsoft

Azure Face API

Da Microsoft den fertigen Azure-Gesichtserkennungsdienst für Kunden zur Verfügung stellt, müssen lediglich Einstellungen über die Location, die Preisstufe oder auch den Speicherort festgelegt werden. Allerdings sind diese an das Unternehmen angepasst, welches der Anlagebank WBHD die Nutzung von Power Apps ermöglicht hat. Innerhalb der Face API wird eine Resource Group erstellt, die zur Ressourcenverwaltung dient.

Power Automate Flow

Für die Erstellung der App muss vorerst ein Flow in Power Automate erstellt werden. Der Flow dient zur Übertragung der Bilder in einen SharePoint, um diese später mit der schon verfügbaren Face API von Microsoft auszuwerten. Genauer wird der Flow durch einen Button in der App ausgelöst. Damit die Daten der Bilder in den SharePoint gelangen muss ein Dokument erstellt werden, das lediglich die Daten des Bildes und nicht das Bild selber enthält. Im Weiteren wird das erstellte Dokument in eine binäre Funktion konvertiert, was wichtig für den nächsten Schritt ist. Mit der HTTP Komponente kann nun der API Endpoint aufgerufen werden, welcher durch die individuellen Schlüssel aus dem Microsoft Azure Portal ausgelesen wird. Ebenso werden hier die benötigten Attribute für die Analyse angegeben, wie beispielsweise das Alter oder die Emotion. Zuletzt wird eine Response Komponente eingefügt, die das JSON Schema aus der Face API Dokumentation enthält. Damit kann der Flow die benutzerdefinierte HTTP Komponente interpretieren.



Power App

Nachdem der Flow erstellt wurde kann die OnSelect Funktion des Analyse Buttons zusammengesetzt werden, der mit dem erstellten Bild zusammenhängt: `ClearCollect(colFaceData, FaceDetection.Run(First(colPhoto).Url))`

Hierbei wird eine Datenquelle (Collection) erstellt, die eine Tabelle als geänderte Sammlung zurückgibt. Das Kamerafenster ist ebenso mit einer Datenquelle verbunden, die bei Berührung des Fensters befüllt wird.

Die analysierten Werte werden durch die Datenquelle über sogenannte Text inputs ausgegeben. Für das Alter beispielsweise sieht der Befehl wie folgt aus:

```
First(colFaceData).faceAttributes.age
```

Die restlichen Text inputs haben dementsprechend Befehle mit den Emotionsattributen Neutral, Happiness, Surprise, Contempt, Disgust, Fear, Sadness und Anger.

Nun können die Produkte den einzelnen Emotionen zugeordnet werden. Dazu wird die erstellte Produktmatrix genutzt. Ein Beispiel für einen Bausparvertrag sieht folgendermaßen aus:

```
If(Value(TextInputHappinessValue.Text) > 0.8 &&
Value(TextInputAgeValue.Text) >= 18 &&
Value(TextInputAgeValue.Text) <= 30, RGBA(168, 0, 0, 1),
If(Value(TextInputContemptValue.Text) > 0.8 &&
Value(TextInputAgeValue.Text) >= 18 &&
```

```

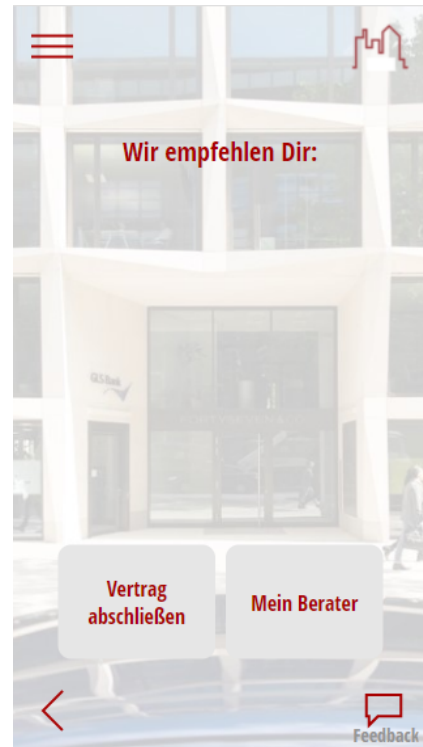
Value (TextInputAgeValue.Text)  <=  30,  RGBA (168,  0,  0,  1) ,
If (Value (TextInputDisgustValue.Text)      >      0.6      &&
Value (TextInputAgeValue.Text)      >=      18      &&
Value (TextInputAgeValue.Text)  <=  30,  RGBA (168,  0,  0,  1) ,
If (Value (TextInputFearValue.Text)      >      0.8      &&
Value (TextInputAgeValue.Text)      >=      18      &&
Value (TextInputAgeValue.Text)  <=  30,  RGBA (168,  0,  0,  1) ,
RGBA (0,0,0,0) ) ) ) )

```

Wie im Code zu erkennen ist, muss für einen Bausparvertrag die Person zwischen 18 und 30 Jahren sein sowie Glück, Verachtung, Ekel oder Angst verspüren. Trifft die Bedingung zu ändern sich die Farben der Elemente, damit es für den Nutzer sichtbar ist.



Neben dem Analyse Button befinden sich noch andere Buttons, die dem normalen Gebrauch einer Banking App entsprechen. Weitere Funktionen sind beispielsweise das Unternehmenslogo oben rechts, das bei Berührung auf die Startseite führt. Mit dem Pfeil unten rechts kann man auf die vorherige Seite zurückkehren.



Fazit und kritische Reflektion

Trotz Schwierigkeiten bei dem Testen des Gesichtsanalyse-Modells von Deepface, konnte das Team durch schnelles und entschlossenes Handeln und Suche nach Alternativen eine geeignete App entwickeln. Diese kann sowohl das Alter, als auch die Emotionen der aufgenommenen Gesichter genau genug für den Anwendungsfall der Banking App erkennen. Durch die regelmäßigen Absprachen und die gute Kommunikation im Team, konnte so eine Verspätung der Fertigstellung verhindert werden, ohne dabei an Qualität einbüßen zu müssen. Ebenso konnte stets auf die Hilfe und Unterstützung des Unternehmens gezählt werden, welches dem Team die Durchführung des Projekts mit den Microsoft Produkten ermöglicht hat. Hervorzuheben ist auch, dass ebenfalls auf den Datenschutz geachtet werden konnte. Die Server speichern die Bilder nicht ab, und befinden sich in Europa und somit mit deutschen Datenschutzrecht konform. Hinzuzufügen ist ebenso der Versuch eine reale Person als Finanzinfluencer zu engagieren, jedoch kamen dabei keine positiven Rückmeldungen zurück. Somit bleibt es eine fiktive Person.