# Prüfungsleistung

### Team Namen:

Startdatum: 11 Dezember 2024

Abgabedatum: 09 Februar 2025

## 1. Aufgabe

Ihr könnt aus einer Liste von Datensätzen wählen und müsst aus einem dieser einen Business-Case erzeugen. Diesen Business-Case müsst ihr dann in Form eines Dashboards ausarbeiten und am Ende präsentieren.

Beispiele, die euch helfen sollen, das ganze einordnen zu können:

- Ihr wählt einen Datensatz zu Rotten Tomates Movie Rating und leitet daraus den Business-Case ab, welche Filme Hollywood Produzenten vermehrt produzieren sollten. Dazu formuliert ihr ein Problem-Statement, bereinigt die Daten und stellt diese sinnvoll in einem Dashboard zur Verfügung und erläutert anhand sinnvoller visueller Daten, warum z.B. vermehrt Action-Filme gedreht werden sollten. Die Ausarbeitung muss plausibel und sinnvoll anhand der Daten eure Entscheidung belegen können
- Ihr wählt einen Datensatz zu Fake Job Postings. Für diesen Datensatz leitet
  ihr den Business-Case einer Start-Up Gründung ab, welches sich darauf
  spezialisiert eine Webseite zur Verfügung zu stellen, die Fake Angebote
  identifizieren kann und dem Benutzer sinnvolle Job Postings anbietet, die
  seinem Gesuch entsprechen. Das Ganze belegt ihr anhand eines Dashboards
  mit sinnvoll aufgearbeiteten Daten

## 2. Prüfungsvoraussetzungen

Ihr müsst als Gruppe sowohl eine Präsentation mit Live-Demo Eurer Applikation vorbereiten als auch eine schriftliche Abgabe, die ca. 12-15 Seiten umfassen sollte.

#### Die schriftliche Abgabe (Richtlinien):

- Folgt dem schriftlichen Standard, der von der FH vorgegeben ist
- Abgabe besteht aus eurem Code (in einer .zip) und aus einer .pdf (Es sind nur 2 Dateien erlaubt in der Abgabe)
- Benutzt ein Inhaltsverzeichnis, ein Deckblatt, eine Eidesstattliche Erklärung (muss unter anderem Umfassen, dass ihr keine KI-Tools benutzt habt) und eine Quellenangabe (diese 4 Seiten zählen nicht zu den 12-15 Seiten Ausarbeitung)
- Wurden Code und schriftliche Abgabe nicht bis zum 09.02.25 bis 23:59 hochgeladen sein oder eines von beiden fehlen, ist die Gruppe mit 5.0 durchgefallen
- Markiert, wer welchen Absatz ausgearbeitet hat und markiert im Code, wer was gecodet hat (beides wird individuell bewertet)
- Themen, die abgedeckt sein müssen, umfassen: Problemstatement, Arbeit im Projektteam (Methode, Organisation, Ablauf, ...), kurze Beschreibung eures Codes (Architektur/Aufbau), eure gewählten Daten und deren Aufbereitung (ETL-Implementierung erklären), das Dashboard und eueren Business-Case

#### Am Präsentationstag:

- 5 Minuten pro Person Vortrag (individuelle Bewertung (z.B. 1 Minute Redeanteil wird mit 5.0 bewertet)
- Es müssen sowohl eure Arbeit im Projektteam (z.B. zeigen eines KanBan Boards, Teamorganisation, Zeitplan, etc.), als auch euer Produkt, inklusive Code vorgestellt werden
- 5 Minuten pro Person Fragen (individuelle Bewertung)
- Gesamtzeit ergibt sich also aus Gruppenmitglieder x 10 (Bei 5 Leuten sind das 50 Minuten Prüfung)

#### 3. Hinweise

- Versucht bei der schriftlichen Ausarbeitung darauf zu achten, dass jeder sowohl im schriftlichen Teil als auch beim Code ungefähr einen gleichen Anteil hat
- Achtet bei der Präsentation darauf, dass der Redeanteil gleich ist und vom Inhalt relevant (macht z.B. eine Person nur Vorstellung und Fazit, aber hat keinen Inhalt, wird er eine deutlich schlechtere Note bekommen als die Kommilitonen)
- Beachtet, dass bei den anschließenden Fragen, Stoff aus der gesamten Vorlesung rankommen kann

# 4. Python - ETL und Visualisierung

- Ihr benötigt mindestens 2 verschieden Programme
  - ETL, welches von einer .csv Daten einliest und anschließend transformiert und abschließend in eine Datenbank lädt (sollte keine Datenbank benutzt werden oder keine Transformation vorliegen, dann bekommt ihr als Gruppe eine 5.0)
  - 2. Visualisierung, die aus der Datenbank die transformierten Daten lädt und in einem Dashboard darstellt
- Es muss im code klar ersichtlich sein, welche Transformationsschritte gemacht werden
- Ihr müsst keine Unit Tests schreiben, aber:
- Je mehr Abstraktion (Klassen, Design Patterns, Decorator, Funktionen, usw.), Dokumentation, PEP8 Befolgung, Unit-Tests und so weiter benutzt werden, desto wahrscheinlicher wird eine bessere Bewertung, beachtet hier aber, dass ersichtlich sein muss, dass Ihr versteht, was ihr da macht; allgemein ist es nicht schlimm, wenn der Code simpel ist, solange er gut leserlich ist und alle anderen Kriterien erfüllt, spricht nichts dagegen, dass ihr eine sehr gute Note für den Code bekommt

## 5. Noten

Die Note wird am Folgetag der Präsentation ans Prüfungsamt gemeldet und setzt sich aus den folgenden 4 individual Leistungen zusammen:

- 25% Schriftliche Ausarbeitung
- 25% Code
- 25% Vortrag (Inhalt, Präsentationsfolien, Stil, ...)
- 25% Beantwortung der Fragen
- Ist einer der 4 Teile eine 5.0, ist man komplett durchgefallen
- Z.B. 1.0 Schriftliche Ausarbeitung, 2.0 Code, 3.0 Vortrag und 4.0 Beantwortung der Fragen = (1.0+2.0+3.0+4.0) / 4 = 2.5 ~= 2.3
- Am Ende wird die Verbesserung angerechnet, falls alle Blätter bearbeitet wurden, aus dem Beispiel oben, würde ich dann eine 2.0 ans Prüfungsamt melden