

Projektstruktur:

Projektsetup & Git-Workflow:

- **Repository-Initialisierung:**
Wir erstellen einen GitHub-Repository mit einer initialen README-Datei, die die Projektziele und Details zum Datensatz beschreibt.
 - **Forking und Cloning:**
Ihr klonst das Repository auf eure lokale Maschinen.
 - **Branching-Strategie:**
Ihr erstellt separate Branches für eure Aufgaben (z. B. data-preprocessing, eda, modeling).
-

Datenvorverarbeitung:

- **Aufgabenteilung:**
 - **Matthias:** Lädt den Datensatz und behandelt fehlende oder inkonsistente Daten.
 - **Annett:** Führt Datenumwandlungen durch, z. B. Kodierung kategorialer Variablen und Normalisierung numerischer Merkmale.
 - **Zusammenarbeit:**
Nach Abschluss ihrer Aufgaben überprüft gegenseitig euer Code und führt die Branches in den main-Branch zusammen.
-

Explorative Datenanalyse (EDA):

- **Visualisierungen:**
Erstellen von Diagrammen und Grafiken zur Visualisierung von Trends in Freelancer-Einnahmen, Jobkategorien und anderen relevanten Metriken.
 - **Erkenntnisse:**
Wichtige Erkenntnisse aus den Visualisierungen zusammenfassen, um Faktoren zu identifizieren, die Freelancer-Einnahmen beeinflussen.
-

Maschinelles Lernen:

- **Ziel:**
Ein Vorhersagemodell entwickeln, um Freelancer-Einnahmen basierend auf verfügbaren Merkmalen zu schätzen.

- **Aufgabenteilung:**
 - **Matthias:** Teilt die Daten in Trainings- und Testdaten auf und implementiert Basis-Modelle (z. B. lineare Regression).
 - **Annett:** Experimentiert mit fortgeschrittenen Modellen (z. B. Entscheidungsbäume, Random Forests) und führt eine Hyperparameter-Optimierung durch.
 - **Evaluation:**

Bewertung der Modellleistung anhand geeigneter Metriken (z. B. RMSE, R^2) und Vergleich der Ergebnisse.
-

Dokumentation & Berichterstattung:

- **Code-Dokumentation:**

Sicherstellen, dass alle Skripte gut kommentiert und organisiert sind.
 - **Projektbericht:**

Gemeinsam einen Bericht verfassen, der die Schritte der Datenvorverarbeitung, die EDA-Ergebnisse, die Modellansätze, die Resultate und die Schlussfolgerungen beschreibt.
-

Richtlinien zur Git-Zusammenarbeit:

- **Commit-Nachrichten:**

Klare und aussagekräftige Commit-Nachrichten verwenden.
- **Pull Requests:**

Pull Requests erstellen, um Branches zusammenzuführen, und Peer-Reviews anfordern.
- **Issue-Tracking:**

GitHub-Issues nutzen, um Aufgaben, Fehler und Verbesserungen zu dokumentieren.
- **Regelmäßige Synchronisierung:**

Regelmäßig Änderungen aus dem main-Branch abrufen, um den Code aktuell zu halten.