### **Project Structure Explanation**

The project comprises the following files and directories:

* **convert to parquet.ipynb** – A Jupyter Notebook that converts all data (excluding image encodings) into Parquet files and organizes them. This script must be executed after downloading the dataset from Kaggle (if using the Kaggle version rather than the provided dataset). [**Kaggle Dataset Link**](https://www.kaggle.com/datasets/kaborg15/vibrent-clothes-rental-dataset)*.*
* **Requirements.docx** – A document detailing the steps required to run the project. It is essential to read this file before executing any code.
* **Declaration of Tools and Resources.docx** – A document listing the tools and resources utilized in the project.
* **article.pdf** – The research article derived from this study.
* **Archive folder** – Contains the dataset, structured as follows:
  + **Data folder** – Stores the converted Parquet files:
    - orders.parquet
    - picture\_triplets.parquet
    - outfit\_tags.parquet
    - outfits.parquet
  + **Additional files**
    - outfits.csv
    - user\_activity\_triplets.csv – A subset of the orders dataset.
    - picture\_triplets.csv – Maps outfits to their corresponding images.
  + **additional\_tabular\_data folder** – Contains supplementary datasets, of which only the first file was utilized in this project:
    - original\_orders.csv – The remaining order data.
    - third\_chance.csv
    - subscription\_plan.csv
    - spot\_rentals.csv
  + **embeddings folder** – Stores image embeddings:
    - EfficientNet\_V2\_L\_final\_dict.pkl – A file facilitating the loading of embeddings.
    - \*.npy files – The outfit embeddings saved as NumPy arrays.
* **Exploring folder** – Contains notebooks for data exploration:
  + orders\_outfit\_tags.ipynb – Analyzes outfit orders across categories and their associated tags.
  + outfits\_tags.ipynb – Examines outfit categories and their respective tags.
  + orders.ipynb – Investigates order data.
  + outfits.ipynb – Explores outfit data.
  + picture\_triplets.ipynb – Analyzes the picture\_triplets dataset.
  + outfits\_tags\_groups.ipynb – Examines outfit categories and their groupings.
  + repeat.ipynb – Identifies cases where customers rent the same outfit multiple times.
  + repeat\_group.ipynb – Analyzes instances where customers rent from the same category multiple times.
* **Models folder** – Contains model training, evaluation, and supporting functionalities:
  + embedding.ipynb – Generates various types of outfit embeddings and encodings.
  + my\_encoding.ipynb – Implements a custom approach to outfit encoding based on price and category.
  + prepare\_train\_test\_splits.ipynb – Splits data into training, validation, and test sets chronologically.
  + evaluate\_models.ipynb – Implements methods for model performance evaluation.
  + Collaborative\_Filtering.ipynb – Implements a collaborative filtering recommendation approach.
  + baseline.ipynb – Implements multiple baseline recommendation methods.
  + content\_based\_KNN.ipynb – Uses different encoding techniques (*see embedding.ipynb for details*) to generate recommendations.
  + Rating.ipynb – Applies collaborative filtering techniques, using the frequency of outfit rentals as an implicit rating.
  + content\_based\_KNN\_with\_repeat\_count.ipynb – Enhances content\_based\_KNN.ipynb by incorporating rental frequency into recommendations.
  + order\_by\_previous\_rents.ipynb – Modifies predict\_previous\_rental and predict\_rental\_and\_most\_popular (*from the baseline methods*) by incorporating sorting based on rental frequency.
  + **Results and supporting files:**
    - results.xlsx – Documents the performance results of all models.
    - val.json, val\_outfit.json, val\_group.json – Files required for hyperparameter tuning.
    - user\_splits.parquet – Stores the train-validation-test splits (*generated in prepare\_train\_test\_splits.ipynb*).
    - outfits\_embeddings.parquet – Stores outfit embeddings for content-based recommendations (*generated in embedding.ipynb*).
  + **failed\_attempts folder** – Contains preliminary modeling attempts that did not yield results.