Project Structure Overview . . .

1. \*\*data\_processing\*\*: This folder contains scripts and files related to preparing the data for training.

2. \*\*main\*\*: This folder contains the main scripts to build, train, and evaluate the neural network model.

3. \*\*notebooks\*\*: This folder contains Jupyter notebooks for exploring the data and experimenting with models.

4. \*\*results\*\*: This folder stores the results of your training and evaluation, like logs and saved models.

5. \*\*README.md\*\*: This file explains what your project is about and how to use it.

6. \*\*requirements.txt\*\*: This file lists the Python packages needed to run your project.

Detailed Breakdown

1. \*\*data\_processing Folder\*\*:

- \*\*preprocess\_data.py\*\*: This script processes your raw data by filling in missing values, removing any data anomalies, and normalizing the data (making sure it has a consistent scale). It also splits the data into training, validation, and test sets.

- \*\*raw\_data.csv\*\*: This is your original data file.

- \*\*processed\_data.csv\*\*: This is your data after it has been cleaned and prepared.

- \*\*train\_data.csv, valid\_data.csv, test\_data.csv\*\*: These files are your data divided into three parts: one for training your model, one for validating it during training, and one for testing it after training.

- \*\*normalization\_params.csv\*\*: This file saves the parameters used to normalize your data (like the mean and standard deviation).

2. \*\*main Folder\*\*:

- \*\*preprocess\_functions.py\*\*: This file contains functions for preprocessing the data (cleaning, filling missing values, normalizing).

- \*\*process\_new\_data.py\*\*: This script uses the saved normalization parameters to preprocess new data, ensuring it’s processed in the same way as your training data.

- \*\*train\_model.py\*\*: This script trains your neural network model using the training data.

- \*\*evaluate\_model.py\*\*: This script evaluates how well your trained model performs using the validation and test data.

- \*\*model.py\*\*: This file defines the structure of your neural network.

- \*\*utils.py\*\*: This file contains utility functions that help with tasks like loading data and calculating performance metrics.

3. \*\*notebooks Folder\*\*:

- \*\*exploratory\_data\_analysis.ipynb\*\*: This notebook lets you explore and understand your raw data.

- \*\*model\_training.ipynb\*\*: This notebook lets you experiment with different model configurations and training procedures.

- \*\*model\_evaluation.ipynb\*\*: This notebook lets you evaluate the performance of your trained model.

4. \*\*results Folder\*\*:

- \*\*training\_logs/\*\*: This folder stores logs generated during model training.

- \*\*models/\*\*: This folder stores the saved versions of your trained models.

- \*\*evaluation\_results/\*\*: This folder stores results from evaluating your model, like performance metrics and confusion matrices.

Key Files and Scripts

1. \*\*preprocess\_data.py\*\*: This script will preprocess your raw data. Here’s what it does step-by-step:

- Load the raw data.

- Fill in any missing values.

- Remove any data anomalies (e.g., unrealistic values).

- Split the data into training, validation, and test sets.

- Calculate normalization parameters (like mean and standard deviation) from the training data.

- Normalize the training, validation, and test data using these parameters.

- Save the processed data and normalization parameters to files.

2. \*\*preprocess\_functions.py\*\*: This file contains functions that are used by `preprocess\_data.py` and other scripts to handle different preprocessing steps.

3. \*\*process\_new\_data.py\*\*: This script will preprocess new data using the same normalization parameters saved during the initial preprocessing. This ensures that any new data is processed in the same way as your training data.

4. \*\*train\_model.py\*\*: This script will train your neural network using the preprocessed training data.

5. \*\*evaluate\_model.py\*\*: This script will evaluate your trained model on the validation and test data to see how well it performs.

Documentation

- \*\*README.md\*\*: This file provides an overview of your project, instructions on how to run the scripts, and explanations of what each script does.

- \*\*requirements.txt\*\*: This file lists all the Python packages needed to run your project, which can be installed using a package manager like `pip`.

By organizing your project in this way, you ensure that it’s clear, reproducible, and easy for others to understand and use.