

Directory organization structure for aimo-competition project(files):

Project Directory Structure:

```
aimo-competition/ ├── README.md # Project documentation ├── requirements.txt # Project
dependencies ├── setup.py # Package setup configuration ├── configs/ # Configuration
files | ├── dev_config.yaml # Development settings | ├── prod_config.yaml # Production
settings | └── test_config.yaml # Testing settings ├── src/ # Source code | ├── core/
# Core agent functionality | | ├── agent_zero.py | | └── basic_reasoning.py | ├──
data/ # Data processing | | ├── ingestion.py | | ├── preprocessor.py | | ├──
data_cleaner.py # Your data cleaning script | | ├── validation_script.py # Your
validation script | | ├── parsed_ArtOfProblemSolving.csv # Original dataset | | └──
cleaned_dataset.csv # Cleaned dataset | ├── validation/ # Validation tools | | ├──
validator.py | | └── quality_checks.py | ├── utils/ # Utilities | ├── logging_utils.py
| ├── submission_helper.py ├── extensions/ # Extension modules | ├──
advanced_reasoning/ | ├── memory_enhancement/ | ├── accuracy_tools/ ├── notebooks/ #
Jupyter notebooks | ├── development/ # Development notebooks | | ├──
01_data_exploration.ipynb | | ├── 02_model_development.ipynb | | └──
03_validation_tests.ipynb | └── submission/ # Submission notebooks | ├──
final_submission.ipynb | └── submission_analysis.ipynb ├── tests/ # Test suite | ├──
test_core/ # Core tests | ├── test_data/ # Data processing tests | └──
test_validation/ # Validation tests └── docs/ # Documentation ├── api/ # API
documentation └── CONTRIBUTING.md # Contribution guidelines
```

Where to Save Your Files:

1. Data Files:
 - Original Dataset (parsed_ArtOfProblemSolving.csv):
 - Place this file in src/data/.
 - Path: aimo-competition/src/data/parsed_ArtOfProblemSolving.csv
 - Cleaned Dataset (cleaned_dataset.csv):
 - Save the cleaned dataset in the same directory after processing.
 - Path: aimo-competition/src/data/cleaned_dataset.csv
 - Reference Problems and Solutions (if any):
 - If you have a CSV or any structured format of the reference problems, place it in src/data/.
 - For example, aimo-competition/src/data/reference_problems.csv
2. Data Cleaning and Validation Scripts:
 - Data Cleaning Script (data_cleaner.py):
 - Save this script in src/data/.
 - Path: aimo-competition/src/data/data_cleaner.py
 - Validation Script (validation_script.py):
 - Place this script in src/data/ or src/validation/ if it's more aligned with validation tools.
 - Path: aimo-competition/src/data/validation_script.py or aimo-competition/src/validation/validation_script.py
3. Notebooks:
 - Development Notebooks:
 - If you have notebooks for data exploration or processing, save them in notebooks/development/.
 - Example: aimo-competition/notebooks/development/01_data_exploration.ipynb

- Submission Notebooks:
 - Final notebooks for submission go in notebooks/submission/.
 - Example: aimo-competition/notebooks/submission/final_submission.ipynb
4. Utilities and Helpers:
- Any utility scripts that assist with logging, submission, or other tasks should be in src/utils/.
 - Example: aimo-competition/src/utils/submission_helper.py
5. Configuration Files:
- If your scripts use configuration files, place them in configs/.
 - Example: aimo-competition/configs/dev_config.yaml
6. Tests:
- Place any test scripts or test data in the tests/ directory.
 - Data-specific tests can go in tests/test_data/.
 - Example: aimo-competition/tests/test_data/test_data_cleaner.py

Adjusting Your Scripts for the Directory Structure:

To ensure your scripts work correctly within this structure, you'll need to update the file paths accordingly.

Data Cleaning Script (data_cleaner.py):

File Location: • aimo-competition/src/data/data_cleaner.py

Updated Script with Correct Paths:

```
import pandas as pd
import re
import os
```

Define the paths

```
DATA_DIR = os.path.dirname(os.path.abspath(file))
INPUT_FILE = os.path.join(DATA_DIR, 'parsed_ArtOfProblemSolving.csv')
OUTPUT_FILE = os.path.join(DATA_DIR, 'cleaned_dataset.csv')
```

Load the dataset

```
df = pd.read_csv(INPUT_FILE)
```

... [Rest of your data cleaning code] ...

Save the cleaned dataset

```
df_cleaned.to_csv(OUTPUT_FILE, index=False)
```

Validation Script (validation_script.py):

File Location: • aimo-competition/src/data/validation_script.py or aimo-competition/src/validation/validation_script.py

Updated Script with Correct Paths:

```
import pandas as pd
import re
import os
```

Define the paths

```
CURRENT_DIR = os.path.dirname(os.path.abspath(file))
CLEANED_DATA_FILE = os.path.join(CURRENT_DIR, '../data/cleaned_dataset.csv')
REFERENCE_DATA_FILE = os.path.join(CURRENT_DIR, '../data/reference_problems.csv')
```

Load the cleaned dataset

```
df_cleaned = pd.read_csv(CLEANED_DATA_FILE)
```

Load the reference problems and solutions

```
df_reference = pd.read_csv(REFERENCE_DATA_FILE)
```

... [Rest of your validation code] ...

Note: Adjust the relative paths ('../data/cleaned_dataset.csv') based on the location of your script.

Additional Tips:

- Importing Modules Across Directories:
- If you need to import modules from other directories within src/, ensure you adjust the import statements and manage the Python path.
- Example:

```
import sys
sys.path.append(os.path.abspath(os.path.join(CURRENT_DIR, '..')))
```

```
from data.data_cleaner import clean_data_function
```

- Using Virtual Environments:
- It's good practice to use a virtual environment for your project to manage dependencies specified in requirements.txt.
- Update README.md:
- Document the directory structure and any important information about where files are located and how to run your scripts.
- Version Control:
- Use Git or another version control system to track changes in your project.

Validating Data with Updated Paths:

In your validation_script.py, make sure to update the paths when loading data:

Define the paths

```
CURRENT_DIR = os.path.dirname(os.path.abspath(file)) CLEANED_DATA_FILE =  
os.path.join(CURRENT_DIR, '../data/cleaned_dataset.csv') REFERENCE_DATA_FILE =  
os.path.join(CURRENT_DIR, '../data/reference_problems.csv') # If applicable
```

Load the cleaned dataset

```
df_cleaned = pd.read_csv(CLEANED_DATA_FILE)
```

Load the reference problems and solutions

If the reference data is in a different format, adjust accordingly

```
df_reference = pd.read_csv(REFERENCE_DATA_FILE)
```

Final Notes:

- Data Validation:
- As per your request, the validation is more about verifying the dataset against the competition requirements rather than cleaning. Ensure that after adjusting the file paths, your validation script runs correctly and prints out the results.
- Testing:
- Place any unit tests or validation tests in the tests/ directory. This helps in maintaining code quality and ensures your scripts perform as expected.
- Documentation:
- Consider adding docstrings and comments to your scripts to explain the functionality, which is helpful for future maintenance and for others who may work on your project.

By organizing your files according to this structure and updating your scripts with the correct file paths, you'll have a clean, maintainable project that's easy to navigate and work with.

If you have any further questions or need assistance with specific parts of your project, feel free to ask!