

```

multistepreasoning_system/
%
% % README.md
% % requirements.txt
% % .gitignore
%
% % data/
% % % test.csv
% % % predictions.csv
%
% % src/
% % % init.py
% % % data_loader.py      # code to load datasets
% % % model.py            # model definition and training
% % % predictor.py        # code to generate predictions
% % % utils.py            # helper functions
%
% % reports/
% % % technical_report.pdf # PDF or markdown report
1p README.md Example
markdown

```

Copy code

Multi-Step Reasoning System

Overview

This project implements a system for multi-step reasoning on structured data. It includes:

- Data preprocessing
- Model training and evaluation
- Generating predictions

Folder Structure

- data/: input test dataset and predictions
- src/: Python modules for model and prediction
- reports/: technical report

How to Run

1. Install dependencies:

```
bash
```

```
pip install -r requirements.txt
```

Run prediction:

```
bash
```

Copy code

```
python src/predictor.py
```

yaml

Copy code

2b requirements.txt Example

pandas

numpy

scikit-learn

joblib

yaml

Copy code

3b Sample Code Snippets

`src/data_loader.py`

python

```
import pandas as pd
```

```
def loadtestdata(path="data/test.csv"):
```

```
    return pd.read_csv(path)
```

src/model.py

python

Copy code

```
from sklearn.ensemble import RandomForestClassifier
```

```
import joblib
```

```
class MultiStepModel:
```

```
    def init(self):
```

```
        self.model = RandomForestClassifier()
```

```
    def train(self, X, y):
```

```
        self.model.fit(X, y)
```

```
def predict(self, X):  
    return self.model.predict(X)
```

```
def save(self, path="model.pkl"):  
    joblib.dump(self.model, path)
```

```
def load(self, path="model.pkl"):  
    self.model = joblib.load(path)  
src/predictor.py
```

python

Copy code

```
from data_loader import load_test_data  
from model import MultiStepModel  
import pandas as pd
```

```
def main():  
    test_data = load_test_data()  
    model = MultiStepModel()  
    model.load("model.pkl")
```

```
predictions = model.predict(test_data)
```

```
pd.DataFrame({"Prediction": predictions}).to_csv("data/predictions.csv", index=False)  
print("Predictions saved to data/predictions.csv")
```

```
if name == "main":
```

```
    main()
```

4p ðŸ“šreports/technical_report.pdf (Template in Markdown)

markdown

Copy code

Technical Report

1. System Design and Architecture

- Data preprocessing
- Model training
- Prediction module

2. Problem Decomposition & Reasoning Approach

- Multi-step reasoning
- Feature selection
- Model choice justification

3. Results and Evaluation

- Accuracy, F1 score, etc.
- Observations