Modules & Packages

1️⃣ Module কি?

- যেকোনো .py ফাইল = একটি **Module**
- যেমন:

```
data_loader.py

model.py

utils.py
```

এগুলো আলাদা কাজ করবে, পরে main ফাইলে import করে ব্যবহার করা হবে।

2️⃣ Package কি?

- একটি **folder** যার ভিতরে একটি __init__.py ফাইল থাকে
- Example:

```
ml_project/
    data/
        __init__.py
        loader.py
    models/
         __init__.py
         Linear_regression.py
    utils/
         __init__.py
```

metrics.py

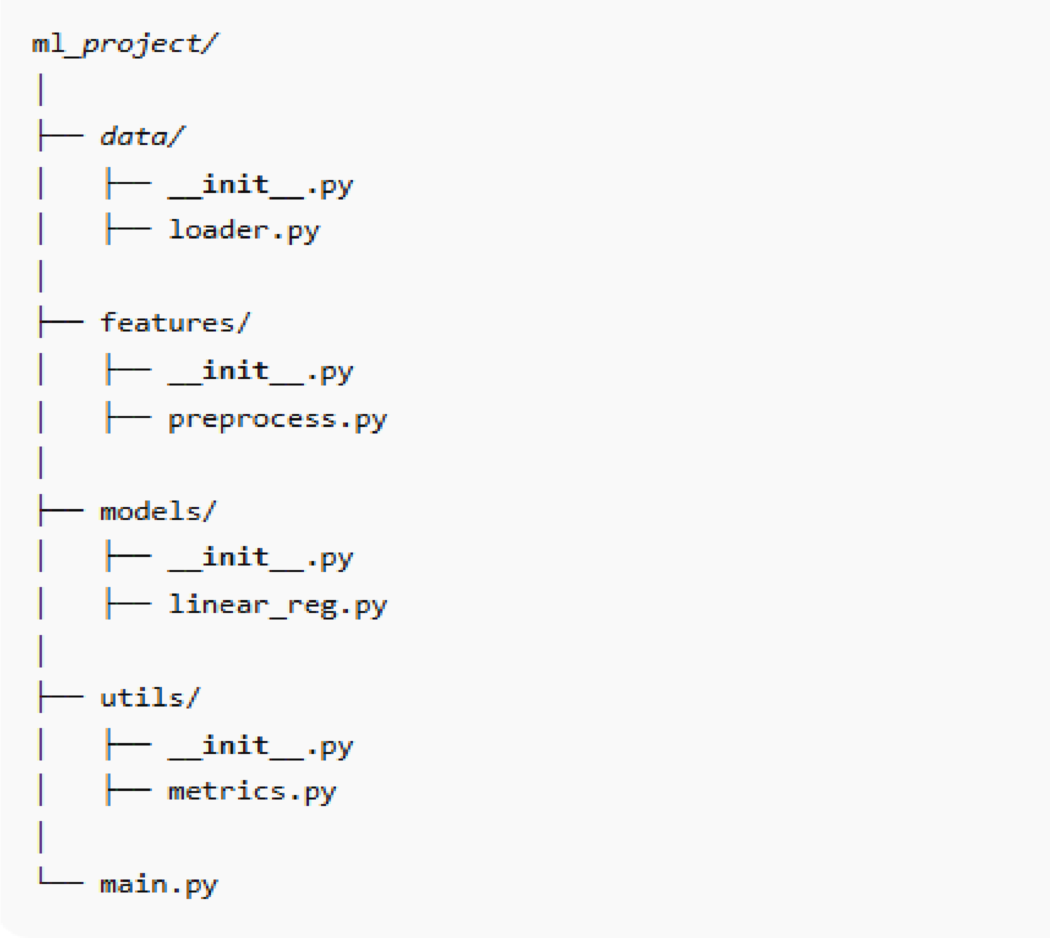
main.py

এটাই ব্যবহার করবে huge ML projects-এ।

3️⃣ কেন AI/ML এ Modules & Packages জরুরি?

- ✔ কোড clean ও reusable থাকে
- ✔ বড় datasets → আলাদা loader module
- ✔ feature engineering → আলাদা module
- ✔ Model train/test/predict → আলাদা packages
- ✔ Experiment pipeline control সহজ হয়
- ✔ Production-level ML code লিখতে রেডি যাও

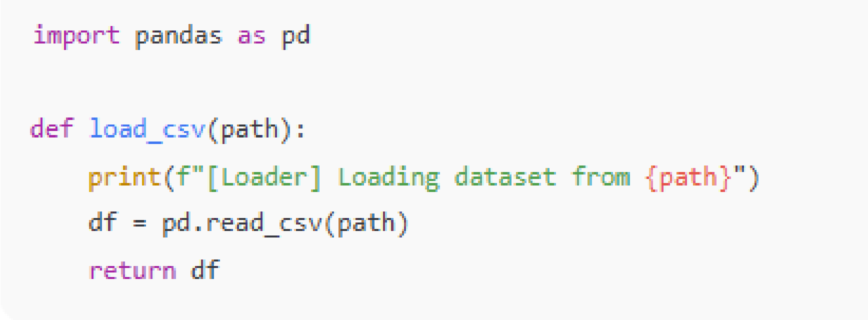
4️⃣ AI/ML Style Project Structure (Best Practice)



```
ml_project/
│
├── data/
│   ├── __init__.py
│   ├── loader.py
│
├── features/
│   ├── __init__.py
│   ├── preprocess.py
│
├── models/
│   ├── __init__.py
│   ├── linear_reg.py
│
├── utils/
│   ├── __init__.py
│   ├── metrics.py
│
└── main.py
```

5️⃣ প্রতিটি Module-এর কোড (AI/ML Ready Example)

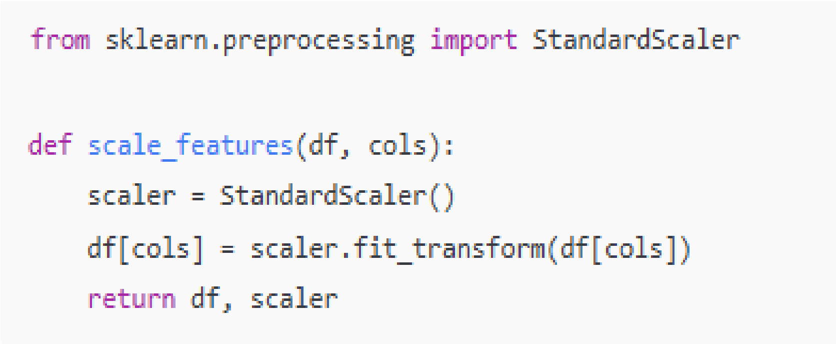
🔵 data/loader.py



```python
import pandas as pd

def load_csv(path):
    print(f"[Loader] Loading dataset from {path}")
    df = pd.read_csv(path)
    return df
```

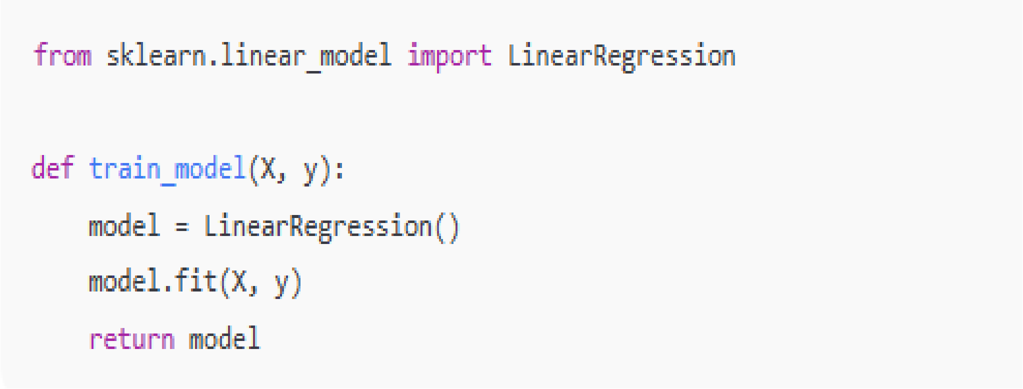
🟣 features/preprocess.py



```python
from sklearn.preprocessing import StandardScaler

def scale_features(df, cols):
    scaler = StandardScaler()
    df[cols] = scaler.fit_transform(df[cols])
    return df, scaler
```

🔴 models/linear_reg.py



```python
from sklearn.linear_model import LinearRegression

def train_model(X, y):
    model = LinearRegression()
    model.fit(X, y)
    return model
```

🟡 utils/metrics.py

```
from sklearn.metrics import mean_squared_error, r2_score

def evaluate(model, X, y):
    y_pred = model.predict(X)
    return {
        "mse": mean_squared_error(y, y_pred),
        "r2": r2_score(y, y_pred)
    }
```

6️⃣ main.py → সব মডিউল IMPORT করে ব্যবহার

```
from data.loader import load_csv
from features.preprocess import scale_features
from models.linear_reg import train_model
from utils.metrics import evaluate

def main():

    # 1) Load data
    df = load_csv("data.csv")

    # 2) Preprocessing
    df, scaler = scale_features(df, ["age", "salary"])

    # 3) Train
    X = df[["age", "salary"]]
    y = df["target"]
    model = train_model(X, y)

    # 4) Evaluate
    scores = evaluate(model, X, y)
    print("Model Scores:", scores)

if __name__ == "__main__":
    main()
```

7️⃣ MODULE IMPORT করার বিভিন্ন পদ্ধতি (AI Engineers Must Know)

✔️ Basic import

import math

✔️ From-import

from math import sqrt

✔️ Alias (short name)

import numpy as np

✔️ Import local module

from utils.metrics import evaluate

✔️ Relative import (within package)

from .metrics import evaluate

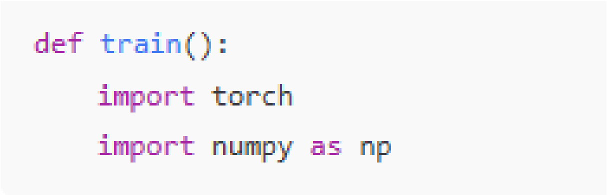
✔️ Import whole package

import features

8️⃣ Advanced Concepts for AI/ML

🔥 (1) Lazy Importing

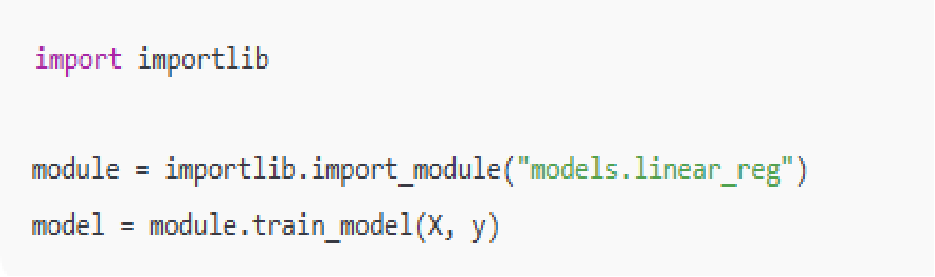
Large ML project → speed optimize:



```
def train():
    import torch
    import numpy as np
```

🔥 (2) Dynamic Import

When many possible models exist:



```python
import importlib

module = importlib.import_module("models.linear_reg")
model = module.train_model(X, y)
```

🔥 (3) Package init.py Re-exports

models/__init__.py:

    from .linear_reg import train_model

Then use:

    from models import train_model

9️⃣ AI/ML Engineers এর MODULE Writing Tips

✔ 1. একটি module → একটি responsibility
✔ 2. main.py শুধু pipeline control করবে
✔ 3. Function ছোট ও পুনর্ব্যবহারযোগ্য রাখো
✔ 4. Global variable ব্যবহার কমাও
✔ 5. Logs ব্যবহার করো (print নয়)
✔ 6. Constant → config.json এ রাখো
✔ 7. Experiment tracking → modules আলাদা করো