# Machine Learning Project Structure

Sunday, 20 April 2025 12:14 PM

## 1. Problem Definition
- Clearly state the problem you're trying to solve.
- Define success metrics (accuracy, F1-score, RMSE, business KPIs, etc.).
- Identify constraints (data, computation, time).

## 2. Data Collection
- Gather relevant data (public datasets, APIs, company data, etc.).
- Assess data availability, quality, and size.

## 3. Data Exploration & Preprocessing
- **Exploratory Data Analysis (EDA):**
  - Summary statistics
  - Visualizations (histograms, scatter plots, correlation matrices)
- **Data Cleaning:**
  - Handle missing values
  - Remove duplicates
  - Correct errors/inconsistencies
- **Feature Engineering:**
  - Create new features
  - Encode categorical variables
  - Feature scaling/normalization
  - Dimensionality reduction (PCA, etc.)
- **Data Splitting:**
  - Train/validation/test split (commonly 60/20/20 or 70/15/15)

## 4. Model Selection
- Choose baseline models (Linear Regression, Decision Trees, etc.).
- Choose advanced models (Random Forest, XGBoost, Neural Networks, etc.).
- Set up cross-validation.

## 5. Model Training
- Train models on training data.
- Tune hyperparameters (Grid Search, Random Search, Bayesian Optimization).

## 6. Model Evaluation
- Evaluate models on validation data.
- Use appropriate metrics (classification: accuracy, precision, recall, ROC-AUC; regression: RMSE, MAE, $R^2$).
- Analyze error cases.

## 7. Model Interpretation
- Feature importance analysis
- Partial dependence plots
- SHAP, LIME, or similar tools for explainability

## 8. Model Deployment (if applicable)
- Export model (pickle, ONNX, TensorFlow SavedModel, etc.).
- Build an API endpoint (Flask, FastAPI, etc.).
- Integrate with existing systems.
- Monitor model performance in production.

## 9. Iterative Improvement
- Gather feedback from test or production environment.
- Refine features, try new models, collect more data.
- Re-train and re-evaluate as needed.

## 10. Documentation & Reporting
- Document all steps, code, and findings.
- Create visualizations and summary tables.
- Prepare a final report or presentation.

```
project_name/
│
├── data/             # Raw and processed data
├── notebooks/        # Jupyter notebooks for EDA, prototyping
├── src/              # Source code (preprocessing, models, utils)
├── models/           # Saved models
├── reports/          # Generated analysis, figures, final report
├── requirements.txt  # Python dependencies
├── README.md         # Project overview
└── .gitignore
```

## 1. Problem Definition
- construct the regime switching model
- the current market situation and forecast the subsequent market conditions
- can decide on the appropriate trading strategies
- the mean and variance of the return in different market states are different
- information criterion like AIC, log likelihood and BIC

## 2. Data Collection
- Gather relevant data (public datasets, APIs, company data, etc.).
- Assess data availability, quality, and size.

## 3. Data Exploration & Preprocessing
- **Exploratory Data Analysis (EDA):**
  - Summary statistics
  - Visualizations (histograms, scatter plots, correlation matrices)
- **Data Cleaning:**
  - Handle missing values
  - Remove duplicates
  - Correct errors/inconsistencies
- **Feature Engineering:**
  - Create new features
  - Encode categorical variables
  - Feature scaling/normalization
  - Dimensionality reduction (PCA, etc.)
- **Data Splitting:**
  - Train/validation/test split (commonly 60/20/20 or 70/15/15)

## 4. Model Selection
- Choose baseline models (Linear Regression, Decision Trees, etc.).
- Choose advanced models (Random Forest, XGBoost, Neural Networks, etc.).
- Set up cross-validation.

## 5. Model Training
- Train models on training data.
- Tune hyperparameters (Grid Search, Random Search, Bayesian Optimization).

## 6. Model Evaluation
- Evaluate models on validation data.
- Use appropriate metrics (classification: accuracy, precision, recall, ROC-AUC; regression: RMSE, MAE, $R^2$).
- Analyze error cases.

## 7. Model Interpretation
- Feature importance analysis
- Partial dependence plots
- SHAP, LIME, or similar tools for explainability

## 8. Model Deployment (if applicable)
- Export model (pickle, ONNX, TensorFlow SavedModel, etc.).
- Build an API endpoint (Flask, FastAPI, etc.).
- Integrate with existing systems.
- Monitor model performance in production.

## 9. Iterative Improvement
- Gather feedback from test or production environment.
- Refine features, try new models, collect more data.
- Re-train and re-evaluate as needed.

## 10. Documentation & Reporting
- Document all steps, code, and findings.
- Create visualizations and summary tables.
- Prepare a final report or presentation.

```
project_name/
│
├── data/             # Raw and processed data
├── notebooks/        # Jupyter notebooks for EDA, prototyping
├── src/              # Source code (preprocessing, models, utils)
├── models/           # Saved models
├── reports/          # Generated analysis, figures, final report
├── requirements.txt  # Python dependencies
├── README.md         # Project overview
└── .gitignore
```