Lesson 45: Project Work Session 1

Data Science with Python – BSc Course

Data Science Program

BSc Course

45 Minutes

Module 10: Capstone & Ethics

L45: Project Work 1 – plan and start building
L46: Project Work 2 – from analysis to story
L47: ML Ethics – the responsibility of prediction
L48: Final Presentations – showtime

Time to apply EVERYTHING you have learned.

48 lessons. 9 modules completed. Now: put it all together.

Learning Objectives

The Context: You have learned Python, pandas, visualization, ML models, NLP, and deployment. Now it is time to put it ALL together in a real project.

After this session, you will be able to:

- Define a clear project scope with measurable goals
- Select appropriate data sources and ML approaches
- Create a structured project plan with milestones
- Begin implementation with a working data pipeline

Finance Application: End-to-end ML project from data to deployment

You Have 4 Weeks. Build a Real Data Science Product.

The Challenge:

- Pick a real problem. Find real data. Build a real model. Deploy it.
- This is NOT a homework exercise – it is a portfolio piece
- Future employers will ask: "Show me something you built"

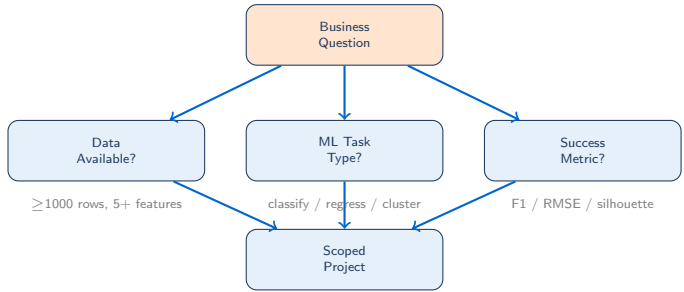
What "Real" Means:

- Real data (not `sklearn.datasets.make_classification`)
- Real question (not "predict everything about stocks")
- Real deployment (accessible via URL, not just a notebook)

The Bar: Could you demo this in a job interview and feel proud?

A well-executed simple project beats a broken ambitious one every time

Project Scope Definition



Bad: "Analyze the stock market"
Bad: "Predict Bitcoin price"

Good: "Predict loan default (F1 $>$ 0.7)"
Good: "Cluster stocks by return patterns (k=5)"

Good scope = specific question + available data + measurable success

Finding Data Sources

Free Finance Data

- **Yahoo Finance** (via `yfinance`): Stock prices, fundamentals
- **FRED**: Macroeconomic indicators, interest rates, GDP
- **Kaggle**: Credit scoring, fraud detection, loan datasets
- **SEC EDGAR**: Company filings (10-K, 10-Q text for NLP)

General ML Datasets

- **UCI ML Repository**: Classic, well-documented datasets
- **Hugging Face**: NLP datasets and pre-trained models
- **Google Dataset Search**: Aggregator across many sources

Data Quality Checklist:

- $\geq$ 1000 rows (ideally 5000+), $\geq$ 5 features
- Target variable clearly defined
- Balanced classes (or you handle imbalance – L28)

Spend time finding good data – garbage in, garbage out applies doubly to projects

EDA Checklist

Before You Touch a Model: Understand Your Data

EDA is not optional – it reveals data problems before they become model problems

Feature Engineering Steps

Transform Raw Data into Model-Ready Features

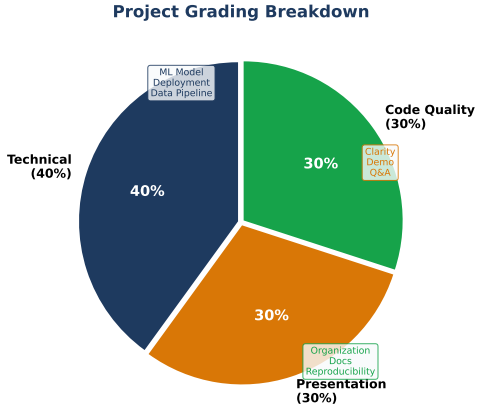
1. **Handle missing values** (L07): Drop columns $> 50\%$ missing, impute the rest
2. **Encode categoricals** (L08): One-hot for < 10 categories, label encode otherwise
3. **Scale numericals** (L22): StandardScaler for linear models; not needed for trees. **Critical:** Fit on training data ONLY
4. **Create domain features:** Finance: log returns, volatility, Sharpe, P/E. General: interactions, polynomial terms

---

Domain features often matter more than model choice – use your finance knowledge

Model Selection Criteria

Pick the Right Tool for Your Problem



Start simple (logistic regression, random forest). Complexity is earned, not assumed.

Checkpoint: What Makes a Project BSc-Worthy?

Pause and Reflect

A BSc-worthy project is NOT:

- Running `model.fit(X, y)` on a Kaggle dataset and reporting accuracy
- A Jupyter notebook with no structure

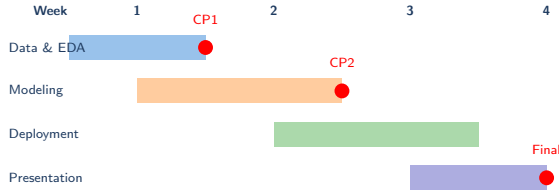
A BSc-worthy project IS:

- A clear question answered with evidence
- Thoughtful feature engineering from domain knowledge
- A justified model choice with proper evaluation
- A deployed product someone else can use

Ask yourself: Would I be comfortable defending every decision in a Q&A?

Understanding WHY is worth more than a high accuracy score

Project Timeline



CP = Checkpoint (show progress to instructor)

Overlap is intentional – start modeling before EDA is "perfect"

Finance Project Ideas

Proven Project Ideas (Students Have Succeeded With These)

- **Credit scoring:** Predict default from borrower features (classification, German Credit dataset)
- **Stock clustering:** Group S&P 500 stocks by return patterns (K-Means, yfinance)
- **Sentiment trading:** News sentiment $\rightarrow$ next-day return direction (NLP + classification)
- **Portfolio dashboard:** Risk-return optimizer with efficient frontier (Streamlit)
- **Fraud detection:** Identify fraudulent transactions (imbalanced classification, Kaggle)
- **Earnings surprise:** Predict if earnings beat estimates (10-K text + fundamentals)

Avoid These Traps:

- "Predict stock price" – everyone tries, nobody succeeds at BSc level
- Proprietary data you cannot share or reproduce
- Scope creep – pick ONE question, not five

Pick something you find genuinely interesting – motivation matters for project quality

Data Quality Assessment

Your Data Is Not as Clean as You Think

- **Completeness:** What percentage of values are missing per column?
- **Consistency:** Are dates in the same format? Are units consistent?
- **Validity:** Are there negative ages? Prices of $0? Duplicates?
- **Relevance:** Do features actually relate to your target variable?

Quick Quality Check (First 10 Minutes):

- `df.info()` – types and non-null counts
- `df.describe()` – ranges, means, suspicious values
- `df.isnull().sum()` – missing value counts
- `df.duplicated().sum()` – duplicate rows

Finance-Specific:

- Check for survivorship bias (only successful companies in data?)
- Look-ahead bias (using future information as a feature?)

---

20 minutes of data quality checks saves 2 hours of debugging bad model results

Reproducibility: Notebooks and requirements.txt

Your Project Must Run on Someone Else's Machine

The Test: Clone your repo, run `pip install -r requirements.txt`, execute – does it work?

Reproducibility Checklist:

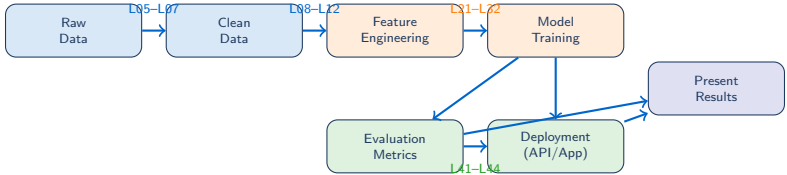
- `requirements.txt` with pinned versions (`pandas==2.1.0`)
- `random_state=42` in every `train_test_split` and model
- No hardcoded paths – use `pathlib.Path` or relative paths
- Data included or download script provided
- Clear README: "How to run this project in 3 steps"

Common Failures:

- "It works on my machine" (missing dependency)
- Data file not in repo (too large? use `.gitignore` + download script)
- Paths like `C:\Users\me\data.csv` (absolute Windows paths)

Reproducibility is a professional skill. Non-reproducible results are non-results.

The Full Project Pipeline



Every lesson in this course maps to a step in this pipeline

Project Structure Best Practices

Recommended Folder Layout

- `data/` – Raw and processed datasets
- `notebooks/` – Jupyter notebooks for EDA
- `src/` – Python modules (data loading, model, utils)
- `app/` – FastAPI or Streamlit deployment code
- `models/` – Saved model files (.joblib)
- `requirements.txt` – All dependencies with versions
- `README.md` – Setup instructions and project description

Code Quality Expectations

- Functions with docstrings (not one giant script)
- Reproducible: `random_state=42` everywhere
- No hardcoded paths – use `pathlib.Path`
- Version control: commit early, commit often

Good project structure makes debugging easier and impresses reviewers

Hands-On: Define Your Project (25 min)

Task: Complete These Three Deliverables

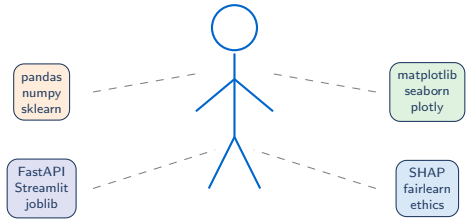
1. **Problem Statement** (fill in the blanks):
   - "I will predict [____] using [____] from [____]"
   - "Success means [____] > [____] on the test set"
2. **Data Loaded and Explored:**
   - `df.shape`, `df.dtypes`, `df.describe()`, missing values
   - At least 2 EDA visualizations (target distribution + correlation)
3. **Project Repo Initialized:**
   - Folder structure created, `requirements.txt` started
   - First commit pushed to GitHub

If Stuck: Use Kaggle's German Credit dataset + predict default with Random Forest

---

Done is better than perfect – get a working baseline, then iterate

The Capstone Mindset



You already have every tool you need.

Now: put them all together.

The capstone is integration, not new learning. You are ready.

Lesson Summary

What You Should Have Done Today:

- Defined a clear, scoped project problem with success metric
- Found and loaded real-world data
- Created initial EDA visualizations
- Set up project folder structure with version control

Before Next Session (L46):

- Complete data cleaning and feature engineering
- Run at least one baseline model
- Decide: API (FastAPI) or dashboard (Streamlit)?

Next Session: Project Work 2 (L46) – from analysis to story

Memory: Start simple, iterate. A working simple model beats a broken complex one.

Looking Ahead: L46

Project Work 2: From Analysis to Story

- Complete model training and evaluation
- Build deployment (API or dashboard)
- Draft presentation outline
- Learn to communicate results to non-technical audiences

Come Prepared With:

- Clean dataset loaded in a notebook
- At least one baseline model trained
- Initial ideas for deployment approach

L46 is execution day – come ready to build