Lesson 46: Project Work Session 2

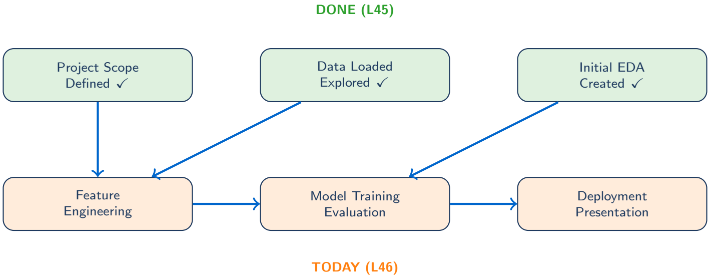
Data Science with Python – BSc Course

Data Science Program

BSc Course

45 Minutes

Previously in L45...



You have the foundation. Today: build the house.

Learning Objectives

The Context: You have your data and initial EDA from L45. Now build the ML model, evaluate it properly, and start telling the story.

After this session, you will be able to:

- Train and compare at least two ML models properly
- Communicate results to both technical and non-technical audiences
- Build a deployment that others can interact with
- Prepare a presentation that makes people CARE about your results

Finance Application: Complete ML pipeline from features to deployed prediction

Your Model Works. Now: Make Others CARE.

The Hard Truth:

- A model with 92% accuracy means nothing if nobody understands it
- Your manager does not care about your loss function
- Your client cares about "Will I get the loan?" not "The AUC is 0.87"

Two Audiences, Two Languages:

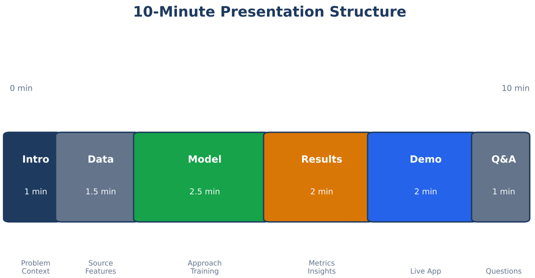
- **Technical:** "Random Forest outperformed logistic regression (F1: 0.84 vs 0.71) due to non-linear feature interactions"
- **Executive:** "Our model correctly identifies 84% of default cases, saving an estimated 2M annually in bad loans"

Today's Goal: Build the model AND the story around it

The best model in the world is worthless if nobody understands or trusts it

Result Visualization

Show, Do Not Tell



One clear chart is worth 100 lines of printed metrics

Writing the Executive Summary

One Page That Tells the Whole Story

Structure (4 paragraphs):

1. **Problem:** What business question did you address?
2. **Approach:** What data and methods did you use?
3. **Results:** What did you find? (1–2 key metrics)
4. **Impact:** What should the reader DO with this information?

Finance Example:

- "We built a credit scoring model using 10,000 historical loan applications. A Random Forest classifier identifies likely defaults with 84% precision, reducing expected losses by 23%. We recommend integrating this model into the loan approval workflow with human review for borderline cases."

Rule: If your executive summary needs more than one page, it is not a summary.

Write the executive summary FIRST – it forces you to clarify your own thinking

Document Your Process So Others Can Reproduce It

What to Include:

- **Data source:** Where, when, how much, any filtering applied
- **Preprocessing:** Missing value strategy, encoding, scaling
- **Feature engineering:** What you created and why
- **Model:** Algorithm, hyperparameters, cross-validation strategy
- **Evaluation:** Metrics chosen and why, train/test split ratio

Do NOT Write:

- "I imported pandas and loaded the data" (obvious)
- "I tried 12 models" without explaining why you settled on one
- Vague claims: "The model performed well" (compared to what?)

Methodology = your recipe. Someone else should be able to reproduce your results.

Explaining Model Choices

Why THIS Model? (The Question You Will Be Asked)

Strong Justification:

- "I chose Random Forest because my features have non-linear relationships (visible in EDA scatter plots) and I have a mix of numerical and categorical features"
- "I started with logistic regression as a baseline, then tried Random Forest. RF improved F1 from 0.71 to 0.84"

Weak Justification:

- "Random Forest is popular" (so what?)
- "I tried everything and RF was best" (why was it best?)
- "My friend used it" (not a reason)

The Pattern: Baseline → why it falls short → better model → evidence

Model choice should follow from your data and problem, not from popularity

Evaluation: Choose the Right Metric

Classification Projects

- Balanced classes → Accuracy is fine
- Imbalanced classes → F1-score, precision, recall
- Cost-sensitive → Custom threshold on ROC curve
- Finance example: Fraud detection needs high recall

Regression Projects

- General: RMSE (same units as target), R^2
- Outlier-heavy: MAE (more robust than RMSE)
- Finance: Directional accuracy (predict the sign correctly?)

For Your Presentation:

- Report at least 2 metrics
- Show confusion matrix (classification) or residual plot (regression)
- Compare train vs test performance (overfitting check)

Metric choice should reflect your business problem, not just convenience

Checkpoint: Technical vs Executive Audience

Same Results, Different Framing

Consider this result: Your fraud detection model catches 91% of fraud but flags 8% of legitimate transactions.

For a Data Science Team:

- Recall = 0.91, Precision = 0.72, F1 = 0.81
- Trade-off: Lowering threshold to 0.3 improves recall to 0.95 but drops precision to 0.58
- Recommend threshold optimization based on cost matrix

For the VP of Risk:

- "We catch 91% of fraudulent transactions automatically"
- "8% of legitimate transactions get flagged for manual review"
- "Estimated annual savings: 1.2M in prevented fraud"

Which changes? Language, detail level, emphasis. **What stays?** Honesty, evidence.

Know your audience. Same data, different story. Both truthful.

Interpretation vs Prediction

What Does Your Model Actually Tell You?

Prediction: "This loan applicant has a 73% probability of default"

- Useful for: automated decisions, scoring, ranking
- Does NOT tell you: why, or what to change

Interpretation: "High debt-to-income ratio and short employment history are the main risk factors"

- Useful for: understanding, policy, advising the applicant
- Requires: feature importance, SHAP values (L47)

For Your Project:

- Report BOTH: what the model predicts AND which features drive predictions
- Use `model.feature_importances_` (trees) or SHAP for any model

Prediction without interpretation is a black box. Add SHAP or feature importance.

Limitations and Future Work

Every Project Has Limitations. Acknowledge Them.

Common Limitations (Be Honest):

- "Dataset is from 2018 – market conditions have changed"
- "Only 2000 samples – may not generalize to larger populations"
- "Model assumes stationarity – may degrade in regime changes"
- "No external validation on out-of-sample data"

Future Work (Show You Thought Ahead):

- "Would test with more recent data from 2024"
- "Would add SHAP explanations for regulatory compliance"
- "Would A/B test against current manual process"

Why This Matters: Acknowledging limitations shows maturity. Hiding them shows inexperience.

An honest "I do not know" earns more respect than a fabricated answer

Finance: Presenting Backtest Results

Special Rules for Finance Projects

Backtest Presentation Must Include:

- Time period and data frequency (daily? monthly?)
- Transaction costs assumed (even if zero, state it)
- Benchmark comparison (S&P 500, buy-and-hold)
- Drawdown analysis (worst-case loss periods)

Red Flags Reviewers Look For:

- Look-ahead bias (using future data in features)
- Survivorship bias (only successful companies in dataset)
- Overfitting to specific time period (in-sample vs out-of-sample)
- "My model beats the market" without transaction costs

Honest Statement: "On historical data, the strategy returned X% annually with Y% maximum drawdown, compared to Z% for buy-and-hold, before transaction costs."

In finance, an honest backtest with modest returns beats an inflated one every time

Report Structure

1. **Executive Summary** – one paragraph, the whole story → Write LAST
2. **Problem & Data** – question, source, EDA highlights
3. **Methodology** – preprocessing, features, model choice
4. **Results** – metrics, comparison table, key charts → Most time here
5. **Deployment** – how to use the API / dashboard
6. **Limitations & Future Work** – honest assessment
7. **Appendix** – code, additional charts, references

Executive summary written last but placed first. Results section gets the most attention.

Code Documentation

Your Code IS Part of Your Submission

Minimum Documentation:

- `README.md`: How to install, run, and use your project (3 steps)
- Docstrings on every function: inputs, outputs, purpose
- Comments for non-obvious logic (not `# increment counter`)
- Type hints for function signatures

README Template:

- **Title**: One-line project description
- **Setup**: `pip install -r requirements.txt`
- **Run**: `streamlit run app.py`
- **Data**: Where to get it (link or included)
- **Results**: Summary of findings

If the instructor cannot run your project in 2 minutes, you lose points

Reproducibility Checklist

Before You Submit: The Final Quality Gate

- ☐ Fresh environment test: `pip install -r requirements.txt` + run
- ☐ All random seeds set (`random_state=42`)
- ☐ No absolute file paths (only relative or `pathlib`)
- ☐ Model saved with `joblib.dump()` in `models/`
- ☐ Scaler fit on train data ONLY (no data leakage)
- ☐ At least 2 models compared on same metric
- ☐ Confusion matrix or residual plot included
- ☐ README explains setup in ≤ 3 steps

Bonus: Clean git history, `.gitignore` for data/cache, type hints

Treat your project like production code – your future employer might see it

Peer Review Process

Before Presenting: Get Feedback from a Classmate

Peer Review Steps:

1. Swap project repos with a partner
2. Clone their repo and try to run it (5 min)
3. Review their EDA: Do the charts tell a story?
4. Check their model: Is the evaluation fair?
5. Give 2 compliments and 2 suggestions

What to Look For:

- Can you run the project? (reproducibility)
- Do you understand the problem from the README?
- Is the model evaluation honest (no data leakage)?
- Would the demo work in a presentation?

Peer review catches problems you cannot see in your own work

From Analysis to Story

```
accuracy: 0.847
precision: 0.812
recall: 0.891
f1: 0.850
auc: 0.923
```

Your Job →

"Our model catches 89% of defaults while maintaining 81% precision, saving an estimated 1.5M per year in bad loan losses."

Raw Numbers

The Story

Data science is the art of turning numbers into decisions. Tell the story.

Lesson Summary

What You Should Have Done Today:

- Completed feature engineering and model training
- Evaluated at least 2 models with appropriate metrics
- Started building deployment (API or dashboard)
- Drafted executive summary and presentation outline

Before Presentation Day (L48):

- Finalize deployment – make sure demo works
- Create 3–5 presentation slides
- Practice your 5-minute presentation
- Prepare for Q&A (anticipate obvious questions)

Next Session: ML Ethics (L47) – fairness, bias, and responsible AI

Memory: Working demo > perfect slides. Prepare your backup plan (screenshots).

Looking Ahead: L47

ML Ethics: The Responsibility of Prediction

- Your model denies a loan. Was it fair?
- Bias in data, fairness metrics, explainability
- GDPR and EU AI Act requirements
- Apply ethics concepts to your own project

Think About:

- Could your project's predictions harm someone?
- Do your features include proxy variables for protected attributes?
- Can you explain WHY your model makes a specific prediction?

Ethics is not optional – it is a constraint, like memory or compute