COMP 333 — Final Project Guidelines

End-to-End Data Analytics Pipeline

Prof: Ph.D. Alexis Yanez   TAs: Ph.D. Students Sami Ben Brahim, Jashan Sidhu

Overview

This final project integrates all COMP 333 skills: data retrieval, wrangling, exploratory analysis, predictive modeling, unsupervised learning, and evaluation. Teams of 2–4 students will implement a complete analytics pipeline on a **real-world dataset $\geq$ 1 GB**.

Three Phases:

- **Phase 1** [5%]: Data acquisition, exploration, baseline analysis — **Due: Sunday, March 1, 2026 (Week 7)**
- **Phase 2** [5%]: Advanced modeling and comparative evaluation — **Due: Sunday, April 5, 2026 (Week 12)**
- **Phase 3** [20%]: Complete pipeline and demonstration — **Due: Final exam day/time**

Mandatory: All team members must attend and participate in the live demonstration or receive **zero marks** for the entire project.

Project Goals and Rationale

This final project is designed as the culminating experience of COMP 333, where you synthesize all concepts, techniques, and best practices learned throughout the semester into a single, comprehensive data analytics project. Unlike individual lab assignments that focus on specific skills in isolation, this project challenges you to navigate the complete data science lifecycle from raw data acquisition to actionable insights and model deployment considerations.

Why This Project Matters:

In professional data analytics roles, you will rarely work with pre-cleaned, perfectly formatted datasets. Real-world data is messy, incomplete, and often distributed across multiple sources. This project simulates authentic industry scenarios where you must:

- **Make strategic decisions** about which data sources to use and how to integrate them
- **Navigate technical challenges** such as API rate limits, authentication, inconsistent formats, and missing documentation
- **Balance competing priorities** between model complexity, interpretability, computational cost, and predictive performance
- **Communicate technical findings** to both technical and non-technical audiences through visualizations, reports, and live demonstrations
- **Work collaboratively** in a team environment, dividing labor effectively while maintaining code quality and reproducibility

What Makes a Successful Project:

A successful final project goes beyond simply applying algorithms to data. Exceptional projects demonstrate:

- **Depth of understanding:** You can explain not just *how* your methods work, but *why* you chose them and *when* they are appropriate
- **Critical thinking:** You identify limitations, potential biases, and ethical considerations in your analysis
- **Technical excellence:** Your code is modular, well-documented, reproducible, and follows software engineering best practices
- **Analytical rigor:** You don't just report results; you validate them through cross-validation, and robustness checks
- **Clear communication:** Your visualizations are professional, your writing is concise, and your demonstrations are compelling

- **Intellectual curiosity:** You explore interesting patterns in your data, ask probing questions, and connect findings to real-world implications

The Learning Journey:

This project is structured in three progressive phases to support your learning and ensure consistent progress throughout the semester:

- **Phase 1** establishes the foundation. You'll acquire and explore your dataset, identify data quality issues, and establish baseline performance. This phase helps you understand your data deeply before attempting complex modeling. The idea is your selected database will be used in all the project, the teaching team will validate your choice in this first phase son you can continue using this same database for the next phases. Prior to selecting your database and submitting your phase 1 deliverables **READ CAREFULLY** the entire project's guideline, so your database satisfies the requirements and it is also aligned with the next tasks (i.e. suitable for supervised and unsupervised machine learning).
- **Phase 2** builds sophistication. You'll implement advanced machine learning algorithms, engineer informative features, and apply unsupervised learning techniques. This phase emphasizes comparative analysis and model interpretation.
- **Phase 3** integrates everything. You'll refactor your work into a cohesive, production-ready pipeline, conduct comprehensive evaluations, and defend your choices in a live demonstration. This phase simulates presenting to stakeholders or technical reviewers.

Portfolio-Worthy Work:

View this project as an opportunity to create a portfolio piece that demonstrates your capabilities to future employers or graduate programs. Many students have successfully highlighted their COMP 333 final projects in job interviews, using them to showcase both technical skills and problem-solving abilities. Invest the time to produce work you'll be proud to share.

Embracing AI Tools Responsibly:

As third or fourth-year computer science students entering the workforce, you'll regularly use AI coding assistants and large language models (LLM). This project explicitly permits and even encourages their use—but with a critical requirement: **you must deeply understand every line of code you submit**. The live demonstration component ensures this understanding is genuine, as you'll be asked to explain, modify, and defend your implementation choices in real time.

Learning Objectives

Assess your ability to: (a) Retrieve/integrate data from databases using SQL/Python/APIs; (b) Apply data cleaning: missing values, imputation, outliers, feature selection; (c) Design experimental setups: train/val/test splits, cross-validation, bias-variance trade-off; (d) Implement/evaluate supervised learning (classification/regression) with AUC, Brier score, RMSE, R^2; (e) Implement/interpret unsupervised learning: clustering, dimensionality reduction, matrix factorization; (f) Justify method selection based on data and research questions.

Team & Dataset Requirements

Team: 2–4 students; all must contribute meaningfully; include division-of-labour statement in each deliverable.

Dataset Requirements:

- **Size:** ≥ 1 GB (uncompressed)
- **Source:** Kaggle, UCI ML Repository, data.gov, APIs, web scraping (legal/ethical), database dumps
- **Complexity:** Requires substantial wrangling/cleaning
- **Research:** Supports both supervised and unsupervised tasks, that's means you have a labeled target class as feature for supervised approach, for the unsupervised approach you can just remove that column/feature (e.g. in Titanic database we have the column survived), you can work over a binary class or multi-class problem depending on the research question or context.
- **Prohibited:** Lab datasets (tips, Titanic), pre-cleaned toy datasets, synthetic datasets

- **Database Usage:** You must use the whole database over your project development, you can do so splitting in train/val/test subset, you may use a subset for EDA/DDA in the first tasks.

Phase 1: Data Acquisition & Baseline [5%] — Due: Mar 1, 2026

Deliverables:

1. **Data Retrieval:** Document sources; programmatic retrieval (SQL/API/scraping); handle challenges (rate limits, auth); store raw data.
2. **Wrangling/Cleaning:** Initial audit (missing values, duplicates, outliers); reproducible pipeline; use `quantDDA()/vizDDA()` from Lab 2.
3. **EDA:** Summary statistics; uni/bivariate visualizations; correlation analysis; formulate 2 research questions (1 supervised, 1 unsupervised).
4. **Baseline Model:** Simple model (logistic/linear regression, decision tree); train/val/test split (you can define a default proportion as 70%/15%/15%, if you use another percentages please justify); evaluate with appropriate metrics; discuss performance.
5. **Report:** Jupyter notebook with code, output, visualizations, Markdown explanations, division-of-labour, references.

Rubric (50 marks): Data Retrieval (10), Wrangling (10), EDA (10), Baseline Model (12), Documentation (8).

Phase 2: Advanced Modeling [5%] — Due: Apr 5, 2026

Deliverables:

1. **Advanced Supervised Learning:** Implement ≥ 2 models (Random Forest, XGBoost, SVM, MLP/Neural Networks); hyperparameter tuning; evaluate with multiple metrics; systematic comparison (tables, ROC curves, AUC, confusion matrices); justify best model.
2. **Feature Engineering:** New features (polynomial, interactions, domain-specific, text/time features); feature selection (filter, wrapper, embedded methods); compare performance.
3. **Unsupervised Learning:** Implement dimension reduction SVD or PCA; determine optimal clusters; visualize; evaluate quality; justify appropriateness.
4. **Interpretation:** Feature importance, partial dependence plots; discuss insights; relate to research questions.
5. **Report:** Updated notebook with Phase 1 (revised) + Phase 2 work; updated division-of-labour.

Rubric (50 marks): Advanced Models (15), Feature Engineering (8), Unsupervised Learning (12), Model Comparison (8), Interpretation (7).

Phase 3: Complete Pipeline & Demo [20%] — Due: Final Exam

Deliverables:

1. **End-to-End Pipeline:** Integrate all phases; refactor into modular functions/classes; error handling; reproducible on new data.
2. **Comprehensive Evaluation:** Test set performance; robustness analysis; results tables/visualizations.
3. **Bonus (Optional, +10 marks):** Deep learning, ensemble stacking, time series, anomaly detection, NLP, REST API, interactive dashboard.
4. **Ethical Considerations:** Discuss bias, fairness, privacy; identify limitations; suggest future work.
5. **Final Report:** Comprehensive notebook; executive summary (1–2 pages); table of contents; professional visualizations; complete division-of-labour; references.
6. **Live Demonstration (15–20 min, Mandatory):** Present research questions/dataset; demonstrate pipeline live; highlight findings; discuss challenges; answer questions. Each member presents and answers individual questions.

Rubric (200 marks): Pipeline (30), Evaluation (25), Code Quality (20), Visualizations (15), Insights (20), Ethics (10), Report (20), Demo (50), Individual Contribution (10). **Bonus:** +10 marks possible.

Submission Guidelines

For Each Phase — ZIP file: `Team<X>-Phase<N>.zip` containing:

- Jupyter Notebook (`.ipynb`, all cells executed) + PDF export
- Python scripts (if refactored into modules)
- Provide download instructions in `README.md`
- `README.md`: team names/IDs, reproduction instructions, dependencies, data source links
- `requirements.txt` or `environment.yml`

Code Requirements: Reproducible (no hard-coded paths); relative paths/config files; random seeds; runs without errors.

Demo Booking: Sign up via Moodle; last week of classes (early) or final exam period; all members present.

Use of AI and Coding Agents

Permitted: Use AI tools (GitHub Copilot, ChatGPT, Claude) for boilerplate code, debugging, exploring implementations, documentation. Document AI usage in your report.

Requirements: **You must understand every line of code.** During demos, you'll explain code, modify it on the spot, justify decisions. Inability to explain = **zero marks + academic integrity investigation**.

Prohibited: Submitting code you don't understand; copying notebooks/pipelines without attribution; sharing code with other teams; using prior course materials.

Academic Integrity

Zero-Tolerance: Plagiarism = **zero marks (30% of grade) + departmental discipline**.

Plagiarism includes: Copying from other teams; unattributed online code; outsourcing work; code sharing between teams.

Detection: Automated similarity tools; manual review; individual questioning during demos.

How to avoid it?: Write your own code (AI-assisted OK); cite sources; don't share code; ensure all members understand all parts.

Suggested Resources

Dataset Sources: Kaggle (`https://www.kaggle.com/datasets`), UCI ML Repo, Data.gov, Google Dataset Search, AWS Open Data, Twitter/Reddit APIs, OpenWeatherMap, Yahoo Finance, NASA Open Data.

Dataset Ideas (> 1 GB): E-commerce transactions, social media posts, healthcare records, financial data, transportation logs, energy/climate data, text corpora, image/video datasets, IoT sensor data.

Tools: pandas, NumPy, Dask — Matplotlib, Seaborn, Plotly — scikit-learn, XGBoost, LightGBM, CatBoost — TensorFlow, PyTorch — SHAP, LIME — SQLAlchemy, pymongo — requests, BeautifulSoup, Scrapy.

Grading Summary

Phase	**Marks**	**% of Grade**
Phase 1	50	5%
Phase 2	50	5%
Phase 3	200	20%
Total	**300**	**30%**
Bonus	+10	+1%

Support

Get Help: Moodle Forum (requirements, datasets, tools); Office Hours (conceptual questions); Discussion Sections.

TAs Will: Clarify requirements; suggest approaches; review high-level design.

TAs Won't: Debug code line-by-line; write code for you; make design decisions.

Team Dynamics

Expected: Regular communication (weekly meetings); fair work distribution; mutual support; meet team deadlines.

Conflicts: 1) Resolve within team. 2) Document + consult TAs. 3) Escalate to instructor. Non-contribution addressed via peer evaluation, individual demos, adjusted marks.

This project is your opportunity to demonstrate data analytics mastery end-to-end.

Approach it as a portfolio piece showcasing your skills, analytical thinking, and teamwork.

Be ambitious, be rigorous, and be proud of your work. Good luck!