Big Data Final Project Instructions

Andrii Muzychuk

2025-11-17

Table of contents

1 Overview

This final project is your opportunity to apply the concepts, tools, and techniques learned throughout the course to a real-world big data problem of your choice. You will design and implement a complete big data solution from data ingestion to final deliverable.

2 Project Objectives

- Apply data engineering and analytics skills to solve a meaningful problem
- Communicate insights effectively through a data product
- Document and present your work professionally

3 Topic Selection

3.1 Requirements

- **Choose a domain that interests you**: Healthcare, Finance, E-commerce, Social Media, IoT, Transportation, Energy, Sports, Entertainment, Education, etc.
- **Ensure data availability**: You must have access to a sufficiently large dataset (100K+ records)
- **Define a clear problem statement**: What question are you trying to answer or what problem are you solving?

3.2 Suggested Data Sources

- Public datasets (Kaggle, UCI ML Repository, data.gov, Google Dataset Search)
- APIs (Twitter, Reddit, Financial APIs, Weather APIs)
- Web scraping (ensure compliance with terms of service)
- Synthetic data generation (if appropriate)
- Open data portals (World Bank, WHO, NASA, etc.)

3.3 Technologies Requirements

Although using Databricks is convenient choice, please feel free to explore and use any tools that help with your end goal!

4 Main Deliverables

- Deliverable 1: Data Processing Pipeline (40%)
- Deliverable 2: Data Analysis (30%)
- Deliverable 3: Data Product (30%)

> **Note**
>
> Please note that above is a suggested breakdown, and you can adjust the focus based on your project goals. That is, if data of choice is already clean and well-structured, you can focus more on analysis and product development.

4.1 Deliverable 1: Data Processing Pipeline

Build an end-to-end data pipeline that includes:

- Data Ingestion
  - Collect data from one or more sources
  - Handle batch and/or streaming data as appropriate
  - Implement error handling and data validation
- Data Storage
  - Choose appropriate storage solution(s) (S3, Data Lake, etc.)
  - Justify your storage architecture decisions
  - Implement data partitioning/sharding strategy if applicable/necessary
- Data Processing & Transformation
  - Clean and preprocess raw data
  - Handle missing values, outliers, and data quality issues
  - Perform feature engineering or data enrichment
  - Implement using appropriate framework (pySpark)
- Pipeline Orchestration
  - Automate pipeline execution
  - Tools: Databricks workflows (or similar, e.g. Airflow)
  - [optional] Include scheduling and monitoring capabilities

> Tip
>
> **Deliverable Format**: Documentation
>
> - Architecture diagram showing data flow
> - Code with clear comments
> - README with setup and execution instructions
> - Discussion of scalability considerations

4.2 Deliverable 2: Data Analysis

Conduct comprehensive analysis of your processed data:

- Exploratory Data Analysis (EDA)
  - Statistical summaries and distributions
  - Correlation analysis
  - Trend identification and pattern discovery
  - Visualization of key findings
- Advanced Analytics
  - Time series analysis (if applicable)
  - Clustering, classification, or regression (as relevant)
  - A/B testing or hypothesis testing (if applicable)
  - Network analysis or graph analytics (if applicable)
- Insights & Findings
  - Answer your original problem statement
  - Identify actionable insights
  - Discuss limitations and assumptions

> Tip
>
> **Deliverable Format**:
>
> - Databricks/Jupyter Notebook or similar interactive document
> - Minimum 5 meaningful visualizations
> - Clear narrative explaining your analytical process

4.3 Deliverable 3: Data Product

Data product is something that your team will craft together :) Below are options that may inspire you:

- Option A: Predictive Model
  - Train and evaluate a machine learning model
  - Perform hyperparameter tuning
  - Document model performance metrics (accuracy, precision, recall, F1, RMSE, etc.)
  - Implement model serving/deployment (REST API, batch scoring, etc.)
  - Include model monitoring strategy
  - **Technologies**: SparkML, Scikit-learn, TensorFlow, PyTorch, MLlib, XGBoost, MLflow
- Option B: Interactive Dashboard (Databricks + Streamlit + Genie)
  - Create a web-based visualization dashboard
  - Include multiple interactive visualizations
  - Implement filters, drill-downs, and dynamic updates
  - Ensure responsive design and good UX
  - Add Agent (this can be Databricks genie - should be properly configured)
  - Deploy to an accessible URL (if possible)
  - **Technologies**: Databricks, Streamlit, Plotly, Genie (or alike)
- Option C: Data Application
  - Build a functional web or mobile application
  - Integrate with your data pipeline
  - Provide user interface for data interaction
  - Include real-time or near-real-time capabilities (optional)
  - Deploy with proper error handling and logging
  - **Technologies**: Databricks API, Flask, FastAPI, Django, React, Shiny, Streamlit

> Tip
>
> **Deliverable Format**
>
> - User guide or demo video (3-5 minutes)
> - Technical documentation
> - Deployment instructions
> - Screenshots or live demo link

5 Submission Requirements

5.1 Final Submission Package

- Code Repository (GitHub/GitLab)
  - All source code
  - Configuration files
  - Requirements/dependencies file
  - Comprehensive README.md
- Documentation (PDF or Markdown)
  - Executive summary (1 page)
  - Technical architecture (2-3 pages)
  - Analysis report (2-5 pages)
  - Data product guide (1-3 pages)
  - References and data sources
- Presentation (15 minutes + 5 min Q&A)
  - Problem statement & motivation
  - Architecture overview
  - Key findings from analysis
  - Data product demo
  - Challenges & lessons learned
  - Future improvements
  - End-to-end demonstration

6 Academic Integrity

- All work must be your own (or properly attributed in team projects)
- Cite all data sources and external code/libraries
- You may use online resources but must document usage
- Plagiarism will result in project failure
- Use of AI assistants (ChatGPT, etc.) must be disclosed

7 Evaluation Criteria

- Data Processing Pipeline (~40 points)
  - **Functionality**: Pipeline executes successfully end-to-end
  - **Architecture**: Well-designed, scalable, follows best practices
  - **Code Quality**: Clean, documented, maintainable code
- Data Analysis (~30 points)
  - **Depth**: Thorough and sophisticated analysis
  - **Visualizations**: Clear, informative, and professional
  - **Insights**: Meaningful conclusions and actionable findings
- Data Product (~30 points)
  - **Functionality**: Product works as intended
  - **User Experience**: Intuitive, polished, professional
  - **Technical Implementation**: Properly integrated with pipeline
- Additional Factors
  - **Documentation & Presentation**
  - **Complexity & Ambition**

Good luck with your project!