CS 439: Intro to Data Science

Course Project Guidelines

Created by Naina Chaturvedi

1 General Instruction

You should work on a project that is related to the theme of the course: Intro to Data Science. The scope of data science is broadly defined and can include any aspect of data collection, storage, integration, ML models, algorithms etc. I encourage you to pick a problem you are excited about and will be flexible if the project is relevant to topics and research papers in lectures.

2 Group Formation and Submit the Project Topic

You can form a project group (of max 3 people) and decide/submit the project topic by Nov 13th.

3 Project Guidelines

3.1 Components

Your project should cover both Data Science and ML components.

Project - Formulate your own project question related to Intro to Data Science and attempt to provide a solution, which can be partial/complete with preliminary results. Your project can be computational, theoretical, experimental, or empirical. A project can be done individually (not recommended) or by a group of max 3 students (recommended, but no more than 3).

3.2 Extra Notes

- **Start Early:** Start thinking about your project early and spend enough time developing it.

4 Deliverables

4.1 1. Project Proposal

Due: 23:59pm, Nov 13th, 2025 (encouraged to submit early)

Purpose: The written proposal should define your project/research question and explain what you are planning to do.

Length: 2 pages (no more than 3), typed, single-space.

Content:

- **Define Project:** What problem are you solving? What strategic aspects are involved? How does your project relate to the lectures/papers we discussed?
- **Novelty and Importance:** Why is your project important? Why are you excited about it? What are some existing issues in current data science practices? Are there any prior related works? Provide a brief summary.
- **Plan:** Be specific and succinct.
  - What kind of data will you use (if any)? How will you get it? Will you create or simulate it?
  - What models/techniques/algorithms do you plan to use or develop?
  - What would be your implementation steps? How will you evaluate your method? How will you test and measure success?

4.2 2. Final Report

Due: 23:59pm, Dec 9th, 2025

Length: Normally, a well-explained project would take 6-8 pages, typed, single-space.

Content:

- **Project Definition:**
  - What problem are you solving? What strategic aspects are involved? How does your project relate to the lectures/papers we discussed?
- **Novelty and Importance:**
  - Why is your project important? Why are you excited about it? What are some existing issues in current data science practices? Are there any prior related works? Provide a brief summary.
  - Depending on your individual case, the above two aspects can be an extended or revised version of what you have written in your proposal.
- **Progress and Contribution:**
  - What kind of data did you use (if any)? How did you get it?
  - What models/techniques/algorithms did you use or develop?
  - What experiments did you design?
  - What are the key findings or results from your project? Did they verify or refute your original hypothesis? How did you evaluate your method?
  - Discuss the advantages and limitations of your approach.
- **Changes After Proposal:**
  - If your final report differs from your proposed project, discuss the differences, why you made certain changes, and the bottlenecks that prevented you from proceeding with the proposed project.

Note to all: You may use tools to help with your writing but do not use generated contents directly. Please cite any tools, web sources, papers, and textbooks you consult/use. You are responsible for the content of your writing, including its originality and correctness. Plagiarism is not allowed.

5 Project Suggestions/Examples

Refer to projects on Cubits (Links to be shared later).

6 Example Projects and Their Structure

Note: These are sample projects and their structure. You can define your own structure for your project.

6.1 1. Customer Churn Prediction for a Telecom Company

Objective: Predict customer churn using historical customer data and machine learning techniques.

Steps and Technologies:

A. Data Collection:

- Extract data from company's CRM system via API
- Collect customer reviews through web scraping
- Set up real-time data collection from service logs
- Implement data quality checks at collection points

B. Data Storage Strategy:

- Store structured data with proper organization
- Implement storage solutions for unstructured data (customer reviews, call logs)
- Create data archival procedures for historical data
- Organize data files with appropriate naming conventions

C. Data Integration:

- Create data pipelines for combining multiple data sources
- Implement data synchronization procedures
- Set up real-time data integration monitoring

- Use workflow management tools for automation

D. Data Cleaning:

- Handle missing values using Pandas
- Remove duplicates and standardize formats
- Create data quality reports
- Implement automated data cleansing procedures

E. Data Transformation:

- Create feature engineering pipelines
- Normalize and scale features
- Generate derived metrics for churn prediction
- Create aggregated views for analysis

F. Exploratory Data Analysis:

- Visualize customer behavior patterns using Matplotlib
- Create distribution analysis using Seaborn
- Generate correlation studies
- Build interactive dashboards

G. Model Building:

- Develop machine learning models (logistic regression, random forests)
- Implement cross-validation procedures
- Create model validation pipelines
- Set up model versioning

H. Evaluation and Deployment:

- Create model performance metrics
- Evaluate model accuracy and effectiveness
- Deploy model for prediction
- Monitor model performance

6.2 2. Real-Time Sentiment Analysis on Social Media

Objective: Analyze and visualize the sentiment of social media posts (tweets) in real-time with robust data processing capabilities.

6.2.1 Detailed Steps and Technologies

1. Data Collection System

a) Twitter API Integration:

- Set up Twitter Developer account
- Implement OAuth authentication
- Create API rate limiting handling
- Set up error handling and retry logic
- Implement streaming connection management

b) Data Collection Pipeline:

- Create tweet filtering by keywords
- Implement language detection
- Set up geolocation tracking
- Create user metadata collection
- Implement hashtag tracking

c) Real-time Data Validation:

- Create input data validation rules
- Implement duplicate detection
- Set up data quality checks
- Create error logging system
- Implement data correction procedures

2. Data Streaming Architecture

a) Apache Kafka Setup:

- Configure Kafka clusters
- Create topic partitioning strategy

- Set up consumer groups
- Implement message serialization
- Configure retention policies

b) Stream Processing:

- Implement Kafka Streams for processing
- Create real-time filtering logic
- Set up stream aggregations
- Implement windowing operations
- Create fault tolerance handling

3. Data Cleaning Pipeline

a) Text Preprocessing:

- Remove URLs and special characters
- Clean hashtags and mentions
- Handle emoji conversions
- Implement language-specific cleaning
- Create text normalization procedures

b) Data Standardization:

- Implement timestamp standardization
- Create user ID normalization
- Set up location data standardization
- Implement metadata formatting
- Create data validation checks

4. Natural Language Processing

a) Text Processing:

- Implement tokenization
- Create stop word removal
- Set up lemmatization/stemming

- Implement part-of-speech tagging
- Create named entity recognition

b) Sentiment Analysis:

- Implement VADER sentiment analyzer
- Create TextBlob integration
- Set up custom sentiment rules
- Implement sentiment score normalization
- Create confidence scoring

5. Real-time Processing System

a) Stream Processing Pipeline:

- Create real-time processing workers
- Implement parallel processing
- Set up batch processing for backlog
- Create processing queue management
- Implement error handling

b) Performance Optimization:

- Implement caching strategies
- Create load balancing
- Set up resource scaling
- Implement performance monitoring
- Create optimization feedback loops

6. Data Storage and Management

a) Raw Data Storage:

- Implement tweet storage procedures
- Create metadata management
- Set up historical data archiving
- Implement data compression

- Create storage optimization

b) Processed Data Management:

- Create sentiment results storage
- Implement aggregation storage
- Set up trend data management
- Create data access layers
- Implement data lifecycle management

7. Analysis and Aggregation

a) Real-time Analytics:

- Create rolling sentiment averages
- Implement trend detection
- Set up anomaly detection
- Create volume analysis
- Implement topic clustering

b) Statistical Analysis:

- Create sentiment distribution analysis
- Implement correlation studies
- Set up significance testing
- Create predictive modeling
- Implement pattern recognition

8. Visualization System

a) Dashboard Development (Plotly Dash):

- Create real-time sentiment graphs
- Implement trend visualizations
- Set up geographic visualizations
- Create interactive filters
- Implement custom visualizations

b) Reporting System:

- Create automated report generation
- Implement custom metrics calculation
- Set up visualization templates
- Create export functionality

6.3 3. Sales Forecasting for an E-commerce Platform

Objective: Develop a comprehensive sales forecasting system to predict future sales patterns using historical data with robust data science and real-time updating capabilities.

6.3.1 Detailed Steps and Technologies

1. Data Storage Architecture

a) Data Organization:

- Design file structure for products catalog management
- Organize sales transactions data with time-series optimization
- Structure customer data for buyer behavior analysis
- Set up inventory management data files
- Create forecast results storage
- Implement seasonal patterns data files

b) Data Performance Optimization:

- Configure efficient data indexing strategies
- Implement data partitioning by date ranges
- Optimize data access patterns
- Create efficient data structures
- Implement data caching mechanisms

c) Data Management:

- Set up automated backup procedures
- Create data archiving policies
- Implement monitoring and alerting

- Configure data redundancy
- Create disaster recovery procedures
- Set up data retention policies

2. Data Collection and Integration

a) Historical Data Collection:

- Extract historical sales data
- Collect product catalog information
- Gather customer purchase histories
- Import inventory movement data
- Collect pricing history
- Extract promotional campaign data

b) Real-time Data Collection:

- Set up real-time sales tracking
- Implement inventory level monitoring
- Create price change tracking
- Monitor customer behavior
- Track website analytics
- Implement promotional activity tracking

c) Data Integration:

- Create data flow automation
- Set up source system connections
- Implement data transformation rules
- Create error handling procedures
- Set up data validation checks
- Implement logging and monitoring

3. ETL Pipeline Development

a) Data Extraction:

- Create data connectors

- Implement API integrations
- Set up file system readers
- Create data validation rules
- Implement error handling
- Set up extraction scheduling

b) Data Transformation:

- Implement data cleaning rules
- Create data standardization procedures
- Set up data enrichment processes
- Create calculation pipelines
- Implement data quality checks
- Create transformation logging

c) Data Loading:

- Create loading procedures
- Implement transaction management
- Set up error handling
- Create loading validation
- Implement performance optimization
- Set up loading monitoring

4. Data Preprocessing and Cleaning

a) Data Cleaning:

- Handle missing values
- Remove duplicates
- Clean outliers
- Standardize formats
- Validate data types
- Implement data quality rules

b) Feature Engineering:

- Create time-based features
- Generate seasonal indicators
- Calculate moving averages
- Create lag features
- Implement interaction features
- Generate categorical encodings

5. Time Series Analysis

a) Pattern Analysis:

- Identify seasonal patterns
- Analyze trends
- Detect cyclical components
- Study irregular variations
- Analyze special events impact
- Create pattern documentation

b) Statistical Analysis:

- Perform decomposition analysis
- Calculate correlation studies
- Implement statistical tests
- Create distribution analysis
- Generate summary statistics
- Perform variance analysis

6. Forecasting Model Development

a) Model Selection and Implementation:

- Implement ARIMA models
- Create Prophet forecasting
- Set up exponential smoothing
- Implement machine learning models
- Create ensemble methods

- Set up model comparison framework

b) Model Training:

- Create training pipelines
- Implement cross-validation
- Set up hyperparameter tuning
- Create model validation
- Implement performance testing
- Set up model versioning

7. Model Evaluation System

a) Performance Metrics:

- Calculate MAE (Mean Absolute Error)
- Implement RMSE calculations
- Create MAPE analysis
- Set up accuracy metrics
- Implement bias checking
- Create confidence intervals

b) Model Validation:

- Create validation procedures
- Implement backtesting
- Set up cross-validation
- Create benchmark comparisons
- Implement scenario testing
- Set up sensitivity analysis

8. Real-time Prediction System

a) Prediction Pipeline:

- Create real-time forecasting
- Implement model serving

- Set up prediction scheduling
- Create update procedures
- Implement error handling
- Set up monitoring system

b) Performance Optimization:

- Implement caching
- Create load balancing
- Set up resource scaling
- Implement request queuing
- Create performance monitoring
- Set up optimization procedures

9. Visualization and Reporting

a) Dashboard Development (Flask/Django):

- Create forecast visualizations
- Implement trend displays
- Set up comparative analysis views
- Create performance metrics display
- Implement interactive features
- Set up customizable reports

b) Reporting System:

- Create automated reports
- Implement export functionality
- Set up scheduling system
- Create custom report templates
- Implement notification system
- Set up report distribution