Learning from Data - Final Project

Project Overview

Text based classification, Segmentation, Clustering, Prediction

Course: Learning from Data
Project Type: Team (2-3 members)
Duration: 4 weeks

Objectives

This project has designed to assess your ability to:

1. **Collect and preprocess** real-world text data through web scraping
2. **Apply multiple ML/DL algorithms** to solve classification, segmentation, or clustering problems
3. **Compare and analyze** the performance of different approaches
4. **Present findings** in a clear, professional manner with proper visualizations

Project Requirements

1. Data Collection (20 points)

You must collect your own dataset through web scraping. Minimum requirements:

- **Size:** At least 2,000 text samples (minimum 1,500 for training, 500 for testing)
- **Quality:** Clean, meaningful text with appropriate labels (for supervised tasks)
- **Legality:** Follow website Terms of Service before scrapping

Recommended Data Sources:

SOURCE TYPE	EXAMPLES	TASK SUGGESTIONS
PRODUCT REVIEWS	Amazon, Yelp, Google Reviews	Sentiment classification (positive/negative/neutral)
MOVIE REVIEWS	IMDb, Rotten Tomatoes, Metacritic	Rating prediction, genre classification
SOCIAL MEDIA	Twitter/X, Reddit threads	Topic classification, spam detection
NEWS ARTICLES	News websites, RSS feeds	Category classification, topic clustering
BOOK TEXT	Project Gutenberg, book excerpts	Author identification, genre classification
Q&A FORUMS	Stack Overflow, Quora	Question quality classification

Data Collection Tools you may use:

- BeautifulSoup4 (HTML parsing)
- Scrapy (structured scraping)
- Selenium (dynamic content)
- APIs when available (Twitter API, Reddit API, etc.)

2. Problem Definition (10 points)

Choose ONE of the following:

A. Classification Problem

- **Binary Classification:** 2 classes (e.g., spam vs. ham, positive vs. negative)
- **Multi-class Classification:** 3+ classes (e.g., news categories, star ratings)
- **Required:** Clear class labels, balanced or documented imbalance

B. Clustering Problem

- **Task:** Group similar documents without predefined labels
- **Examples:** Topic discovery in news, customer review themes
- **Required:** Interpretable clusters with clear separation

C. Segmentation Problem

- **Task:** Identify boundaries or segments within text
- **Examples:** Sentence topic segmentation, opinion target extraction
- **Required:** Clear segmentation criteria

3. Implementation Requirements (40 points)

You must implement and compare **at least 4 different approaches** from the following categories:

Category A: Traditional ML (Choose at least 2)

- **Linear Models:**
    - Perceptron / Pocket Algorithm
    - Logistic Regression
    - Linear SVM
- **Non-linear Models:**
    - SVM with RBF kernel
    - k-Nearest Neighbors
    - Radial Basis Functions (RBF Networks)
    - Decision Trees / Random Forests

Category B: Deep Learning (Choose at least 1)

- **Neural Networks:**
  - Multi-layer Perceptron (MLP)
  - Convolutional Neural Networks (CNN) for text
  - Recurrent Neural Networks (RNN/LSTM)
  - Transformer-based models (optional: pre-trained embeddings)

Category C: Unsupervised/Hybrid (If doing clustering)

- k-Means clustering
- Gaussian Mixture Models (GMM)
- Hierarchical clustering

Feature Engineering Requirements:

- Bag-of-Words (BoW)
- TF-IDF
- Word embeddings (Word2Vec, GloVe, or similar)
- At least ONE custom feature set based on domain knowledge

4. Methodology & Analysis (20 points)

Data Preprocessing:

- ✓ Text cleaning (remove HTML, special characters, etc.)
- ✓ Tokenization
- ✓ Stop word removal (justify your choices)
- ✓ Stemming/Lemmatization
- ✓ Handling class imbalance (if applicable)

Training Strategy:

- Train/Validation/Test split (e.g., 70/15/15 or 60/20/20)
- Cross-validation (at least 5-fold)
- Hyperparameter tuning (show your process)

Evaluation Metrics:

For Classification:

- Accuracy
- Precision, Recall, F1-Score

- Confusion Matrix
- ROC curve and AUC (for binary)

For Clustering:

- Silhouette Score
- Within-cluster sum of squares
- Cluster coherence analysis
- Visual inspection (t-SNE/PCA plots)

Analysis Requirements:

- **Learning curves:** Plot training vs. validation error
- **Bias-Variance analysis:** Discuss overfitting/underfitting
- **Comparison table:** All algorithms with key metrics
- **Error analysis:** Analyze misclassifications with examples
- **Computational analysis:** Training time, memory usage

5. Regularization & Overfitting Prevention (10 points)

Demonstrate understanding of overfitting and show at least 2 techniques:

- **L1/L2 Regularization** (weight decay)
- **Early stopping** (for iterative algorithms)
- **Dropout** (for neural networks)
- **Feature selection** or dimensionality reduction
- **Validation-based model selection**

Show learning curves and discuss the bias-variance tradeoff in your models.

Deliverables

1. Code Submission (via GitHub/GitLab)

- Clean, well-documented Python code
- Jupyter notebook with full pipeline
- requirements.txt with all dependencies

- README with setup instructions
- Raw data or clear instructions to reproduce data collection

2. Written Report (PDF, 8-10 pages)

Structure:

a. **Introduction (1 page)**
   - Problem motivation
   - Dataset description
   - Project objectives

b. **Data Collection & Preprocessing (1-2 pages)**
   - Scraping methodology
   - Data statistics and visualization
   - Preprocessing pipeline
   - Challenges encountered

c. **Methodology (2-3 pages)**
   - Feature engineering approaches
   - Algorithm descriptions and justifications
   - Hyperparameter tuning process
   - Training strategy

d. **Results & Analysis (3-4 pages)**
   - Comprehensive comparison table
   - Learning curves and visualizations
   - Error analysis with examples
   - Statistical significance tests (optional but recommended)

e. **Discussion (1-2 pages)**
   - Interpretation of results
   - Bias-variance analysis
   - Limitations and future work
   - Lessons learned

3. Presentation (10 minutes + 5 min Q&A)

- Problem and motivation (1-2 slides)
- Data collection approach (1 slide)
- Methods comparison (2-3 slides)

- Key results (2-3 slides)
- Conclusions and insights (1 slide)

Grading

Component	Points	Criteria
Data Collection	20	Quality, size, legality, documentation
Implementation	40	Algorithm variety, correctness, code quality
Methodology	20	Preprocessing, validation, hyperparameter tuning
Regularization	10	Overfitting prevention, analysis
Report	30	Clarity, analysis depth, visualizations
Presentation	20	Organization, clarity, time management
Code Quality	10	Documentation, reproducibility, organization
Bonus	+10	Novel approaches, exceptional analysis
TOTAL	**150**	(Normalized to 100)

Good luck! This project is an opportunity to apply everything you've learned in this course to a real-world problem. Make it count!

Updated: December 2025 Instructor: Cumali Türkmenoğlu Course: Learning from Data