Jordan University of Science and Technology

DS231 Final project Winter 2025

Deadline: May 18, Noon! Discussions: May 20. Each group must attend based on schedule!

Final Project description

In this project, you will apply advanced unsupervised learning techniques and anomaly detection methods on assigned real-world datasets. You must analyze, cluster, visualize, validate, and extract insights from complex, high-dimensional data.

You are expected to deeply explore the data, perform extensive sensitivity analysis, and properly document your methodology and findings.

Assigned Datasets

You **must** use the following datasets (both required):

1. Credit Card Fraud Detection Dataset (Kaggle)
   - Transactions labeled as fraud or not fraud.
   - Highly imbalanced, anonymized data with 30 features.
2. Mall Customer Segmentation Dataset
   - Data on customer age, income, spending score, etc.
   - A simpler dataset to practice clustering more visibly.

Techniques and Algorithms You Must Apply

1. Dimensionality Reduction

- Apply **PCA** on both datasets:
  - Analyze explained variance.
  - Choose the minimum number of components preserving 90–95% variance.
  - Visualize reduced space (2D scatter plots).
- Apply **t-SNE** for visualization:
  - Visualize clusters formed after PCA.
  - Create 2D t-SNE plots **before** and **after** clustering

2. Clustering Analysis

- **K-Means Clustering**:
  - Use the **Elbow method** and **Silhouette score** to determine best k.
  - Plot **Inertia vs.** ***k*** and **Silhouette score vs.** ***k***.
  - Create **Silhouette diagrams** for at least 3 values of k.
- **K-Means++ Initialization**:
  - Compare standard K-Means and K-Means++.
  - Show how initialization impacts convergence and results.
- **Mini-Batch K-Means**:
  - Apply on both datasets.
  - Compare speed, memory usage, and accuracy with standard K-Means.
- **DBSCAN Clustering**:
  - Tune *eps* and *min_samples* carefully.
  - Analyze how noise points are classified.
  - Plot clusters and compare to K-Means clusters.

3. Anomaly Detection

- Apply to **Credit Card Fraud Dataset** ONLY:
  - **Isolation Forest** for anomaly detection.
  - **One-Class SVM** for anomaly detection.
- Evaluate using:
  - **Precision**, **Recall**, **F1-Score**.
  - **Confusion Matrix** (use actual fraud labels).
- Analyze:
  - How unsupervised models detect fraud vs. true labels.

  - Which method performs better, and why.

Sensitivity Analysis Requirements

For **each clustering technique** (KMeans, MiniBatch KMeans, DBSCAN):

- Vary important hyperparameters:
  - For KMeans: k = 2 to 20
  - For DBSCAN: grid search on eps and min_samples
- Plot and interpret:
  - **Inertia graphs**
  - **Elbow curves**
  - **Silhouette score plots**
  - **Silhouette diagrams**

Additionally:

- Compare clustering performance **with and without** dimensionality reduction (PCA).
- Compare clustering in full space vs. reduced 2D space.

Special Tasks (Mandatory)

- Create a **comparison table** summarizing:
  - Inertia values
  - Silhouette scores
  - Execution times (for KMeans vs MiniBatch)
  - Number of clusters found (for DBSCAN)
- Document:
  - How scaling (StandardScaler) affects clustering.
  - How random seed affects results (especially for t-SNE and KMeans).
- Feature engineering:
  - If applicable, create at least **two new features** based on existing ones.

  - Justify their usefulness.

Deliverables

1. **Technical Report** (embedded in Jupyter itself):
   - Introduction (motivation, datasets description)
   - Data preprocessing and feature engineering
   - Dimensionality reduction (PCA, t-SNE)
   - Clustering experiments (KMeans, MiniBatch KMeans, DBSCAN)
   - Anomaly detection (Isolation Forest, One-Class SVM)
   - Sensitivity analysis results
   - Tables, charts, graphs, interpretation
   - Final insights and discussion
   - Challenges and how you solved them
   - Findings!
   - Work load distribution between team members! If we find something different from what you say, all team members might get the lowest grade in the team!
2. Fully executable Jupyter Notebook:
   - Well-commented
   - Organized sections
   - Code that runs from start to finish

General Important Notes:

- We KNOW that you can get some suggested codes from Kaggle AND other sources, HOWEVER, be very careful that *the requirements of the project are VERY detailed*! Every step counts!
- You will have to do sufficient preprocessing, as needed, tables, figures, plots, and then discussions that support your argument!
- We will check in between the lines, so be ready to be tested in whatever you write and around these concepts to check your level of understanding!

- **Failure to attend the presentation will result in a ZERO no matter what your solution is!**
- **There will be NO MAKEUPS!**