Report of Interplay between Supervision and Learning in Classification

Task 1

1.1 The algorithm for Task 1 draws inspiration from linear Support Vector Machines (SVMs) with two main modifications. Firstly, traditional linear SVMs are designed for binary classification, which doesn't suit task 1 involving three classes. To address this, we employ the One-vs-Rest method, creating one classifier for each class that distinguishes it from the others. For *K* classes, *K* binary classifiers are trained, each determining if a sample belongs to its class or any other class. Secondly, the original linear SVM objective function includes a regularization term, the square root of the sum of the absolute squares of matrix elements. To make it a Linear Problem, we eliminate the regularization term while retaining the Hinge Loss Term, achieving satisfactory results.

$$Minimize \sum_{i=1}^{M} \sum_{k=1}^{K} \zeta_{ik} \tag{1a}$$

$$s.t. \quad y_{ik} \cdot (W_k \cdot X_i + b_k) \geq 1 - \zeta_{ik} \tag{2a}$$

$$\zeta_{ik} \geq 0 \tag{3a}$$

Specifically, the objective function (1a), which is the sum of slack variables, quantifies the overall margin violation across all classes and samples. The slack variables ζ_{ik} gauge how much a sample I in class k violates the margin. A larger value signifies a significant violation or misclassification, whereas zero indicates no violation. Minimizing this sum encourages the classifier to accurately classify as many samples as possible while maintaining the specified margin. This objective function strikes a balance between a wide margin and correct training sample classification, providing flexibility for non-separable cases.

The constraint (1b) ensures that for each sample I and class k if the sample belongs to class k (i.e., y_{ik} = 1), the linear decision function $W_k \cdot X_i + b_k$ should be greater than $1 - \zeta_{ik}$, which represents the desired margin. For samples that don't belong to class k (i.e., y_{ik} = -1), the constraint becomes $-W_k \cdot X_i - b_k \geq 1 - \zeta_{ik}$, allowing these samples to be within or on the wrong side of the margin, with the slack variable measuring this violation. Furthermore, the constraint (1c) ensures that the slack variables are non-negative. This is because they are introduced as penalties for margin violations which measure the degree of violation or misclassification.

1.2 Please refer to the Appendix for the accuracy results of the classifier.

Task 2

2.1 To formulate this task, we use L2-norm as the distance matrix. Firstly, we use Integer Linear Programming (ILP) where the binary indicator is 0 or 1. Then we relax the indicator to all real values in the range [0,1]. This is possible because it will represent the majority vote method where some centroids all contribute to the clustering label. But the largest contribution centroid will be the final label class.

This formulation captures the essence of K-means clustering, aiming to minimize the total within-cluster variation by iteratively updating cluster assignments and cluster centers until convergence. In this algorithm, we first randomly choose the centroids. Then we use Linear Programming (LP) method to assign the cluster and then use the new cluster to compute the mean as new centroids. And we will continue the loop until the cluster assignment doesn't change.

Objective function:

$$Minimize \sum_{i=1}^{M} \sum_{k=1}^{K} \mathrm{Z}_{ij} * \left\| x_i - c_j \right\|^2 \tag{2a}$$

Assignment Constraints: Each data point must be assigned to exactly one cluster.

$$\sum_{j=1}^{K} z_{ij} = 1, \text{for } i = 1,2,\dots,N \tag{2b}$$

Cluster Center Update: Update cluster centers based on the assigned data points.

$$c_j = \frac{1}{|S_j|}\sum_{i \in S_j} x_i, \text{ for } j = 1,2,\dots,K \tag{2c}$$

Where, $X = \{x_1, x_2, \dots, x_N\}$ represents the set of data points, $C = \{c_1, c_2, \dots, c_K\}$ represents the set of cluster centers, Zij is a binary variable indicating whether data point Xi belongs to cluster c_j, Sj represents the set of data points assigned to cluster c_j

2.2 Discussion of the results:

(i) Clustering performance among different K values:

In Figure 2-2, we observe that increasing K leads to higher accuracy, but there are specific reasons for this phenomenon. A larger K enhances cluster granularity, capturing subtle data variations. It also reduces the data spread within clusters, leading to improved accuracy.

However, it is noteworthy that the Normalized Mutual Information (NMI) remains nearly constant irrespective of K. Figure 2-1 shows that this discrepancy between low NMI and reasonable accuracy arises from different clustering interpretations. Accuracy focuses on overall correct classifications, while NMI measures mutual dependence between true labels and cluster assignments. A low NMI implies limited information alignment between clustering and true labels.

Additionally, class imbalances may impact results, as dominant classes can influence accuracy while obscuring issues with minority classes. In some cases, clusters may predominantly contain data from one class but be labeled differently, leading to high accuracy but low NMI due to minimal mutual information.

(ii) Compare the classification results with what we get in Task 1:

Based on Table 1 and Figure 2-2, we observe that the performance comparison between the K-means algorithm and the modified Linear SVM algorithm depends on the dataset characteristics.

For the synthetic dataset, the modified Linear SVM outperforms the K-means algorithm with K values of 3, 5, and 10. This superiority arises from the synthetic dataset's linear separability, aligning well with the Linear SVM's capabilities.

In contrast, for the MNIST dataset, we find that the K-means algorithm with K = 10 excels in performance compared to the modified Linear SVM algorithms. There are two key reasons for this:

Increasing K in K-means generally enhances its ability to learn and adapt to dataset clustering, benefiting performance. The MNIST dataset used in this project has a total of 3 classes and is high-dimensional. The modified Linear SVM, lacking a regularization term, is susceptible to overfitting in such cases.

Task 3

3.1 In this Task, we developed a Linear Programming model for the decision of label selection, intending to maximize the representation of different clusters identified by K-means clustering. The k-means implemented in this task were encapsulated by Sklearn for robust clustering results.

Specifically, we used a binary decision variable for each sample in the training set and the cluster as well. For instance, a cluster k, k_{select} is 1 if at least one point from that cluster is selected for labeling, and 0 otherwise. The objective function is shown as follows:

$$Maximize \sum_{k=1}^{K} k_{select} \tag{3a}$$

For the constraints, the first one was defined by the fixed number of sample labels, it could be limited by the label ratio 5%, 10%, 20%, 50%, and 100%. Also, a cluster representation constraint was implemented, it links the selection of samples to k_{select} for each cluster. Moreover, if any point in cluster k is selected, k_{select} must be 1.

$$\sum_{i=1}^{N} i_{select} = ratio * total\ samples \tag{3b}$$

$$k_{select} \leq \sum_{i \in k}^{N} i_{select} \tag{3c}$$

3.2 As mentioned in 3.1, the data label selection ratio has been pre-defined with multiple values to verify the performance of semi-supervised tasks. Specifically, we used the classifier developed in Task 1 and trained on the labeled data which was decided in Task 3.1 As a result, The accuracy achieved results similar to Task 1 on both datasets, and the higher the label rate, the higher the accuracy. Please refer to the Appendix for more details of the results.

Task 4

4.1 The relationship between unsupervised and supervised learning regarding dataset size and performance is nuanced. Firstly, unsupervised learning serves different purposes than supervised learning. While supervised learning maps inputs to outputs (labels), unsupervised learning uncovers hidden patterns in unlabeled data. Their objectives and success metrics differ. However, more data can benefit unsupervised learning, and data quality and relevance matter significantly. Irrelevant or low-quality data can hinder performance. For example, we simulated the theory for Task 2, please refer to Figure 4-1 for results.

4.2 In general, increasing sample size tends to improve results due to a more comprehensive representation of the data distribution, enabling the detection of subtle patterns and enhancing robustness to outliers and noise. However, this can lead to higher computational complexity, slowing down the clustering process and requiring more resources. Yet, the impact on clustering algorithm performance depends on factors like data structure, chosen algorithm, and clustering goals, which can occasionally result in poorer performance with more samples. Please refer to Figure 4-2 in Appendix for the results.

4.3 Yes, this information can enhance classifier design for several reasons. Firstly, soft clustering provides probabilities indicating each sample's degree of membership in each cluster. These probabilities can serve as additional features for a classifier, offering richer information compared to binary hard labels. Also, soft assignments express data uncertainty, which is valuable for classifiers. In regions where cluster boundaries are ambiguous, soft decisions offer a more nuanced perspective than hard clustering.

Additionally, achieving better performance with a smaller sample size is possible for some reasons. First of all, soft clustering allows leveraging unlabeled data, often more abundant than labeled data, which is particularly useful when labeled data is scarce. In addition, soft clustering helps mitigate overfitting, especially when dealing with limited labeled samples, promoting better model generalization.

4.4 We recommend employing semi-supervised learning, which utilizes a small set of labeled data and a larger set of unlabeled data. The model learns from the labeled data, makes predictions on the unlabeled data, and refines itself using these predictions as pseudo-labels. Task 3 demonstrates that we can achieve strong performance while reducing the need for labeled data.

Alternatively, we can opt to select a small subset of data points from the clusters and then request true labels for only those specific points. This approach helps decrease our reliance on the complete set of true labels while still preserving performance.

Another strategy is to conduct multiple runs of the K-means algorithm with different initializations. Subsequently, we can combine or assign labels based on a consensus between these runs. This approach minimizes the sensitivity to a single set of labels, thereby maintaining strong performance.

Appendix – Experiment Results & Discussion

Task 1

Table 1 Result of Task 1

	Test Accuracy
Synthetic	0.97
MNIST	0.84

Additionally, 2D plots depicting the decision boundary of the classifier and test points for the synthetic dataset, can be found in Figure 1-1 and Figure 1-2.

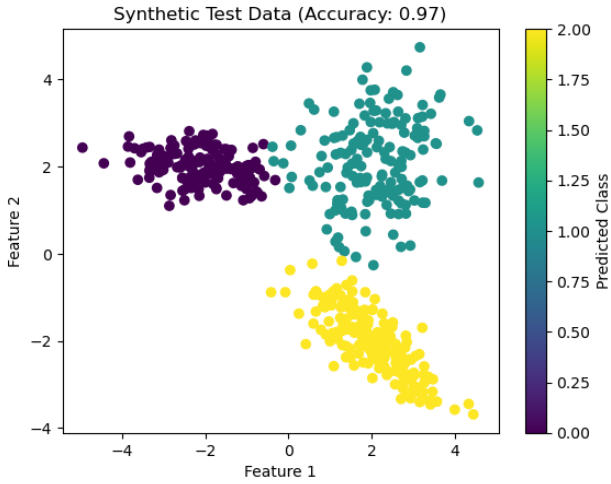

Figure 1-1 Synthetic Test Data Accuracy (Accuracy: 0.97)

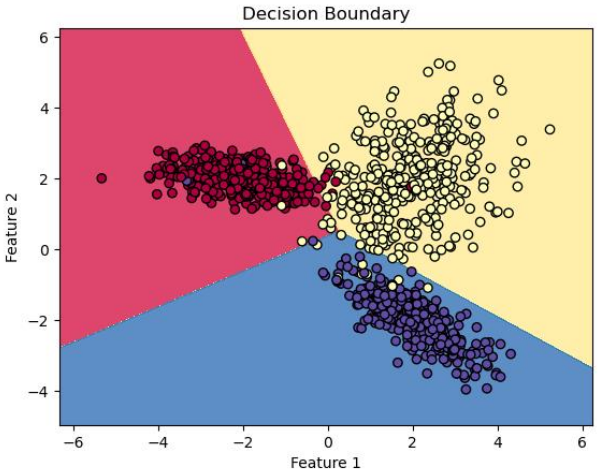

Figure 2-2 Decision Boundary for Synthetic Dataset

Task 2

Table 2 Result of Task 2

Dataset	K	Clustering NMI	Classification Accuracy
Syn Data	3	0.0009698308725354794	0.536
Syn Data	5	0.0037225416263362023	0.702
Syn Data	10	0.007295479809290849	0.92
MNIST	3	0.0	0.302
MNIST	10	0.0032257749162201577	0.444
MNIST	32	0.01041003111548222	0.592

Table 1 shows that the modified Linear SVM algorithm achieves better performance on the Synthetic dataset compared to the MNIST dataset. Several factors contribute to this difference.

Firstly, the MNIST dataset has significantly higher dimensionality (784 dimensions) compared to the Synthetic dataset (2 dimensions). Additionally, the Synthetic dataset is linearly separable, making it inherently easier to classify. Furthermore, the modified Linear SVM is prone to overfitting on the MNIST dataset due to the absence of regularization, which typically helps prevent overfitting by penalizing larger weight values. This lack of regularization can lead to the model fitting the training data too closely and not generalizing well to unseen data, resulting in reduced performance.

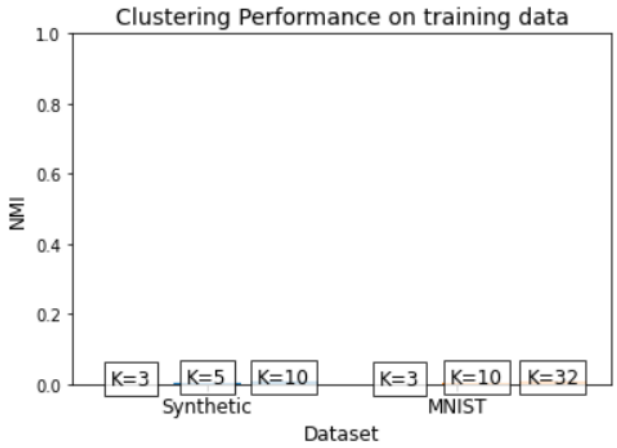

Figure 2-1 NMI Values among Different K Values and Datasets

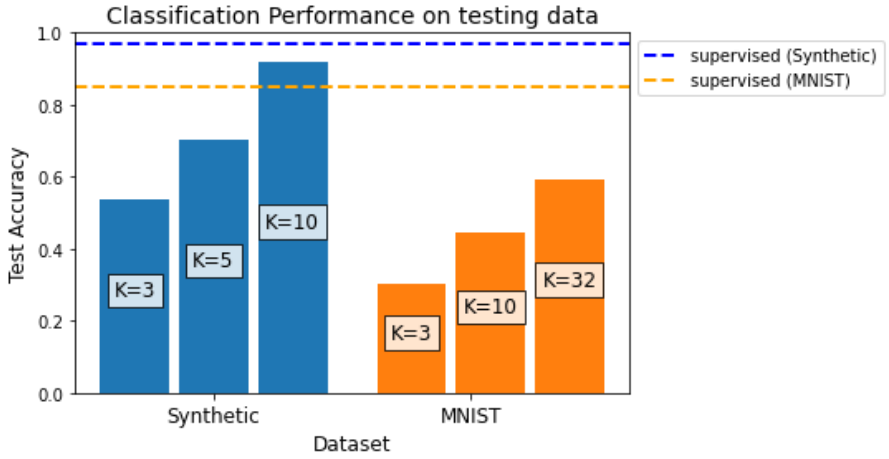

Figure 2-2 K-means Unsupervised Learning Accuracy

Task 3

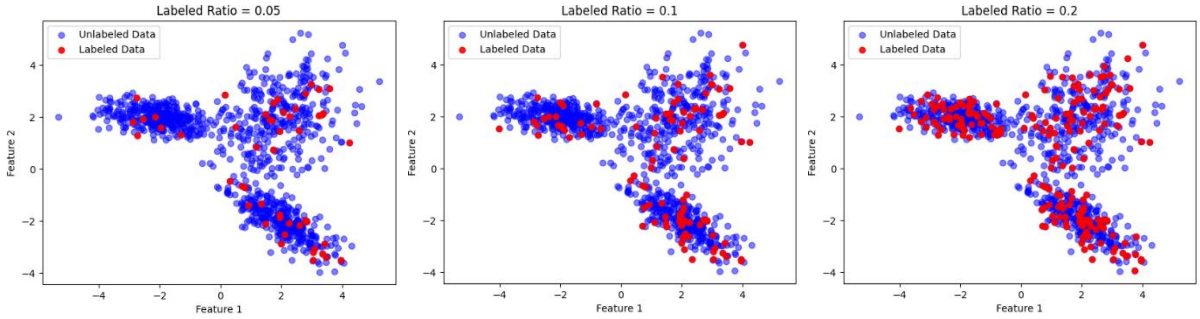

Figure 3-1 Different Label Ratio

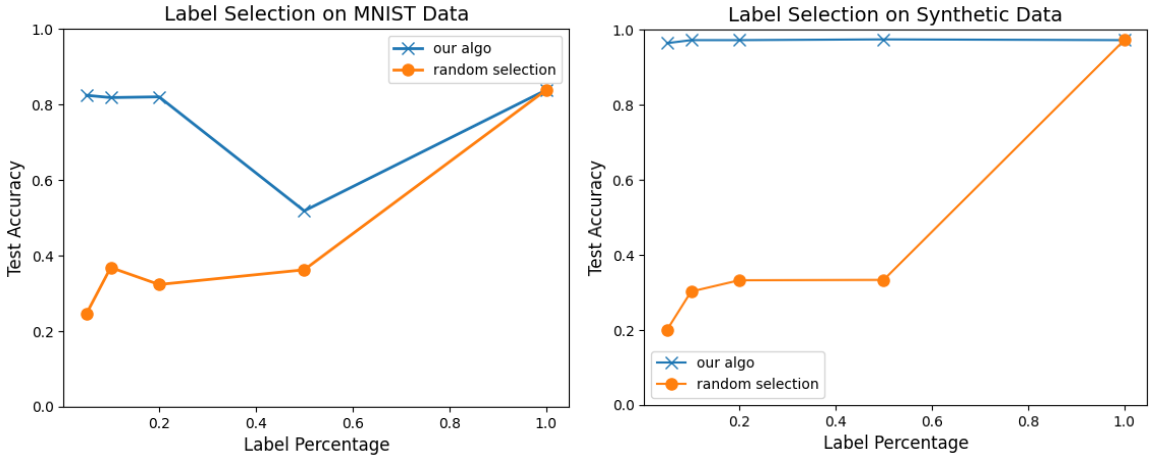

Figure 3-2 Semi-supervised Learning Accuracy

Task 4

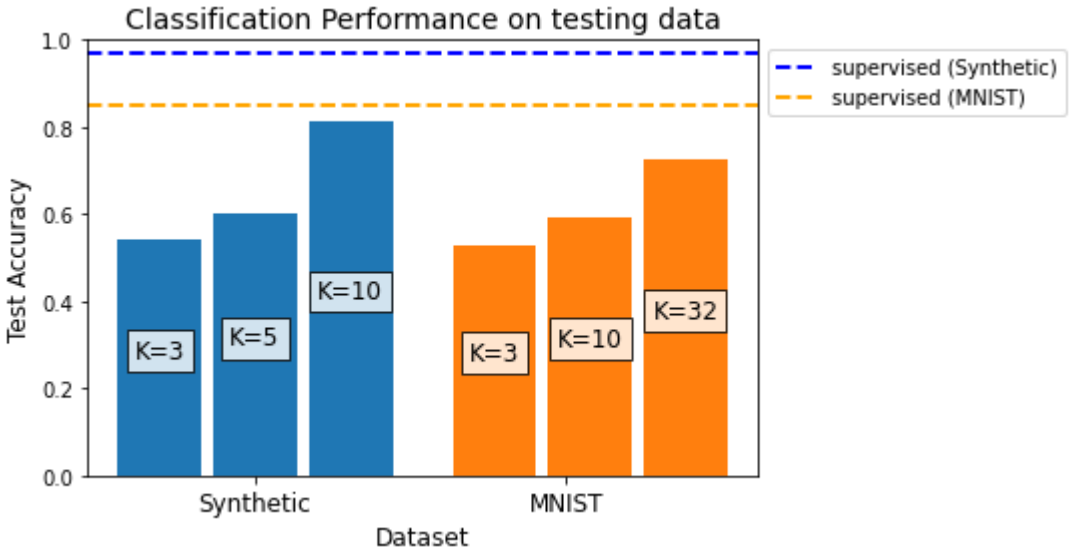

Figure 4-1-1 Unsupervised Learning Accuracy – Decreased to 700 Samples

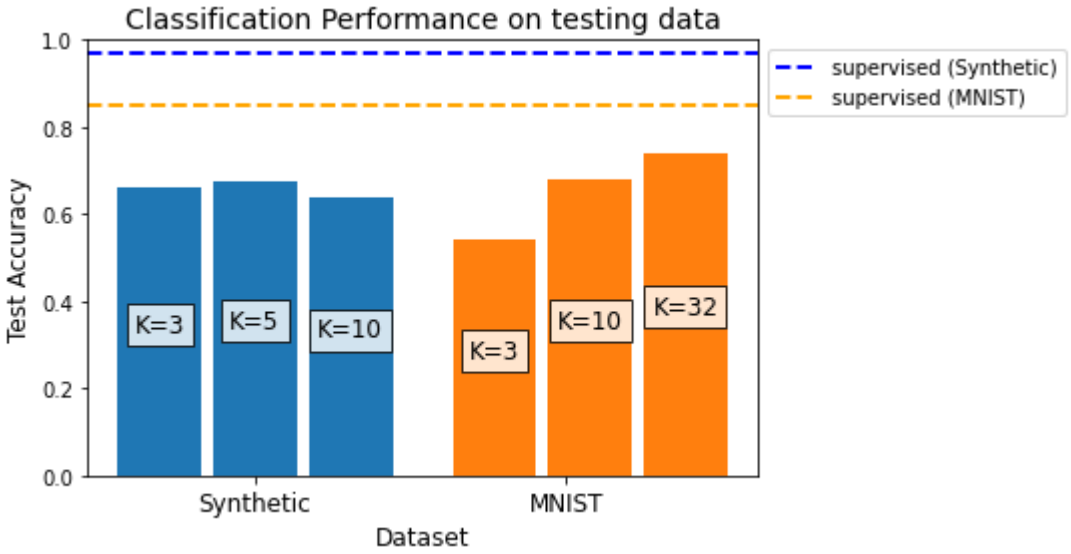

Figure 4-1-2: K-means Unsupervised Learning Accuracy - Increased to 1200 Samples

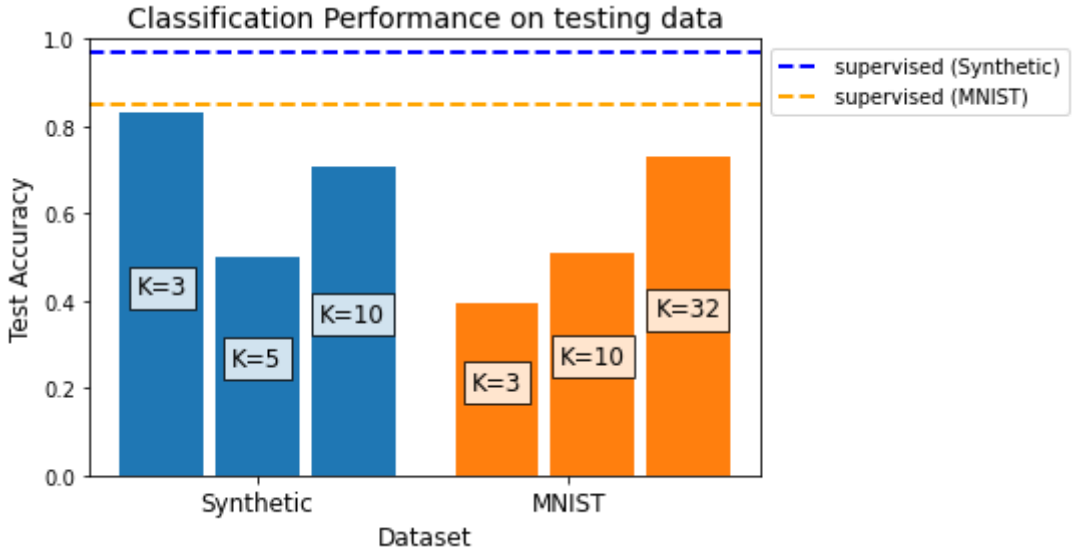

Figure 4-2: K-means Unsupervised Learning Accuracy – Decreased 300 Samples