**Name: Rahul Bhagat**

Module 3 – Project: Classification Methods along with PCA

## Abstract

This project focused on the classification of the MNIST dataset using multiple machine learning algorithms, along with the application of Principal Component Analysis (PCA) to study its effect on classification performance. The project compares different classifiers and provides insights into their pros and cons, supported by confusion matrix heatmaps and performance evaluations.

## Project Overview

The project applies four classification algorithms on the MNIST dataset, which consists of vectorized 28x28 pixel images resulting in 784 features per sample. A subset of 2000 samples is used, split into 80% training and 20% testing data. The classification methods considered are:

- Logistic Regression
- Support Vector Machine (SVM) with Radial Basis Function (RBF) kernel
- K-Nearest Neighbors (KNN)
- Decision Tree

PCA is employed as a dimensionality reduction technique with varying numbers of components to observe its impact on the classifiers' performance. The study includes performance comparisons and visualizations through confusion matrix heatmaps. The implementation utilizes the scikit-learn library with a pipeline approach to streamline model execution.

## Classification Algorithms Details

### Logistic Regression

This algorithm learns a linear decision boundary by estimating probabilities for class membership. It is suitable for problems where classes are linearly separable or approximately.

### Support Vector Machine (SVM) with RBF Kernel

SVM with RBF kernel finds a maximum-margin nonlinear boundary by applying kernel functions, enabling it to handle complex, nonlinearly separable data effectively.

### K-Nearest Neighbors (KNN)

KNN classifies a sample based on the majority class among its nearest neighbours in the feature space, making it a simple and intuitive method that relies on distance metrics .

### Decision Tree

This method makes hierarchical decisions using feature-based rules, recursively partitioning the data to classify samples. It is interpretable and can capture nonlinear relationships.

## Dimensionality Reduction and Performance Evaluation

PCA is applied with varying numbers of components to reduce the high dimensionality of the MNIST dataset. The project evaluates how reducing dimensionality affects the accuracy and performance of each classification method. Performance metrics are compared, and confusion matrix heatmaps are used to visualize classification results, aiding in understanding strengths and weaknesses of each approach.

## Tools and Implementation

The entire modelling and evaluation process is implemented using the scikit-learn library, leveraging its pipeline capabilities to ensure modular and efficient execution of preprocessing, PCA, and classification steps.

## Baseline performance

Following results are obtained from executing the pipeline

| **Algorithm** | **Accuracy** |
|---|---|
| Logistic Regression | 0.8625 |
| SVM | 0.8825 |
| KNN | 0.8300 |
| Decision Tree | 0.6550 |

## Classification Accuracy vs PCA

We applied for PCA with components [10, 20, 40, 80, 160] and following results were obtained.

| PCA Components | Logistic Regression | SVM (RBF) | KNN | Decision Tree |
|---|---|---|---|---|
| 10 | 0.7575 | 0.855 | 0.825 | 0.7075 |
| 20 | 0.835 | 0.9025 | 0.868 | 0.7 |
| 40 | 0.84 | 0.915 | 0.898 | 0.695 |
| 80 | 0.8525 | 0.91 | 0.885 | 0.655 |
| 160 | 0.84 | 0.9075 | 0.858 | 0.65 |

## Observations

- Logistic Regression peaks at **80 components**
- SVM peaks at **40 components**
- KNN peaks at **40 components**
- Decision Tree performance decreases as PCA components increase

Following graph shows the impact of PCA components on classification.