Pumpkin Seeds Dataset
SVM Classification Model
Development Report

Name: Na Yeon. Park
Student ID:2270097
2025.04.18

1.Introduction (Problem Definition)

In today's agricultural industry, automated seed variety identification has emerged as a crucial task for improving both productivity and quality control. In this project, I developed a classification model using the publicly available Pumpkin Seeds Dataset to predict seed varieties based on their physical characteristics. By leveraging the Support Vector Machine (SVM) algorithm, the aim was to accurately determine the class of each seed using measurements such as area, perimeter, major axis length, and minor axis length. The motivation behind this work is to enable a more efficient seed selection and quality management process in agricultural operations.

2.Dataset Description

Data Source:

The Pumpkin Seeds Dataset was obtained from publicly accessible data repositories such as Kaggle.
https://www.kaggle.com/datasets/muratkokludataset/pumpkin-seeds-dataset

Data Structure & Features:

Total Samples: Approximately 2,500 entries

Number of Columns: 13 features, including attributes such as Area, Perimeter, Major_Axis_Length, Minor_Axis_Length, Convex_Area, Equiv_Diameter, Eccentricity, Solidity, Extent, Roundness, Aspect_Ratio, Compactness, and the target variable, Class.

Target Variable:

Class – This column indicates the seed variety. The dataset includes three distinct classes, with approximately 1,300, 800, and 400 samples in each class, respectively.

Preprocessing Steps:

- Rows with missing values were removed using dropna(), ensuring only complete records remained.
- The Class column was used directly as the target variable, with no additional encoding required since it already represents the variety labels.

3.Model Description

Model Selection & Rationale:

For this project, I chose to employ a Support Vector Machine (SVM) due to its proven effectiveness in handling moderate-sized datasets and producing clear decision boundaries. SVMs are particularly suitable for classification tasks that require precision—an essential factor in accurately classifying seed varieties for agricultural applications.

Data Preprocessing:

Considering that SVMs are sensitive to feature scale, all input features were normalized using StandardScaler. This ensures that each feature contributes equally to the model by converting them to have zero mean and unit variance.

Implementation Approach:

• A pipeline was constructed using StandardScaler for normalization and an RBF-kernel SVM (C=1.0, gamma='scale') to train the model.

• Evaluation metrics such as Accuracy, Confusion Matrix, Precision, Recall, and F1-Score were computed to gauge the performance comprehensively.

4. Evaluation and Results

● Confusion Matrix & Classification Report:

The confusion matrix demonstrated that the model correctly classified most instances in both classes. The class Çerçevelik achieved a higher recall (93%) compared to Ürgüp Sivrisi (85%), indicating slightly better sensitivity in detecting the former. Overall, the precision, recall, and F1-scores averaged around 89%, showcasing the model's ability to generalize effectively for seed variety classification.

Confusion Matrix:

	Predicted: Çerçevelik	Predicted: Ürgüp Sivrisi
Actual: Çerçevelik	242	18
Actual: Ürgüp Sivrisi	36	204

Classification Report:

Class Label	*Precision*	*Recall*	*F1-score*	*Support*
Çerçevelik	0.87	0.93	0.90	260
Ürgüp Sivrisi	0.92	0.85	0.88	240
Accuracy			0.89	500

Macro Avg	0.89	0.89	0.89	500
Weighted Avg	0.89	0.89	0.89	500

● Significance of Evaluation Metrics:

In the context of seed classification tasks, accurate differentiation between varieties is crucial for quality assurance and operational effectiveness. Misclassification could lead to incorrect seed labeling or compromised agricultural outcomes. Therefore, in addition to accuracy, we emphasized evaluation metrics such as Precision, Recall, and F1-score. These metrics help ensure that the model minimizes classification errors and maintains consistent performance under real-world usage conditions.

5.Development Environment

Operating System (OS): macOS

Python Version: 3.12.3

Key Packages & Libraries:
The project relies on a set of packages captured in the requirements.txt file. Some of the core packages include:

- pandas==2.2.2
- numpy==1.26.4
- scikit-learn==1.5.0
- openpyxl==3.1.5
- opencv-python==4.10.0.84
- matplotlib==3.9.0

In addition, various other packages such as absl-py, aiohttp, and aiomsignal are also part of the environment. The complete package list is available in the submitted requirements.txt file.

6.Remote Repository

https://github.com/devcathy/mlops_ass1