# SEED QUALITY CLASSIFICATION SYSTEM

### Automated Pumpkin Seed Classification Using Machine Learning & Flask Web Application

## 1. PROJECT OVERVIEW

The Seed Quality Classification System is a machine learning–based web application designed to classify pumpkin seeds into predefined quality categories based on their physical and morphological characteristics. The system leverages supervised learning algorithms trained on a structured dataset containing seed attributes such as area, perimeter, axis lengths, solidity, roundness, and compactness.

The project integrates a trained ML model with a Flask-based web interface, allowing users to input seed parameters and instantly receive a prediction regarding the seed class. This automated approach assists agricultural analysts, researchers, and seed quality inspectors in making faster and more accurate classification decisions.

**Project Member :**

Sujal Rajkumar Chougule

## 2. OBJECTIVES

- To develop an accurate machine learning model for pumpkin seed classification.
- To analyze and preprocess seed morphology data for effective training.
- To compare multiple supervised learning algorithms and select the best-performing model.
- To deploy the trained model using a Flask web framework.
- To design a clean, user-friendly web interface for data input and result visualization.
- To enable real-time prediction of seed class based on user inputs.

## 3. KEY FEATURES

- User-Friendly Web Interface: Simple and intuitive UI for entering seed parameters.
- Machine Learning–Based Prediction: Accurate classification using a trained ML model.
- Multiple Feature Inputs: Supports morphological attributes such as area, perimeter, eccentricity, and compactness.
- Real-Time Prediction: Instant classification results upon form submission.
- Model Persistence: Pre-trained model loaded using pickle for efficient inference.
- Clean UI Flow: Home page → Prediction page → Result display.

## 4. USE CASE SCENARIOS

**Scenario 1: Agricultural Research**

Researchers can input measured parameters of pumpkin seeds obtained from imaging systems. The system classifies the seeds into predefined categories, assisting in seed quality analysis and crop research.

**Scenario 2: Seed Quality Inspection**

Seed processing units can use the application to quickly verify the quality category of seeds before packaging and distribution, ensuring consistency and quality assurance.

**Scenario 3: Educational Demonstration**

Students and learners can use the project to understand how machine learning models are trained, evaluated, and deployed in real-world agricultural applications.

## 5. TECHNICAL APPROACH

**Frontend: Flask + HTML/CSS**

- **Framework:** Flask (Python)
- **Technologies:** HTML5, CSS3
- **Features:** Form-based input, responsive layout, conditional result rendering

**Backend: Machine Learning Model**

- **Algorithms Used:** Logistic Regression, Support Vector Machine, Random Forest (during experimentation)
- **Final Model:** Best-performing algorithm selected based on evaluation metrics
- **Libraries:** NumPy, Pandas, Scikit-learn

**Data Processing**

- Data cleaning and preprocessing
- Feature scaling using StandardScaler
- Label encoding for class labels

**System Architecture**

- **Presentation Layer:** HTML/CSS templates
- **Application Layer:** Flask routes and request handling
- **Model Layer:** Pre-trained ML model loaded via pickle

## 6. IMPLEMENTATION PLAN

| Phase | Activities | Timeline |
|---|---|---|
| Requirement Analysis | Dataset understanding, feature selection | 2 Days |
| Data Preprocessing | Cleaning, scaling, encoding | 3 Days |
| EDA | Statistical & visual analysis | 3 Days |
| Model Training | Train multiple ML models | 3 Days |
| Model Evaluation | Accuracy comparison & tuning | 3 Days |
| Deployment | Flask integration | 3 Days |
| Testing | UI & prediction testing | 3 Days |

## 7. BENEFITS

**For Agriculture Sector**

- Faster and more consistent seed classification
- Reduced manual inspection effort
- Improved decision-making accuracy

**For Researchers & Students**

- Practical exposure to ML model deployment
- Understanding end-to-end ML workflow
- Real-world dataset usage

## 8. PROJECT FLOW

- User Input: User enters seed parameters through the web interface.
- Data Handling: Inputs are collected and formatted by Flask backend.
- Preprocessing: Input data is scaled using the same scaler used during training.
- Prediction: The ML model predicts the seed class.
- Result Display: Predicted seed category is shown on the prediction page.

# 9. REQUIREMENTS SPECIFICATION

### System Requirements

- Python 3.8 or above
- Windows / Linux / macOS
- Web browser (Chrome, Edge, Firefox)

### Python Packages

- flask
- numpy
- pandas
- scikit-learn
- pickle-mixin

### Hardware Requirements

- Minimum 4 GB RAM
- Standard processor

## 10. RISKS AND MITIGATIONS

| Risk | Impact | Mitigation |
|---|---|---|
| Incorrect Inputs | Wrong prediction | Display expected ranges in input fields |
| Model Overfitting | Poor generalization | Cross-validation & tuning |
| Deployment Errors | App failure | Modular code & testing |
| User Misunderstanding | Invalid data entry | Clear UI hints and placeholders |

## 11. FUTURE ENHANCEMENTS

- Integration with image-based seed detection
- Support for multiple seed types
- Advanced visualization of prediction confidence
- Database storage for historical predictions
- REST API for third-party integration

## 12. CONCLUSION

The Seed Quality Classification System successfully demonstrates the application of machine learning in agricultural quality analysis. By combining data preprocessing, model training, and Flask-based deployment, the project delivers a complete end-to-end solution for real-time seed classification. The system is scalable, educational, and practical, making it suitable for academic, research, and industry-level use.

## REFERENCES

1. Scikit-learn Documentation – https://scikit-learn.org/
2. Flask Documentation – https://flask.palletsprojects.com/
3. UCI Machine Learning Repository – Pumpkin Seeds Dataset
4. Python Official Documentation – https://www.python.org/