PUMPKIN SEED PREDICTOR

MACHINE LEARNING PROJECT

1. INTRODUCTION

In recent years, Machine Learning (ML) has emerged as a powerful tool for solving complex real-world problems by enabling systems to learn patterns from data and make intelligent decisions. Agriculture and food technology are two major domains that benefit significantly from ML-based solutions. One such application is the classification of agricultural products based on their physical and morphological characteristics.

Pumpkin seeds, although small in size, have high nutritional and commercial value. They are widely used in food products, health supplements, and agricultural research. Different varieties of pumpkin seeds exhibit variations in size, shape, texture, and other physical properties. Identifying and classifying these varieties manually is time-consuming, error-prone, and requires domain expertise.

This project focuses on building a Machine Learning-based Pumpkin Seed Classification system that automatically classifies pumpkin seed varieties using measurable physical attributes. The trained model is then deployed using a Flask web application, allowing users to input seed parameters and receive predictions through a user-friendly interface.

2. PROBLEM STATEMENT

Manual classification of pumpkin seed varieties based on physical observation is inefficient and subjective. Farmers, researchers, and food industries require a reliable, fast, and automated system to classify pumpkin seeds accurately.

Problems Identified:

1. Manual classification requires expert knowledge
2. High chance of human error
3. Time-consuming process
4. Lack of scalable digital solutions

Proposed Solution:

Develop a Machine Learning-based classification system that:

5. Uses numerical features of pumpkin seeds

6. Trains multiple ML models
7. Selects the best-performing model
8. Deploys the model as a web application using Flask

3. OBJECTIVES OF THE PROJECT

The main objectives of this project are:

1. To collect and analyse pumpkin seed data
2. To perform Exploratory Data Analysis (EDA) for understanding data patterns
3. To pre-process data (outlier handling and feature scaling)
4. To train multiple machine learning models
5. To compare model performance and select the best model
6. To deploy the trained model using Flask
7. To provide a simple web interface for prediction

4. SCOPE OF THE PROJECT

This project demonstrates how machine learning can be applied in agricultural data analysis. The scope includes:

9. Classification of pumpkin seed varieties using physical attributes
10. Use of supervised learning algorithms
11. Development of a real-time prediction web application
12. Educational and research purposes

The project can be extended further to include:

13. Image-based seed classification
14. Mobile application integration
15. Cloud deployment
16. Larger datasets for improved accuracy

5. LITERATURE REVIEW

Several studies have explored the use of machine learning in agricultural classification problems. Researchers have applied algorithms such as Decision Trees, Random Forests, Support Vector Machines, and Neural Networks to classify seeds, grains, and crops based on physical and chemical attributes.

Previous works indicate that ensemble models such as Random Forest and Gradient Boosting often outperform single models due to their ability to reduce variance and bias. However, model performance heavily depends on data pre-processing and feature scaling.

This project builds upon existing research by implementing and comparing multiple algorithms and deploying the best-performing model as a web application.

6. SYSTEM ARCHITECTURE

6.1 High-Level Architecture

The system consists of the following components:

17. Dataset (Online Google Sheets)
18. Data Pre-processing Module
19. Machine Learning Model Training
20. Model Evaluation & Selection
21. Model Serialization (Pickle)
22. Flask Web Application
23. User Interface (HTML + CSS)

6.2 Workflow

24. User provides input through the web interface
25. Flask server receives the input
26. Input data is scaled using the saved scalar
27. The trained ML model predicts the class
28. Prediction is displayed on the UI

7. TOOLS AND TECHNOLOGIES USED

7.1 Programming Language

29. Python 3

7.2 Libraries and Frameworks

30. NumPy
31. Pandas
32. Matplotlib
33. Seaborn
34. Scikit-learn
35. Flask
36. Pickle

7.3 Development Tools

37. Visual Studio Code
38. Git & GitHub
39. Jupyter Notebook

8. DATASET DESCRIPTION

The dataset used in this project is publicly available and hosted online as a Google Spreadsheet. It contains 2,500 pumpkin seed samples with numerical features describing physical characteristics of pumpkin seeds.

8.1 Features Used

40. Area
41. Perimeter
42. Major Axis Length
43. Minor Axis Length

44. Convex Area
45. Equivalent Diameter
46. Eccentricity
47. Solidity
48. Extent
49. Roundness
50. Aspect Ratio
51. Compactness

8.2 Target Variable

52. Class (Pumpkin Seed Variety): Çerçevelik (Class 0) and Ürgüp Sivrisi (Class 1)

9. DATA PREPROCESSING

Data pre-processing is a crucial step to ensure high-quality model performance.

9.1 Handling Missing Values

The dataset was analyzed for missing values using pandas functions. No significant missing values were found in the dataset.

9.2 Outlier Detection and Removal

Outliers were detected using the Interquartile Range (IQR) method, particularly in the Area feature. Box plots were used to visualize outliers, and values beyond the IQR boundaries were filtered out to improve model reliability.

9.3 Feature Scaling

Min-Max Scaling was applied to normalize feature values into a range of 0 to 1. This ensures equal importance of all features during model training. The features scaled include Area, Perimeter, and Major Axis Length.

9.4 Feature Selection

Unnecessary columns (Convex_Area, Equiv_Diameter, Eccentricity, Minor_Axis_Length) were dropped after analysis to reduce dimensionality and improve model efficiency.

10. EXPLORATORY DATA ANALYSIS (EDA)

EDA was performed to understand the distribution and relationships of features.

10.1 Univariate Analysis

53. Distribution plots were used to analyze individual features
54. Count plots were used to observe class distribution (balanced dataset with approximately equal samples)

10.2 Bivariate Analysis

55. Scatter plots revealed strong positive correlations between features such as Area and Perimeter

10.3 Multivariate Analysis

56. Correlation heatmaps were used to identify feature relationships
57. Pair plots visualized interactions between multiple features across different seed classes

11. MACHINE LEARNING MODELS USED

The following supervised learning algorithms were implemented and compared:

8. Logistic Regression
9. Decision Tree Classifier
10. Random Forest Classifier
11. Multinomial Naive Bayes Classifier
12. Support Vector Machine (SVM)
13. Gradient Boosting Classifier

Each model was trained using the same train-test split (80-20) and evaluated using accuracy scores and classification reports.

12. MODEL EVALUATION AND COMPARISON

12.1 Evaluation Metrics

58. Accuracy Score
59. Precision
60. Recall
61. F1-score
62. Classification Report

12.2 Model Comparison

The performance of all models was systematically compared. Ensemble models such as Random Forest (88.1% accuracy) and Gradient Boosting Classifier (88.5% accuracy) achieved the highest accuracy, followed by Support Vector Machine (86.5%) and Logistic Regression (86.1%). The Decision Tree Classifier achieved 84.5% accuracy, while Multinomial Naive Bayes had the lowest performance at 70.2%.

Based on the comparison, Random Forest was selected as the final model due to its balance of high accuracy, robustness, and generalization capability.

13. MODEL DEPLOYMENT

The best-performing model (Random Forest) was serialized using the Python pickle library and integrated into a Flask web application for real-time predictions.

13.1 Flask Application

63. Backend: Python Flask framework
64. Frontend: HTML5 and CSS3 for responsive design
65. Model Loading: Pickle file (model.pkl) loaded at server startup

13.2 User Interface

66. Input form for entering 8 seed measurement parameters
67. Prediction result page displaying the classified seed variety
68. Clean, professional, and responsive UI design

13.3 Application Workflow

14. User enters seed measurements through the web form

15. Flask server receives and validates the input data
16. Input features are converted to a NumPy array
17. The trained model predicts the seed class (0 or 1)
18. Result is displayed: 'Çerçevelik class' or 'Ürgüp Sivrisi class'

14. RESULTS AND DISCUSSION

The developed system successfully classifies pumpkin seed varieties based on numerical features with 88% accuracy. The web application provides real-time predictions with a user-friendly interface, demonstrating the practical applicability of machine learning in agriculture.

Key findings include:

69. Ensemble methods (Random Forest, Gradient Boosting) outperformed individual classifiers
70. Feature scaling significantly improved model performance
71. Outlier removal enhanced model reliability
72. The Flask deployment enables practical real-world usage

15. LIMITATIONS OF THE PROJECT

73. Dataset size is limited to 2,500 samples
74. Only two seed varieties are classified
75. Only numerical features are used (no image-based classification)
76. Model performance depends on data quality and measurement accuracy
77. Local deployment only (not cloud-based)

16. FUTURE ENHANCEMENTS

78. Image-based pumpkin seed classification using Convolutional Neural Networks (CNN)
79. Mobile application development for on-field usage
80. Cloud deployment (AWS, Azure, or Google Cloud Platform)

81. Expansion to classify additional pumpkin seed varieties
82. Integration of larger and more diverse datasets
83. Advanced hyperparameter tuning using GridSearchCV or RandomizedSearchCV
84. Real-time camera integration for automated seed scanning

17. CONCLUSION

This project demonstrates an end-to-end implementation of a machine learning classification system, from data collection and pre-processing to model training and web deployment. The Pumpkin Seed Classification system successfully automates the identification of seed varieties using physical measurements and highlights the potential of machine learning in agriculture and food technology domains.

Through comprehensive data analysis, multiple algorithm comparison, and practical deployment, the project achieves its objectives of creating an accessible and accurate classification tool. The Random Forest model, with 88% accuracy, provides reliable predictions that can assist farmers, researchers, and industry professionals in seed quality assessment and variety identification.

The successful integration of the model into a Flask web application demonstrates the practical viability of deploying machine learning solutions for real-world agricultural applications. This project serves as a foundation for future enhancements and showcases how technology can modernize traditional agricultural practices.

18. REFERENCES

85. Scikit-learn Documentation: https://scikit-learn.org/
86. Pandas Documentation: https://pandas.pydata.org/
87. Flask Documentation: https://flask.palletsprojects.com/
88. Machine Learning Research Papers on Agricultural Classification