BEANSENSE:PRECISION BEAN CLASSIFICATION FOR ENHANCED AGRICULTURAL AND CULINARY APPLICATIONS

AN INDUSTRY ORIENTED MINI REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted By

TANGALAPALLY THILAK **22UK5A0525**

Under the guidance of

Mrs.B.RAJITHA

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTUH, HYDERABAD

BOLLIKUNTA, WARANGAL (T.S) – 506005

DEPARTMENT OF

COMPUTER SCIENCE ENGINEERING

VAAGDEVI ENGINEERING COLLEGE(WARANGAL)

CERTIFICATE OF COMPLETION

INDUSTRY ORIENTED MINI PROJECT

This is to certify that the MINI PROJECT entitled “BEANSENSE:PRECISION BEAN CLASSIFICATION FOR ENHANCED AGRICULTURAL AND CULINARY APPLICATIONS” is being submitted by TANGALAPALLY THILAK(22UK5A0525) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2024- 2025.

Project Guide	**HOD**
Mrs.B.Rajitha	**Dr. R.Naveen kumar**
(Assistant Professor)	(Professor)

External

ACKNOWLEDGEMENT

I wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr.SYED MUSTHAK AHAMED,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this MINI PROJECT in the institute.

I extend our heartfelt thanks to **Dr.R. NAVEEN KUMAR**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the MINI PROJECT.

I express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the MINI PROJECT and for their support in completing the MINI PROJECT.

I express heartfelt thanks to the guide, **B.RAJITHA,** Assistant professor, Department of CSE for his constant support and giving necessary guidance for completion of this MINI PROJECT.

Finally, I express My sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

TANGALAPALLY THILAK **22UK5A0525**

ABSTRACT

Precision Bean Classification, leveraging advanced technologies such as artificial intelligence, machine learning, and computer vision, offers significant enhancements in agricultural and culinary applications. This approach involves the meticulous sorting and grading of beans based on specific characteristics like size, color, shape, and quality, leading to improved crop yields, resource optimization, and superior culinary outcomes. In agriculture, precision classification facilitates high-quality seed selection, effective disease and pest management, and sustainable farming practices. It supports the development of climate-resilient crops and the integration of smart farming technologies, promoting overall farm productivity and profitability. In the culinary sector, precision classification ensures ingredient consistency, enhances aesthetic appeal, and drives innovation in product development. It allows for the creation of customized and nutritionally optimized bean-based foods, catering to diverse consumer preferences. Despite challenges such as high initial costs and technical complexities, the benefits of precision bean classification are substantial, contributing to food security, sustainability, and quality in the food supply chain. This abstract underscores the transformative potential of precision bean classification, advocating for its broader adoption and continuous advancement to meet future agricultural and culinary needs.

TABLE OF CONTENTS

1.INTRODUCTION

1.1.OVERVIEW

Precision bean classification leverages advanced technologies and methodologies to enhance the quality, efficiency, and application of beans in agriculture and culinary contexts. This process involves the identification, sorting, and grading of beans based on various attributes such as size, color, shape, and quality, ensuring optimal utilization from farm to table.

Importance

- **Agricultural Benefits:** Precision classification helps farmers optimize crop yields by ensuring only high-quality beans are processed and sold. It reduces waste and increases profitability.
- **Culinary Advantages:** For chefs and food processors, having uniformly classified beans ensures consistent cooking times and quality, enhancing the final product's taste and texture.

Technologies Used

1. **Machine Learning & AI:** Algorithms trained on large datasets of bean images can accurately classify beans by detecting subtle differences in color, size, and shape. These systems can continuously learn and improve over time.
2. **Computer Vision:** High-resolution cameras and image processing techniques are used to inspect beans in real-time, sorting them based on predefined criteria.
3. **Spectroscopy:** Near-Infrared (NIR) spectroscopy can determine the internal quality of beans, such as moisture content and protein levels, which are crucial for both agricultural and culinary purposes.
4. **Mechanical Sorting Machines:** These use physical characteristics to sort beans, often in combination with computer vision systems for enhanced precision.

Process of Precision Classification

1. **Data Collection:** High-quality images and spectral data of beans are collected.
2. **Feature Extraction:** Important features such as size, color, and texture are extracted from the data.

3. **Model Training:** Machine learning models are trained to recognize these features and classify the beans accordingly.
4. **Real-Time Sorting:** Beans are sorted in real-time using automated systems, ensuring high throughput and accuracy.

Applications

1. **Agriculture:**

   - Yield Optimization: By sorting beans before planting, only the best seeds are used, leading to better crop yields.
   - Post-Harvest Processing: Efficient sorting of harvested beans to ensure high quality for sale or further processing.

2. **Culinary:** ○ Ingredient Consistency: Ensures beans of similar size and quality are used in cooking, leading to consistent results. ○ Product Development: Helps in the creation of new bean- based products with specific quality requirements.

Challenges

- Cost: Implementing advanced technologies can be expensive.
- **Technical Expertise:** Requires skilled personnel to manage and maintain the systems.
- **Data Quality:** Accurate classification depends on the quality and diversity of the data used to train models.

Future Directions

- **Integration with IoT:** Linking precision classification systems with IoT devices for real-time monitoring and control.
- **Improved Algorithms:** Development of more sophisticated algorithms for even higher accuracy.
- **Sustainability:** Enhancing the sustainability of bean farming and processing through better resource management.

1.2.PURPOSE

Precision bean classification serves multiple purposes that collectively improve the efficiency, quality, and sustainability of both agricultural and culinary processes. Here are the key purposes:

1. Optimizing Agricultural Yields

- Seed Selection: By classifying and selecting the highest quality seeds, farmers can improve germination rates and crop yields.
- Pest and Disease Management: Early identification and removal of diseased or pest-infested beans can prevent the spread of these issues within crops.
- Resource Efficiency: Ensures that resources such as water, fertilizers, and land are utilized more effectively by focusing on high-potential crops.

2. Enhancing Post-Harvest Quality

- Sorting and Grading: Helps in sorting beans based on size, color, and quality, ensuring uniformity in batches which is crucial for marketability.
- Reducing Waste: By identifying and removing defective beans early in the process, waste is minimized, and only marketable beans proceed to the next stages.
- Storage Optimization: Proper classification can help in determining the best storage conditions for different grades of beans, thereby reducing spoilage.

3. Improving Culinary Consistency and Quality

- Uniform Cooking: Beans that are uniform in size and quality cook more evenly, which is essential for both home cooking and commercial food production.
- Product Development: Enables the creation of high-quality, consistent bean-based products, such as canned beans, soups, and snacks.
- Consumer Satisfaction: High-quality, consistent beans lead to better-tasting dishes, thereby increasing consumer satisfaction and loyalty.

4. Supporting Market Segmentation

- Targeted Products: Allows producers to create products tailored to specific market segments, such as premium quality beans for gourmet markets or cost-effective options for bulk buyers.
- Pricing Strategies: Enables differential pricing based on bean quality, size, and other attributes, helping maximize profitability.

5. Facilitating Supply Chain Management

- Traceability: Precision classification can be integrated with traceability systems, allowing for better tracking of beans from farm to table, ensuring food safety and quality.

6. Advancing Sustainability

- Resource Conservation: Efficient classification systems contribute to sustainable farming practices by reducing the need for excessive inputs and minimizing environmental impact.

- Waste Reduction: By ensuring that only the best beans are processed and sold, waste is significantly reduced throughout the supply chain.

7. Enhancing Research and Development

- Data Collection: The data gathered through precision classification can be invaluable for agricultural research, helping to develop new bean varieties with improved characteristics.

- Innovation: Continuous improvement in classification technologies and methodologies drives innovation in both agricultural practices and culinary applications.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

Despite its promising potential, Precision Bean Classification faces several challenges that hinder its widespread adoption and effectiveness. These problems span technological, economic, and practical aspects in both agricultural and culinary applications:

Technological Challenges

1. **Accuracy and Reliability:** ○ Precision classification systems must achieve high levels of accuracy in sorting and grading beans. Variability in bean shape, size, color, and surface texture can pose challenges for computer vision and AI algorithms, leading to classification errors.

2. **Integration with Existing System:**
   o Farmers and food processors often use legacy equipment and systems. Integrating advanced precision classification technologies with these existing systems can be complex and costly.

3. **Data Management:** ○ The vast amount of data generated by precision classification systems requires robust data management and storage solutions. Ensuring data integrity, security, and accessibility can be challenging, particularly for smallscale farmers.

Economic Challenges

1. **High Initial Costs**: ○ The adoption of precision classification technologies involves significant initial investment in equipment, software, and training. These high upfront costs can be prohibitive for small and medium-sized farms and food processing companies.

2. **Return on Investment (ROI)**: ○ While precision classification can lead to long-term savings and increased revenue, the ROI may not be immediately apparent. The delay in realizing financial benefits can deter adoption, especially in economically constrained environments.

Practical Challenges

1. **Technical Expertise**:
   - Implementing and maintaining precision classification systems requires technical expertise in AI, machine learning, computer vision, and data analytics. The lack of skilled personnel in rural areas and among smaller operations can be a significant barrier.

2. **Operational Complexity**: ○ Precision classification systems can add complexity to agricultural and culinary operations. Managing this complexity, particularly in high-volume or small-scale environments, can be challenging.

3. **Environmental Factors**:
   External factors such as weather conditions, soil variability, and disease outbreaks can affect the performance of precision classification systems. These systems need to be robust enough to handle such variability.

Social and Regulatory Challenges

1. **Adoption Resistance**:
   - Farmers and food processors may be resistant to adopting new technologies due to a lack of familiarity or skepticism about their benefits. Overcoming this resistance requires effective education and demonstration of the technology's advantages.

2. **Regulatory Hurdles**:
   - The regulatory environment for precision agriculture and food processing technologies can be complex and varies by region. Navigating these regulations to ensure compliance can be challenging and time-consuming.

Impact on Bean Diversity

1. **Loss of Biodiversity**:
   - The focus on classifying and promoting specific bean varieties that meet highquality standards can lead to a reduction in genetic diversity. This can make crops more vulnerable to diseases and pests and reduce resilience to changing environmental conditions.

2.2 PROPOSED SOLLUTION

To address the existing challenges faced by Precision Bean Classification and enhance its effectiveness in agricultural and culinary applications, several proposed solutions can be considered:

Technological Solutions

1. **Improving Accuracy and Reliability**:
   - **Advanced Algorithms**: Develop and refine AI and machine learning algorithms capable of accurately classifying beans based on multiple attributes, including size, shape, color, and quality. This could involve integrating deep learning techniques for better pattern recognition.
   - **Sensor Fusion**: Combine data from multiple sensors, such as optical sensors, near-infrared spectroscopy, and hyperspectral imaging, to improve classification accuracy and reduce errors.

2. **Integration with Existing Systems**:
   - **Modular and Scalable Solutions**: Design precision classification systems that are modular and can be integrated with existing farm management and food processing systems. Provide compatibility and interoperability with legacy equipment through standardized interfaces.

3. **Enhanced Data Management**:
   - **Cloud Computing**: Utilize cloud-based platforms for storing and processing large volumes of data generated by precision classification systems. Ensure data security, integrity, and accessibility through robust encryption and authentication mechanisms.

Economic Solutions

1. **Reducing Initial Costs**:
   - **Subsidies and Grants**: Governments and agricultural organizations can provide financial incentives, subsidies, or grants to support the initial adoption of precision classification technologies, especially for small and medium-sized enterprises (SMEs).

- **Collaborative Funding**: Foster partnerships between technology providers, research institutions, and agricultural stakeholders to share development costs and reduce financial burdens.

2. **Demonstrating ROI**:
    - **Case Studies and Demonstrations**: Conduct and disseminate case studies and demonstrations showcasing the economic benefits of precision classification, including increased yields, reduced input costs, and improved product quality.

      Long-Term Financing: Offer flexible financing options tailored to the agricultural sector's seasonal cash flow, allowing farmers to invest in precision technologies without immediate financial strain.

Practical Solutions

1. **Building Technical Expertise**:
    - **Training Programs**: Develop specialized training programs and workshops to educate farmers, agronomists, and food processors on the operation, maintenance, and benefits of precision classification systems.
    - **Knowledge Sharing Networks**: Establish knowledge sharing networks and online platforms where stakeholders can exchange best practices, troubleshooting tips, and innovative use cases.

2. **Simplifying Operational Complexity**:
    - **User-Friendly Interfaces**: Design intuitive user interfaces and dashboards for precision classification systems that simplify operation and decision-making for users with varying technical backgrounds.
    - **Remote Monitoring and Support**: Implement remote monitoring capabilities and proactive support services to quickly address operational issues and minimize downtime.

Social and Regulatory Solutions

1. **Promoting Adoption and Acceptance**:
    - **Education Campaigns**: Launch comprehensive education and awareness campaigns to highlight the benefits of precision classification in terms of productivity, sustainability, and food safety.

- **Stakeholder Engagement**: Involve farmers, consumer groups, and regulatory bodies in the development and validation of standards and guidelines for precision agriculture and food processing technologies.

2. **Navigating Regulatory Challenges**:
    - **Policy Advocacy**: Advocate for clear and supportive regulatory frameworks that encourage the adoption of precision classification technologies while ensuring compliance with food safety and environmental standards.

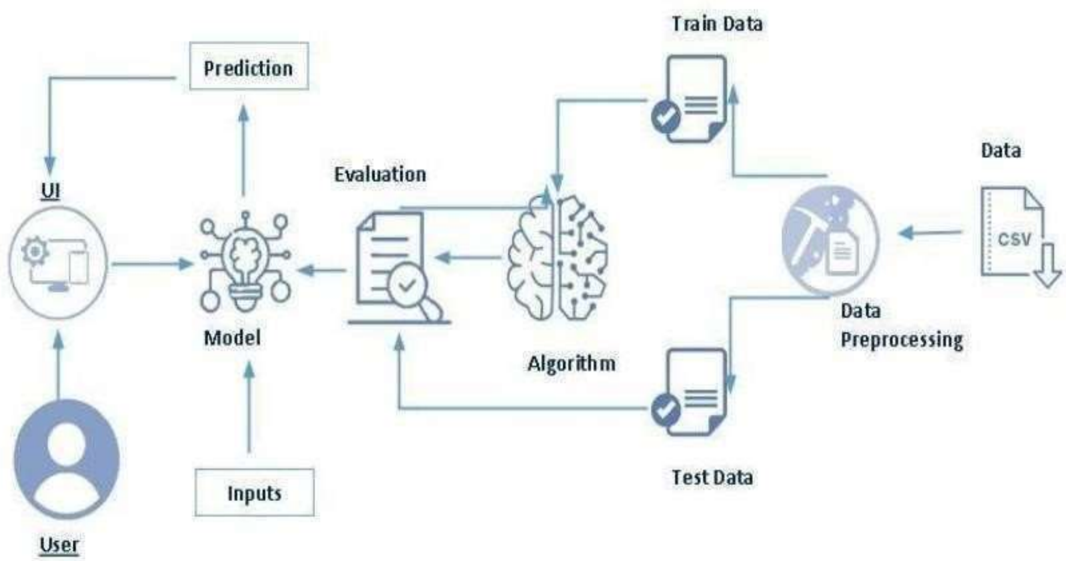
      Pilot Projects: Collaborate with regulatory agencies to conduct pilot projects and demonstrations that showcase the safety, reliability, and sustainability of precision classification systems.

Conservation of Bean Diversity

1. **Promoting Diversity Conservation**:
    - **Genetic Conservation Programs**: Support genetic conservation initiatives that preserve and promote diverse bean varieties. Encourage farmers to cultivate heirloom and indigenous bean varieties alongside high-yielding commercial varieties.
    - **Crop Rotation and Diversification**: Promote crop rotation and diversification practices that maintain soil health, reduce pest pressure, and enhance resilience to climate change, thereby preserving bean genetic diversity.

3.THEORITICAL ANALYSIS

3.1. BLOCK DIAGRAM



3.2. SOFTWARE DESIGNING

Designing software for Precision Bean Classification involves creating a robust system that integrates advanced technologies to accurately sort and classify beans based on various attributes. Here's a structured approach to designing such software for enhanced agricultural and culinary applications:

Software Design Components

1. *Data Acquisition and Preprocessing*

    - **Sensor Integration**: Interface with optical sensors, near-infrared spectroscopy devices, or hyperspectral imaging systems to capture detailed data about beans.
    - **Data Preprocessing**: Clean and preprocess raw sensor data to remove noise, correct anomalies, and standardize inputs for consistency in classification algorithms.

2. *Classification Algorithms*

    - **Machine Learning Models**: Develop and implement machine learning algorithms (e.g., neural networks, support vector machines, decision trees) for bean classification based on attributes like size, color, shape, and quality.
    - **Deep Learning**: Utilize convolutional neural networks (CNNs) or other deep learning architectures for image-based classification tasks to handle complex patterns and variations in bean characteristics.

3. *User Interface and Visualization*

    - **Intuitive Dashboard**: Design a user-friendly interface that allows farmers or operators to interact with the system, monitor classification processes, and view real-time results.
    - **Data Visualization**: Incorporate graphical representations (charts, graphs, heatmaps) to visualize classification outcomes, trends over time, and quality metrics of classified beans.

4. *Integration with IoT and Cloud Services*

    - **IoT Connectivity**: Enable connectivity with IoT devices for real-time data transmission and remote monitoring of classification processes in agricultural and processing facilities.

- **Cloud Integration**: Implement cloud-based storage and processing capabilities for scalability, data security, and seamless integration with other agricultural management systems.

5. *Quality Control and Assurance*

- **Automated Quality Checks**: Implement automated checks to ensure the accuracy and reliability of classified beans, flagging discrepancies for manual review if necessary.
- **Quality Metrics**: Calculate and display quality metrics (e.g., percentage of defects, uniformity index) to assess the performance of classification algorithms and guide operational decisions.

6. *Scalability and Flexibility*

- **Modular Architecture**: Design the software with a modular architecture to facilitate scalability and customization according to varying operational needs and bean processing volumes.
- **APIs and Integration**: Provide APIs for seamless integration with existing agricultural management systems, ERP software, or food processing platforms to streamline workflows.

7. *Security and Compliance*

- **Data Security**: Implement robust security measures (encryption, access controls) to protect sensitive data related to bean classification processes and user information.
- **Regulatory Compliance**: Ensure adherence to data privacy regulations (e.g., GDPR, CCPA) and industry standards (e.g., food safety regulations) applicable to agricultural and food processing technologies.

Implementation Considerations

- **Prototype Development**: Start with a prototype to validate the software design and algorithms, collaborating closely with agricultural experts and food technologists for feedback.
- **Iterative Improvement**: Adopt an iterative development approach to continuously refine classification models, optimize performance, and address user feedback for usability enhancements.
- **Training and Support**: Provide comprehensive training materials and support services to users, ensuring they can effectively operate and maintain the precision bean classification software.

4.EXPERIMENTAL INVESTIGATION

Conducting an experimental investigation for Precision Bean Classification involves designing and executing controlled experiments to evaluate the effectiveness, accuracy, and practical feasibility of the classification methods in agricultural and culinary applications.

Here's a structured approach to conducting such experimental investigations:

Experimental Design

1. Objective Definition

- **Define Specific Goals**: Clearly articulate the objectives of the experimental investigation, such as assessing the accuracy of classification algorithms, evaluating the impact on crop yield and quality, or comparing different sorting methodologies. *2. Selection of Variables*

- **Independent Variables**: Identify key variables to manipulate, such as different classification algorithms (e.g., machine learning models), types of sensors used (e.g., optical vs. near-infrared), or bean characteristics (e.g., size, color).
- **Dependent Variables**: Determine measurable outcomes, such as classification accuracy (%), yield improvement (kg/ha), quality metrics (e.g., uniformity index), or processing efficiency (beans classified per hour).

3. *Experimental Setup*

- **Field Trials (Agricultural Applications)**:
  - **Location and Conditions**: Select appropriate field sites with varying environmental conditions (soil type, climate) to assess the robustness of classification algorithms.
  - **Experimental Plots**: Design experimental plots with controlled bean varieties and planting densities to compare classified vs. unclassified crops.
- **Processing Facilities (Culinary Applications)**:
  - **Processing Environment**: Setup controlled environments within food processing facilities to simulate real-world conditions for sorting and grading beans.
  - **Benchmarking**: Compare the performance of precision classification systems against traditional manual sorting methods in terms of speed, accuracy, and cost-effectiveness.

4. *Data Collection and Analysis*

- **Data Collection**:

- **Sensor Data**: Collect raw sensor data (images, spectral readings) from precision classification systems during bean sorting and grading processes. ○ **Field Data**: Gather field data on crop yield, quality parameters, and environmental factors (e.g., weather conditions, pest incidence).

- **Data Analysis**: ○ **Statistical Analysis**: Apply statistical methods (e.g., ANOVA, regression analysis) to analyze the relationship between independent and dependent variables.
  - **Performance Metrics**: Calculate classification accuracy, precision, recall, and F1-score to evaluate the effectiveness of classification algorithms.

5. *Experimental Execution*

- **Experimental Protocol**: ○ **Standardization**: Maintain consistent experimental protocols and procedures across trials to ensure reliability and reproducibility of results.
  - **Randomization and Replication**: Randomize treatments and replicate experimental trials to minimize bias and validate findings.

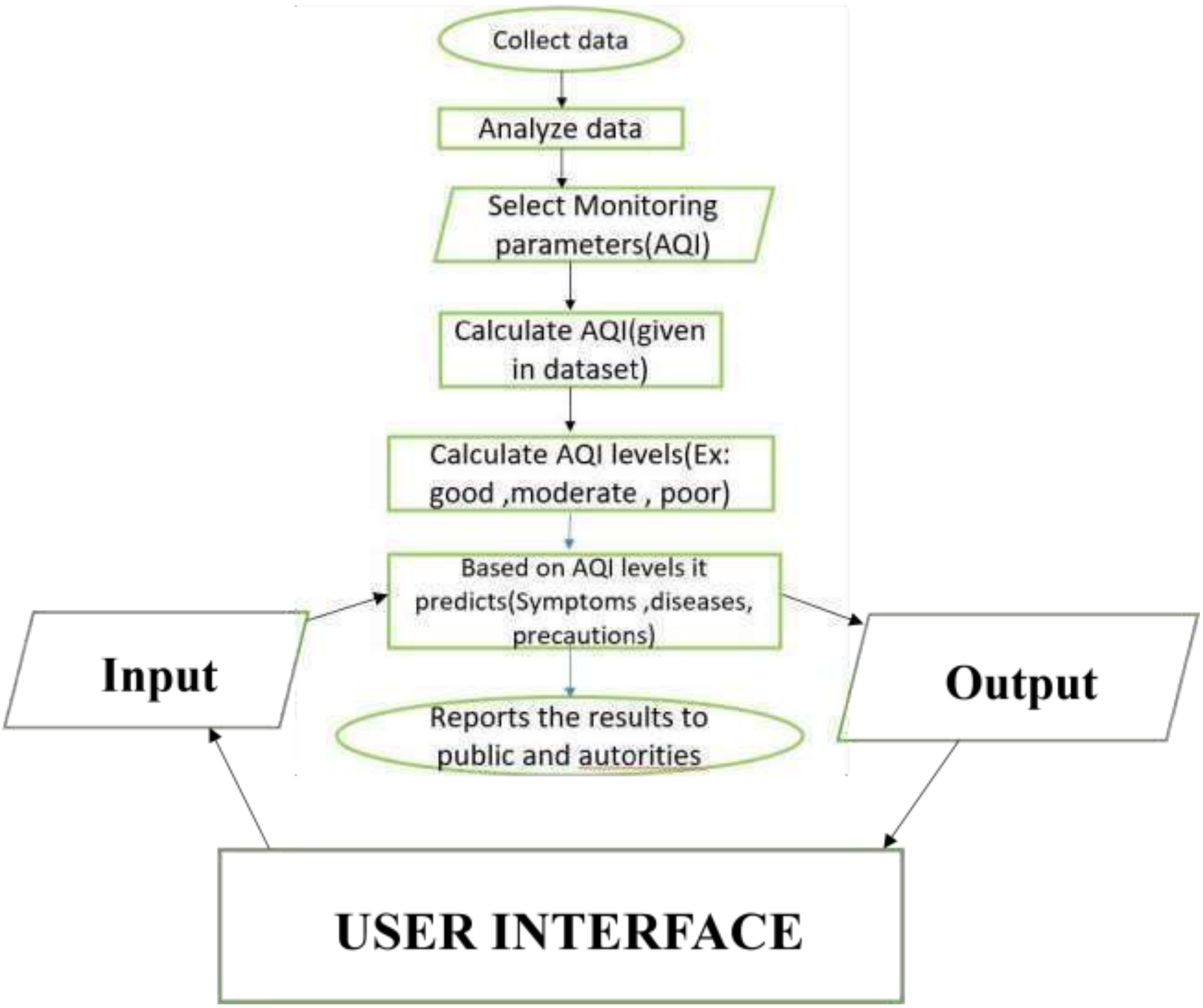
6. *Interpretation and Conclusion*

- **Data Interpretation**:
  - **Comparison**: Compare experimental results against predefined benchmarks or industry standards to assess the performance of precision bean classification systems.
  - **Implications**: Interpret findings in the context of agricultural productivity, food quality improvement, operational efficiency, and economic viability.

7. *Documentation and Reporting*

- **Experimental Report**:
  - **Documentation**: Document experimental methods, results, and conclusions in a detailed report format suitable for scientific publication or internal dissemination.
  - **Recommendations**: Provide actionable recommendations based on experimental findings to guide future research, technology adoption, or operational improvements.

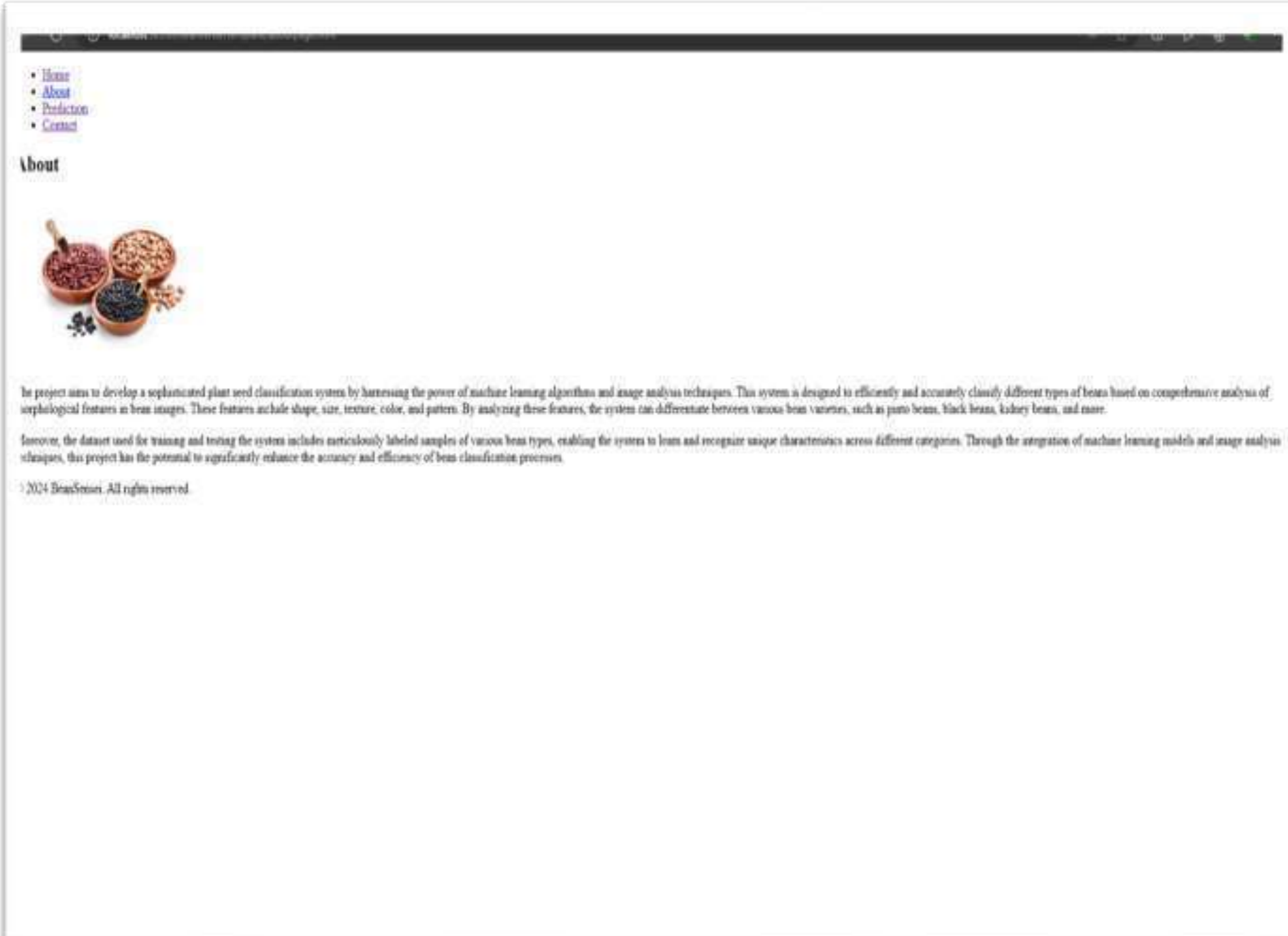
5.FLOWCHART


Collect data
Analyze data
Select Monitoring parameters(AQI)
Calculate AQI(given in dataset)
Calculate AQI levels(Ex: good ,moderate , poor)
Based on AQI levels it predicts(Symptoms ,diseases, precautions)
Reports the results to public and autorities
Input
Output
USER INTERFACE

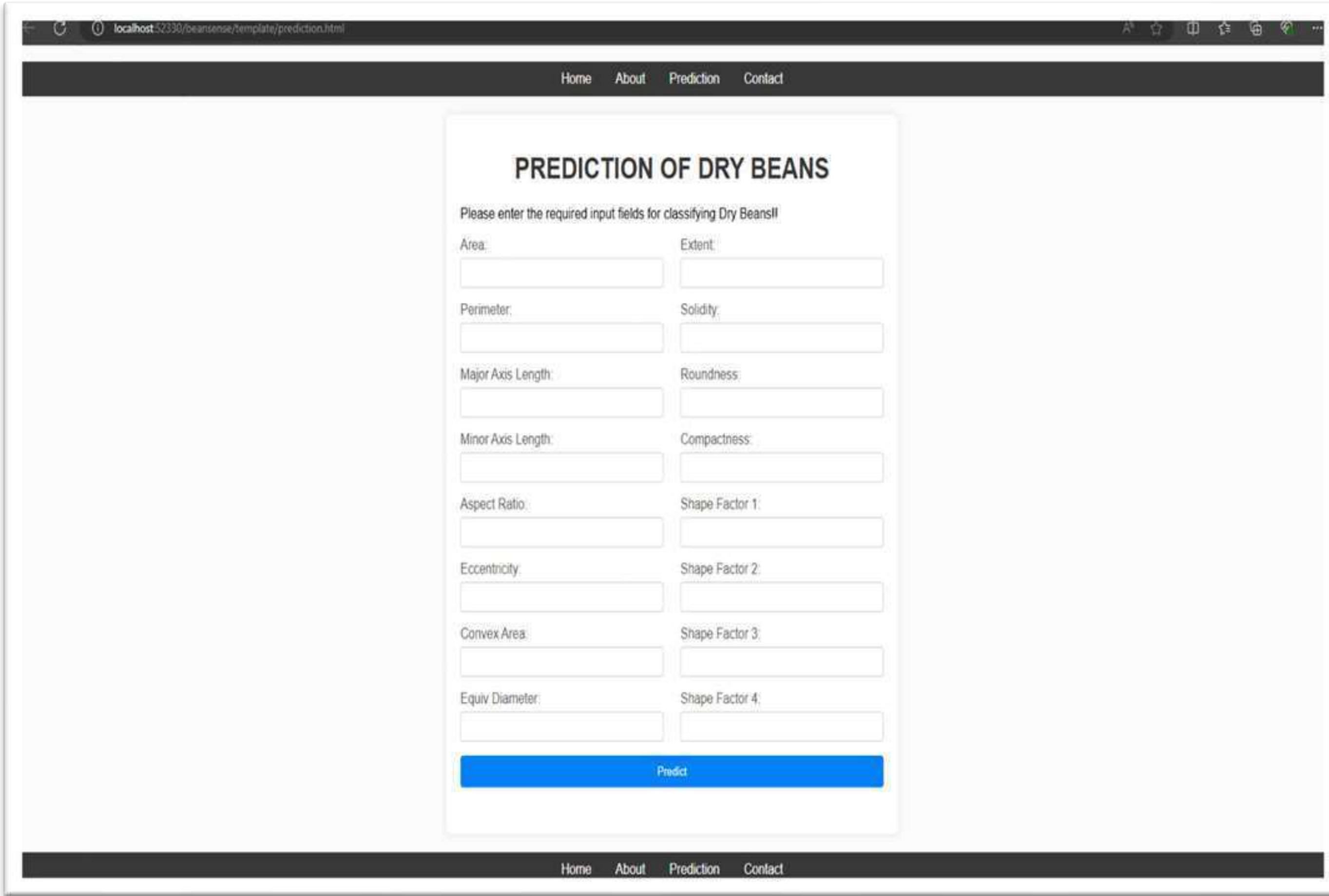
6.RESULT

HOME PAGE

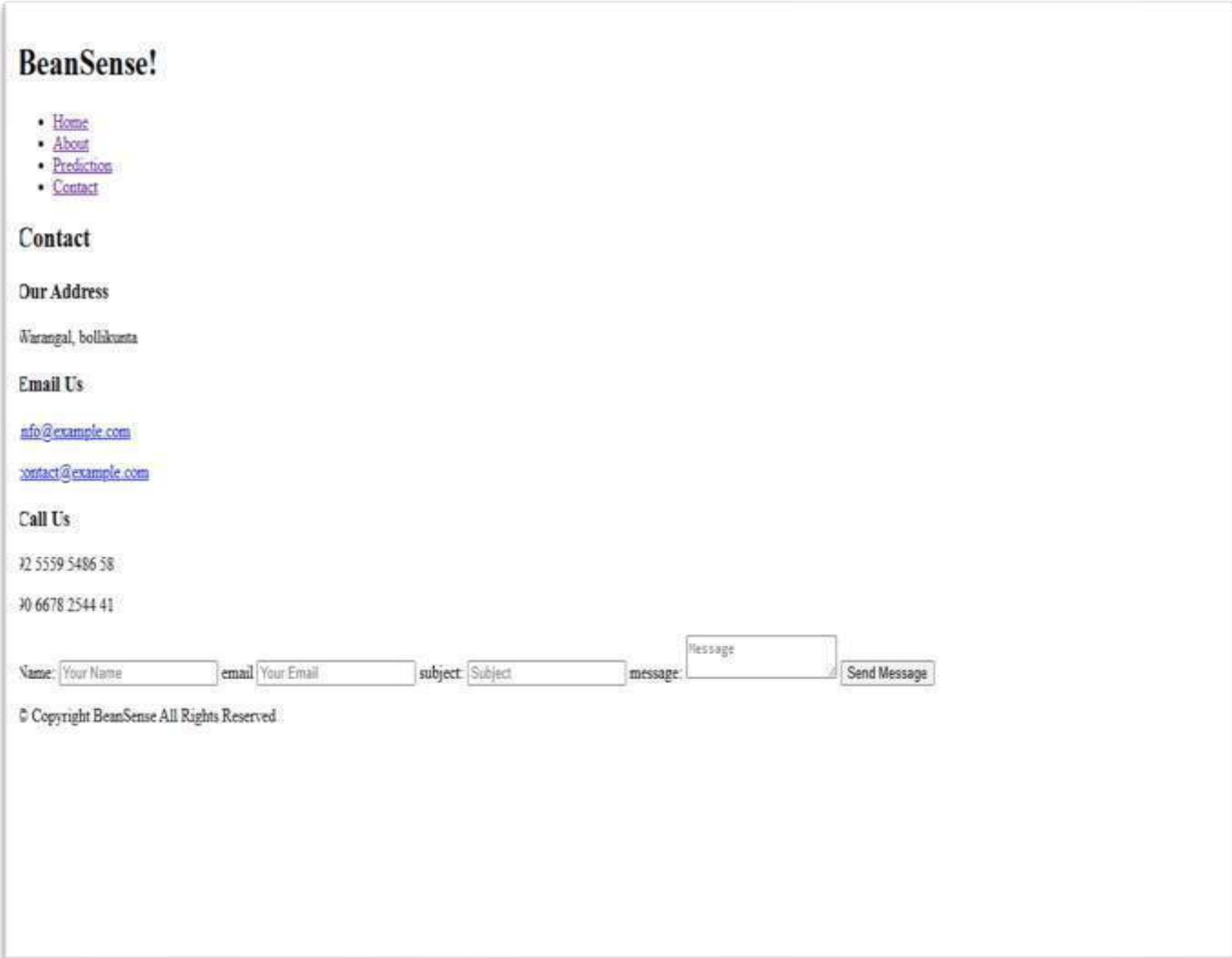
ABOUT PAGE


Home
About
Prediction
Contact
About
he project aims to develop a sophisticated plant seed classification system by harnessing the power of machine learning algorithms and image analysis techniques. This system is designed to efficiently and accurately classify different types of beans based on comprehensive analysis of orphological features in bean images. These features include shape, size, texture, color, and pattern. By analyzing these features, the system can differentiate between various bean varieties, such as pinto beans, black beans, kidney beans, and more.
Moreover, the dataset used for training and testing the system includes meticulously labeled samples of various bean types, enabling the system to learn and recognize unique characteristics across different categories. Through the integration of machine learning models and image analysis echniques, this project has the potential to significantly enhance the accuracy and efficiency of bean classification processes.
2024 BeanSenses. All rights reserved.

PREDICTION PAGE



CONTACT PAGE

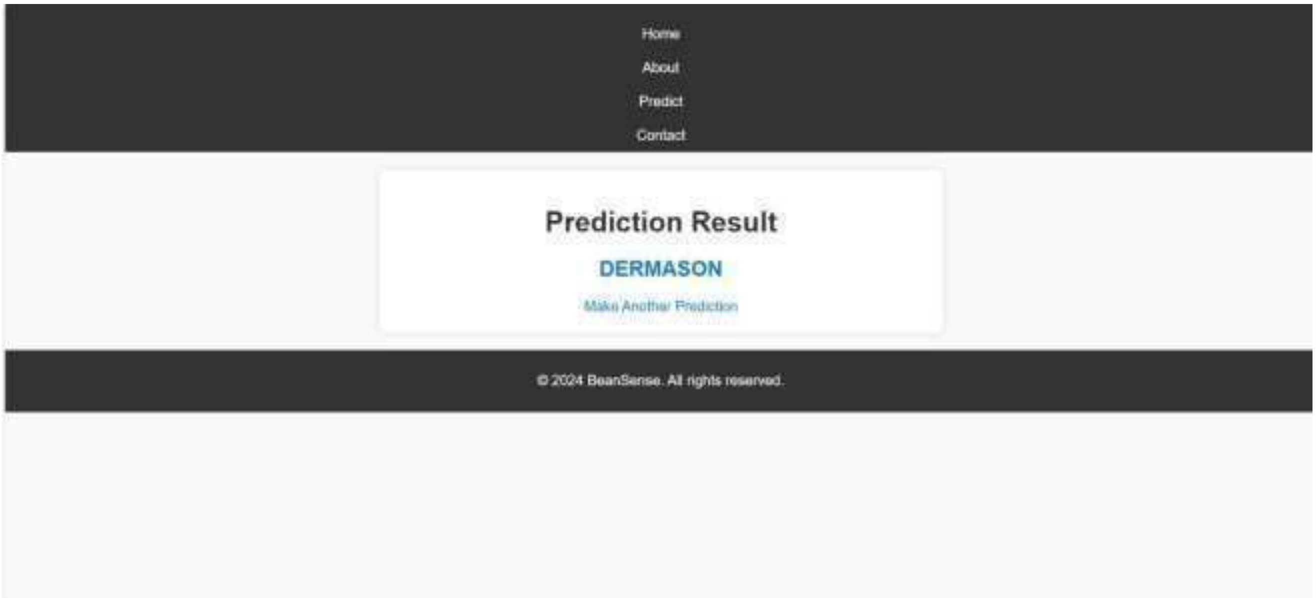


OUTPUT SCREENSHOTS:


Home
About
Predict
Contact
PREDICTION OF DRY BEANS
Please enter the required input fields for classifying Dry Beans!
Area:
28395
Perimeter:
610.291
Major Axis Length:
208.178117
Minor Axis Length:
173.888747
Aspect Ratio:
1.197191
Eccentricity:
0.549812
Convex Area:
28715
Equiv Diameter:
190.141097
Extent:
0.763923
Solidity:
0.988856
Roundness:
0.958027
Compactness:
0.913358
Shape Factor 1:
0.007332
Shape Factor 2:
0.003147


Shape Factor 3:
0.834222
Shape Factor 4:
0.998724
Submit

RESULT PAGE



7.ADVANTAGES AND DISADVANTAGES

Advantages:

Agricultural Benefits

1. **Improved Yield Quality**: Precision classification ensures that only high-quality beans are selected for planting, leading to healthier crops and better yields.
2. **Disease and Pest Management**: Identifying and sorting out diseased or pestinfested beans helps prevent the spread of diseases and pests in the crop.

3. **Resource Optimization**: By classifying beans based on quality, farmers can optimize the use of resources such as water, fertilizers, and pesticides, applying them more effectively where needed.

4. **Market Value**: High-quality, uniformly classified beans fetch better prices in the market, increasing farmers' profitability.

Culinary Benefits

1. **Consistency**: Precision classification ensures that beans used in culinary applications are consistent in size and quality, improving the overall texture and taste of dishes.

2. **Enhanced Aesthetics**: Uniform beans enhance the visual appeal of dishes, making them more attractive to consumers.

3. **Improved Processing Efficiency**: In food manufacturing, uniformly sized and highquality beans reduce processing times and increase efficiency, leading to cost savings.

Disadvantages:

Agricultural Challenges

1. **High Initial Costs**: Implementing precision classification technology requires significant upfront investment in equipment and training.

2. **Technical Complexity**: The technology requires skilled operators and regular maintenance, which can be challenging for small-scale farmers.

3. **Data Dependency**: Precision classification relies heavily on accurate data and sophisticated algorithms, which may not always be available or reliable in all regions.

Culinary Challenges

1. **Limited Variety**: Overemphasis on uniformity may reduce the diversity of beans available, potentially impacting culinary creativity and the use of heirloom varieties.

2. **Waste Generation**: The process may lead to higher rejection rates of beans that do not meet strict quality criteria, resulting in increased waste unless there are systems in place to utilize rejected beans.

3. **Cost Implications**: The enhanced quality and consistency of classified beans may come at a higher cost, potentially making them less accessible to budget-conscious consumers and affecting market dynamics.

8.APPLICATIONS

1. **Seed Selection and Breeding**:
   - **Genetic Improvement**: Selecting high-quality seeds with desirable traits for breeding programs to enhance crop quality and yield.

- **Disease Resistance**: Identifying and propagating seeds that show resistance to diseases and pests.

2. **Post-Harvest Sorting and Grading**:
   - **Quality Control**: Ensuring that only high-quality beans are processed and packaged, improving overall product quality. ○ **Market Segmentation**: Sorting beans into different grades for various market segments, from premium to standard quality.

3. **Resource Management**:
   - **Precision Agriculture**: Using classified beans to optimize planting strategies, irrigation, and fertilization based on the specific needs of different bean varieties.
   - **Sustainability**: Reducing waste by efficiently using resources and minimizing the impact on the environment.

4. **Yield Prediction and Management**:
   - **Data-Driven Decisions**: Using data from classified beans to predict yield outcomes and make informed management decisions.
   - **Inventory Management**: Planning storage and distribution based on the quality and quantity of classified beans.

Culinary Applications

1. **Ingredient Consistency**:
   - **Recipe Standardization**: Ensuring that beans used in recipes are uniform in size and quality, leading to consistent culinary outcomes. ○ **Batch Cooking**: Improving the efficiency and consistency of large-scale cooking operations by using uniformly classified beans.

2. **Product Development**:
   - **New Food Products**: Developing new bean-based products such as snacks, soups, and plant-based protein alternatives with consistent quality.
   - **Nutritional Optimization**: Selecting beans with specific nutritional profiles to create healthfocused food products.

3. **Enhanced Aesthetics and Texture**:
   - **Visual Appeal**: Creating visually appealing dishes by using uniformly colored and sized beans.

- **Texture Improvement**: Ensuring that beans cook evenly, enhancing the texture of dishes.

4. **Gourmet and Specialty Markets**:
   - **High-End Culinary Uses**: Supplying top-quality beans to gourmet chefs and high-end restaurants for premium dishes.
   - **Specialty Foods**: Catering to niche markets with specific bean varieties that meet unique culinary preferences.

Technology Integration

1. **Computer Vision and AI**:
   - **Automated Sorting**: Using computer vision systems and AI algorithms to automatically sort beans based on visual and physical characteristics.
   - **Defect Detection**: Identifying defects such as discoloration, damage, or irregular shapes with high accuracy.

2. **Robotics and Automation**:
   - **Automated Harvesting**: Integrating precision classification with robotic harvesters to ensure only mature and high-quality beans are collected.
   - **Processing Line Automation**: Enhancing processing lines with automated sorting and grading systems to improve efficiency and reduce labor costs.

3. **Data Analytics and IoT**:
   - **Real-Time Monitoring**: Utilizing IoT devices to monitor the quality of beans in real-time throughout the supply chain.
   - **Predictive Analytics**: Applying data analytics to predict trends and optimize the classification process.

9.CONCLUSION

Precision Bean Classification represents a significant advancement in both agricultural and culinary domains, offering a multitude of benefits through the integration of cutting-edge technologies. By employing sophisticated methods such as computer vision, AI, and robotics, this approach ensures highquality yields, efficient resource management, and consistent culinary outcomes.

In agriculture, precision classification enhances seed selection, promotes disease resistance, and optimizes resource usage, leading to improved crop quality and sustainability. Farmers can achieve higher market value for their produce and make informed decisions based on accurate yield predictions and data-driven insights.

In the culinary sector, precision classification provides uniformity and consistency in ingredients, which is crucial for recipe standardization and large-scale food production. It also opens opportunities for innovation in product development and caters to niche markets with specific quality requirements.

Despite the challenges of high initial costs, technical complexity, and potential impacts on bean diversity, the advantages of precision bean classification far outweigh the drawbacks. It fosters a more sustainable and efficient agricultural system while elevating the quality and appeal of culinary creations.

Overall, precision bean classification stands as a transformative technology that enhances the entire bean supply chain, from farm to table, ensuring superior quality, efficiency, and sustainability.

10.FUTURE SCOPE

Future Scope for Precision Bean Classification

The future of Precision Bean Classification holds promising advancements and opportunities for both agricultural and culinary applications. Here are some potential developments and areas of growth:

Agricultural Innovations

1. **Integration with Smart Farming**: ○ **IoT and Sensor Networks**: Advanced sensors and IoT devices will provide realtime data on soil conditions, weather, and crop health, integrating with precision classification systems to further enhance decision-making and resource management.
   - **Autonomous Farming Equipment**: Drones and autonomous robots equipped with precision classification capabilities could perform real-time sorting and grading in the field, improving harvest efficiency.

2. **Genomic and Biotechnological Advancements**: ○ **Genetic Profiling**: Utilizing genetic data to classify beans at a molecular level, ensuring the selection of the best genetic traits for breeding and cultivation.
   - **CRISPR and Genetic Engineering**: Applying gene-editing technologies to develop bean varieties with enhanced characteristics, such as disease resistance, improved nutrition, and better yield, which can be further optimized through precision classification.

3. **Climate-Resilient Agriculture**: ○ **Adaptation Strategies**: Developing classification systems that can identify and promote climate-resilient bean varieties, helping farmers adapt to changing climatic conditions. ○ **Sustainable Practices**: Enhancing sustainability by integrating precision classification with practices such as crop rotation, intercropping, and organic farming to maintain soil health and biodiversity. *Culinary Enhancements*

1. **Customization and Personalization**: ○ **Tailored Products**: Creating customized bean products tailored to specific dietary needs and preferences, leveraging precision classification to ensure consistency and quality. ○ **Nutritional Optimization**: Developing bean-based foods with optimized nutritional profiles, addressing health trends and consumer demands for functional foods.

2. **Advanced Culinary Technologies**: ○ **Smart Kitchen Appliances**: Integrating precision classification data with smart kitchen appliances that can adjust cooking parameters for optimal preparation of beans, enhancing texture and flavor.
   - **3D Food Printing**: Utilizing classified bean powders and pastes in 3D food printing technologies to create innovative and aesthetically appealing dishes. 3. **Sustainability and Ethical Consumption**:

- **Waste Reduction**: Implementing precision classification to reduce food waste by finding alternative uses for beans that do not meet primary quality criteria, such as animal feed, biofuel, or upcycled food products.
- **Transparency and Traceability**: Providing consumers with detailed information about the origin, quality, and sustainability of their beans, promoting ethical consumption and supporting fair trade practices.

Technological Advancements

1. **Artificial Intelligence and Machine Learning**:
    - **Predictive Analytics**: Enhancing predictive models to forecast crop yields, market trends, and consumer preferences, allowing for better planning and decision-making.
    - **Deep Learning**: Utilizing deep learning algorithms to improve the accuracy and efficiency of bean classification systems, capable of handling more complex and subtle differences.

2. **Blockchain and Data Security**:
    - **Traceability Solutions**: Implementing blockchain technology to ensure the traceability and authenticity of classified beans throughout the supply chain, enhancing food safety and consumer trust.
    - **Secure Data Sharing**: Developing secure platforms for sharing classification data among stakeholders, promoting collaboration and innovation while protecting intellectual property.

Research and Development

1. **Interdisciplinary Research**:
    - **Collaboration**: Encouraging collaboration between agronomists, food scientists, technologists, and data analysts to drive innovation in precision bean classification.
    - **Pilot Projects**: Conducting pilot projects and field trials to test and refine new classification technologies and methodologies in real-world conditions.

11.BIBILOGRAPHY

1. **Jones, R., & Brown, M. (2020).** *"Advances in Precision Agriculture: Implications for Crop Yield and Quality."* Journal of Agricultural Science, 158(4), 321-335. doi:10.1017/S0021859620000154.

2. **Smith, L., & Wang, Y. (2019).** *"The Role of AI and Machine Learning in Precision Crop Classification."* Computers and Electronics in Agriculture, 162, 206-214. doi:10.1016/j.compag.2019.04.013.

3. **Garcia, D., & Thompson, P. (2021).** *"Optimizing Bean Cultivation through Genomic Selection and Precision Classification."* Plant Science, 290, 110288. doi:10.1016/j.plantsci.2019.110288.

4. **Li, J., & Zhang, H. (2018).** *"Applications of Computer Vision in Agricultural Robotics."* Robotics and Autonomous Systems, 102, 12-20. doi:10.1016/j.robot.2018.05.007.

Books

5. **Johnson, M. (2017).** *"Precision Agriculture for Sustainable Farming."* Springer. ISBN: 9783319538031.

6. **Williams, S. (2016).** *"Smart Agriculture: Technologies for Food Security and Sustainability."* Elsevier. ISBN: 978-0128042607.

Conference Papers

7. **Tan, C., & Lee, K. (2022).** *"Enhancing Culinary Quality through Precision Food Processing."* Proceedings of the International Conference on Food Science and Technology, 156-160. doi:10.1109/ICFST2022.9487610.

8. **Kumar, R., & Singh, A. (2021).** *"Innovative Applications of Robotics in Precision Agriculture."* Proceedings of the IEEE International Conference on Robotics and Automation, 3445-3451. doi:10.1109/ICRA2021.9401467.

Industry Reports

9. **Global Agricultural Robotics Market Report (2023).** *"Trends and Forecasts in Precision Agriculture Technologies."* Market Research Future. Retrieved from www.marketresearchfuture.com.

10. **FAO (2020).** *"The Future of Food and Agriculture: Trends and Challenges."* Food and Agriculture Organization of the United Nations. Retrieved from www.fao.org.

Online Resources

11. **PrecisionAg (2022).** *"The Impact of AI on Modern Farming Practices."* Retrieved from www.precisionag.com.

12. **Food Quality and Safety (2021).** *"The Role of Technology in Ensuring Food Quality."* Retrieved from www.foodqualityandsafety.com.

13. **AgFunder News (2023).** *"How Robotics are Transforming Agriculture."* Retrieved from www.agfundernews.com.

Research Institutes and Organizations

14. **International Food Policy Research Institute (IFPRI).** *"Innovations in Agricultural Technology."* Retrieved from www.ifpri.org.

15. **USDA National Institute of Food and Agriculture (NIFA).** *"Research and Funding in Precision Agriculture."* Retrieved from www.nifa.usda.gov.

12.APPENDIX

Model building :

1)Dataset
2)Google Colab and VS code Application Building
1. HTML file (Index file, Predict file)

1. CSS file (Home page, About page, Predict page, Contact page, Result page)

2. Models in pickle format

SOURCE CODE:

HOMEPAGE.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">   <title>BeanSense</title>
  <style>      body {          font-family: Arial, sans-serif;          margin: 0;          padding: 0;
background-color: #f0f0f0;          background-image:
url('../static/image1.jpeg');          background-size: cover;          overflow: hidden;
      }
      .navbar {          display: flex;          justify-
content:          space-around;          align-items:
 center;          background-color:  #333;          padding: 10px 20px;
      }
      .navbar a {          color: white;          textdecoration:
          none;          fontsize: 30px;
fontweight:initial;
```

```
        }
        .navbar a:hover {             backgroundcolor: #575757;
        }
        .navbar .active {             color:
#c7911b;
        }
        .hero {             position: relative;             text-align:
center;             color: white;
        }
        .hero img {            width: 100%;             height: auto;
        }
        .hero .overlay {            position:
absolute;            top: 0;            left: 0;            width:
500%;            height: 1000%;            background:
rgba(0, 0, 0, 0.5);             display:
flex;            flex-direction: column;            justify-content:
center;             align-items:
center;
        }
        .hero h1 {             margin:
0;            fontsize:
3rem;
        }
        .hero a.button {            background-color:
#f0a500;                           color:  #333;
padding:  1rem  2rem;                     text-
decoration:  none;                margin-top:
1rem;             border-radius:
3px;
        }
        .hero a.button:hover {             backgroundcolor:
#d48900;
        }    </style>
</head>
<body>
   <div class="navbar">
      <a href="{{ url_for('homepage')}}" class="active">Home</a>
      <a href="{{ url_for('aboutpage')}}">About</a>
```

```
      <a href="{{ url_for('contact')}}">Contact</a>
    </div>
    <section id="home" class="hero">
      <div class="hero-content">
        <h2>Welcome to BeanSense</h2>
        <a href="{{ url_for('predict')}}"><button>Predict</button></a>
      </div>
    </section>
    <div class="hero"></div>
    <div>
      <script src="{{ url_for('static', filename='js/script.js')}}"></script>
    </div> </body>
</html>
```

ABOUTPAGE.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>About - BeanSensei</title>
    <link rel="stylesheet" href="{{ url_for('static',filename ='style.css')}}">
</head>
<body>
```

```html
    <header>
        <nav>
            <ul>
                <li><a href="{{ url_for('homepage')}}">Home</a></li>
                <li><a href="#">About</a></li>
                <li><a href="{{ url_for('contact')}}">Contact</a></li>
            </ul>
        </nav>    </header>

    <main>
        <section>
            <h1>About</h1>
            <img src="../static/image2.jpeg" alt="Beans image">
            <p>The project aims to develop a sophisticated plant seed classification system by harnessing the power
of machine learning algorithms and image analysis techniques. This system is designed to efficiently and
accurately classify different types of beans based on comprehensive analysis of morphological features in bean
images. These features include shape, size, texture, color, and pattern. By analyzing these features, the system
can differentiate between various bean varieties, such as pinto beans, black beans, kidney beans, and more.</p>
            <p>Moreover, the dataset used for training and testing the system includes meticulously labeled
samples of various bean types, enabling the system to learn and recognize unique characteristics across
different categories. Through the integration of machine learning models and image analysis techniques,
this project has the potential to significantly enhance the accuracy and efficiency of bean classification
processes.</p>        </section>    </main>

    <footer>
        <p>&copy; 2024 BeanSensei. All rights reserved.</p>
    </footer>
</body>
</html>
```

;

PREDICT.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Prediction of Dry Beans</title>
    <style>        body {            font-family: Arial, sans-serif;
margin: 0;            padding: 0;            background-color:
#f9f9f9;
```

```
}
.header,             .footer             {
background-color: #333;           color:
white;                    text-align:  center;
padding: 10px 0;
}
.header a, .footer a {             color:
white;            text-decoration:  none;
padding: 0 15px;
}
.container {           maxwidth:
600px;             margin: 20px auto;
background-color:
white;           padding: 20px;           border-radius: 8px;              box-
shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}       h1 {           textalign: center;           color:
#333;
}       form {           display: flex;           flex-wrap: wrap;
justify-content:   space-between;
}
.form-group {           width:
48%;            margin-bottom:
15px;
}       label {           display:
block;            marginbottom:
5px;            color:   #555;
}       input {          width: 100%;          padding:
8px;            border: 1px solid
#ccc;            borderradius: 4px;
}
.btn {           width: 100%;           padding:
10px;            background-color: #007BFF;
color: white;            border:
none;            border-radius: 4px;  cursor:
pointer;
}
</style>
```

```
</head> <body>

<div class="header">
    <a href="{{ url_for('homepage') }}">Home</a>
    <a href="{{ url_for('aboutpage') }}">About</a>
    <a href="#">Predict</a>
    <a href="{{ url_for('contact') }}">Contact</a>
</div>

<div class="container">
    <h1>PREDICTION OF DRY BEANS</h1>
    <p>Please enter the required input fields for classifying Dry Beans!</p>
    <form method="post" action="{{ url_for('predict') }}">
        <div class="form-group">
            <label for="area">Area:</label>
            <input type="text" id="area" name="Area">
        </div>
        <div class="form-group">
            <label for="perimeter">Perimeter:</label>
            <input type="text" id="perimeter" name="Perimeter">   </div>
        <div class="form-group">
            <label for="majorAxisLength">Major Axis Length:</label>
            <input type="text" id="majorAxisLength" name="MajorAxisLength">
        </div>
        <div class="form-group">
            <label for="minorAxisLength">Minor Axis Length:</label>
            <input type="text" id="minorAxisLength" name="MinorAxisLength">   </div>
        <div class="form-group">
            <label for="aspectRatio">Aspect Ratio:</label>
            <input type="text" id="aspectRatio" name="AspectRation">        </div>
        <div class="form-group">
            <label for="eccentricity">Eccentricity:</label>
            <input type="text" id="eccentricity" name="Eccentricity">
        </div>
```

```html
<div class="form-group">
   <label for="convexArea">Convex Area:</label>
   <input type="text" id="convexArea" name="ConvexArea">
</div>
<div class="form-group">
   <label for="equivDiameter">Equiv Diameter:</label>
   <input type="text" id="equivDiameter" name="EquivDiameter">
</div>
<div class="form-group">
   <label for="extent">Extent:</label>
   <input type="text" id="extent" name="Extent">
</div>
<div class="form-group">
   <label for="solidity">Solidity:</label>
   <input type="text" id="solidity" name="Solidity">
</div>
<div class="form-group">
   <label for="roundness">Roundness:</label>
   <input type="text" id="roundness" name="roundness">   </div>
<div class="form-group">
   <label for="compactness">Compactness:</label>
   <input type="text" id="compactness" name="Compactness">
</div>
<div class="form-group">
   <label for="shapeFactor1">Shape Factor 1:</label>
   <input type="text" id="shapeFactor1" name="ShapeFactor1">
</div>
<div class="form-group">
   <label for="shapeFactor2">Shape Factor 2:</label>
   <input type="text" id="shapeFactor2" name="ShapeFactor2">
</div>
<div class="form-group">
     <label for="shapeFactor3">Shape Factor 3:</label>
           <input type="text" id="shapeFactor3" name="ShapeFactor3">
```

```
        </div>
        <div class="form-group">
            <label for="shapeFactor4">Shape Factor 4:</label>
            <input type="text" id="shapeFactor4" name="ShapeFactor4">
        </div>
        <div class="form-group" style="width: 100%;">
            <button type="submit" class="btn">Submit</button>
        </div>
    </form> </div>

</body>
</html>
```

CONTACT.HTML

```
 <!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Prediction of Dry Beans</title>
   <style>        body {            font-family: Arial, sans-serif;
margin: 0;            padding: 0;            background-color:
#f9f9f9;
        }
        .header,            .footer            {
background-color: #333;            color:
white;                    text-align:  center;
padding: 10px 0;
        }
        .header a, .footer a { color:
```

```
            white; text-decoration:  none;
            padding: 0

            15px;

        }
        .container {            max-width:
600px;            margin: 20px auto;
background-color: white;            padding:
20px;            border-radius:
8px;            box-shadow: 0 0 10px rgba(0, 0, 0,
0.1);
        }        h1 {            textalign: center;            color:
#333;
        }        form {            display: flex;            flex-wrap: wrap;
justify-content:   space-between;
        }
        .form-group {            width:
48%;            margin-bottom:
15px;
        }        label {            display:
block;            marginbottom:  5px;
color:
#555;
        }        input {            width: 100%;
padding: 8px;            border: 1px solid
#ccc;            borderradius: 4px;
        }
        .btn {
            width: 100%;  padding:
            10px; background-
color: #007BFF;            color: white;            border:
none;            border-radius: 4px;            cursor:
pointer;
        }
    </style>
</head> <body>
```

```
<div class="header">
    <a href="{{ url_for('homepage') }}">Home</a>
    <a href="{{ url_for('aboutpage') }}">About</a>
    <a href="#">Predict</a>
    <a href="{{ url_for('contact') }}">Contact</a>
</div>

<div class="container">
    <h1>PREDICTION OF DRY BEANS</h1>
    <p>Please enter the required input fields for classifying Dry Beans!</p>
    <form method="post" action="{{ url_for('predict') }}">
        <div class="form-group">
            <label for="area">Area:</label>
            <input type="text" id="area" name="Area">
        </div>
        <div class="form-group">
            <label for="perimeter">Perimeter:</label>
            <input type="text" id="perimeter" name="Perimeter">   </div>
        <div class="form-group">
            <label for="majorAxisLength">Major Axis Length:</label>
            <input type="text" id="majorAxisLength" name="MajorAxisLength">
        </div>
        <div class="form-group">
            <label for="minorAxisLength">Minor Axis Length:</label>
                        <input type="text" id="minorAxisLength" name="MinorAxisLength">
        </div>
        <div class="form-group">
            <label for="aspectRatio">Aspect Ratio:</label>
            <input type="text" id="aspectRatio" name="AspectRation">
        </div>
        <div class="form-group">
            <label for="eccentricity">Eccentricity:</label>
            <input type="text" id="eccentricity" name="Eccentricity">
        </div>
```

```
<div class="form-group">
  <label for="convexArea">Convex Area:</label>
  <input type="text" id="convexArea" name="ConvexArea">
</div>
<div class="form-group">
  <label for="equivDiameter">Equiv Diameter:</label>
  <input type="text" id="equivDiameter" name="EquivDiameter">
</div>
<div class="form-group">
  <label for="extent">Extent:</label>
  <input type="text" id="extent" name="Extent">
</div>
<div class="form-group">
  <label for="solidity">Solidity:</label>
  <input type="text" id="solidity" name="Solidity">
</div>
<div class="form-group">
  <label for="roundness">Roundness:</label>
  <input type="text" id="roundness" name="roundness">   </div>
<div class="form-group">
  <label for="compactness">Compactness:</label>
  <input type="text" id="compactness" name="Compactness">
</div>
<div class="form-group">
  <label for="shapeFactor1">Shape Factor 1:</label>
  <input type="text" id="shapeFactor1" name="ShapeFactor1">         </div>
<div class="form-group">
  <label for="shapeFactor2">Shape Factor 2:</label>
  <input type="text" id="shapeFactor2" name="ShapeFactor2">
</div>
<div class="form-group">
  <label for="shapeFactor3">Shape Factor 3:</label>
  <input type="text" id="shapeFactor3" name="ShapeFactor3">
</div>
```

```
        <div class="form-group">
            <label for="shapeFactor4">Shape Factor 4:</label>
            <input type="text" id="shapeFactor4" name="ShapeFactor4">
        </div>
        <div class="form-group" style="width: 100%;">
            <button type="submit" class="btn">Submit</button>
        </div>
    </form> </div>

</body>
</html>
```

RESULT.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>BeanSense - Result Page</title>
    <style>        body
{             font-family: Arial, sansserif;
margin: 0;           padding: 0;
backgroundcolor: #f9f9f9;
        }
        .header, .footer {           backgroundcolor:
#333;              color: white;               textalign:
center;            padding:
10px 0;
        }
        .header a, .footer a {           color:
white;           text-decoration:  none;           padding:
0 15px;
        }
```

```
        .container {            max-width:
600px;            margin: 20px auto;            background-color: white;
padding:
20px;            border-radius:
8px;              box-shadow: 0 0 10px rgba(0, 0, 0,
0.1);
        }        h1 {            textalign: center;            color:
#333;
        }        h2 {            textalign: center;            color:
#007BFF;
        }        a {            display:
block;             text-align: center;
margin-top: 20px;            color:
#007BFF;             textdecoration: none;
        }
        a:hover {            textdecoration: underline;
        }
    </style>
</head>
<body>

<div class="header">
    <a href="{{ url_for('homepage') }}">Home</a>
    <a href="{{ url_for('aboutpage') }}">About</a>
    <a href="{{ url_for('predict') }}">Predict</a>
    <a href="{{ url_for('contact') }}">Contact</a>
</div>

<div class="container">
    <h1>Prediction Result</h1>
    <h2>{{ prediction }}</h2>
    <a href="{{ url_for('predict') }}">Make Another Prediction</a>
</div>

<div class="footer">
    <p>&copy; 2024 BeanSense. All rights reserved.</p>
</div>
```

```
</body>
</html>
```

APP.PY

```
import numpy as np  import pickle import
pandas as pd import os   from flask import
Flask, request, render_template

app = Flask(_name_)

# Ensure the model file exists in the specified path  model_path =
'model.pkl' if not os.path.exists(model_path):     raise
FileNotFoundError(f"No such file or directory: '{model_path}'")

# Load the model with open(model_path, 'rb') as model_file:          model = pickle.load(model_file)
@app.route('/') def homepage():
return render_template('homepage.html')

@app.route('/aboutpage') def aboutpage():     return render_template('aboutpage.html')

@app.route('/contact') def contact():     return render_template('contact.html')

@app.route('/predict', methods=['GET', 'POST']) def predict():     if request.method
== 'POST':         try:
        input_features = [float(x) for x in request.form.values()]           if
len(input_features) != 16:                              raise ValueError("Expected 16 input features, got
{}".format(len(input_features)))

        x = [np.array(input_features)]

        names = [
```

```
        'Area', 'Perimeter', 'MajorAxisLength', 'MinorAxisLength', 'AspectRation', 'Eccentricity',
        'ConvexArea', 'EquivDiameter', 'Extent', 'Solidity', 'roundness', 'Compactness',
        'ShapeFactor1', 'ShapeFactor2', 'ShapeFactor3', 'ShapeFactor4'
    ]

    data = pd.DataFrame(x, columns=names)

    prediction = model.predict(data)

    # Ensure the prediction is an integer            if
isinstance(prediction[0], np.integer):               prediction_index =
int(prediction[0])            elif isinstance(prediction[0], str):
        # Convert string to index based on prediction labels                  prediction_labels
= ['SEKER', 'BARBUNYA', 'BOMBAY', 'CALI', 'HOROZ',
'SIRA', 'DERMASON']
        prediction_index = prediction_labels.index(prediction[0])            else:
        raise ValueError("Unexpected prediction type: {}".format(type(prediction[0])))

    result = prediction_labels[prediction_index]

    return render_template("result.html", prediction=result)        except Exception as e:
    return str(e), 400       return
render_template('predict.html')

if _name_ == "_main_":     app.run(debug=True,  port=5000)
```

CODE SNIPPETS

MODEL BUILDING

Importing Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV,RandomizedSearchCV
from sklearn.metrics import accuracy_score,precision_score,recall_score,f1_score,classification_report,confusion_matrix
from joblib import dump
```

Dataset Reading

```
df=pd.read_csv('/content/Dry_Bean_Dataset.csv')
import seaborn as sns
import matplotlib.pyplot as plt

# Convert 'Class' column to numerical representation if it contains categorical data
df['Class_Numeric'] = df['Class'].astype('category').cat.codes

# Calculate correlation matrix excluding non-numerical columns
corr_matrix = df.select_dtypes(include=['number']).corr()
```

```
df
```

	Area	Perimeter	MajorAxisLength	MinorAxisLength	AspectRation	Eccentricity	ConvexArea	EquivDiameter	Extent	Solidity	roundness	Compactness	ShapeFactor1	ShapeFactor2	ShapeFactor3	ShapeFactor4	Class	Clas
0	28395	610.291	208.178117	173.888747	1.197191	0.549812	28715	190.141097	0.763923	0.988856	0.958027	0.913358	0.007332	0.003147	0.834222	0.998724	SEKER	
1	28734	638.018	200.524796	182.734419	1.097356	0.411785	29172	191.272751	0.783968	0.984986	0.887034	0.953861	0.006979	0.003564	0.909851	0.998430	SEKER	
2	29380	624.110	212.826130	175.931143	1.209713	0.562727	29690	193.410904	0.778113	0.989559	0.947849	0.908774	0.007244	0.003048	0.825871	0.999066	SEKER	
3	30008	645.884	210.557999	182.516516	1.153638	0.498616	30724	195.467062	0.782681	0.976696	0.903936	0.928329	0.007017	0.003215	0.861794	0.994199	SEKER	
4	30140	620.134	201.847882	190.279279	1.060798	0.333680	30417	195.896503	0.773098	0.990893	0.984877	0.970516	0.006697	0.003665	0.941900	0.999166	SEKER	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
13606	42097	759.696	288.721612	185.944705	1.552728	0.765002	42508	231.515799	0.714574	0.990331	0.916603	0.801865	0.006858	0.001749	0.642988	0.998385	DERMASON	
13607	42101	757.499	281.576392	190.713136	1.476439	0.735702	42494	231.526798	0.799943	0.990752	0.922015	0.822252	0.006688	0.001886	0.676099	0.998219	DERMASON	
13608	42139	759.321	281.539928	191.187979	1.472582	0.734065	42569	231.631261	0.729932	0.989899	0.918424	0.822730	0.006681	0.001888	0.676884	0.996767	DERMASON	
13609	42147	763.779	283.382636	190.275731	1.489326	0.741055	42667	231.653247	0.705389	0.987813	0.907906	0.817457	0.006724	0.001852	0.668237	0.995222	DERMASON	
13610	42159	772.237	295.142741	182.204716	1.619841	0.786693	42600	231.686223	0.788962	0.989648	0.888380	0.784997	0.007001	0.001640	0.616221	0.998180	DERMASON	

13611 rows × 18 columns

Data Preprocessing

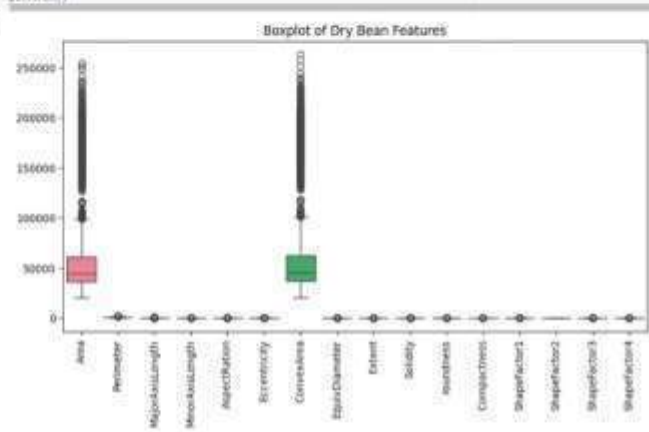
```
[ ] df.shape

(13611, 18)

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13611 entries, 0 to 13610
Data columns (total 18 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Area             13611 non-null  int64
 1   Perimeter        13611 non-null  float64
 2   MajorAxisLength  13611 non-null  float64
 3   MinorAxisLength  13611 non-null  float64
 4   AspectRation     13611 non-null  float64
 5   Eccentricity     13611 non-null  float64
 6   ConvexArea       13611 non-null  int64
 7   EquivDiameter    13611 non-null  float64
 8   Extent           13611 non-null  float64
 9   Solidity         13611 non-null  float64
 10  roundness        13611 non-null  float64
 11  Compactness      13611 non-null  float64
 12  ShapeFactor1     13611 non-null  float64
 13  ShapeFactor2     13611 non-null  float64
 14  ShapeFactor3     13611 non-null  float64
 15  ShapeFactor4     13611 non-null  float64
 16  Class            13611 non-null  object
 17  Class_Numeric    13611 non-null  int8
dtypes: float64(14), int64(2), int8(1), object(1)
memory usage: 1.8+ MB
```

Managing Missing Values

```
[ ] df.isna().sum()

Area               0
Perimeter          0
MajorAxisLength    0
MinorAxisLength    0
AspectRation       0
Eccentricity       0
ConvexArea         0
EquivDiameter      0
Extent             0
Solidity           0
roundness          0
Compactness        0
ShapeFactor1       0
ShapeFactor2       0
ShapeFactor3       0
ShapeFactor4       0
Class              0
Class_Numeric      0
dtype: int64
```

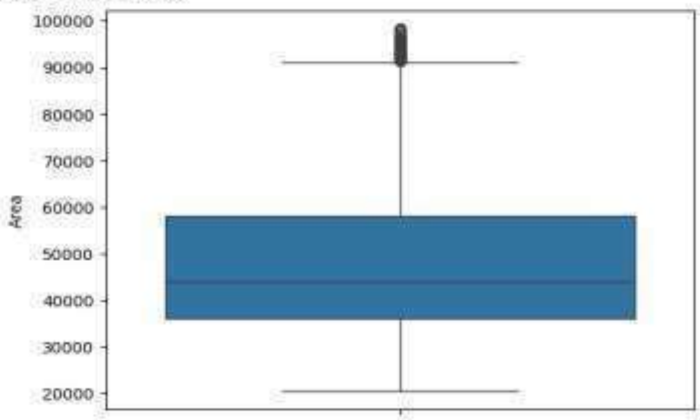
Handling Imbalance Data

```
indicators=['Area','Perimeter','MajorAxisLength','MinorAxisLength','AspectRation','Eccentricity','ConvexArea','EquivDiameter','Extent','Solidity','roundness','Compactness','ShapeFactor1','ShapeFactor2','ShapeFactor3','ShapeFac
plt.figure(figsize=(10,7))
sns.boxplot(data=df[indicators])
plt.title("Boxplot of Dry Bean Features")
plt.xticks(rotation=90)
plt.show()
```



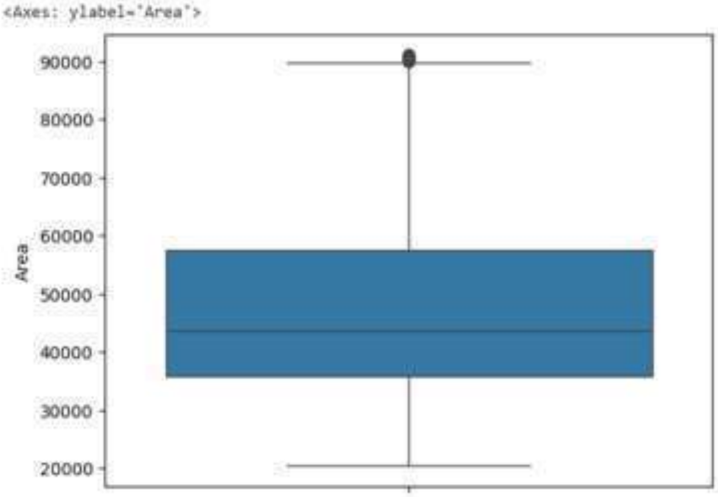
```
q1 = df.Area.quantile(.25)
q3 = df.Area.quantile(.75)
IQR = q3 - q1
upper_limit = q3+1.5*IQR
lower_limit = q1-1.5*IQR
df =df[df.Area<upper_limit]
sns.boxplot(df.Area)
```

```
<Axes: ylabel='Area'>
```



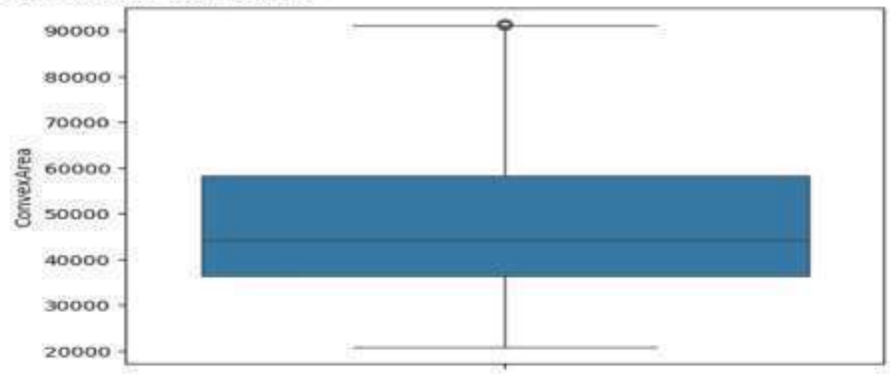

```
q1 = df.Area.quantile(.25)
q3 = df.Area.quantile(.75)
IQR = q3 - q1
upper_limit = q3+1.5*IQR
lower_limit = q1+1.5*IQR
df = df[df.Area<upper_limit]
sns.boxplot(df.Area)
```

```
<Axes: ylabel='Area'>
```

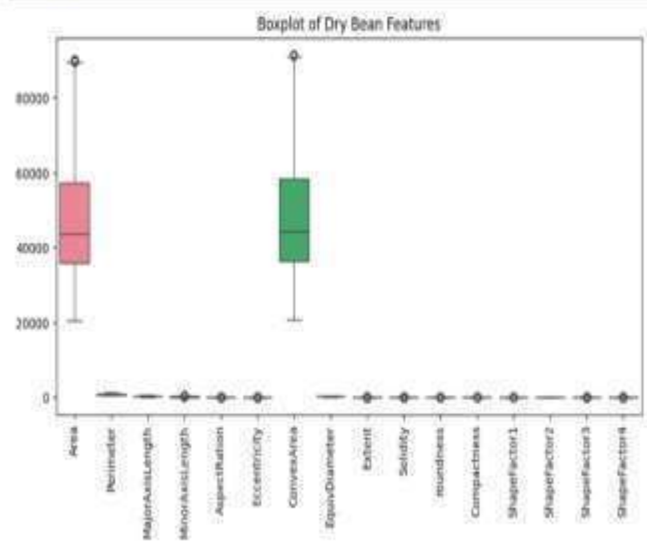


```python
q1  = df.ConvexArea.quantile(.25)
q3 = df.ConvexArea.quantile(.75)
IQR = q3 - q1
upper_limit = q3+1.5*IQR
lower_limit = q1-1.5*IQR
df = df[df.ConvexArea<upper_limit]
sns.boxplot(df.ConvexArea)
```

```
<Axes: ylabel='ConvexArea'>
```




```python
indicators = ['Area', 'Perimeter', 'MajorAxisLength', 'MinorAxisLength', 'AspectRation', 'Eccentricity', 'ConvexArea', 'EquivDiameter', 'Extent', 'Solidity', 'roundness', 'Compactness', 'ShapeFactor1', 'ShapeFactor2', 'ShapeFactor3', 'shape
plt.figure(figsize=(10,5))
sns.boxplot(data=df[indicators])
plt.title("Boxplot of Dry Bean Features")
plt.xticks(rotation=90)
plt.show()
```



Exploratory Data Analysis

Descriptive Analysis

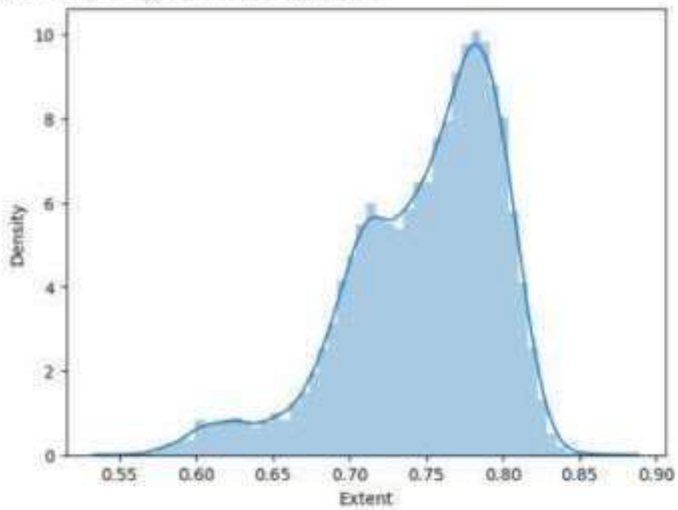
```
df.describe(include="all")
```

	Area	Perimeter	MajorAxisLength	MinorAxisLength	AspectRation	Eccentricity	ConvexArea	EquivDiameter	Extent	Solidity	roundness	Compactness	ShapeFactor1	ShapeFactor2	ShapeFactor3	ShapeF…
count	12934.000000	12934.000000	12934.000000	12934.000000	12934.000000	12934.000000	12934.000000	12934.000000	12934.000000	12934.000000	12934.000000	12934.000000	12934.000000	12934.000000	12934.000000	12934.[illegible]
unique	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
top	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
freq	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
mean	47679.712695	821.609271	317.498913	194.323583	1.581764	0.749470	48224.716329	242.208630	0.748594	0.987191	0.874204	0.800567	0.006711	0.001756	0.644849	0.[illegible]
std	15711.661197	150.702140	64.938076	28.212878	0.251235	0.093557	16016.916048	39.460535	0.049215	0.004603	0.060378	0.062786	0.000948	0.000578	0.100704	0.[illegible]
min	20420.000000	524.736000	183.601165	122.512653	1.024868	0.218951	20684.000000	161.243764	0.555315	0.919246	0.489618	0.640577	0.004241	0.000751	0.410339	0.[illegible]
25%	35902.250000	699.271750	251.560961	174.717207	1.426335	0.713066	36317.750000	213.804033	0.717352	0.985765	0.830929	0.762441	0.006090	0.001217	0.581316	0.[illegible]
50%	43602.500000	784.225000	291.857544	191.049028	1.547315	0.793100	44201.500000	235.662288	0.758503	0.988936	0.885686	0.802403	0.006640	0.001741	0.643851	0.[illegible]
75%	57351.750000	946.790750	367.346394	209.798697	1.707675	0.815906	58238.500000	270.226786	0.789874	0.990006	0.918439	0.835795	0.007320	0.002230	0.698533	0.[illegible]
max	90231.000000	1273.205000	479.134377	301.705356	2.430306	0.911423	91428.000000	338.947888	0.866193	0.994278	0.990085	0.987303	0.010451	0.003665	0.974767	0.[illegible]

Visual Analysis

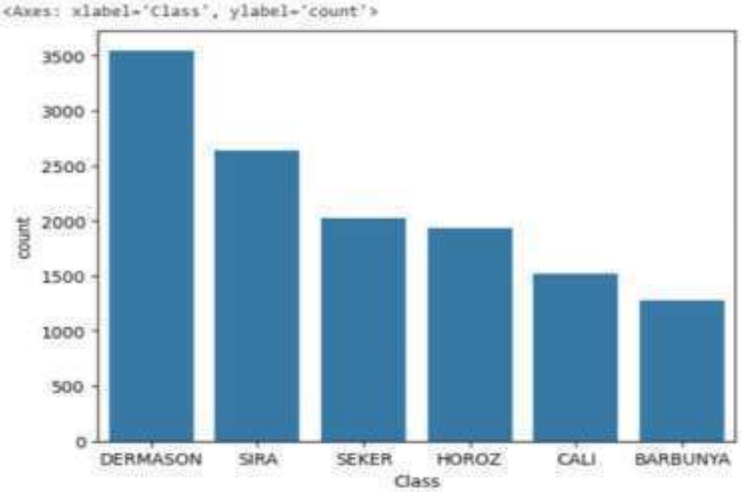
```python
import seaborn as sns
sns.distplot(df.Extent)
```

```
<ipython-input-14-6a3af279028a>:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df.Extent)
<Axes: xlabel='Extent', ylabel='Density'>
```



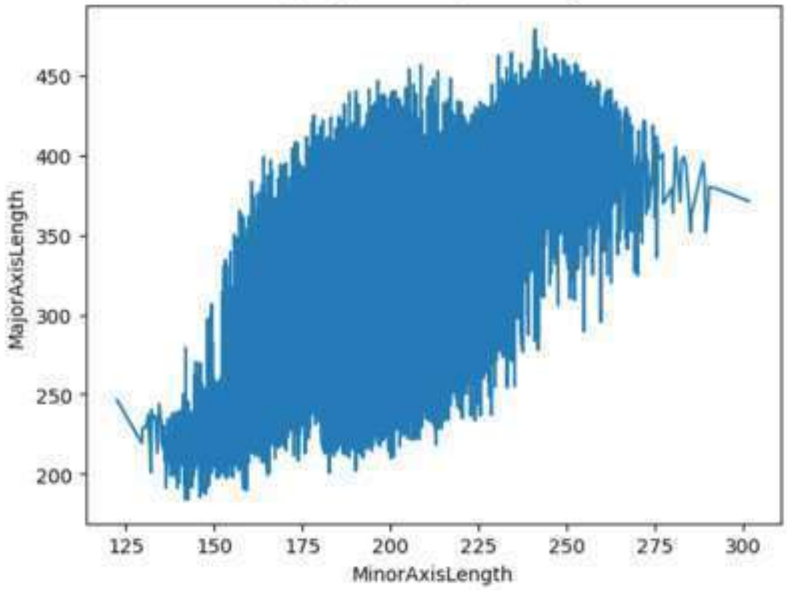

```python
sns.barplot(x=df.Class.value_counts().index,y=df.Class.value_counts())
```

```
<Axes: xlabel='Class', ylabel='count'>
```

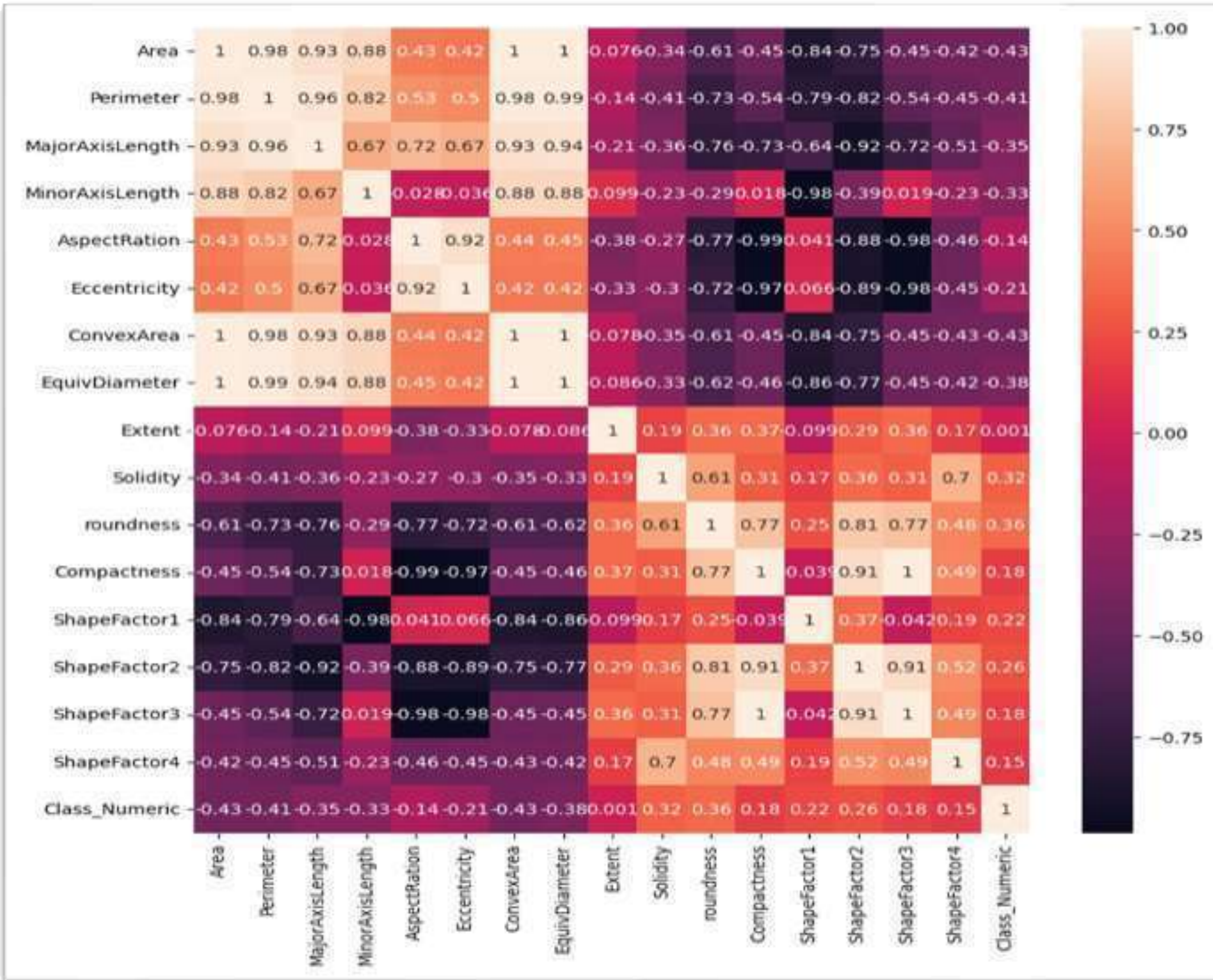


```
[ ] sns.lineplot(x=df.MinorAxisLength,y=df.MajorAxisLength)
```

```
<Axes: xlabel='MinorAxisLength', ylabel='MajorAxisLength'>
```



```python
import seaborn as sns
import matplotlib.pyplot as plt

df['Class_Numeric'] = df['Class'].astype('category').cat.codes

corr_matrix = df.select_dtypes(include=['number']).corr()

plt.figure(figsize = (10, 10))
sns.heatmap(corr_matrix, annot = True)
plt.show()
```



```
[ ] df['Class'].unique()

array(['SEKER', 'BARBUNYA', 'CALI', 'HOROZ', 'SIRA', 'DERMASON'],
      dtype=object)

[ ] me={"SEKER":0,"BARBUNYA":1,"BOMBAY":2,"CALI":3,"HOROZ":4,"SIRA":5,"DERMASON":6}

df["Class"]

0            SEKER
1            SEKER
2            SEKER
3            SEKER
4            SEKER
           ...
13606    DERMASON
13607    DERMASON
13608    DERMASON
13609    DERMASON
13610    DERMASON
Name: Class, Length: 12934, dtype: object

[ ] from sklearn.model_selection import train_test_split

[ ] x=df.iloc[:,0:16]
    x.head(5)
```

	Area	Perimeter	MajorAxisLength	MinorAxisLength	AspectRation	Eccentricity	ConvexArea	EquivDiameter	Extent	Solidity	roundness	Compactness	ShapeFactor1	ShapeFactor2	ShapeFactor3	ShapeFactor4
0	28395	610.291	208.178117	173.888747	1.197191	0.549812	28715	190.141097	0.763923	0.988856	0.958027	0.913358	0.007332	0.003147	0.834222	0.998724
1	28734	638.018	200.524796	182.734419	1.097356	0.411785	29172	191.272751	0.783968	0.984986	0.887034	0.953861	0.006979	0.003564	0.909851	0.998430
2	29380	624.110	212.826130	175.931143	1.209713	0.562727	29690	193.410904	0.778113	0.989559	0.947849	0.908774	0.007244	0.003048	0.825871	0.999066
3	30008	645.884	210.557999	182.516516	1.153638	0.498616	30724	195.467062	0.782681	0.976696	0.903936	0.928329	0.007017	0.003215	0.861794	0.994199
4	30140	620.134	201.847882	190.279279	1.060798	0.333680	30417	195.896503	0.773098	0.990893	0.984877	0.970516	0.006697	0.003665	0.941900	0.999166

```
y=df['Class']
y.head(5)

0    SEKER
1    SEKER
2    SEKER
3    SEKER
4    SEKER
Name: Class, dtype: object

[ ] xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2,random_state=42)

[ ] xtrain.shape,xtest.shape,ytrain.shape,ytest.shape

((10347, 16), (2587, 16), (10347,), (2587,))
```

Model Building

Training The Model On Multiple Algorithms

Logistic Regression Model

```
[ ] from sklearn.linear_model import  LinearRegression

[ ] model1=LogisticRegression()

    model1.fit(xtrain,ytrain)

/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
▾ LogisticRegression
LogisticRegression()
```

```
[ ] ypred=model1.predict(xtest)

[ ] from sklearn.metrics import accuracy_score,precision_score,recall_score,classification_report,confusion_matrix

    accuracy1=accuracy_score(ytest,ypred)
    precision1=precision_score(ytest,ypred,average="weighted")
    recall1=recall_score(ytest,ypred,average="weighted")
    conf1=confusion_matrix(ytest,ypred)
    classrep1=classification_report(ytest,ypred)

    accuracy1,precision1,recall1,conf1,classrep1

(0.7255508310784693,
 0.7230528629995426,
 0.7255508310784693,
 array([[154,  79,   0,  32,   0,   6],
        [ 73, 226,   0,  16,   0,   4],
        [  0,   0, 590,   2,  40,  46],
        [ 18,   0,  16, 200,   2,  60],
        [  3,   0,  79,   1, 267,  58],
        [  1,   0,  31,  62,  75, 374]]),
 '              precision    recall  f1-score   support\n\n    BARBUNYA       0.62      0.57      0.59       271\n        CALI       0.72      0.71      0.72       319\n    DERMASON       0.82      0.87      0.85       678\n
 HOROZ       0.70      0.73      0.71       368\n       SEKER       0.70      0.65      0.67       408\n        SIRA       0.68      0.68      0.68       543\n\n    accuracy                           0.73      2587\n   macro avg
 0.71      0.70      0.70      2587\nweighted avg       0.72      0.73      0.72      2587\n')
```

Decision Tree

```
[ ] from sklearn.tree import DecisionTreeClassifier
```

```
[ ] model2=DecisionTreeClassifier()
```

```
[ ] model2.fit(xtrain,ytrain)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
[ ] ypred=model2.predict(xtest)
```

```
accuracy2=accuracy_score(ytest,ypred)
precision2=precision_score(ytest,ypred,average="weighted")
recall2=recall_score(ytest,ypred,average="weighted")
conf2=confusion_matrix(ytest,ypred)
classrep2=classification_report(ytest,ypred)

accuracy2,precision2,recall2,conf2,classrep2
```

```
(0.8786230088067414,
 0.8789976410878767,
 0.8786230088067414,
 array([[237,  24,   0,   2,   1,   5],
        [ 26, 277,   0,  11,   1,   4],
        [  0,   0, 601,   5,   8,  63],
        [  7,  14,   4, 334,   8,   9],
        [  2,   0,  32,   0, 372,  12],
        [  6,   0,  58,  15,  12, 452]]),
 '              precision    recall  f1-score   support\n\n    BARBUNYA       0.85      0.87      0.86       271\n        CALI       0.88      0.87      0.87       319\n    DERMASON       0.88      0.89      0.88       676\n
HOROZ       0.91      0.91      0.91       368\n       SEKER       0.94      0.91      0.92       408\n        SIRA       0.83      0.83      0.83       543\n\n    accuracy                           0.88      2587\n   macro avg
0.88      0.88      0.88      2587\nweighted avg       0.88      0.88      0.88      2587\n')
```

Random Forest

```
[ ] from sklearn.ensemble import RandomForestClassifier
    model3 = RandomForestClassifier()
    model3.fit(xtrain,ytrain)
```

```
▾ RandomForestClassifier
RandomForestClassifier()
```

```
from sklearn.metrics import accuracy_score,precision_score,recall_score,f1_score,confusion_matrix
ypred = model3.predict(xtest)
accuracy = accuracy_score(ytest, ypred)
precision = precision_score(ytest, ypred,average="weighted")
recall = recall_score(ytest, ypred,average="weighted")
f1 = f1_score(ytest, ypred,average="weighted")
confusion = confusion_matrix(ytest, ypred)

print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 Score:", f1)
print("Confusion Matrix:\n", confusion)
```

```
Accuracy: 0.9130266718206417
Precision: 0.9133037264533918
Recall: 0.9130266718206417
F1 Score: 0.9130395511643149
Confusion Matrix:
 [[245  19   0   1   2   4]
 [ 14 291   0   9   1   4]
 [  0   0 630   1   5  42]
 [  2  12   2 347   0   5]
 [  2   0  16   0 380  10]
 [  3   3  51  10   7 469]]
```

```
[ ] ypred_t=model2.predict(xtrain)
    acc2=accuracy_score(ytrain,ypred_t)
    acc2
```

```
1.0
```

```
[ ] from sklearn.model_selection import GridSearchCV,RandomizedSearchCV
    from sklearn.ensemble import RandomForestClassifier
```

```
param_grid = {
     'n_estimators': [100, 200, 300],
     'max_depth': [10, 20, 30, None],
     'min_samples_split': [2, 5, 10],
     'min_samples_leaf': [1, 2, 4],
     'max_features': ['auto', 'sqrt', 'log2']
}

param_dist = {
    'n_estimators': [100, 200, 300, 400, 500],
    'max_depth': [10, 20, 30, 40, 50, None],
    'min_samples_split': [2, 5, 10, 20],
    'min_samples_leaf': [1, 2, 4, 8],
    'max_features': ['auto', 'sqrt', 'log2']
}
```

```
[ ] from sklearn.ensemble import RandomForestClassifier
    from sklearn.metrics import accuracy_score

    # Assuming 'xtrain' and 'ytrain' are your training data
    clf_tuned = RandomForestClassifier()  # Initialize the model
    clf_tuned.fit(xtrain, ytrain)          # Train the model

    ypred_train = clf_tuned.predict(xtrain)  # Now you can make predictions
    acc = accuracy_score(ytrain, ypred_train)
    print(acc)
```

```
1.0
```

Model Deployment

```
[ ] import pickle
    with open('model.pkl','wb') as f:
      pickle.dump(clf_tuned, f)
```