Project Name: Harvesting Brilliance: A Taxonomic Tale of Pumpkin Seed Varieties

Name: Sanket Manoj Kokate

Roll :190 PRN : 2022011031093

D.Y.Patil Agricultural and Technical University, Talsande.

1.Project Overview

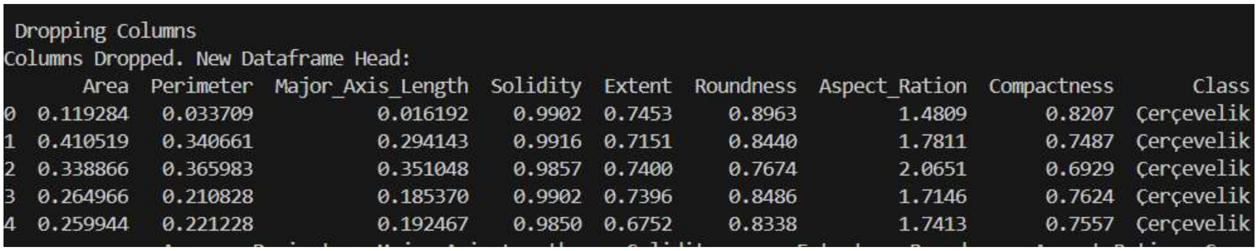
- **Goal:** To classify pumpkin seeds into two distinct varieties ('Çerçevelik' and 'Ürgüp Sivrisi') based on morphological features.
- **Problem Type:** Binary Classification (Supervised Machine Learning).
- **Tech Stack:** Python, Pandas, Scikit-learn, Flask, HTML/CSS.
-

2. Data Collection & Preparation

The foundation of the project involved preparing the raw dataset for analysis.

- **Data Source:**Pumpkin_Seeds_Dataset.xlsx containing 2,500 samples with 13 columns.
- **Loading:** The dataset was loaded using pandas.read_excel().
- **Data Cleaning:**
    - **Null Check:** Verified that the dataset had **0 missing values**.
    - **Outlier Removal:** Used the Interquartile Range (IQR) method to remove outliers from the **'Area'** column. Rows falling outside the range $[Q1 - 1.5*IQR, Q3 + 1.5*IQR]$ were dropped.
- **Preprocessing:**
    - **Scaling:** Applied MinMaxScaler to normalize 'Area', 'Perimeter', and 'Major_Axis_Length' between 0 and 1 to improve model performance.

- **Feature Selection:** Dropped redundant or highly correlated columns: ['Convex_Area', 'Equiv_Diameter', 'Eccentricity', 'Minor_Axis_Length'].
- **Encoding:** Converted the target variable Class from text to integers (0 and 1) using Label Encoding.

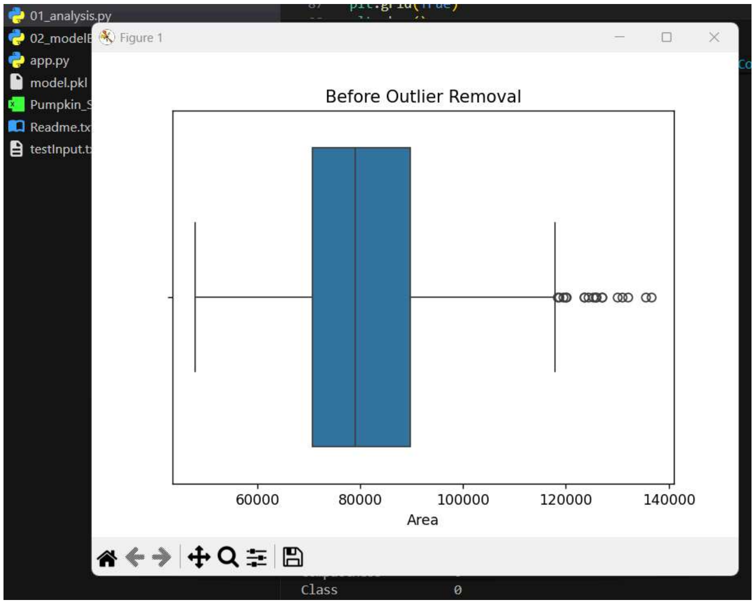


```
 Dropping Columns
Columns Dropped. New Dataframe Head:
       Area  Perimeter  Major_Axis_Length  Solidity  Extent  Roundness  Aspect_Ration  Compactness       Class
0  0.119284   0.033709           0.016192    0.9902  0.7453     0.8963         1.4809       0.8207  Çerçevelik
1  0.410519   0.340661           0.294143    0.9916  0.7151     0.8440         1.7811       0.7487  Çerçevelik
2  0.338866   0.365983           0.351048    0.9857  0.7400     0.7674         2.0651       0.6929  Çerçevelik
3  0.264966   0.210828           0.185370    0.9902  0.7396     0.8486         1.7146       0.7624  Çerçevelik
4  0.259944   0.221228           0.192467    0.9850  0.6752     0.8338         1.7413       0.7557  Çerçevelik
```

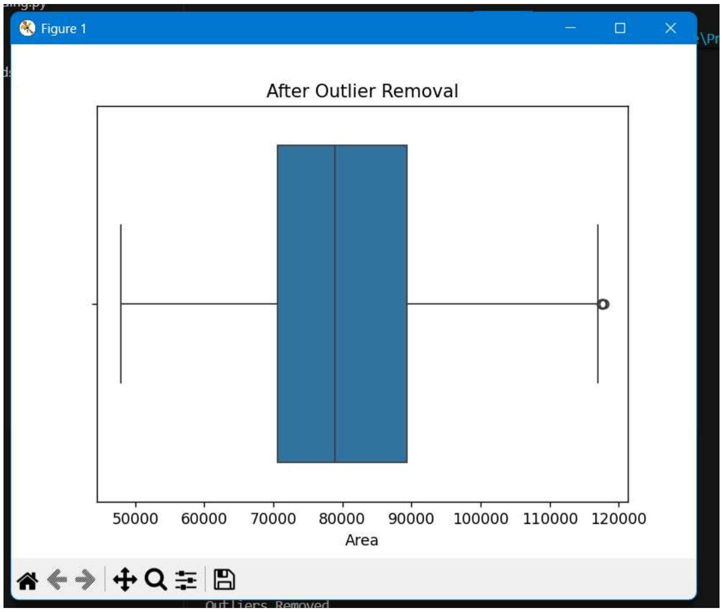
3. Exploratory Data Analysis (EDA)

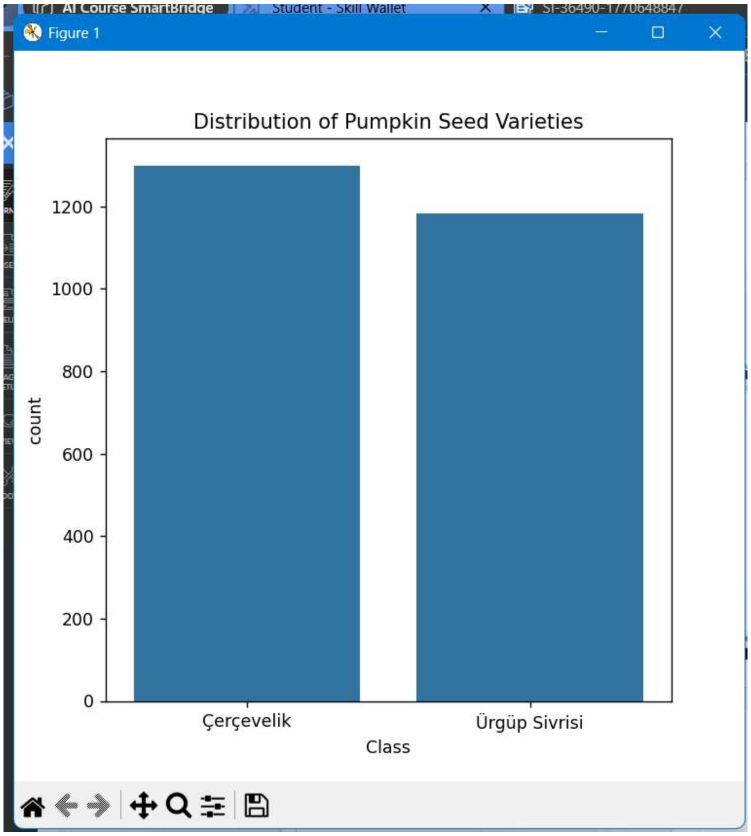
We analyzed the data to understand patterns and separability.

- **Descriptive Statistics:** Generated a statistical summary (Mean, Std, Min, Max) using df.describe() to understand feature distributions.
- **Univariate Analysis:**
  - **Class Balance:** Visualized the target variable using a Countplot, confirming a balanced dataset.
  - **Outliers:** Used Boxplots to visualize the spread of 'Area' before and after cleaning.
- **Bivariate Analysis:** Plotted **Area vs. Perimeter** (Scatter Plot) to observe the positive correlation between seed size and boundary length.
- **Multivariate Analysis:** Created a **Correlation Heatmap** to identify relationships between all features and the target variable

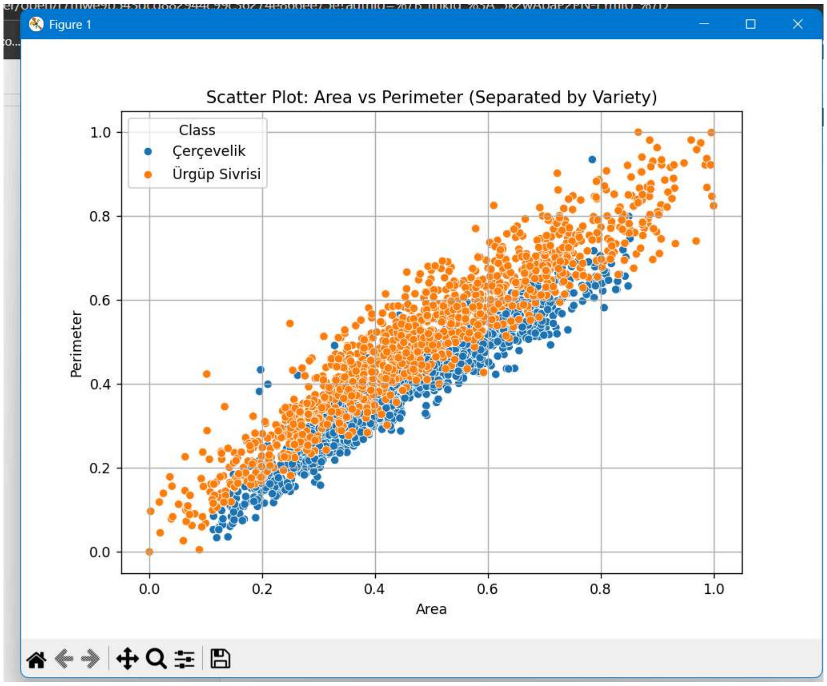


```
Class              0
```

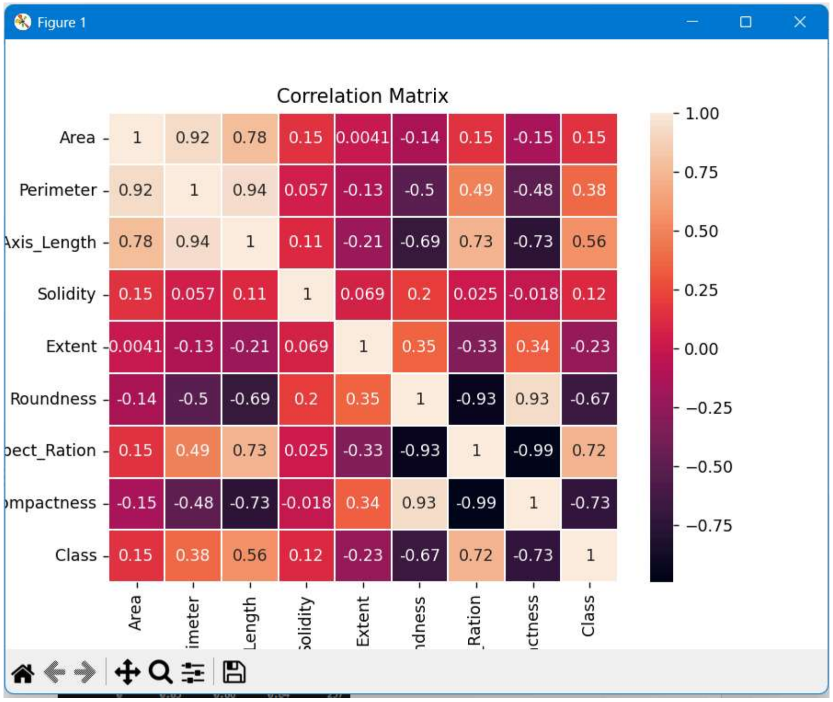




Distribution of Pumpkin Seed Varieties

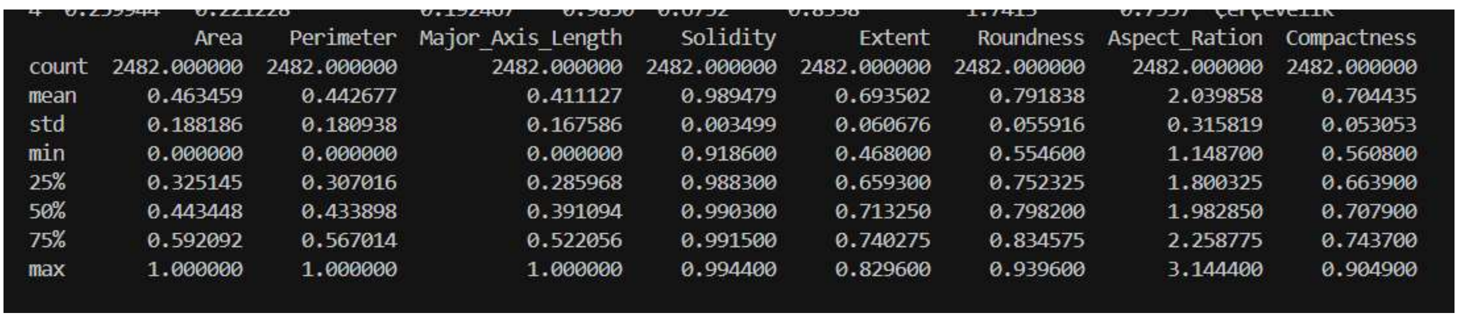


Scatter Plot: Area vs Perimeter (Separated by Variety)



Correlation Matrix

	Area	imeter	Length	olidity	Extent	dness	Ration	ctness	Class
Area	1	0.92	0.78	0.15	0.0041	-0.14	0.15	-0.15	0.15
Perimeter	0.92	1	0.94	0.057	-0.13	-0.5	0.49	-0.48	0.38
Axis_Length	0.78	0.94	1	0.11	-0.21	-0.69	0.73	-0.73	0.56
Solidity	0.15	0.057	0.11	1	0.069	0.2	0.025	-0.018	0.12
Extent	0.0041	-0.13	-0.21	0.069	1	0.35	-0.33	0.34	-0.23
Roundness	-0.14	-0.5	-0.69	0.2	0.35	1	-0.93	0.93	-0.67
pect_Ration	0.15	0.49	0.73	0.025	-0.33	-0.93	1	-0.99	0.72
mpactness	-0.15	-0.48	-0.73	-0.018	0.34	0.93	-0.99	1	-0.73
Class	0.15	0.38	0.56	0.12	-0.23	-0.67	0.72	-0.73	1

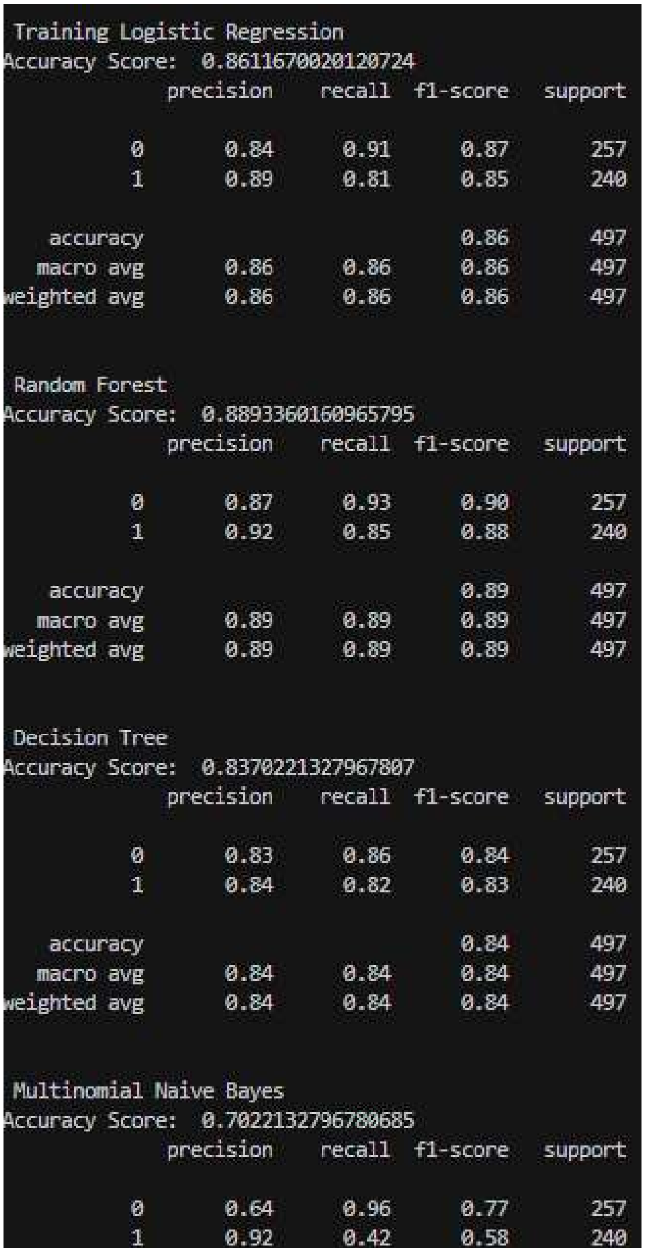


```
4  0.239944  0.221228           0.192407  0.9850  0.6752  0.8338        1.7415  0.7337  Çerçevelik
              Area    Perimeter  Major_Axis_Length     Solidity       Extent    Roundness  Aspect_Ration  Compactness
count  2482.000000  2482.000000        2482.000000  2482.000000  2482.000000  2482.000000    2482.000000  2482.000000
mean      0.463459     0.442677           0.411127     0.989479     0.693502     0.791838       2.039858     0.704435
std       0.188186     0.180938           0.167586     0.003499     0.060676     0.055916       0.315819     0.053053
min       0.000000     0.000000           0.000000     0.918600     0.468000     0.554600       1.148700     0.560800
25%       0.325145     0.307016           0.285968     0.988300     0.659300     0.752325       1.800325     0.663900
50%       0.443448     0.433898           0.391094     0.990300     0.713250     0.798200       1.982850     0.707900
75%       0.592092     0.567014           0.522056     0.991500     0.740275     0.834575       2.258775     0.743700
max       1.000000     1.000000           1.000000     0.994400     0.829600     0.939600       3.144400     0.904900
```

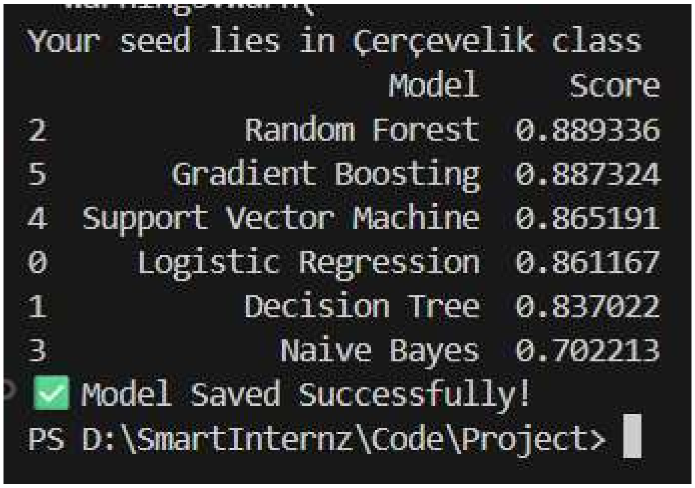
4. Model Building

We trained multiple supervised learning algorithms to find the best classifier.

- **Data Splitting:** The dataset was split into **Training (80%)** and **Testing (20%)** sets using train_test_split with random_state=30.
- **Models Trained:**
    a. **Logistic Regression:** A baseline linear classifier.
    b. **Decision Tree:** A non-linear tree-based model.
    c. **Random Forest:** An ensemble of decision trees (Bagging).
    d. **Naive Bayes:** Probabilistic classifier.
    e. **Support Vector Machine (SVM):** A margin-based classifier.
    f. **Gradient Boosting:** An ensemble boosting technique.



```
 Training Logistic Regression
Accuracy Score:  0.8611670020120724
              precision    recall  f1-score   support

           0       0.84      0.91      0.87       257
           1       0.89      0.81      0.85       240

    accuracy                           0.86       497
   macro avg       0.86      0.86      0.86       497
weighted avg       0.86      0.86      0.86       497


 Random Forest
Accuracy Score:  0.8893360160965795
              precision    recall  f1-score   support

           0       0.87      0.93      0.90       257
           1       0.92      0.85      0.88       240

    accuracy                           0.89       497
   macro avg       0.89      0.89      0.89       497
weighted avg       0.89      0.89      0.89       497


 Decision Tree
Accuracy Score:  0.8370221327967807
              precision    recall  f1-score   support

           0       0.83      0.86      0.84       257
           1       0.84      0.82      0.83       240

    accuracy                           0.84       497
   macro avg       0.84      0.84      0.84       497
weighted avg       0.84      0.84      0.84       497


 Multinomial Naive Bayes
Accuracy Score:  0.7022132796780685
              precision    recall  f1-score   support

           0       0.64      0.96      0.77       257
           1       0.92      0.42      0.58       240
```



```
Your seed lies in Çerçevelik class
                    Model     Score
2           Random Forest  0.889336
5       Gradient Boosting  0.887324
4  Support Vector Machine  0.865191
0     Logistic Regression  0.861167
1           Decision Tree  0.837022
3             Naive Bayes  0.702213
✅ Model Saved Successfully!
PS D:\SmartInternz\Code\Project>
```

5. Performance Testing & Tuning

We evaluated models to select the most accurate one for deployment.

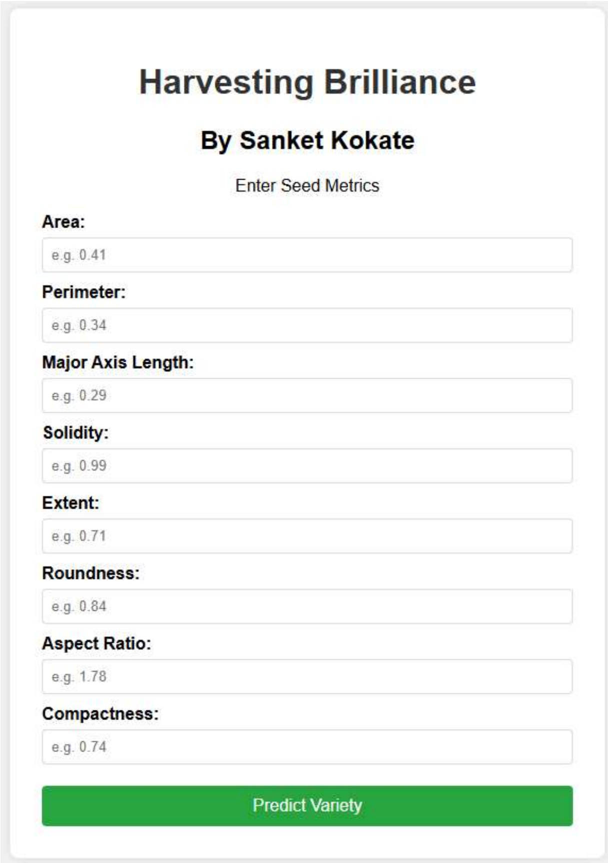
- **Evaluation Metrics:** Used **Accuracy Score** and **Classification Report** (Precision, Recall, F1-Score) for each model.
- **Comparison Results:**
  - Gradient Boosting: **0.885**
  - Random Forest: **0.881**
  - SVM: **0.865**
  - Logistic Regression: **0.861**
  - Naive Bayes: **0.702**
- **Selection:Random Forest** and **Gradient Boosting** were the top performers. Random Forest was selected for deployment due to its robustness.
- **Manual Testing:** Performed a manual prediction test with sample data [0.41, 0.34, ...] which correctly predicted "Çerçevelik".

6.Model Deployment

The final phase involved making the model accessible via a web interface.

- **Serialization:** The trained Random Forest model was saved as a serialized file, **model.pkl**, using the pickle library.
- **Web Framework (Flask):** Created **app.py** to handle web requests. It loads model.pkl and defines a /predict route.
- **Frontend (HTML):**
  - **index.html:** A user-friendly form to accept 8 morphological inputs (Area, Perimeter, Solidity, etc.).
  - **predict.html:** A results page to display the predicted pumpkin seed variety.
- **Integration:** The system successfully takes user input from the browser, processes it through the Python backend, and returns the classification result in real-time.

Project Output :



Harvesting Brilliance

By Sanket Kokate

Enter Seed Metrics

Area: e.g. 0.41

Perimeter: e.g. 0.34

Major Axis Length: e.g. 0.29

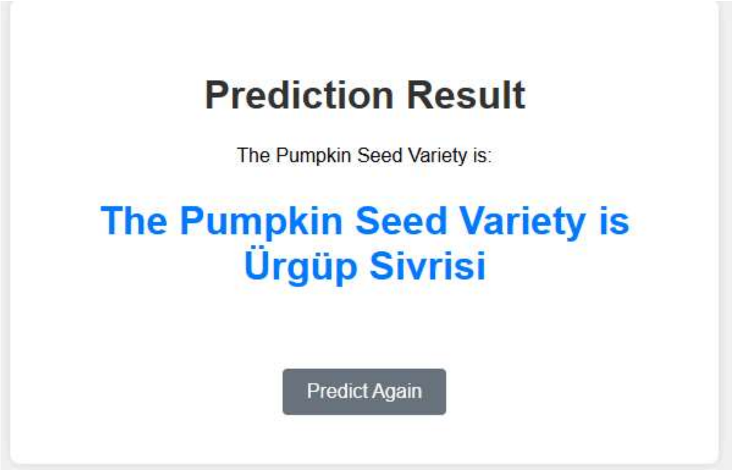
Solidity: e.g. 0.99

Extent: e.g. 0.71

Roundness: e.g. 0.84

Aspect Ratio: e.g. 1.78

Compactness: e.g. 0.74



Prediction Result

The Pumpkin Seed Variety is:

The Pumpkin Seed Variety is Ürgüp Sivrisi