Term End Milestone-2 (Project -1)

IDENTIFICATION OF RICE VARIETIES

Draft White Paper

DSC, Bellevue University

Prashant Raghuwanshi

DSC680-T301 Applied Data Science (2225-1)

Professor Catie Williams

04/02/2022

Identification of Rice Varieties Using Artificial Intelligence Methods

Towards a real-time rice gains sorting system. Identification of vitreous durum rice kernels using ANN based on their morphological, color, wavelet, and gaborlet features.

Business Problem :

The modern "Ready to Eat" Food Processing industry is importing grains (rice) from various international grains distributors. Their producted packed foods mostly depend on quality and varieties of imported raw gains. For them, it is important to keep the gains quality check on high priority.

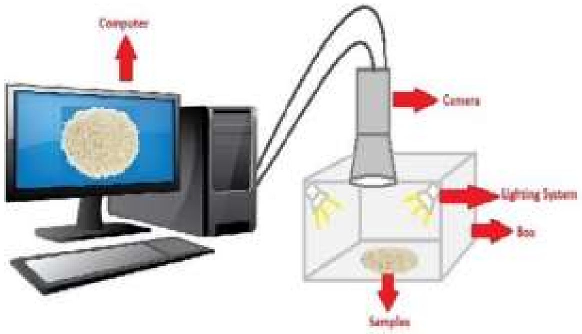
At present the quality check process completely depends on the manual steps performed by experienced grain experts and most of the time manual process is causing a delay in executing production lines.

Moreover, the manual quality check process result is not consistent and not quick, which required grain experts to be present on-site and near the production line.

The processing industry is looking for an Artificial intelligence-based gain sampling instrument that can quickly identify the gain quality and variety with maximum accuracy.

Background :

To solve the above-mentioned critical Business problem of "ready to eat" food processing industries. This project is going to develop the ML Model software which is going to be installed on a grain scanner machine and it will help the machine to identify the variety and quality of passed rice grain samples.



The images of rice samples were obtained first and images were processed using various image processing techniques. The resulting images are first converted to a grayscale image, then converted to a binary image and removed from the noise on the image. In the next phase, various morphological feature inference processes were applied to the obtained images. The classification phase of rice is given in Figure 1. During the modeling phase, the rice classification process was carried out using seven pieces of machine learning techniques. In the last step, the performance of the models used was evaluated.

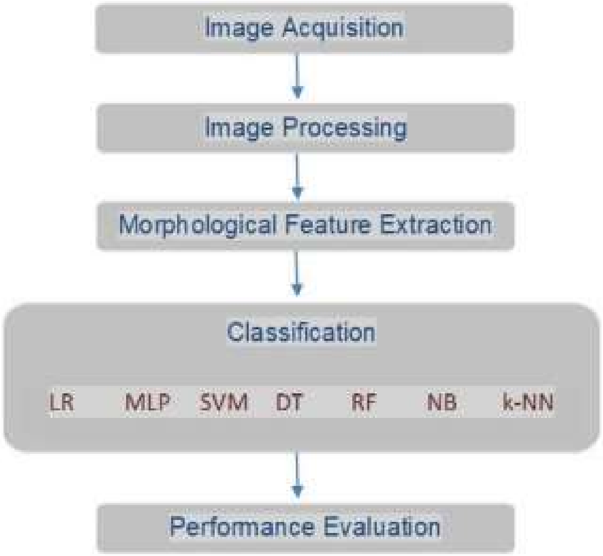
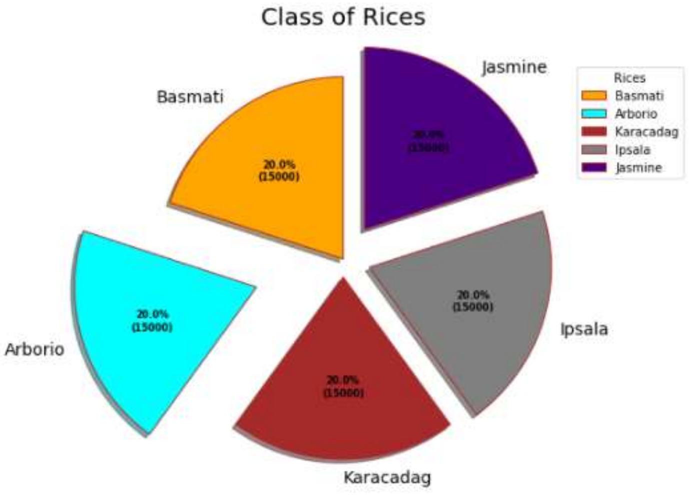


Fig. 1. Rice classification phases

Data Explanation:

A total of 75 thousand pieces of rice grain were obtained, including 15 thousand pieces of each variety of rice (Arborio, Basmati, Uppsala, Jasmine, Karacadag). Preprocessing operations were applied to the images and made available for feature extraction. A total of 106 features were inferred from the images; 12 morphological features and 4 shape features were obtained using morphological features and 90 color features were obtained from five different color spaces (RGB, HSV, Lab*, YCbCr, XYZ).



Attribute Information:

1.) Area: Returns the number of pixels within the boundaries of the rice grain.

2.) Perimeter: Calculates the circumference by calculating the distance between pixels around the boundaries of the rice grain.

3.) Major Axis Length: The longest line that can be drawn on the rice grain, i.e. the main axis distance, gives.

4.) Minor Axis Length: The shortest line that can be drawn on the rice grain, i.e. the small axis distance, gives.

5.) Eccentricity: It measures how round the ellipse, which has the same moments as the rice grain, is.

6.) Convex Area: Returns the pixel count of the smallest convex shell of the region formed by the rice grain.

7.) Extent: Returns the ratio of the region formed by the rice grain to the bounding box pixels.

8.) Class: Cammeo and Osmancik rice

Methods:

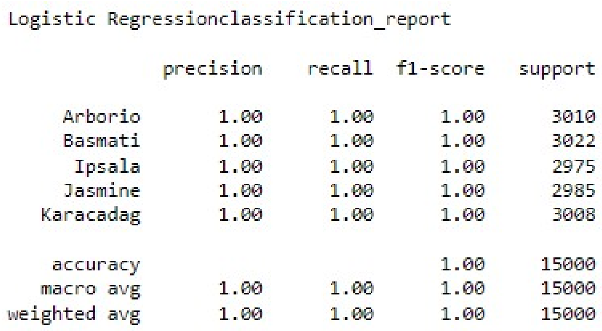
In this project, models were created using LR (Logistic Regression), MLP (Multilayer Perceptron), ABC (AdaBoostClassifier), DT (Decision Tree), RF (Random Forest), GB (Gradient Boosting), and k NN (K Nearest Neighbor) systems to classify rice grains according to their characteristics. (please refer to the code snippet in the appendix section)

Analysis :

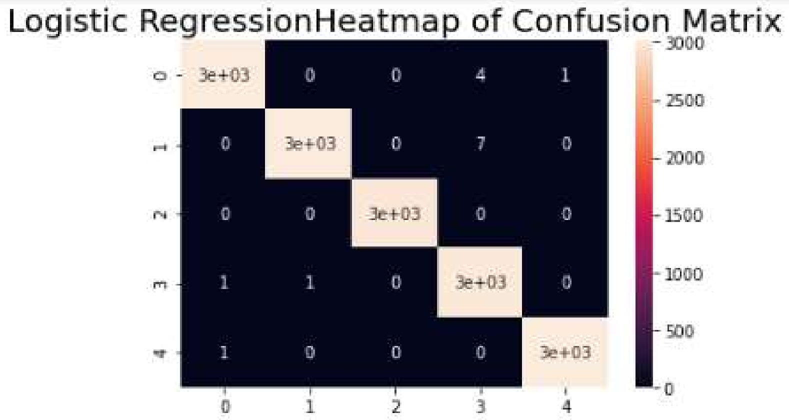
Classification performance measurement results are pasted below. all models except AdaBoostClassifier have been achieved a classification success of over 99%. The 99.93% accuracy achieved in the LR model has the highest value among other models. Also for F1, recall and precision look good for the LR model.

Model Accuracy and F1 scores:

```
Logistic Regression Accuracy: 99.90%  F1-Score: 0.99900
      Decision Tree Accuracy: 99.57%  F1-Score: 0.99567
     Neural Network Accuracy: 99.93%  F1-Score: 0.99927
      Random Forest Accuracy: 99.86%  F1-Score: 0.99860
  Gradient Boosting Accuracy: 99.83%  F1-Score: 0.99833
 AdaBoostClassifier Accuracy: 60.71%  F1-Score: 0.48542
KNeighborsClassifier Accuracy: 99.83%  F1-Score: 0.99827
```



```
Logistic Regressionclassification_report

              precision    recall  f1-score   support

     Arborio       1.00      1.00      1.00      3010
     Basmati       1.00      1.00      1.00      3022
      Ipsala       1.00      1.00      1.00      2975
     Jasmine       1.00      1.00      1.00      2985
   Karacadag       1.00      1.00      1.00      3008

    accuracy                           1.00     15000
   macro avg       1.00      1.00      1.00     15000
weighted avg       1.00      1.00      1.00     15000
```



Logistic RegressionHeatmap of Confusion Matrix

Conclusion:

The logistic regression model is the best fit for rice gains predictions. We can use this classification model to develop Automatic systems that can be designed for many processes such as calibration of rice types and the separation of species from unwanted substances that may be present.

Assumption: Here I am assuming the Preprocessing data extraction operations is not having any limitations and the rice grains images were extracted & preprocessed successfully by image to data conversion tool.

Limitations: This Model is limited to identifying the Rice grains only. This model is suitable to identify Arborio, Basmati, Uppsala, Jasmine, and Karacadag rice varieties and would consider other varieties as unknown gain.

Challenges:

Major Constraints are related to used datasets and processed images, here the used datasets contain a total of 75 thousand rice grain images, including 15 thousand for each variety however due to the rapidly advancing Seed development process, we might not have all full collections of grain records under each gain varieties

Future Uses:

1) By adding additional grains datasets, we can increase the scope of this model to identify large varieties of grains.
2) Programmed in robot scanner machines and enable robots to filter adulterated gains

Recommendations:

Automatic systems can be designed for many processes such as calibration of rice types and the separation of species from unwanted substances that may be present. To increase the success rate in classification, more images can be obtained from species and it is thought that success rates can be increased by using morphological features as well as color and shape features.

Implementation Plan:

To develop ML applications, we have below implementation plan:

IDE's to use: Anaconda Jupyter

Coding languages: Python

Platform: AWS and local PC

Resource Data Science Engineer

Timeline: 03/15/2022 to 4/16/2022

Ethical Considerations:

- This Data contains processed physical images information related to multiple verities of rice and does not contains any PII-related information.
- Datasets and information on data were extracted from the public websites → UCL machine learning repositories.
- This data research is not going to harm any privacy.

References :

https://archive.ics.uci.edu/ml/datasets/Rice+%28Cammeo+and+Osmancik%29

https://www.muratkoklu.com/datasets/

1: KOKLU, M., CINAR, I. and TASPINAR, Y. S. (2021). Classification of rice varieties with deep learning methods. Computers and Electronics in Agriculture, 187, 106285.

DOI: https://doi.org/10.1016/j.compag.2021.106285

2: CINAR, I. and KOKLU, M. (2021). Determination of Effective and Specific Physical Features of Rice Varieties by Computer Vision In Exterior Quality Inspection. Selcuk Journal of Agriculture and Food Sciences, 35(3), 229-243.

DOI: https://doi.org/10.15316/SJAFS.2021.252

3: CINAR, I. and KOKLU, M. (2022). Identification of Rice Varieties Using Machine Learning Algorithms. Journal of Agricultural Sciences.

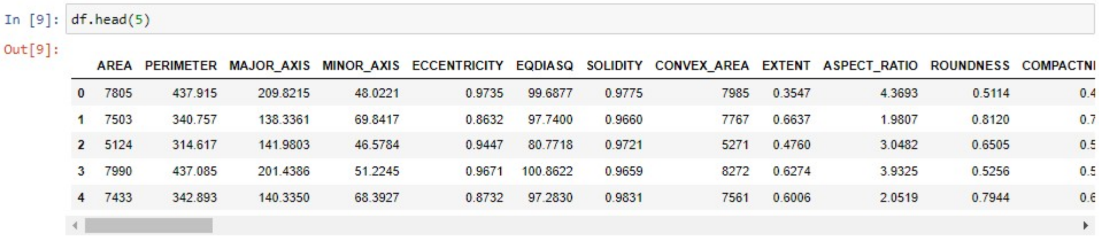
DOI: https://doi.org/10.15832/ankutbd.862482

4: CINAR, I. and KOKLU, M. (2019). Classification of Rice Varieties Using Artificial Intelligence Methods. International Journal of Intelligent Systems and Applications in Engineering, 7(3), 188-194.

DOI: https://doi.org/10.18201/ijisae.2019355381

Appendix :

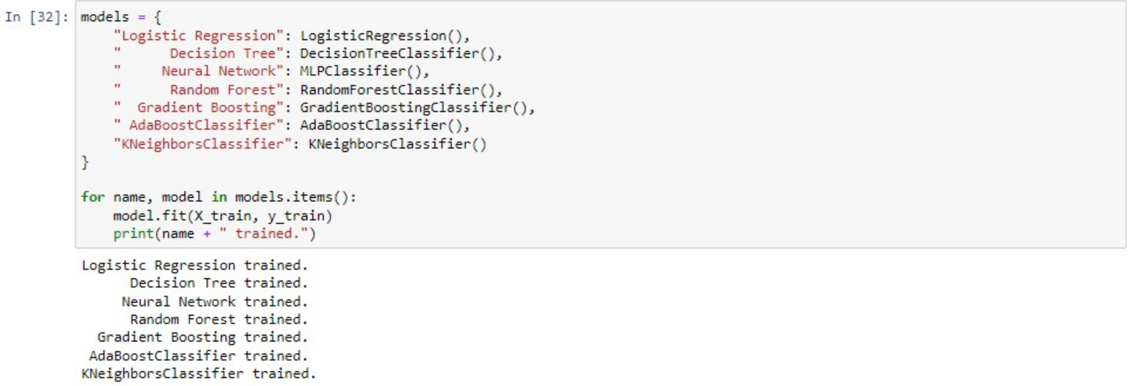
Data set Screenprint :



```
In [9]: df.head(5)
Out[9]:
```

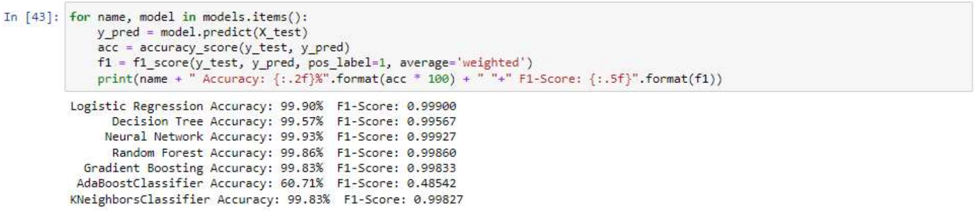
	AREA	PERIMETER	MAJOR_AXIS	MINOR_AXIS	ECCENTRICITY	EQDIASQ	SOLIDITY	CONVEX_AREA	EXTENT	ASPECT_RATIO	ROUNDNESS	COMPACTNI
0	7805	437.915	209.8215	48.0221	0.9735	99.6877	0.9775	7985	0.3547	4.3693	0.5114	0.4
1	7503	340.757	138.3361	69.8417	0.8632	97.7400	0.9660	7767	0.6637	1.9807	0.8120	0.7
2	5124	314.617	141.9803	46.5784	0.9447	80.7718	0.9721	5271	0.4760	3.0482	0.6505	0.5
3	7990	437.085	201.4386	51.2245	0.9671	100.8622	0.9659	8272	0.6274	3.9325	0.5256	0.5
4	7433	342.893	140.3350	68.3927	0.8732	97.2830	0.9831	7561	0.6006	2.0519	0.7944	0.6

Model build screenprint:

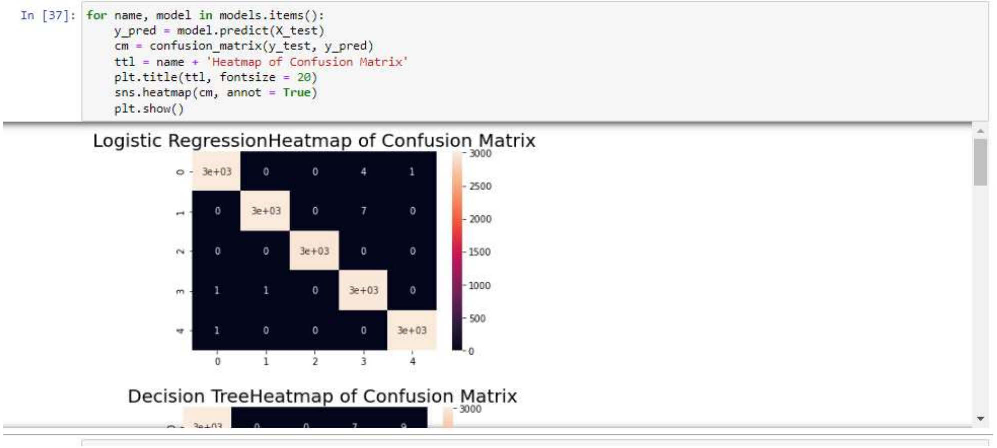


```
In [32]: models = {
             "Logistic Regression": LogisticRegression(),
             "      Decision Tree": DecisionTreeClassifier(),
             "     Neural Network": MLPClassifier(),
             "      Random Forest": RandomForestClassifier(),
             "  Gradient Boosting": GradientBoostingClassifier(),
             " AdaBoostClassifier": AdaBoostClassifier(),
             "KNeighborsClassifier": KNeighborsClassifier()
         }

         for name, model in models.items():
             model.fit(X_train, y_train)
             print(name + " trained.")

         Logistic Regression trained.
               Decision Tree trained.
              Neural Network trained.
               Random Forest trained.
           Gradient Boosting trained.
          AdaBoostClassifier trained.
         KNeighborsClassifier trained.
```

Accuracy computation :



```
In [43]: for name, model in models.items():
             y_pred = model.predict(X_test)
             acc = accuracy_score(y_test, y_pred)
             f1 = f1_score(y_test, y_pred, pos_label=1, average='weighted')
             print(name + " Accuracy: {:.2f}%".format(acc * 100) + " "+" F1-Score: {:.5f}".format(f1))

         Logistic Regression Accuracy: 99.90%  F1-Score: 0.99900
               Decision Tree Accuracy: 99.57%  F1-Score: 0.99567
              Neural Network Accuracy: 99.93%  F1-Score: 0.99927
               Random Forest Accuracy: 99.86%  F1-Score: 0.99860
           Gradient Boosting Accuracy: 99.83%  F1-Score: 0.99833
          AdaBoostClassifier Accuracy: 60.71%  F1-Score: 0.48542
         KNeighborsClassifier Accuracy: 99.83%  F1-Score: 0.99827
```



```
In [37]: for name, model in models.items():
             y_pred = model.predict(X_test)
             cm = confusion_matrix(y_test, y_pred)
             ttl = name + 'Heatmap of Confusion Matrix'
             plt.title(ttl, fontsize = 20)
             sns.heatmap(cm, annot = True)
             plt.show()
```

Logistic RegressionHeatmap of Confusion Matrix
3e+03
0
0
4
1
0
3e+03
0
7
0
0
0
3e+03
0
0
1
1
0
3e+03
0
1
0
0
0
3e+03
0
1
2
3
4
3000
2500
2000
1500
1000
500
0
Decision TreeHeatmap of Confusion Matrix
3000