PISTACHIO CLASSIFICATION

Project By – Manish Sri Sai Surya Routhu (5496)

Document:

Pistachio Classification Using Machine Learning Models and Convolutional Neural Networks

Problem Statement:

Pistachio is a shelled fruit from the anacardiaceous family. The homeland of pistachio is the Middle East. The Kirmizi pistachios and Siirt pistachios are the major types grown and exported in Turkey. Since the prices, tastes, and nutritional values of these types differs, the type of pistachio becomes important when it comes to trade.
So, the goal is to identify these two types of pistachios, which are frequently grown in Turkey, by classifying them via convolutional neural networks.

Data collection:

Kirmizi and Siirt pistachio types were obtained through the computer vision system. The pre-trained dataset includes a total of 2148 images, 1232 of Kirmizi type and 916 of Siirt type.

INTRODUCTION

Pistachio classification is an important task in the trade of pistachios, as different types of pistachios have varying prices, tastes, and nutritional values. This document outlines the approach to identify and classify two major types of pistachios, namely Kirmizi and Siirt, using convolutional neural networks (CNNs). The dataset consists of 2148 images, with 1232 images of Kirmizi type and 916 images of Siirt type.

The dataset of pistachio images was obtained through a computer vision system. The images were categorized into two classes: Kirmizi and Siirt. The dataset contains a sufficient number of images to train and evaluate the performance of the CNN models.

IMPLEMENTATION

About the Data: The data consists of 2148 coloured images that means the shape of the data is (x, y, 3) where 3 represents 3 channels (R, G, B) and x and y are integers representing height and weight of the image.

So, as we have so many images there is a possibility of irregular dimensions and shapes. But it is observed that in all the objects (pistachios) are in the centre of the images which gives an option of cropping the images from all the sides.

Two datasets are taken for classification purpose –

1. Original dataset
2. Cropped dataset (13% cropped above and below, 28% cropped left and right) This is from the observation

Initially, the Implementation is on the original dataset

The Implementation of the project consists of following parts:

1. Preprocessing
2. Working with Machine Learning Models
3. Working with Fully Connected Artificial Neural Network
4. Working with Convolutional Neural Network
5. Working with Xception Model

1. Preprocessing:

Preprocessing consists of various operations

i. Convert the image to array format so that it can be used for further processing

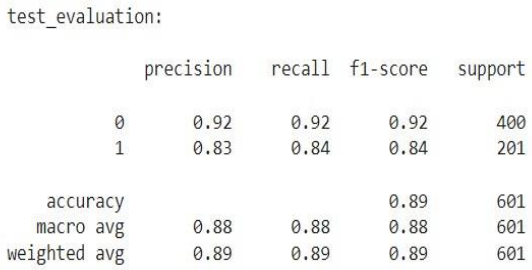
ii. Resize the image to a constant height and width so that all the images will be of uniform size.

iii. Normalise the data from [0, 255] -> [0,1] – It might impact the model because the weights in the neural networks are initialized small.

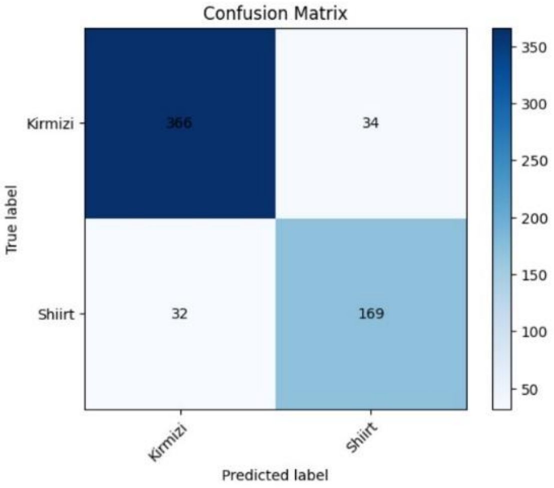
iv. A. flatten the data for the implementation of Machine Learning Models
B. Store the 3 dimensional data for the CNN and other neural network models.

2. Working with Machine Learning Models:

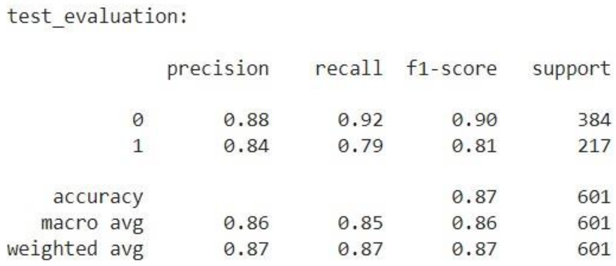
I. Logistic Regression:



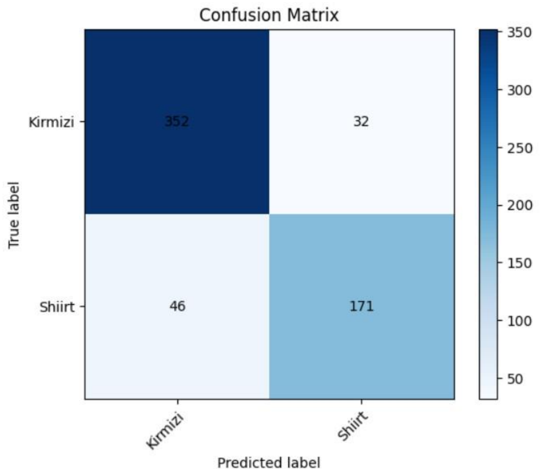
```
test_evaluation:

              precision    recall  f1-score   support

           0       0.92      0.92      0.92       400
           1       0.83      0.84      0.84       201

    accuracy                           0.89       601
   macro avg       0.88      0.88      0.88       601
weighted avg       0.89      0.89      0.89       601
```



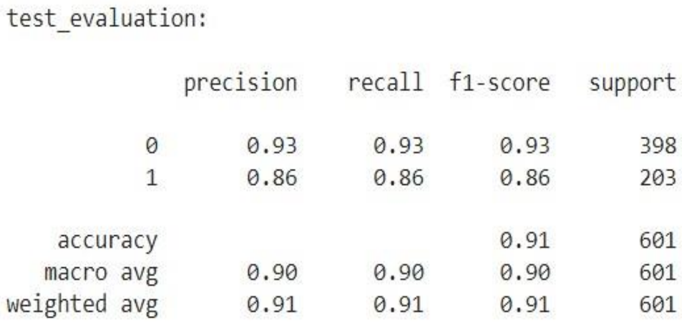
II. Decision Tree Classifier:



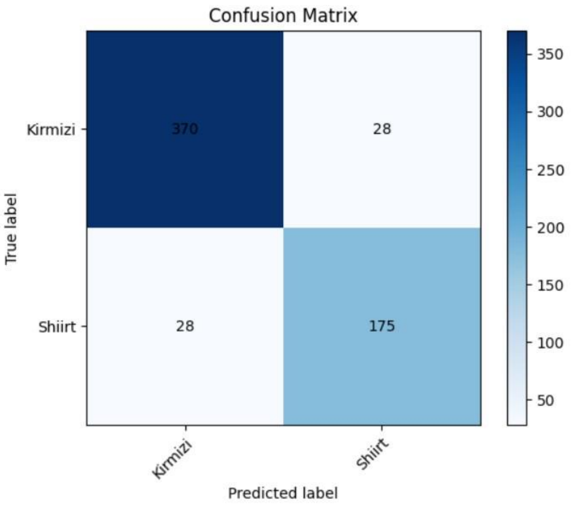
```
test_evaluation:

              precision    recall  f1-score   support

           0       0.88      0.92      0.90       384
           1       0.84      0.79      0.81       217

    accuracy                           0.87       601
   macro avg       0.86      0.85      0.86       601
weighted avg       0.87      0.87      0.87       601
```



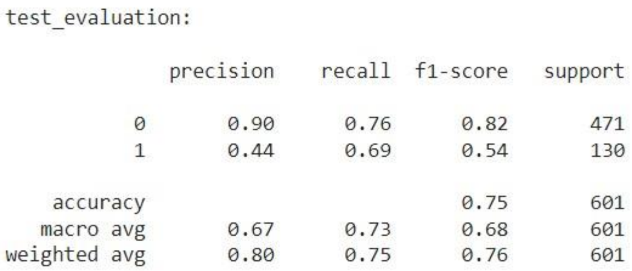
III. Random Forest Classifier:



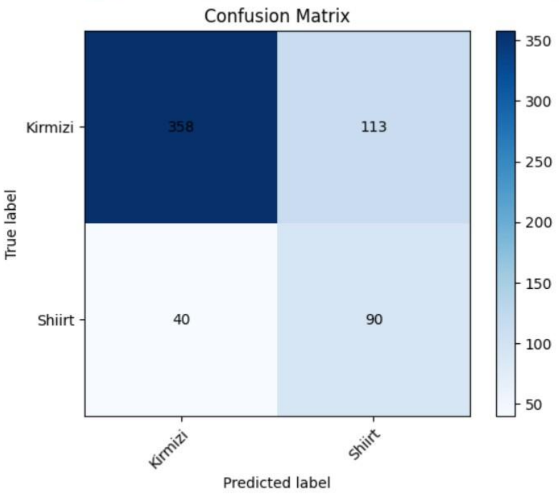
```
test_evaluation:

              precision    recall  f1-score   support

           0       0.93      0.93      0.93       398
           1       0.86      0.86      0.86       203

    accuracy                           0.91       601
   macro avg       0.90      0.90      0.90       601
weighted avg       0.91      0.91      0.91       601
```



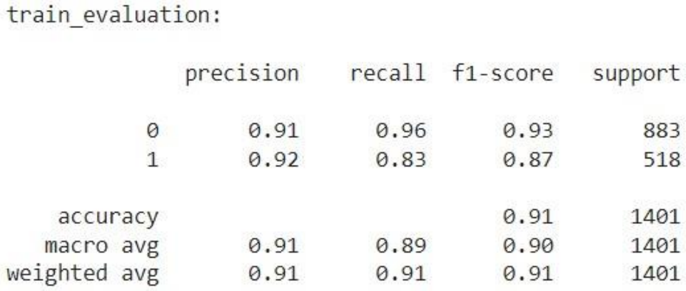
IV. Naïve Bayes Classifier:



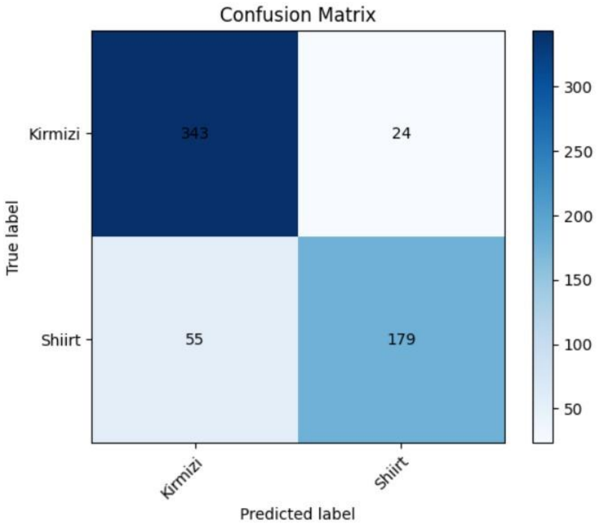
```
test_evaluation:

              precision    recall  f1-score   support

           0       0.90      0.76      0.82       471
           1       0.44      0.69      0.54       130

    accuracy                           0.75       601
   macro avg       0.67      0.73      0.68       601
weighted avg       0.80      0.75      0.76       601
```



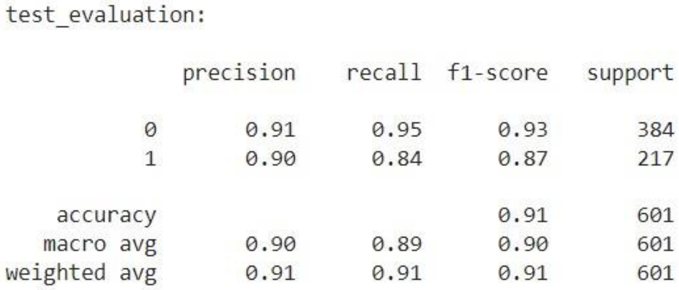
V. K Nearest Neighbours Classifier:



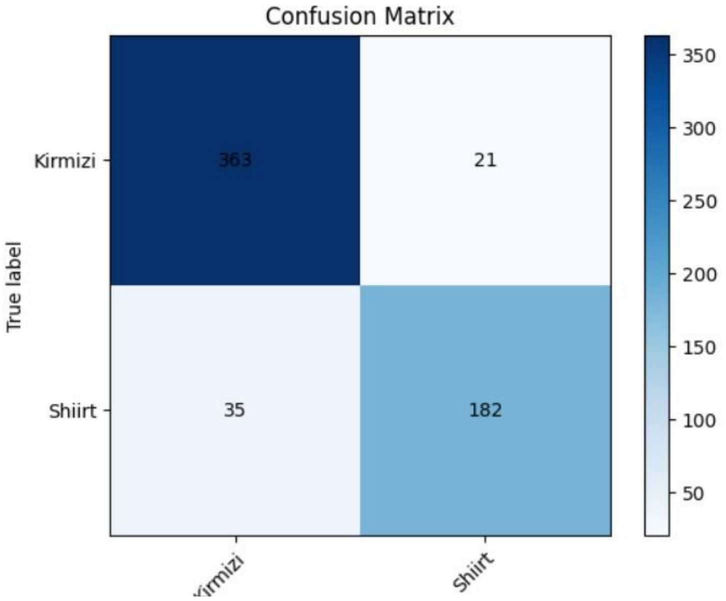
```
train_evaluation:

              precision    recall  f1-score   support

           0       0.91      0.96      0.93       883
           1       0.92      0.83      0.87       518

    accuracy                           0.91      1401
   macro avg       0.91      0.89      0.90      1401
weighted avg       0.91      0.91      0.91      1401
```



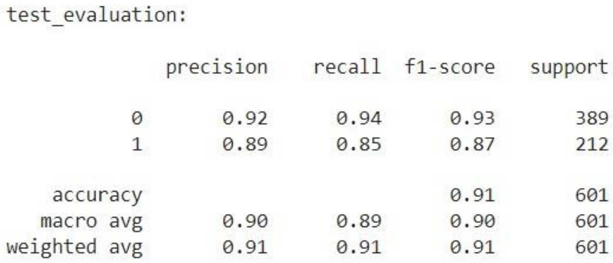
VI. Adaptive boosting Classifier:



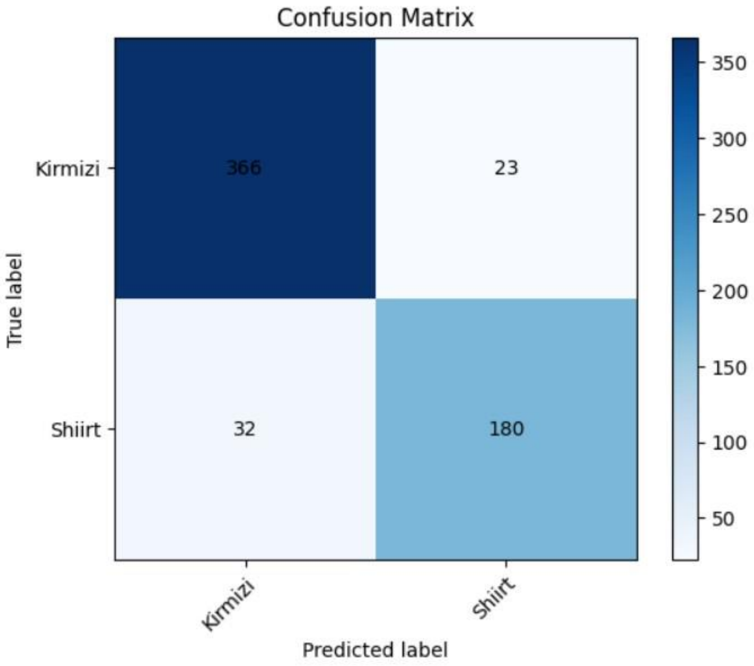
```
test_evaluation:

              precision    recall  f1-score   support

           0       0.91      0.95      0.93       384
           1       0.90      0.84      0.87       217

    accuracy                           0.91       601
   macro avg       0.90      0.89      0.90       601
weighted avg       0.91      0.91      0.91       601
```



VII. Support Vector Classifier:

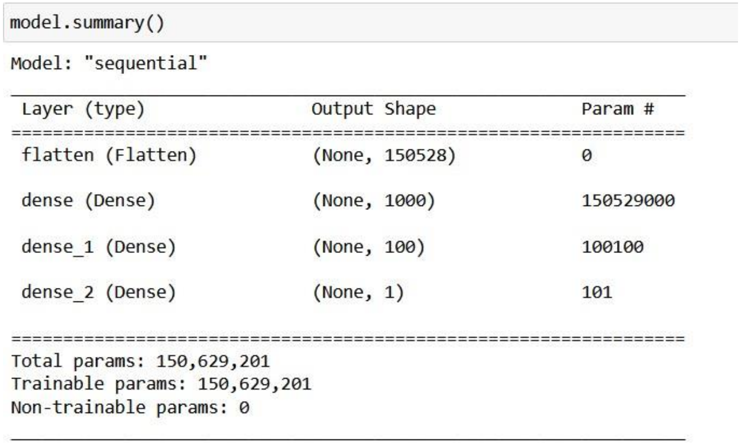


```
test_evaluation:

              precision    recall  f1-score   support

           0       0.92      0.94      0.93       389
           1       0.89      0.85      0.87       212

    accuracy                           0.91       601
   macro avg       0.90      0.89      0.90       601
weighted avg       0.91      0.91      0.91       601
```

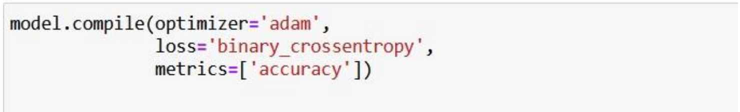


After Experimenting with Machine Learning Models it was found that Machine learning and SVM are giving good accuracy of 91 percent each.

3. Working with Fully Connected Artificial Neural Network:



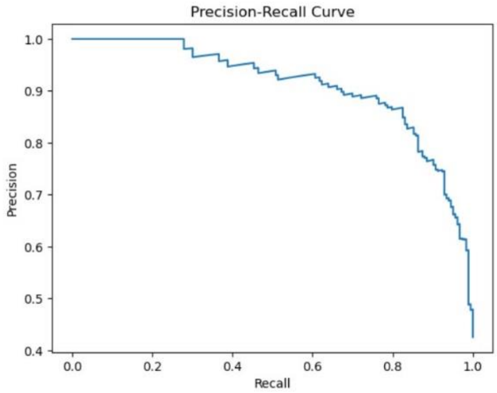
```
model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten (Flatten)           (None, 150528)            0

 dense (Dense)               (None, 1000)              150529000

 dense_1 (Dense)             (None, 100)               100100

 dense_2 (Dense)             (None, 1)                 101

=================================================================
Total params: 150,629,201
Trainable params: 150,629,201
Non-trainable params: 0
_________________________________________________________________
```

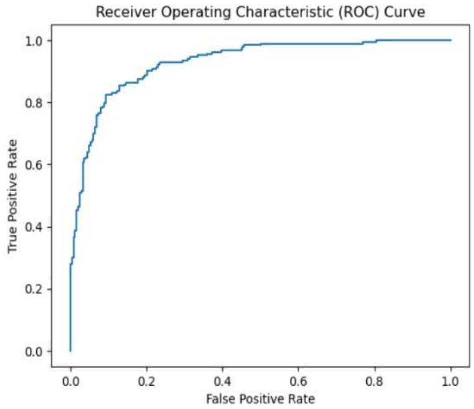


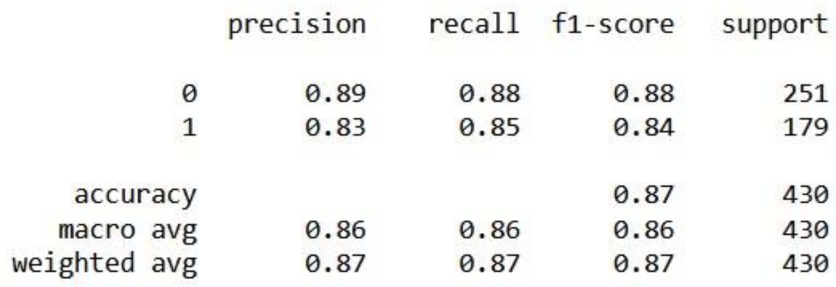
```
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])
```

It is a fully connected Neural Network and the worked pretty better compared to the Convolutional Neural Network which we will be looking next the evaluation metrics are as below:

```
Accuracy score: 0.8651162790697674
Recall score: 0.8491620111731844
```







```
              precision    recall  f1-score   support

           0       0.89      0.88      0.88       251
           1       0.83      0.85      0.84       179

    accuracy                           0.87       430
   macro avg       0.86      0.86      0.86       430
weighted avg       0.87      0.87      0.87       430
```

Confusion matrix:

	Kirmizi	Siirt
Kirmizi	220	31
Siirt	27	152

4. Working with Convolutional Neural Network:

Model Development: Convolutional neural networks (CNNs) are well-suited for image classification tasks due to their ability to capture spatial hierarchies and local patterns. The following steps outline the process of building a CNN model for pistachio classification:

a. Architecture Design: The architecture typically consists of convolutional layers, pooling layers, and fully connected layers.

b. Model Compilation: Define the loss function, optimization algorithm, and evaluation metrics (e.g., accuracy) for the CNN model.

c. Model Training: Split the dataset into training and validation sets. Train the CNN model using the training set and validate its performance on the validation set. Adjust hyperparameters and network architecture as needed.

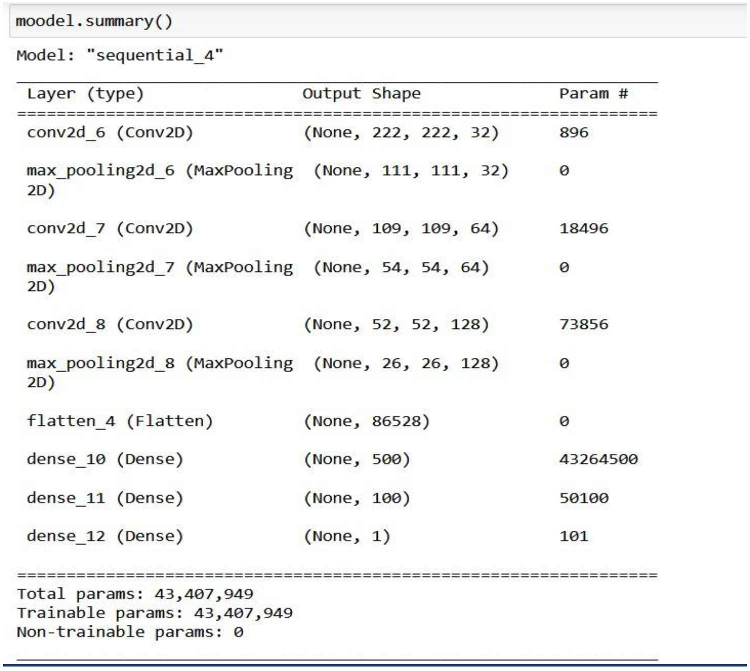
d. Model Evaluation: Evaluation of the training model on a separate test set to assess its performance. Calculation of metrics such as accuracy, precision, recall, and F1-score [classification report]. Additionally, generate a confusion matrix to visualize the classification results. ROC curve and PRC curve are also shown

I. Confusion Matrix: A table that summarizes the classification results by displaying the number of true positives, true negatives, false positives, and false negatives.

II. Accuracy: The ratio of correctly classified samples to the total number of samples.

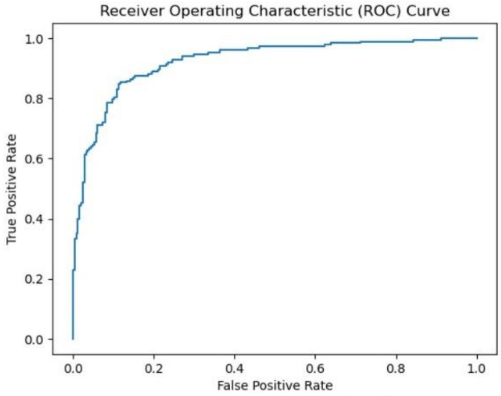
III. Recall (Sensitivity): The ratio of true positives to the sum of true positives and false negatives. It measures the model's ability to correctly identify positive samples.

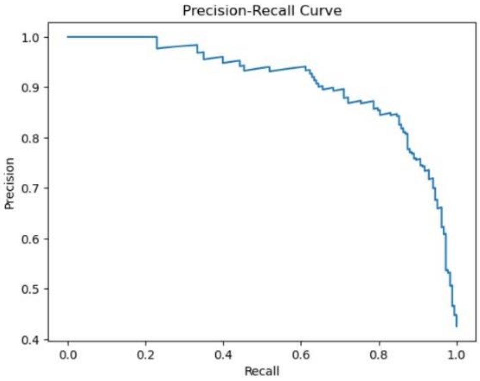
IV. Error Curves: Plotting the training and validation error (loss) over epochs helps in analysing model convergence and potential overfitting.

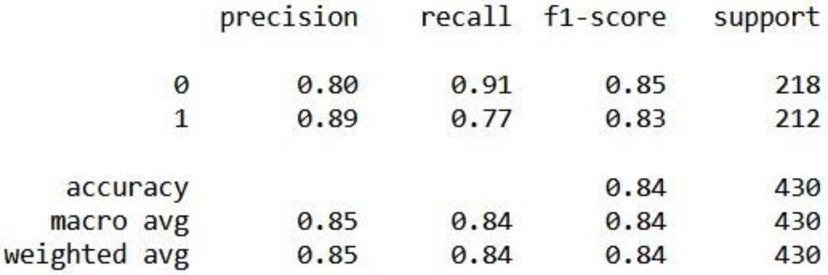


```
moodel.summary()

Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_6 (Conv2D)           (None, 222, 222, 32)      896

 max_pooling2d_6 (MaxPooling  (None, 111, 111, 32)     0
 2D)

 conv2d_7 (Conv2D)           (None, 109, 109, 64)      18496

 max_pooling2d_7 (MaxPooling  (None, 54, 54, 64)       0
 2D)

 conv2d_8 (Conv2D)           (None, 52, 52, 128)       73856

 max_pooling2d_8 (MaxPooling  (None, 26, 26, 128)      0
 2D)

 flatten_4 (Flatten)         (None, 86528)             0

 dense_10 (Dense)            (None, 500)               43264500

 dense_11 (Dense)            (None, 100)               50100

 dense_12 (Dense)            (None, 1)                 101

=================================================================
Total params: 43,407,949
Trainable params: 43,407,949
Non-trainable params: 0
_________________________________________________________________
```

In the above model 3 (convolution + pooling) layers are used and then flattened to send it to the fully connected neural network. The model metrics are as follows:







```
              precision    recall  f1-score   support

           0       0.80      0.91      0.85       218
           1       0.89      0.77      0.83       212

    accuracy                           0.84       430
   macro avg       0.85      0.84      0.84       430
weighted avg       0.85      0.84      0.84       430
```

```
Accuracy score: 0.8395348837209302
Recall score: 0.7688679245283019
```

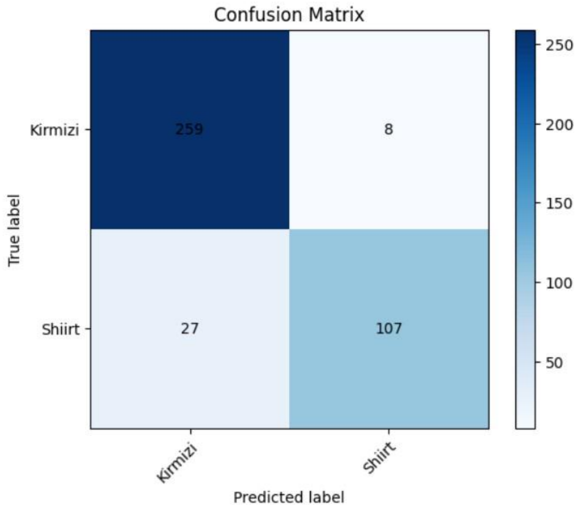
Confusion matrix:

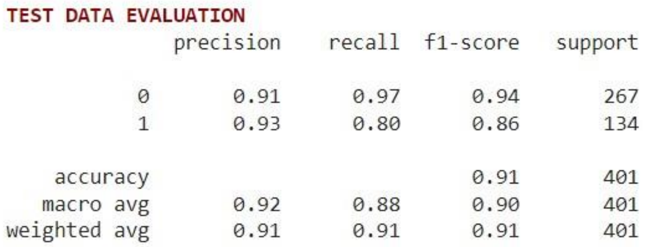
	Kirmizi	Siirt
Kirmizi	198	20
Siirt	49	163

Implementation On Cropped Data:

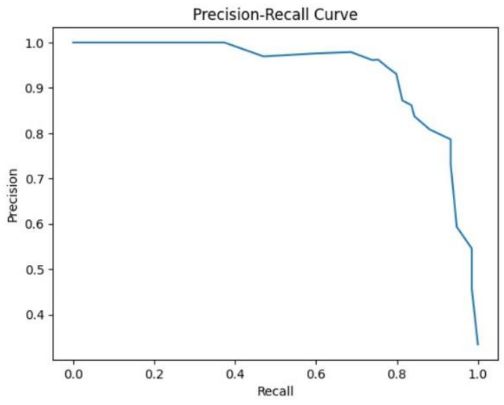
Out of all above models the best model was Random forest and some of the machine learning models. So depending on that 2 experiments are conducted and following are the results

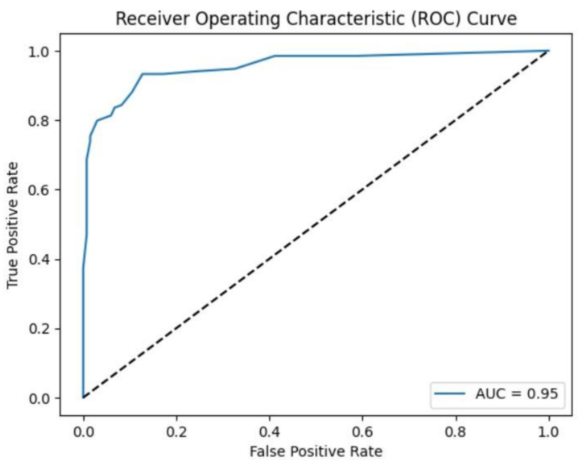
1. Random Forest on cropped data with n_estimators = 30, criterion = entropy, class_weight = balanced.
   Results are as follows:





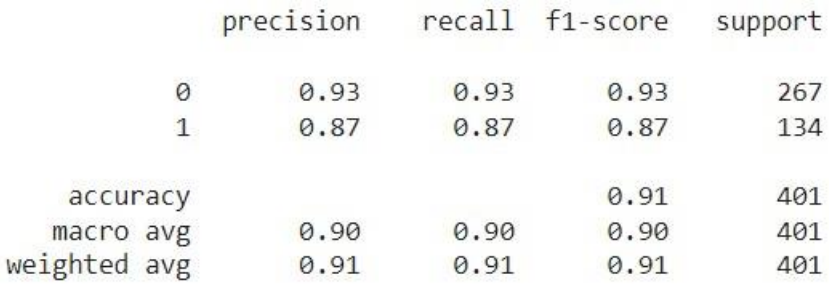
```
TEST DATA EVALUATION
              precision    recall  f1-score   support

           0       0.91      0.97      0.94       267
           1       0.93      0.80      0.86       134

    accuracy                           0.91       401
   macro avg       0.92      0.88      0.90       401
weighted avg       0.91      0.91      0.91       401
```



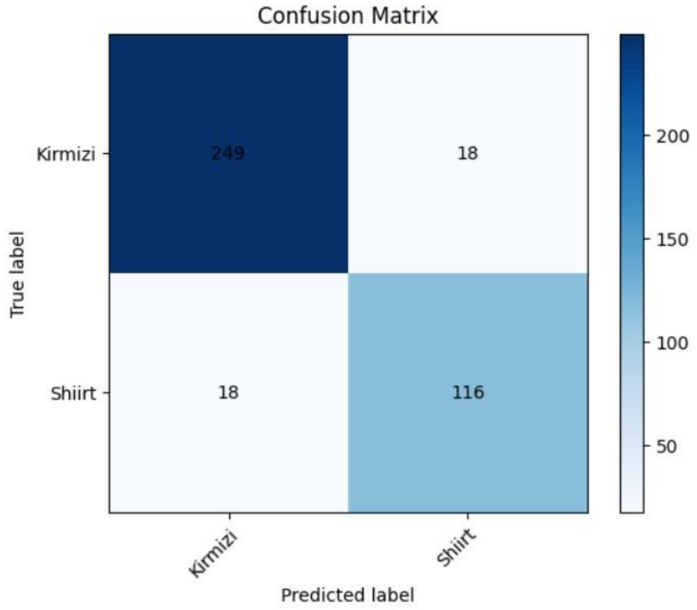


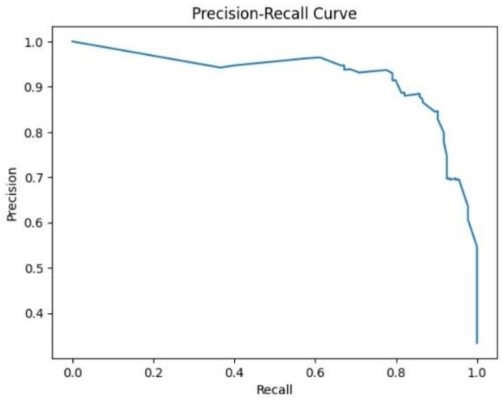
```
ACCURACY:  0.912718204488778
RECALL:  0.7985074626865671
```

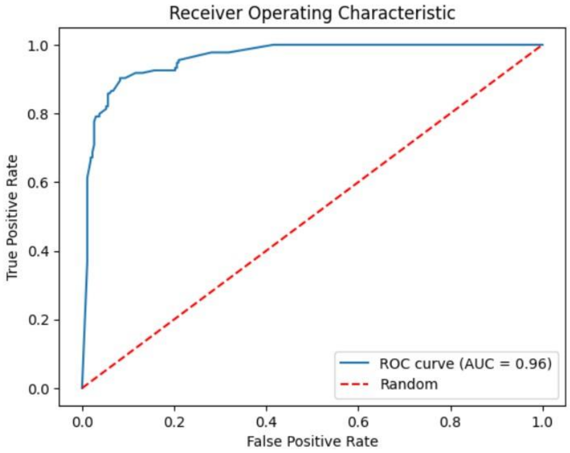
2. Voting Classifier: Random forest, KNN, Logistic Regression
   The results are as follows:



```
              precision    recall  f1-score   support

           0       0.93      0.93      0.93       267
           1       0.87      0.87      0.87       134

    accuracy                           0.91       401
   macro avg       0.90      0.90      0.90       401
weighted avg       0.91      0.91      0.91       401
```







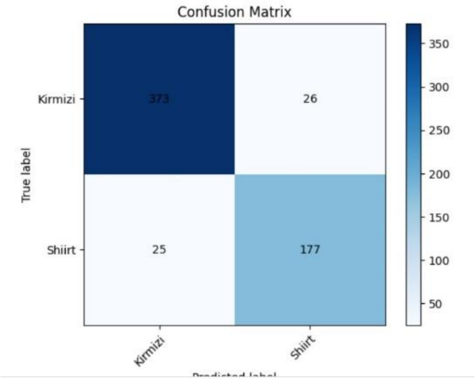
Voting Classifier using 5 models:

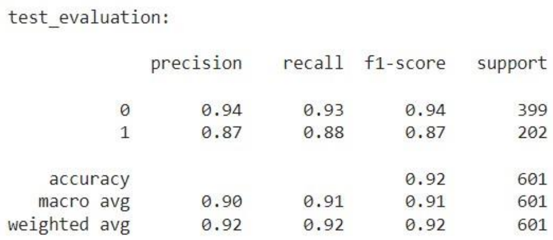
5 best Machine Learning models are

1. Random forest classifier
2. Adaboost

3. SVC
4. N-neighbour classifier
5. Logistic regression

On doing voting classification using these 5 models following are the results





```
test_evaluation:

              precision    recall  f1-score   support

           0       0.94      0.93      0.94       399
           1       0.87      0.88      0.87       202

    accuracy                           0.92       601
   macro avg       0.90      0.91      0.91       601
weighted avg       0.92      0.92      0.92       601
```

Accuracy = 92% (best out of all)

Conclusion

Out of the two categories of pistachio available Kirmizi is more nutritious and costly. Hence, it is better to prefer the model Random Forest Classifier which is predicting Less Kirmizi pistachios as siirt.

Moreover, accuracy point of view voting on ML models does good job where as detecting Kirmizi – random forest does it better.