DAYANANDA SAGAR UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY
KUDLU GATE
BANGALORE - 560068

MINI PROJECT REPORT

ON

"CLASSIFICATION OF RICE SPECIES"

Machine Learning Tools and Techniques (21DS3504)
5th SEMESTER
BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING

Submitted by

ARYAN R G - (ENG21DS0008)
NIKUNJ VIHARI KONAKALLA - (ENG21DS0023)
MIR KHYRUN ALI - (ENG21DS0051)

Under the supervision of

Dr. Kakoli Bora
Associate Professor, Dept. Of CSE(Data Science)

DAYANANDA SAGAR UNIVERSITY

School of Engineering, Kudlu Gate, Bangalore-560068

CERTIFICATE

This is to certify that Mr. Aryan R G, Nikunj Vihari Konakalla, Mir Khyrun Ali bearing USN ENG21DS0008, ENG21DS0023, ENG21DS0051 has satisfactorily completed his/her Mini Project as prescribed by the University for the 5th Semester B.Tech. programme in Computer Science & Engineering (Data Science) during the year at the School of Engineering, Dayananda Sagar University, Bangalore.

Date: 20 December 2023

Signature of the faculty in-charge

Max Marks	Marks Obtained

Signature of Chairman
Department of Computer Science & Engineering

DECLARATION

We hereby declare that the work presented in this mini project entitled - Classification of Rice Species, has been carried out by us and it has not been submitted for the award of any degree, diploma or the mini project of any other college or university.

ARYAN R G - (ENG21DS0008)
NIKUNJ VIHARI – (ENG21DS0023)
MIR KHYRUN ALI - (ENG21DS0051)

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts with success.

We are especially thankful to our **Chairman** **Dr. Shaila S G,** for providing necessary departmental facilities, moral support and encouragement.

We are very much thankful to our **guide**, **Dr. Kakoli Bora** for providing help and suggestions in completion of this mini project successfully.

We have received a great deal of guidance and co-operation from our friends and we wish to thank all that have directly or indirectly helped us in the successful completion of this project work.

ARYAN R G - (ENG21DS0008)
NIKUNJ VIHARI – (ENG21DS0023)
MIR KHYRUN ALI - (ENG21DS0051)

TABLE OF CONTENTS

PROBLEM STATEMENT

Researchers aim to develop accurate and efficient classification models to distinguish between different rice species. Cammeo and Osmancik are two different species of rice from Turkey. The Rice Dataset consists of 3810 instance, each instance of a rice grain. It consists of 2 classes of rice and classification is to be performed to predict the type of rice based on dimensional features.

DATASET DESCRIPTION

The dataset chosen for this project is the Rice (Cammeo and Osmancik) dataset from the UCI Machine Learning Repository. It contains features representing various dimensional parameters of each rice grain. Each instance of the dataset corresponds to data of each rice grain. These dimensions were approximated using the bounded-box method with scanned images of the rice grains. The dataset makes up a combination of two classes/species of rice - Cammeo and Osmancik.

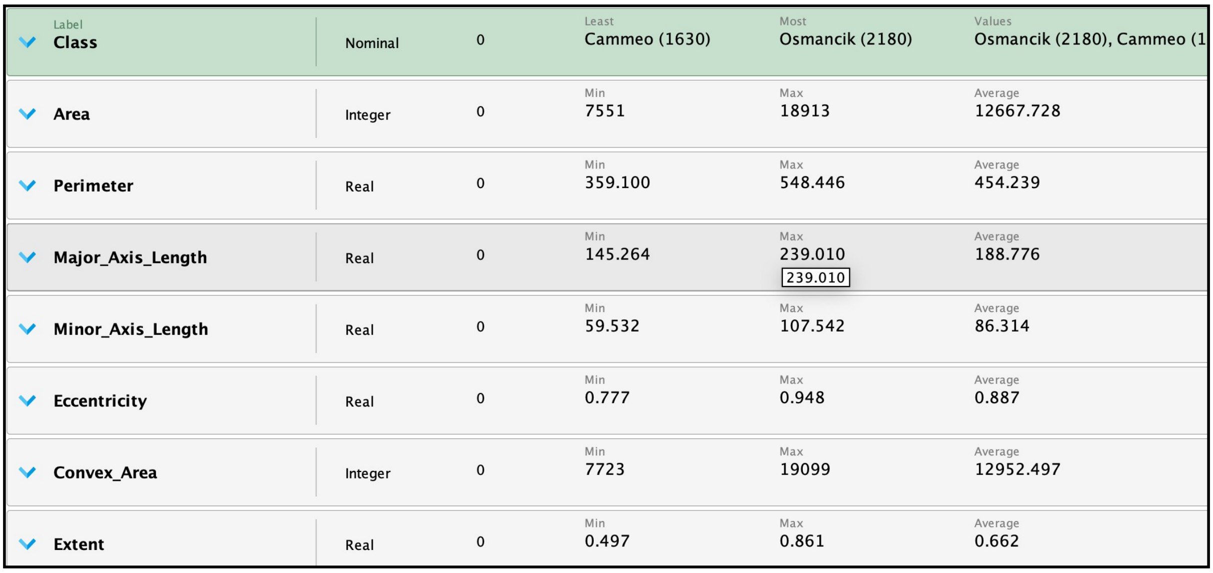
Dataset Type: Multivariate
Feature Type: Real Valued
No. of Instances: 3810
Feature Count: 7
Target: Class

FEATURE DESCRIPTION

Feature Name	**Role**	**Type**	**Description**
Area	Feature	Integer	Area within boundaries of rice grain in pixels
Perimeter	Feature	Continuous	Circumference value calculated by distance between pixels
Major_Axis_Length	Feature	Continuous	Long axis
Minor_Axis_Length	Feature	Continuous	Short axis

Eccentricity	Feature	Continuous	Curvature of the ellipse
Convex_Area	Feature	Integer	Area of the smallest convex shell in pixels
Extent	Feature	Continuous	Ratio of grain region to bounding box
Class	Target	Binary	Cammeo and Osmancik

STATISTICS



Label Class	Nominal	0	Least Cammeo (1630)	Most Osmancik (2180)	Values Osmancik (2180), Cammeo (1
Area	Integer	0	Min 7551	Max 18913	Average 12667.728
Perimeter	Real	0	Min 359.100	Max 548.446	Average 454.239
Major_Axis_Length	Real	0	Min 145.264	Max 239.010 239.010	Average 188.776
Minor_Axis_Length	Real	0	Min 59.532	Max 107.542	Average 86.314
Eccentricity	Real	0	Min 0.777	Max 0.948	Average 0.887
Convex_Area	Integer	0	Min 7723	Max 19099	Average 12952.497
Extent	Real	0	Min 0.497	Max 0.861	Average 0.662

SOLUTION STEPS

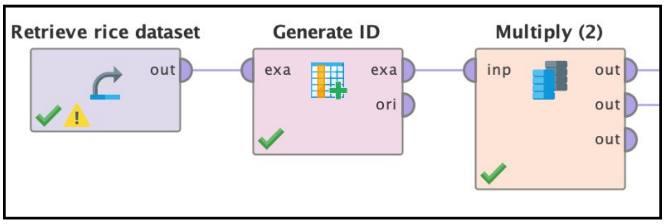
1. **Importing Data :** Using the TurboPrep feature in RapidMiner, the rice.csv file is imported with headers and saved as data object in the local repository. During the import process, the Class attribute's role is set to label. Using the Generate ID operator, we generate IDs for the label.
2. **Correlation Matrix:** The correlation matrix is obtained using the respective operator which also plots a heatmap. We make certain inferences from the heatmap on Strong and Weak Correlations which help in feature/attribute selection.
3. **Preprocessing:**

   1. Normalization : Scaling values of different attributes to similar ranges.
   2. Attribute Selection: Essential attributes are selected as they positively contribute the most in the classification process.

4. **Splitting Example set into training and testing set :** Using the split data operator, we split data into training and testing sets used with models and apply model operator for model training.
5. **Model Training :** The rice dataset is associated with classification. Hence, Random Forest, SVM and a Deep Learning models were used as classifiers for the example set. Additionally clustering was performed on the example set using the K-means method to compare the accuracy and performance. The Apply Model operator was used to fit the example set and validate against the testing set.
6. **Cross Validation Method:** SVM was used with cross validation to obtain better performance on the model.
7. **Output:** The performance operator is used to get the values for different performance measures such as Accuracy, Precision, Recall and AUC from ROC Curve.

SCREENSHOTS OF STEPS AND RESULTS

1. Importing Data



Operator: Retrieve

Used to import example set from local repository.

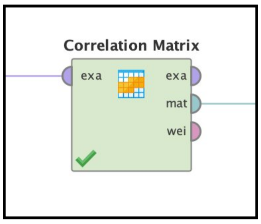
Operator: Generate IDs

This operator generates IDs for all the corresponding values of the label Class.

Operator: Multiply

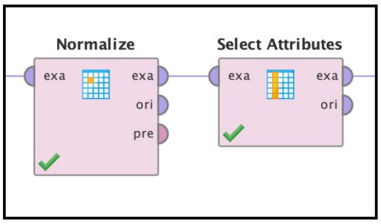
Returns copies of example set

2. Correlation Matrix



Operator: Correlation Matrix
Returns a Correlation Matrix and a correlation heatmap

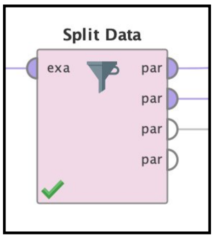
3. Preprocessing



Operator : Normalize
The example set has 7 features with some of them having different ranges. We use the Normalize operator which performs scaling of values of all attributes except ID, to a similar range. Z-Transformation as method of Normalization. Returns an example set of normalized values.

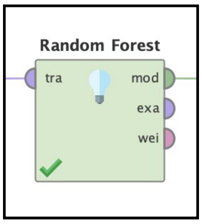
Operator: Select Attributes
The attribute, Extent had the least correlation with all other attributes and is excluded from the example set using this operator.

4. Train Test Split



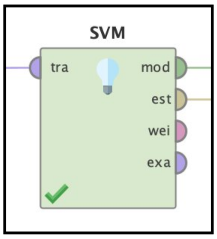
Operator : Split Data
The example set is then split into training and testing sets using operator. It allowed us to specify the ratio as enumeration of (0.8, 0.2). This operator returns partitions of example set partitioned according to the specified ratio.

4. Model Training



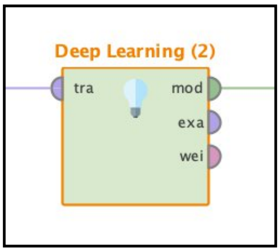
Operator: Random Forest

Training partition from Split Data is given as input to the Random Forest operator. No. of decision trees and maximum depth is set to 100 and 10 respectively. Criterion is set to gain ratio and voting to confidence. Returns a model object.



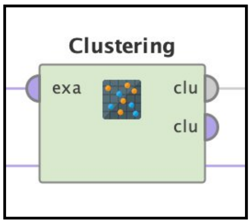
Operator: SVM

Training Partition is given as input to SVM operator. Dot Kernel is selected. Returns a model object and estimated performance vector.



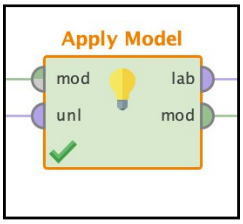
Operator: Deep Learning

Training Partition given as input. No. of epochs is set to 15. Rectifier used as activation function. Two hidden layers each consisting of 50 neurons. Returns a model object.



Operator: Clustering

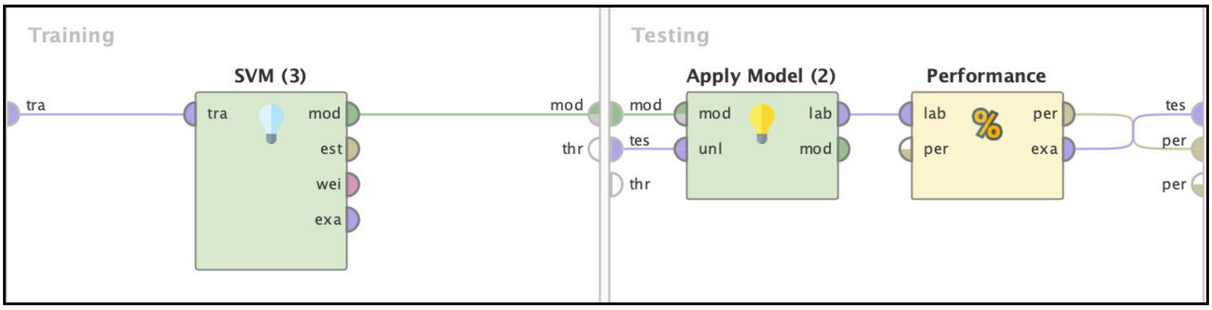
Full Training set is given as input to the Clustering operator. No. of clusters is set to 2 as there are 2 classes. Returns clusters visual and performance as output.



Operator: Apply Model

Takes model as input. Fits training partition to the model and validates with the testing partition. Returns testing example set with predicted values.

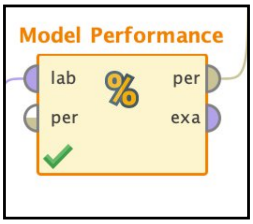
5. Model Training with Cross Validation



Subprocess: Cross Validation

SVM and Deep Learning with cross validation method which returns the example set and confusion matrix.

6. Output

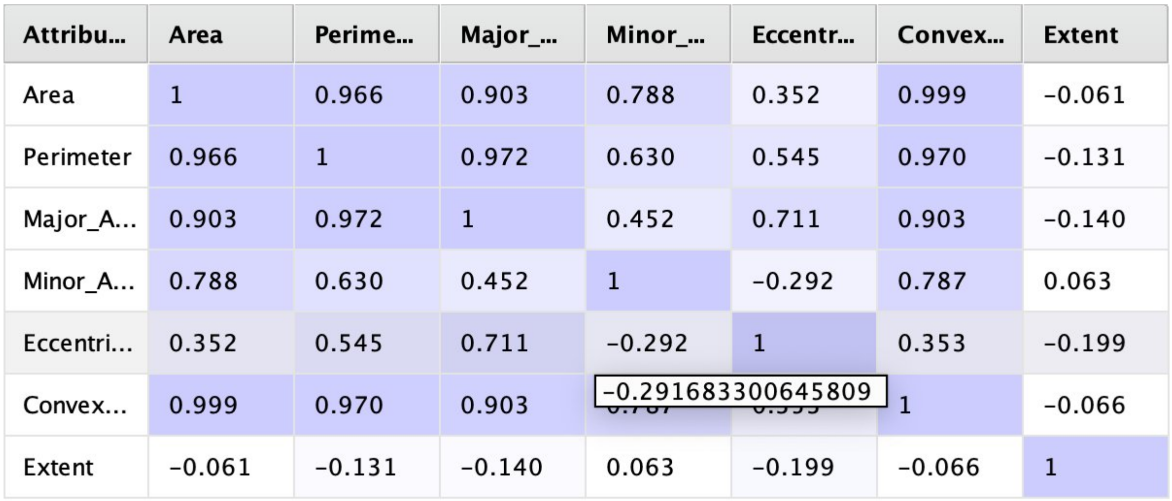


Operator: Model Performance

The performance operator returns a confusion matrix and computes Accuracy, Precision, Recall and plots the ROC curve.

PERFORMANCE EVALUATION

1. Correlation Matrix

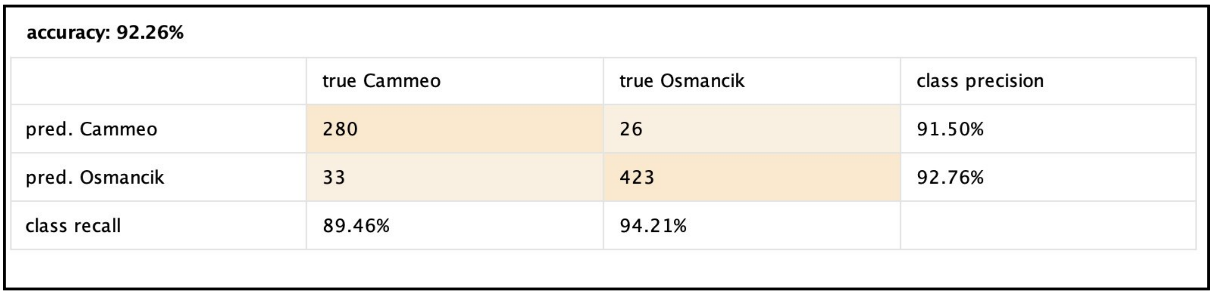


Attribu...	Area	Perime...	Major_...	Minor_...	Eccentr...	Convex...	Extent
Area	1	0.966	0.903	0.788	0.352	0.999	−0.061
Perimeter	0.966	1	0.972	0.630	0.545	0.970	−0.131
Major_A...	0.903	0.972	1	0.452	0.711	0.903	−0.140
Minor_A...	0.788	0.630	0.452	1	−0.292	0.787	0.063
Eccentri...	0.352	0.545	0.711	−0.292	1	0.353	−0.199
Convex...	0.999	0.970	0.903	0.787	0.353	1	−0.066
Extent	−0.061	−0.131	−0.140	0.063	−0.199	−0.066	1

−0.291683300645809

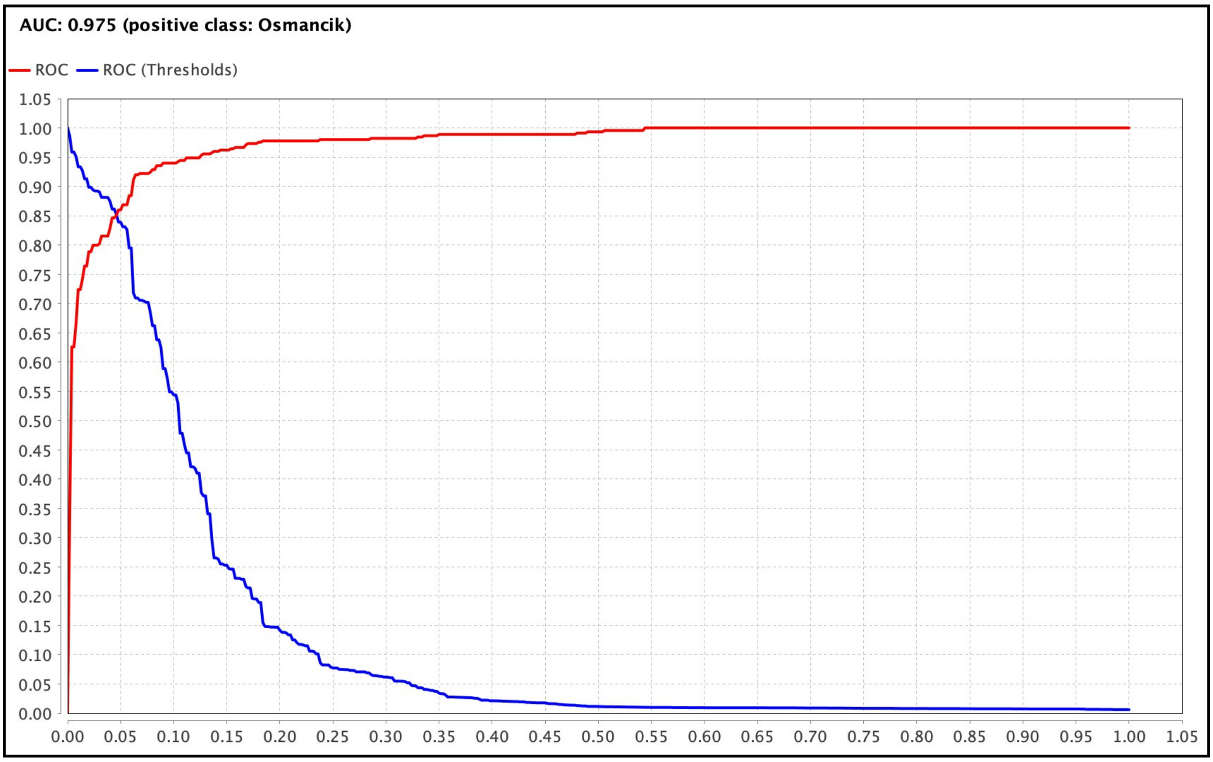
It was observed that the feature, *Extent* had the least correlation ranging from -0.199 to 0.6063 of all features. While most of them having High Positive correlation where the Highest was, Area and Convex_Area had the highest correlation of 0.999 followed by Perimeter and Major Axis Length with value of 0.972. From these results it can be inferred that, most dimensional parameters closely relate to each other. Closely related features help with better classification performance because the model can perform precise assignment of weights.

2. Random Forest



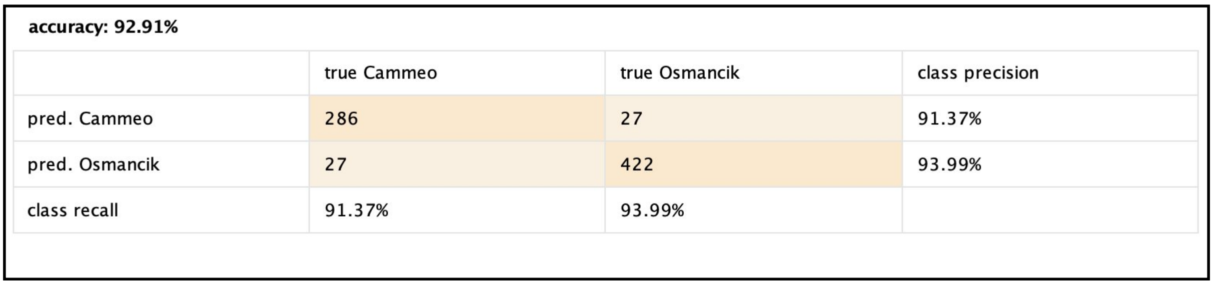
accuracy: 92.26%

	true Cammeo	true Osmancik	class precision
pred. Cammeo	280	26	91.50%
pred. Osmancik	33	423	92.76%
class recall	89.46%	94.21%	



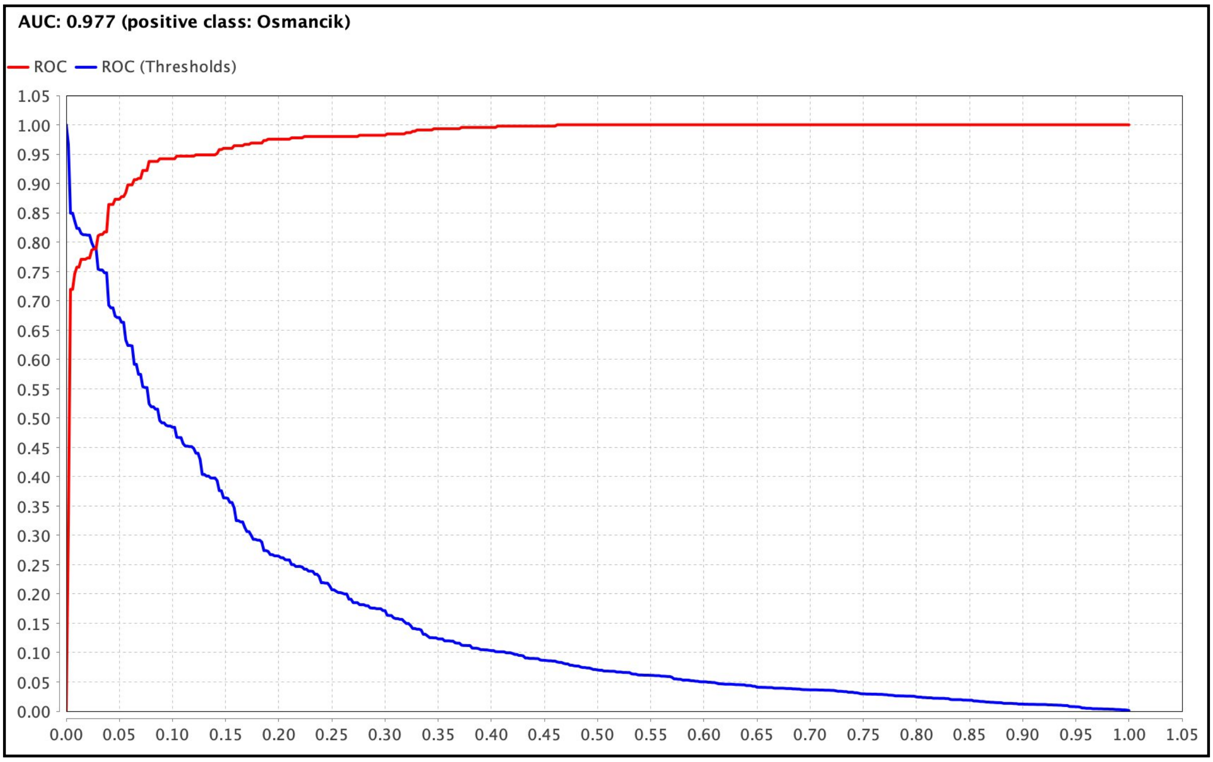
AUC: 0.975 (positive class: Osmancik)

3. SVM

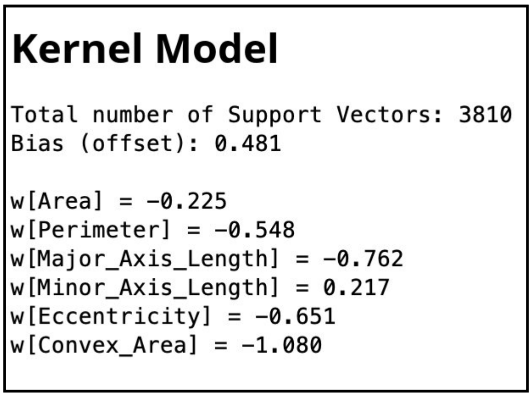


accuracy: 92.91%

	true Cammeo	true Osmancik	class precision
pred. Cammeo	286	27	91.37%
pred. Osmancik	27	422	93.99%
class recall	91.37%	93.99%	



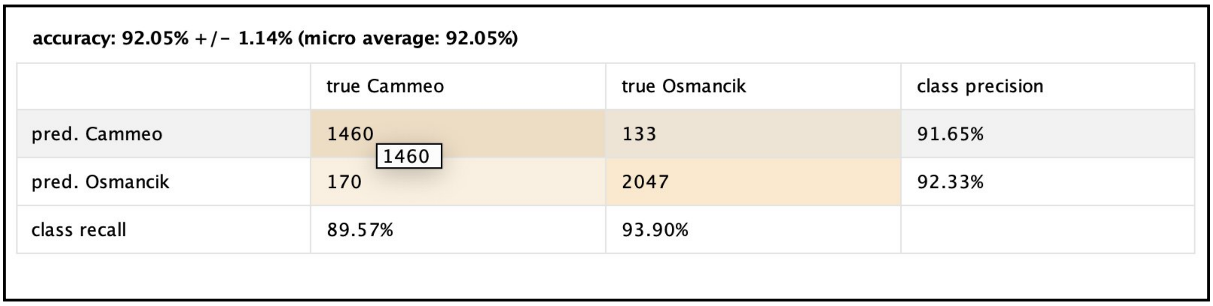
AUC: 0.977 (positive class: Osmancik)



Kernel Model

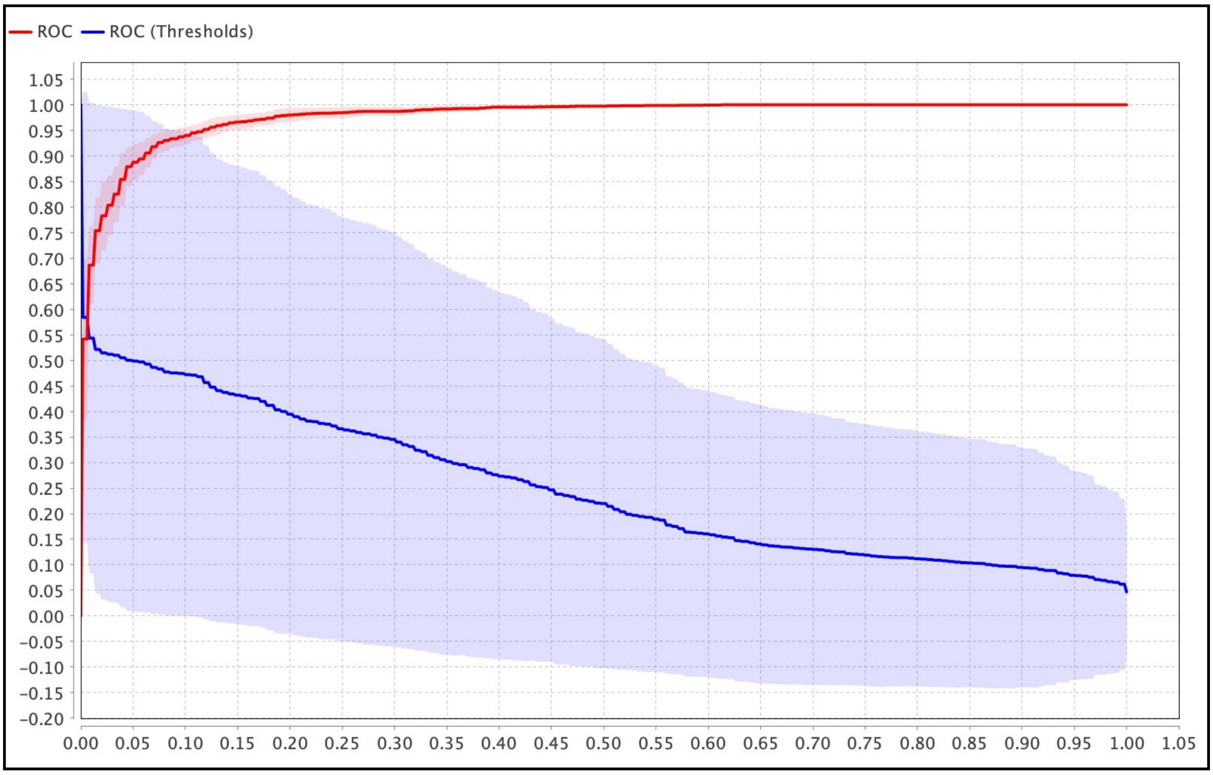
```
Total number of Support Vectors: 3810
Bias (offset): 0.481

w[Area] = -0.225
w[Perimeter] = -0.548
w[Major_Axis_Length] = -0.762
w[Minor_Axis_Length] = 0.217
w[Eccentricity] = -0.651
w[Convex_Area] = -1.080
```

4. Deep Learning



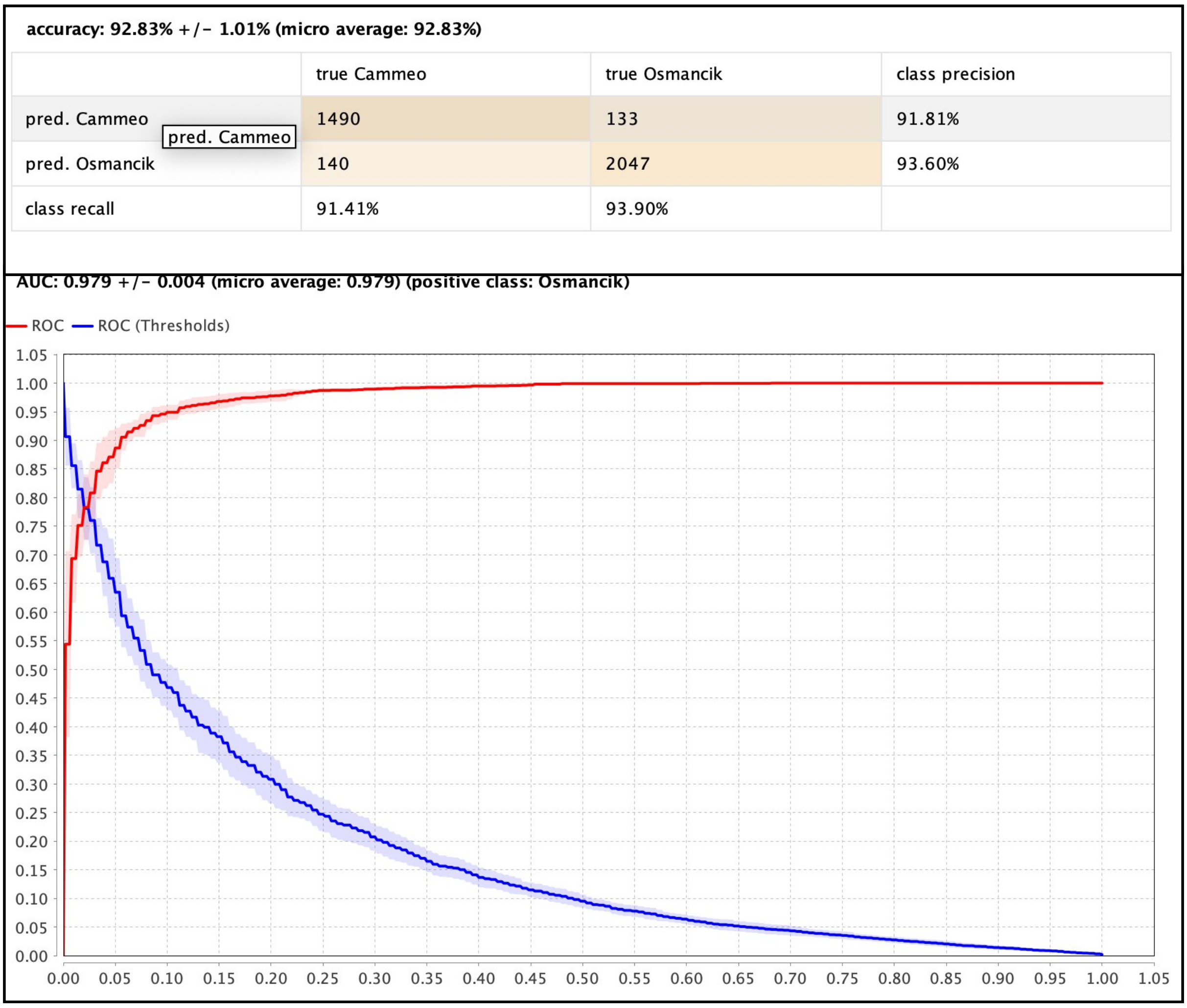
accuracy: 92.05% +/- 1.14% (micro average: 92.05%)

	true Cammeo	true Osmancik	class precision
pred. Cammeo	1460	133	91.65%
pred. Osmancik	170	2047	92.33%
class recall	89.57%	93.90%	



AUC: 0.978 +/- 0.006 (micro average: 0.978) (positive class: Osmancik)

5. Cross Validation



accuracy: 92.83% +/- 1.01% (micro average: 92.83%)

	true Cammeo	true Osmancik	class precision
pred. Cammeo	1490	133	91.81%
pred. Osmancik	140	2047	93.60%
class recall	91.41%	93.90%	

AUC: 0.979 +/- 0.004 (micro average: 0.979) (positive class: Osmancik)

Accuracy:

Random Forest:	92.26 %
SVM:	92.91 %
Deep Learning:	92.05 %
SVM with Cross Validation:	92.83 %

Area Under Curve:

Random Forest:	0.975
SVM:	0.977
Deep Learning:	0.978
SVM with Cross Validation:	0.979

Conclusion

By assessing the performance measure we understand that all models perform similarly with very good accuracy of around 92%. This implies that the models were able to predict the classes – Cammeo and Osmancik 92 out of 100 examples correctly. The AUC obtained is also similar for all the models. It is observed that SVM model’s accuracy is highest by a small margin, with value of 92.91%. AUC values is the highest of 0.979 for SVM with cross validation. Since all models perform well with similar accuracy, to choose the best model we must consider other factors such as model interpretability, available resources, prediction speed and computation efficiency.