Data Science Project Training Report

on

Machine Learning Domain Projects for Regression, Classification and Clustering using Various Datasets

BACHELOR OF TECHNOLOGY

Session 2021-22
in

CSE(Data Science)

By
SANCHI SINGHAL
2000321540051

AATIF JAMSHED
ASSISTANT PROFESSOR

DEPARTMENT OF INFORMATION TECHNOLOGY
ABES ENGINEERING COLLEGE, GHAZIABAD

AFFILIATED TO
DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, U.P., LUCKNOW
(Formerly UPTU)

Student’s Declaration

I hereby declare that the work being presented in this report entitled **“RICE CLASSIFICATION”** is an authentic record of my own work carried out under the supervision of Dr. /Mr. /Ms. **AATIF JAMSHED, Assistant Professor, Information Technology.**

Date: 01 July 2022

Signature of student
(Name: Sanchi Singhal)
(Roll No.: 2000321540051)
Department: CSE (Data Science)

This is to certify that the above statement made by the candidate(s) is correct to the best of my knowledge.

Signature of HOD	**Signature of Teacher**
Dr. Amit Sinha	**Aatif Jamshed**
Information Technology	**Assistant Professor**
	Information Technology

Date:.........................

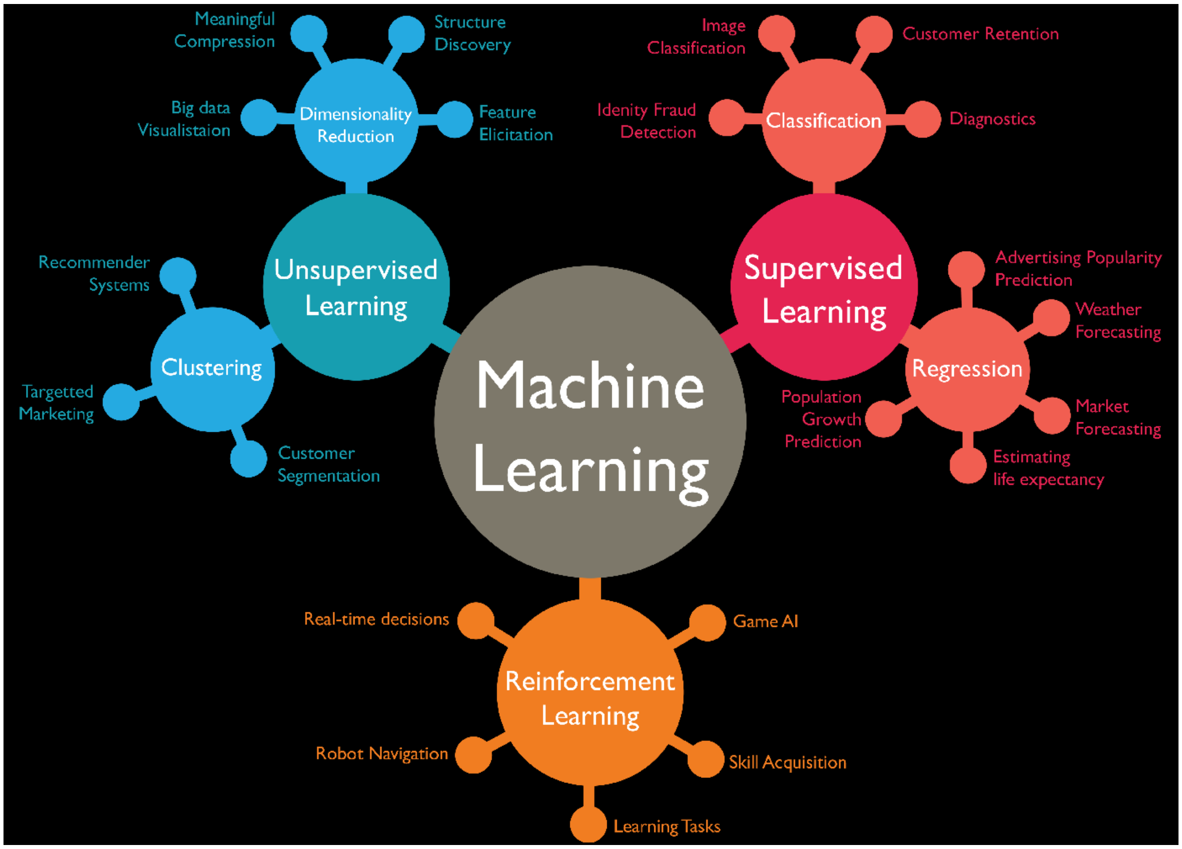
Table of Contents

MACHINE LEARNING

Machine learning is a discipline that deals with programming the systems so as to make them automatically learn and improve with experience. Here, learning implies recognizing and understanding the input data and taking informed decisions based on the supplied data.

Applications of Machine Learning:

- Vision processing
- Language processing
- Forecasting things like stock market trends, weather
- Pattern recognition
- Games
- Data mining
- Expert systems
- Robotics



CLASSIFICATION

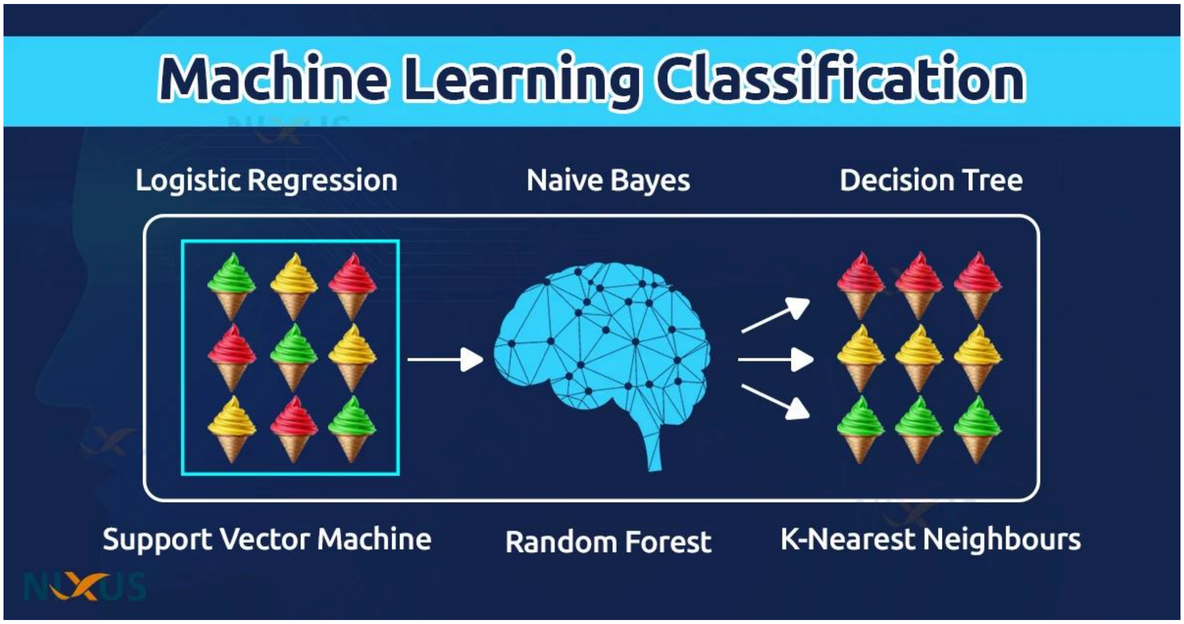
Classification is a process of categorizing a given set of data into classes, It can be performed on both structured or unstructured data. The process starts with predicting the class of given data points. The classes are often referred to as target, label or categories.

The algorithm which implements the classification on a dataset is known as a classifier. There are two types of Classifications:

- **Binary Classifier:** If the classification problem has only two possible outcomes, then it is called as Binary Classifier.
  Examples: YES or NO, MALE or FEMALE, SPAM or NOT SPAM, CAT or DOG, etc.
- **Multi-class Classifier:** If a classification problem has more than two outcomes, then it is called as Multi-class Classifier.
  Example: Classifications of types of crops, Classification of types of music.

In classification algorithm, a discrete output function(y) is mapped to input variable(x).

y=f(x), where y = categorical output



PROJECT

RICE CLASSIFICATION

Abstract:

Rice, which is among the most widely produced grain products worldwide, has many genetic varieties. These varieties are separated from each other due to some of their features. These are usually features such as texture, shape, and color. With these features that distinguish rice varieties, it is possible to classify and evaluate the quality of seeds. In this study, Arborio, Basmati, Ipsala, Jasmine and Karacadag, which are five different varieties of rice often grown in Turkey, were used.

Dataset Used:

https://www.kaggle.com/datasets/muratkokludataset/rice-image-dataset

Features:

- Arborio, Basmati, Ipsala, Jasmine and Karacadag rice varieties were used.
- The dataset has 75K images including 15K pieces from each rice variety.

LIBRARIES USED

1. **Pandas**

Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series.

This module is generally imported as:

```
import pandas as pd
```

2. **Numpy**

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It is open-source software.

This module is generally imported as:

```
import numpy as np
```

3. **Seaborn**

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top of matplotlib library and also closely integrated to the data structures from pandas.

This module is generally imported as:

```
import seaborn as sns
```

4. **Matplotlib**

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack.

This module is generally imported as:

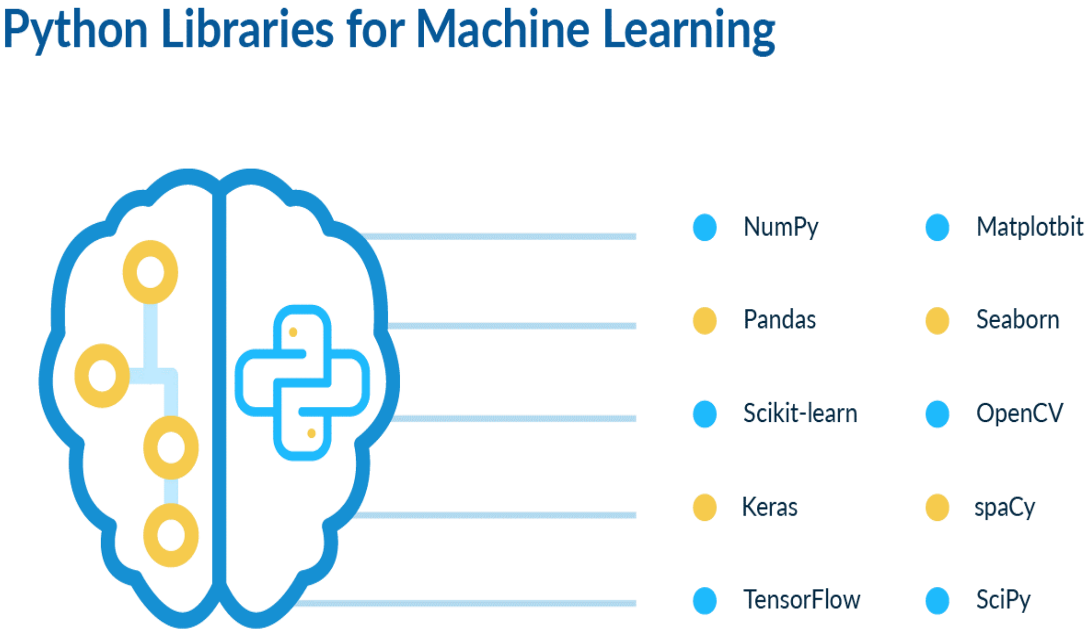
```
import matplotlib.pyplot as plt
```

5. Sklearn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

Benefits of using scikit-learn over some other machine learning libraries(like R libraries):

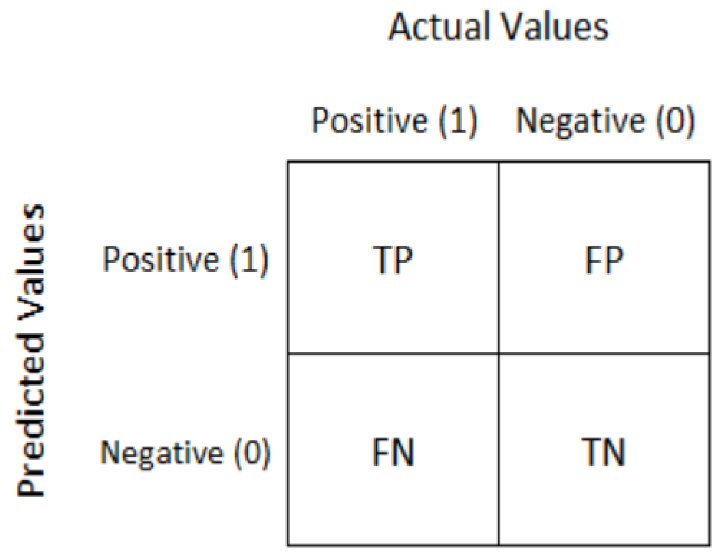
- **Consistent interface** to machine learning models
- Provides many **tuning parameters** but with **sensible defaults**
- Exceptional **documentation**
- Rich set of functionality for **companion tasks**.
- **Active community** for development and support.


Python Libraries for Machine Learning

PREREQUISITES

1. Confusion Matrix

It is a performance measurement for machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values.



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

True Positive:
Interpretation: You predicted positive and it’s true.

True Negative:
Interpretation: You predicted negative and it’s true.

False Positive: (Type 1 Error)
Interpretation: You predicted positive and it’s false.

False Negative: (Type 2 Error)
Interpretation: You predicted negative and it’s false.

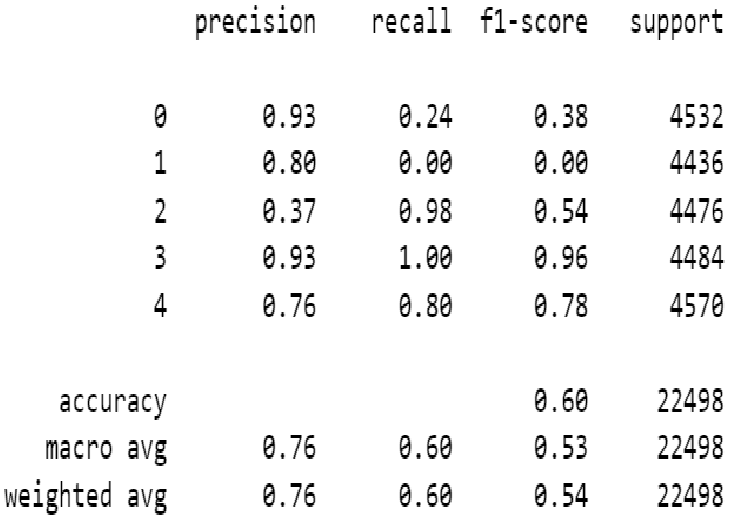
$$\textit{Recall} = \frac{TP}{TP + FN}$$

$$\textit{Precision} = \frac{TP}{TP + FP}$$

$$\textit{F - measure} = \frac{2*Recall*Precision}{Recall + Precision}$$

2. Classification Report

It is the report which explains everything about the classification. This is the summary of the quality of classification made by the constructed ML model. It comprises mainly 5 columns and (N+3) rows. The first column is the class label's name and followed by Precision, Recall, F1-score, and Support. N rows are for N class labels and other three rows are for accuracy, macro average, and weighted average.



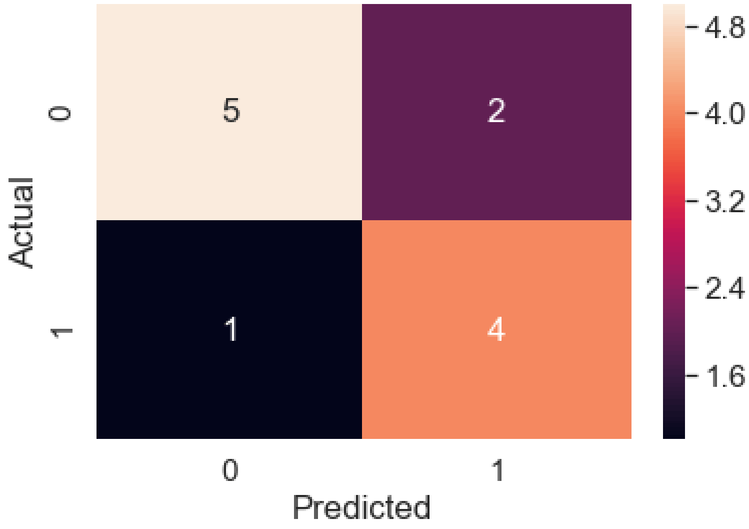
```
              precision    recall  f1-score   support

           0       0.93      0.24      0.38      4532
           1       0.80      0.00      0.00      4436
           2       0.37      0.98      0.54      4476
           3       0.93      1.00      0.96      4484
           4       0.76      0.80      0.78      4570

    accuracy                           0.60     22498
   macro avg       0.76      0.60      0.53     22498
weighted avg       0.76      0.60      0.54     22498
```

3. Heatmap

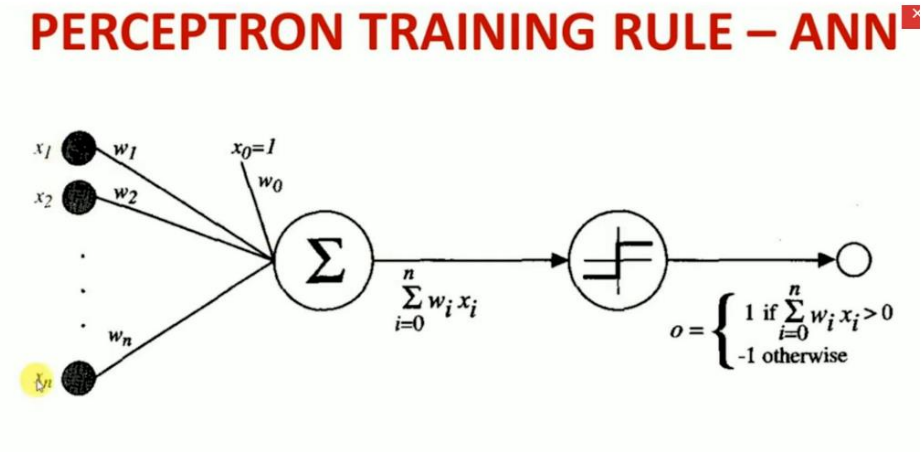
A heatmap contains values representing various shades of the same colour for each value to be plotted. Usually the darker shades of the chart represent higher values than the lighter shade. For a very different value a completely different colour can also be used.



MODELS

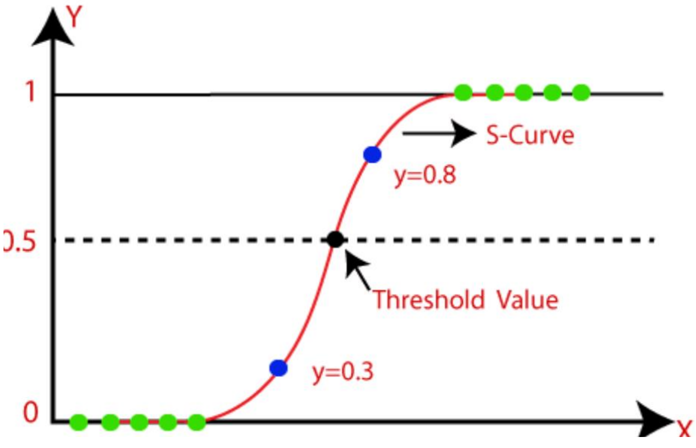
1. **Perceptron:**

Perceptron is Machine Learning algorithm for supervised learning of various binary classification tasks. Further, *Perceptron is also understood as an Artificial Neuron or neural network unit that helps to detect certain input data computations in business intelligence*.


PERCEPTRON TRAINING RULE – ANN
x_1
w_1
x_2
w_2
$x_0=1$
w_0
Σ
$\sum_{i=0}^{n} w_i x_i$
w_n
x_n
$o = \begin{cases} 1 \text{ if } \sum_{i=0}^{n} w_i x_i > 0 \\ -1 \text{ otherwise} \end{cases}$

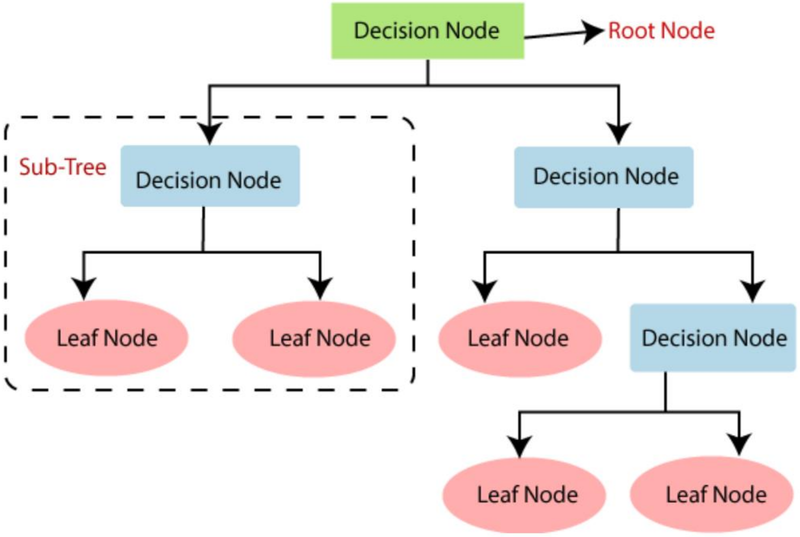
2. **Logistic Regression:**

Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.



3. **<u>Decision Tree Classifier:</u>**

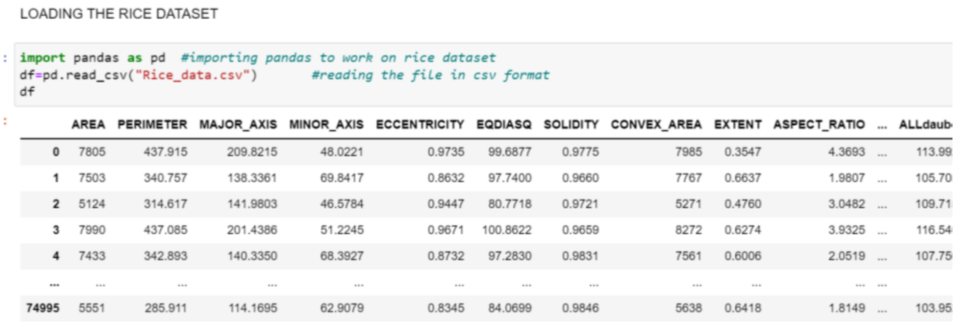
It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node**.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset.



STEPS INVOLVED IN MAKING PROJECT

1. Loading the Dataset

Data is loaded using pandas.



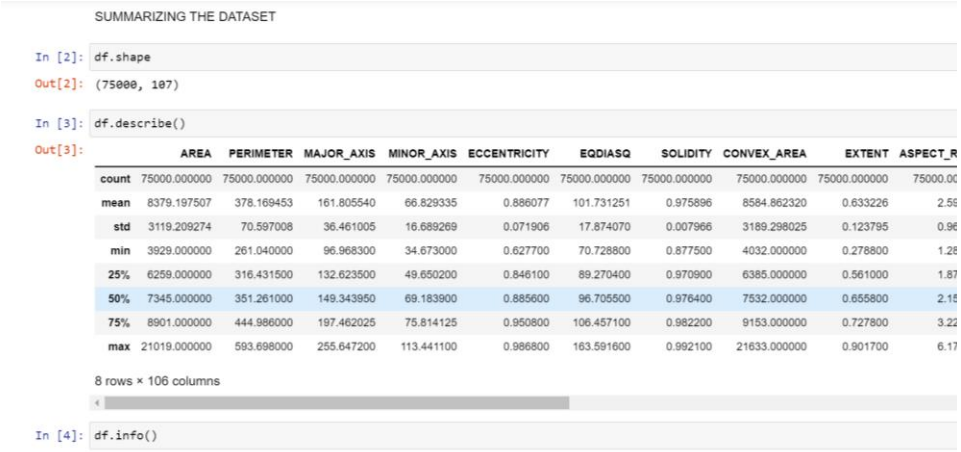
LOADING THE RICE DATASET

```
import pandas as pd  #importing pandas to work on rice dataset
df=pd.read_csv("Rice_data.csv")       #reading the file in csv format
df
```

	AREA	PERIMETER	MAJOR_AXIS	MINOR_AXIS	ECCENTRICITY	EQDIASQ	SOLIDITY	CONVEX_AREA	EXTENT	ASPECT_RATIO	...	ALLdaub
0	7805	437.915	209.8215	48.0221	0.9735	99.6877	0.9775	7985	0.3547	4.3693	...	113.99
1	7503	340.757	138.3361	69.8417	0.8632	97.7400	0.9660	7767	0.6637	1.9807	...	105.70
2	5124	314.617	141.9803	46.5784	0.9447	80.7718	0.9721	5271	0.4760	3.0482	...	109.71
3	7990	437.085	201.4386	51.2245	0.9671	100.8622	0.9659	8272	0.6274	3.9325	...	116.54
4	7433	342.893	140.3350	68.3927	0.8732	97.2830	0.9831	7561	0.6006	2.0519	...	107.75
...	...	...	...	...	...	...	...	...	...	...	...	
74995	5551	285.911	114.1695	62.9079	0.8345	84.0699	0.9846	5638	0.6418	1.8149	...	103.95

2. Summarizing the Dataset

Data is summarized to get familiar with the data.



SUMMARIZING THE DATASET

```
In [2]: df.shape
Out[2]: (75000, 107)

In [3]: df.describe()
Out[3]:
```

	AREA	PERIMETER	MAJOR_AXIS	MINOR_AXIS	ECCENTRICITY	EQDIASQ	SOLIDITY	CONVEX_AREA	EXTENT	ASPECT_R
count	75000.000000	75000.000000	75000.000000	75000.000000	75000.000000	75000.000000	75000.000000	75000.000000	75000.000000	75000.00
mean	8379.197507	378.169453	161.805540	66.829335	0.886077	101.731251	0.975896	8584.862320	0.633226	2.59
std	3119.209274	70.597008	36.461005	16.689269	0.071906	17.874070	0.007966	3189.298025	0.123795	0.96
min	3929.000000	261.040000	96.968300	34.673000	0.627700	70.728800	0.877500	4032.000000	0.278800	1.28
25%	6259.000000	316.431500	132.623500	49.650200	0.846100	89.270400	0.970900	6385.000000	0.561000	1.87
50%	7345.000000	351.261000	149.343950	69.183900	0.885600	96.705500	0.976400	7532.000000	0.655800	2.15
75%	8901.000000	444.986000	197.462025	75.814125	0.950800	106.457100	0.982200	9153.000000	0.727800	3.22
max	21019.000000	593.698000	255.647200	113.441100	0.986800	163.591600	0.992100	21633.000000	0.901700	6.17

8 rows × 106 columns

```
In [4]: df.info()
```

3. Preprocessing the Dataset

a. Removing null values using dropna()

b. Standardizing using StandardScaler():

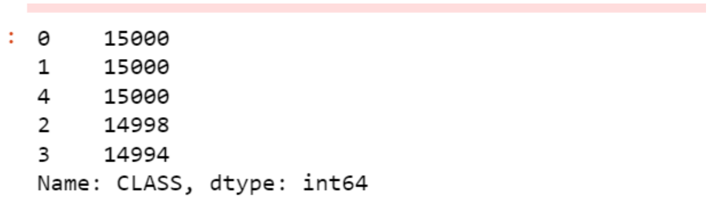
Standardize features by removing the mean and scaling to unit variance. The standard score of a sample x is calculated as:

$$z = (x - u) / s$$

where u is the mean of the training samples and s is the standard deviation of the training samples

4. Labelling the Target Values

The target values are labelled using dictionary mapping.



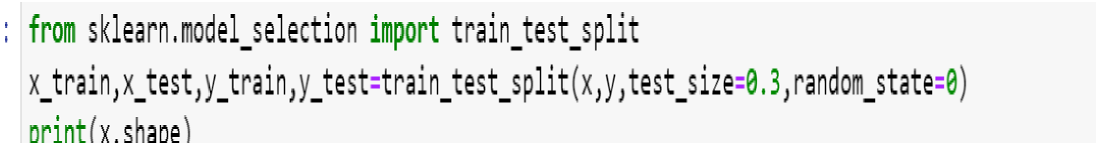
```
: 0    15000
  1    15000
  4    15000
  2    14998
  3    14994
  Name: CLASS, dtype: int64
```

5. Dividing the Data

Data is divided into independent values and target values.

6. Splitting the Data

Splitting the Data for training and testing.



```
: from sklearn.model_selection import train_test_split
  x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=0)
  print(x.shape)
```

7. Applying Models

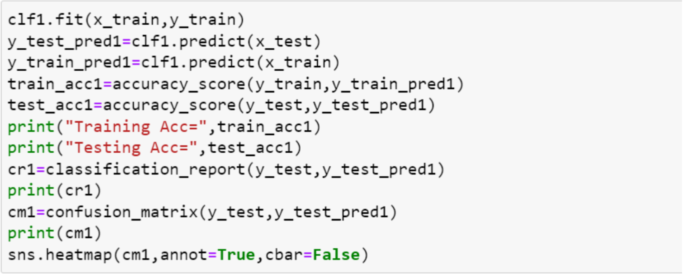
Three models are applied:

a. Perceptron
b. Logistic Regression
c. Decision Tree Classifier

All the three models are test under three scenarios:

A. Without Standardization
B. With Standardization
C. With Hyper Parameter Tuning

Model fitting snippet:

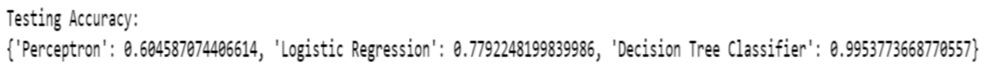


```
clf1.fit(x_train,y_train)
y_test_pred1=clf1.predict(x_test)
y_train_pred1=clf1.predict(x_train)
train_acc1=accuracy_score(y_train,y_train_pred1)
test_acc1=accuracy_score(y_test,y_test_pred1)
print("Training Acc=",train_acc1)
print("Testing Acc=",test_acc1)
cr1=classification_report(y_test,y_test_pred1)
print(cr1)
cm1=confusion_matrix(y_test,y_test_pred1)
print(cm1)
sns.heatmap(cm1,annot=True,cbar=False)
```

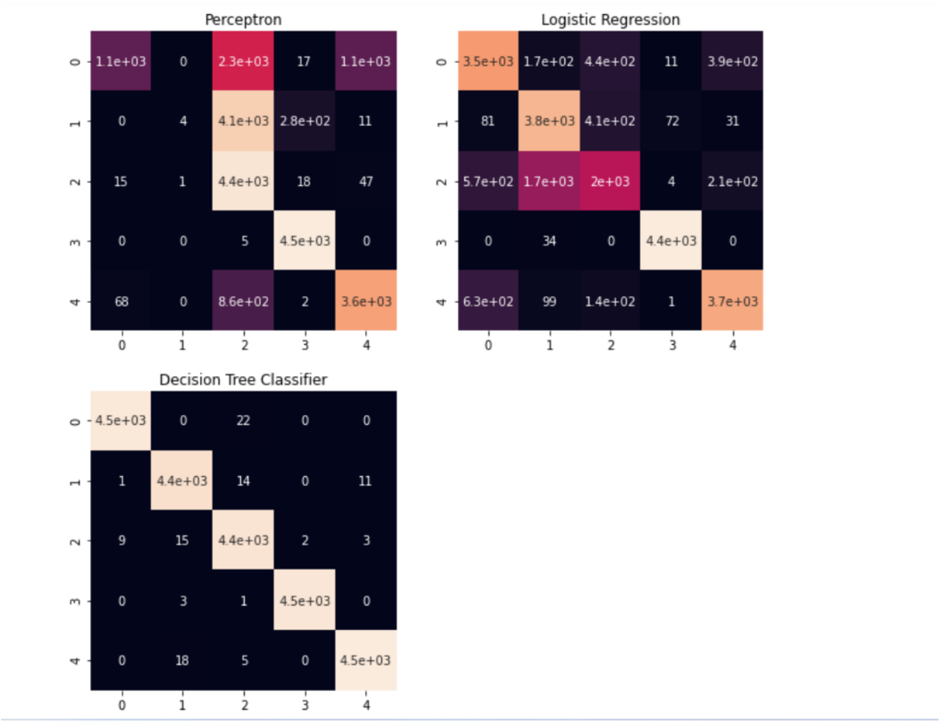
8. Comparison

All the three models are compared in all three scenarios:

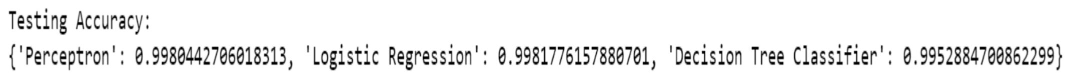
Before Standardizing:



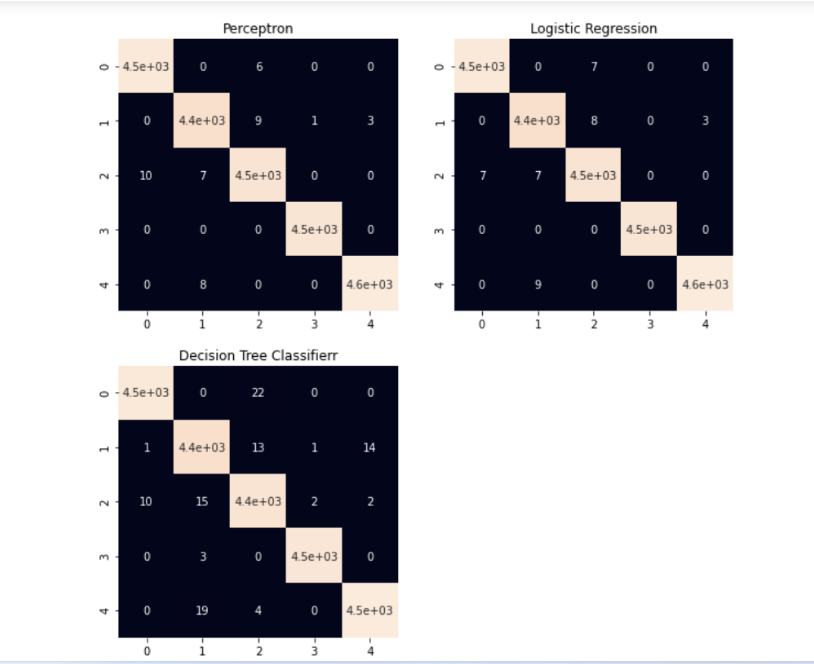
```
Testing Accuracy:
{'Perceptron': 0.604587074406614, 'Logistic Regression': 0.7792248199839986, 'Decision Tree Classifier': 0.9953773668770557}
```



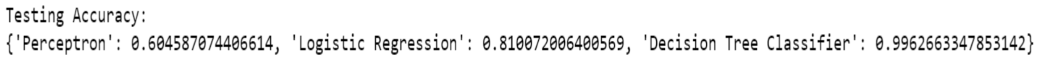
After Standardizing:



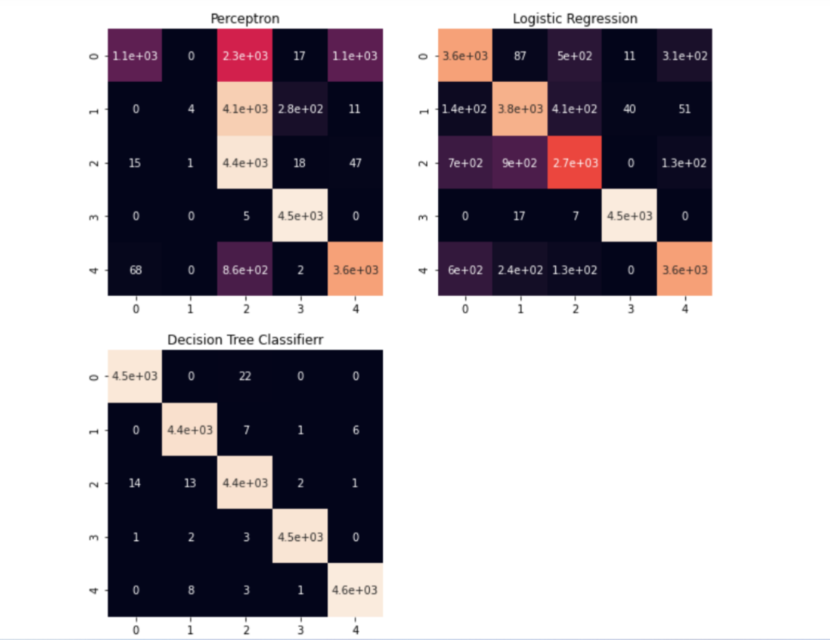
```
Testing Accuracy:
{'Perceptron': 0.9980442706018313, 'Logistic Regression': 0.9981776157880701, 'Decision Tree Classifier': 0.9952884700862299}
```



With Hyper Parameter Tuning:

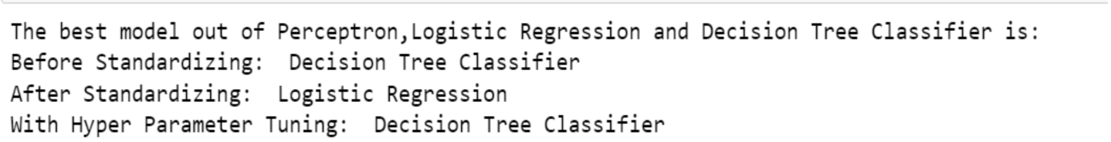


```
Testing Accuracy:
{'Perceptron': 0.604587074406614, 'Logistic Regression': 0.810072006400569, 'Decision Tree Classifier': 0.9962663347853142}
```



9. Conclusion

The model which is best out of all three in all scenarios is concluded:



```
The best model out of Perceptron,Logistic Regression and Decision Tree Classifier is:
Before Standardizing:  Decision Tree Classifier
After Standardizing:  Logistic Regression
With Hyper Parameter Tuning:  Decision Tree Classifier
```

ENCLOSURES

GitHub:

https://github.com/sanchi-singhal/Rice-Data-Classification

Website:

https://sites.google.com/view/sanchi-singhal/home

YouTube:

https://www.youtube.com/watch?v=W8HDzNR8CeE

REFERENCES

Dataset:

https://www.kaggle.com/datasets/muratkokludataset/rice-image-dataset

Libraries:

https://scikit-learn.org/stable/

Learning:

https://www.javatpoint.com/classification-algorithm-in-machine-learning

https://www.geeksforgeeks.org/hyperparameter-tuning/

https://www.tutorialspoint.com/scikit_learn/index.htm