Group 14

Dry Beans Classification

AJAY ATHITYA RAMANATHAN
RIDDHI GUPTA
HIMANI THAKKER
MONIL RAWKA

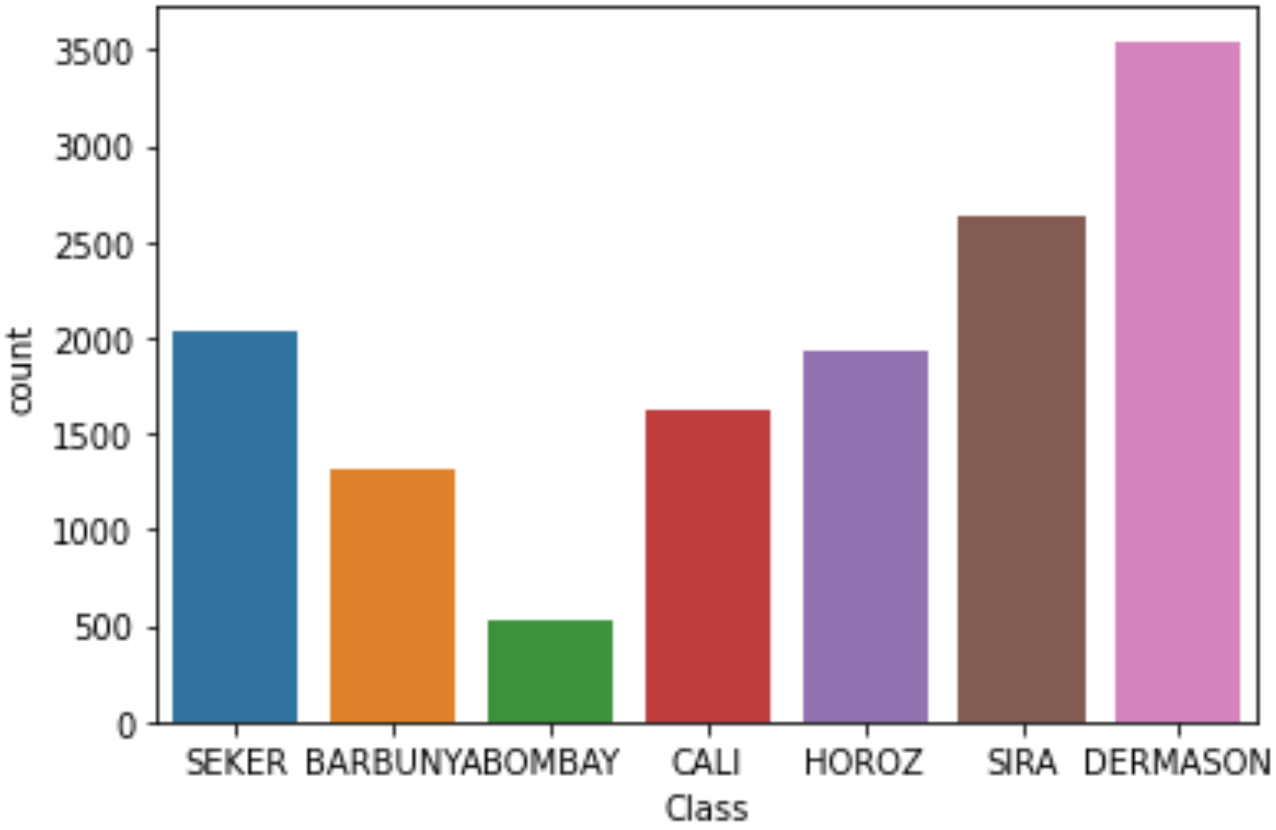
Introduction

– The dry bean, the most widely grown edible legume crop in the world, has a wide variety of genetic variability. There is no doubt that seed quality affects crop production. In order to supply the fundamentals of sustainable agricultural systems, seed classification is crucial for both marketing and production. More than 13,000 samples of dry beans from 7 different species were photographed, and using computer vision techniques, their geometry was determined. We will be using this output dataset generated by the computer vision algorithm. This project’s main goal is to offer a process for obtaining consistent seed varieties from population-based crop production, which prevents the seeds from being verified as belonging to a single variety.

Dataset Description

- 1. Area (A): The area of a bean zone and the number of pixels within its boundaries.
- 2. Perimeter (P): Bean circumference is defined as the length of its border.
- 3. Major axis length (L): The distance between the ends of the longest line that can be drawn from a bean.
- 4. Minor axis length (l): The longest line that can be drawn from the bean while standing perpendicular to the main axis.
- 5. Aspect ratio (K): Defines the relationship between L and l.
- 6. Eccentricity (Ec): Eccentricity of the ellipse having the same moments as the region.
- 7. Convex area (C): Number of pixels in the smallest convex polygon that can contain the area of a bean seed.
- 8. Equivalent diameter (Ed): The diameter of a circle having the same area as a bean seed area.
- 9. Extent (Ex): The ratio of the pixels in the bounding box to the bean area.
- 10.Solidity (S): Also known as convexity. The ratio of the pixels in the convex shell to those found in beans.
- 11.Roundness (R): Calculated with the following formula: (4piA)/(P^2)
- 12.Compactness (CO): Measures the roundness of an object: Ed/L
- 13.ShapeFactor1 (SF1)
- 14.ShapeFactor2 (SF2)
- 15.ShapeFactor3 (SF3)
- 16.ShapeFactor4 (SF4)
- 17.Class (Y): Seker, Barbunya, Bombay, Cali, Dermosan, Horoz and Sira.

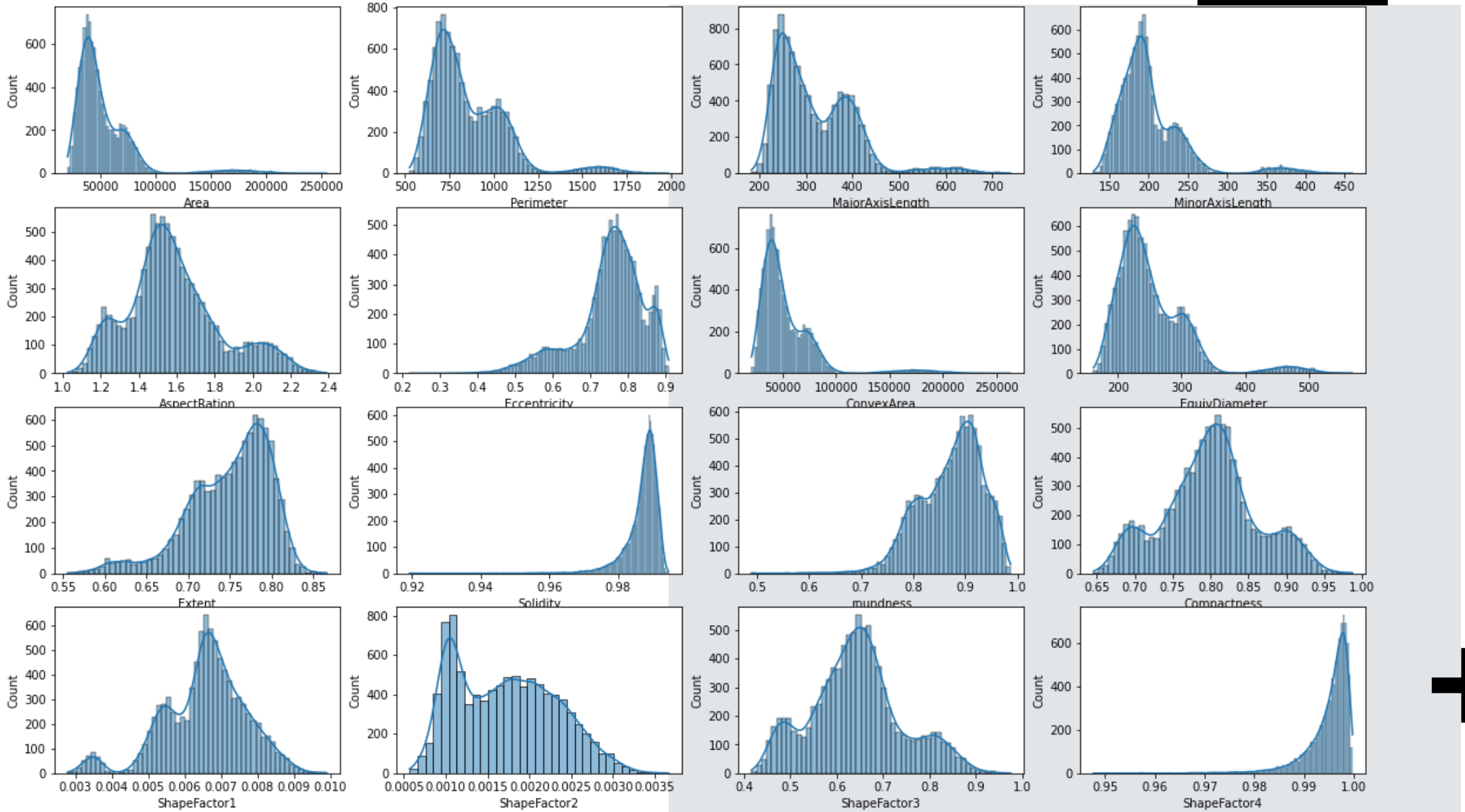
EDA



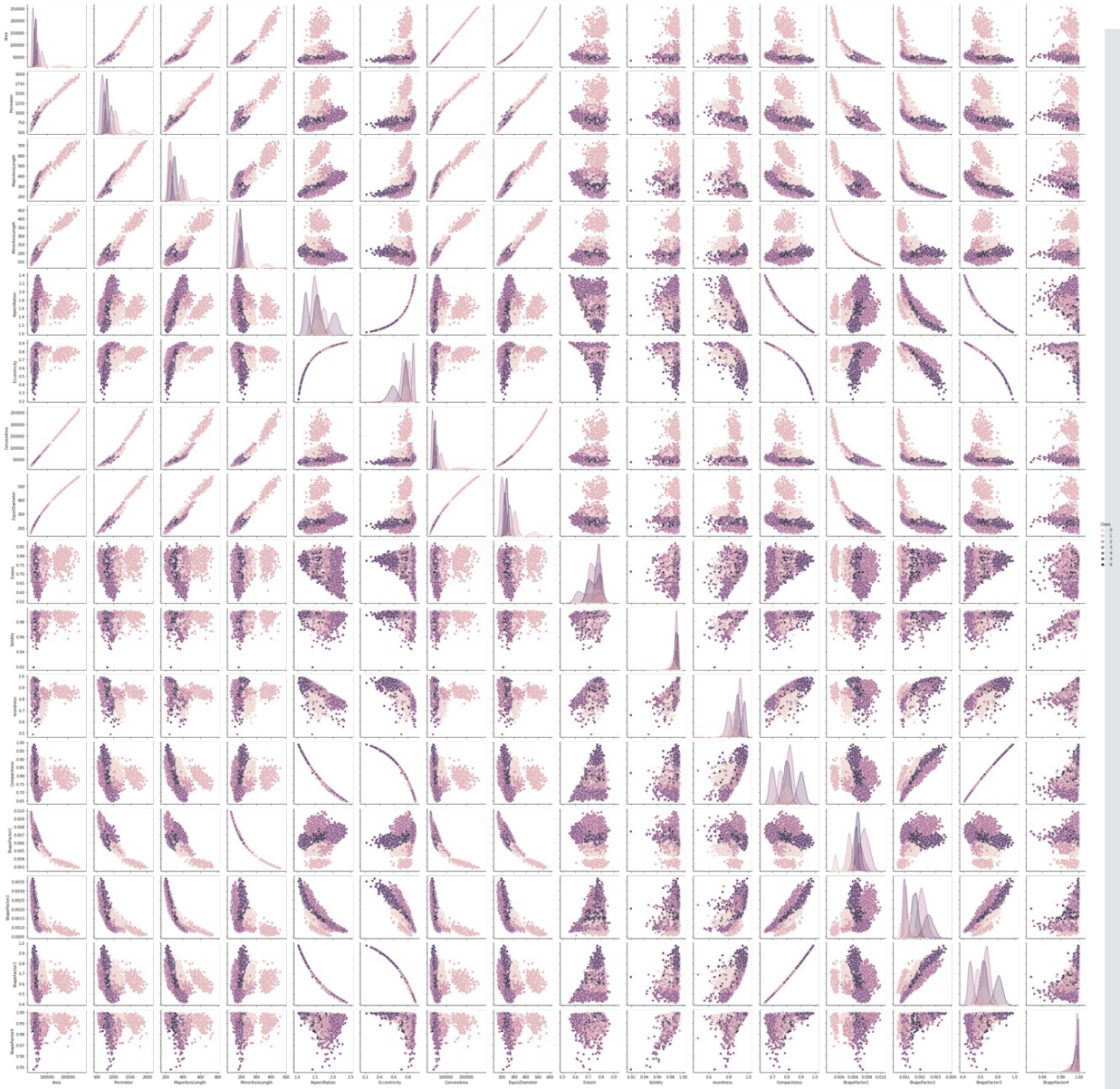
Target Variable Distribution

– According to the graph above, there are different counts for each class. With 3546 beans, Dermason is the most prevalent class. Bombay has 522 beans, making it the least common class. Two classes have a significant distinction that should be considered while developing a model.

Input Variables Distribution

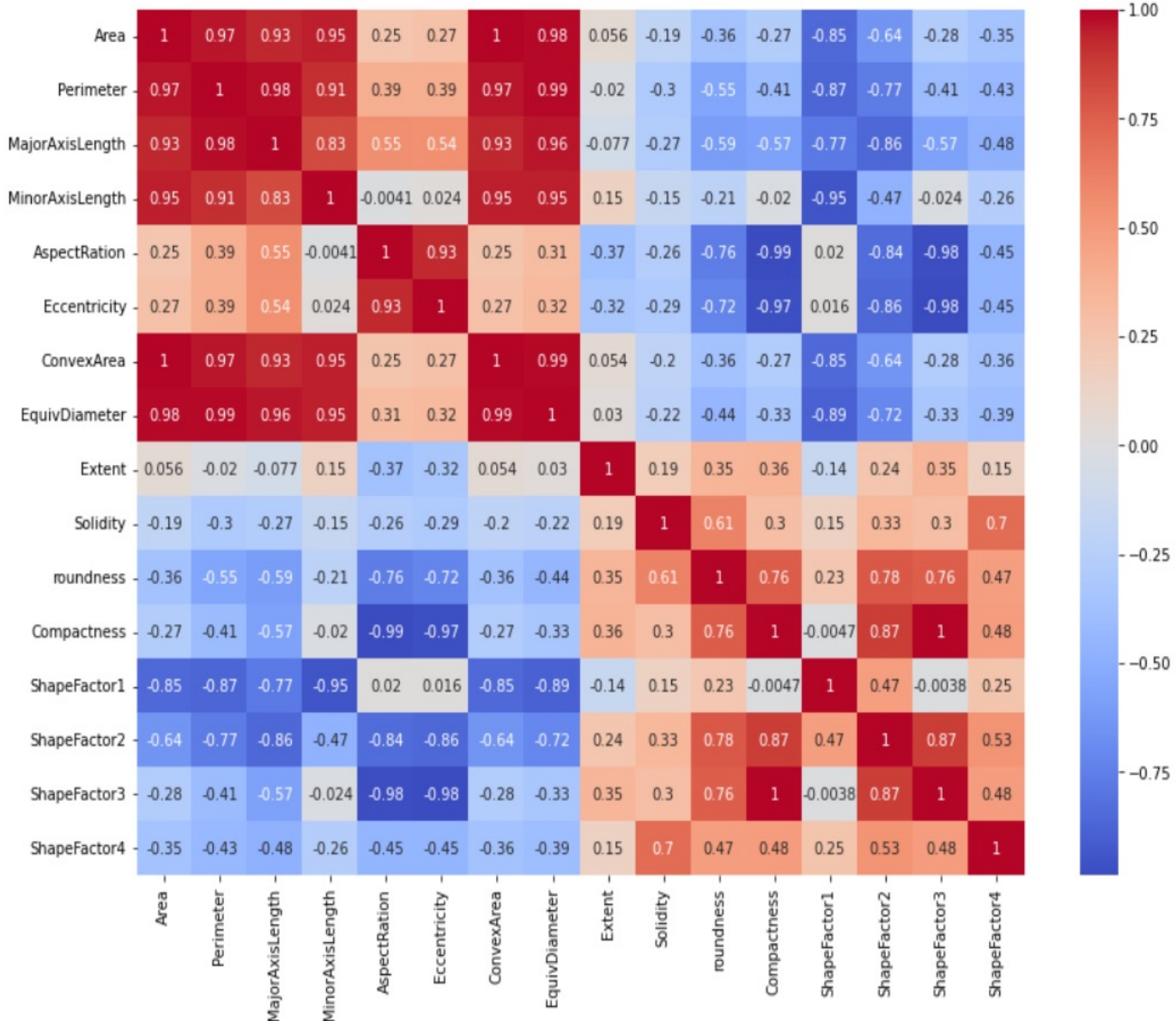


Pair plot



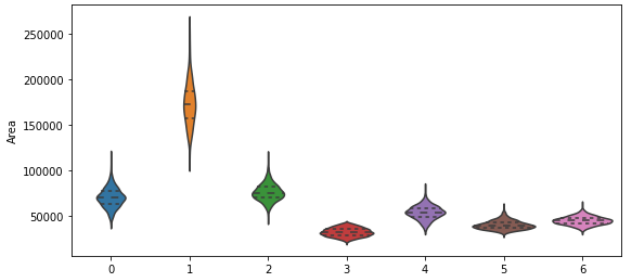
The pair plot shows us how the data is linearly separable when plotted against different dimensions

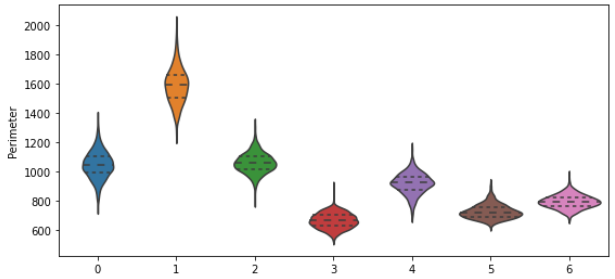
Correlation map of Input Variables

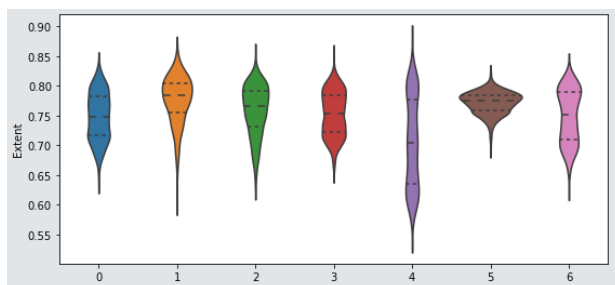


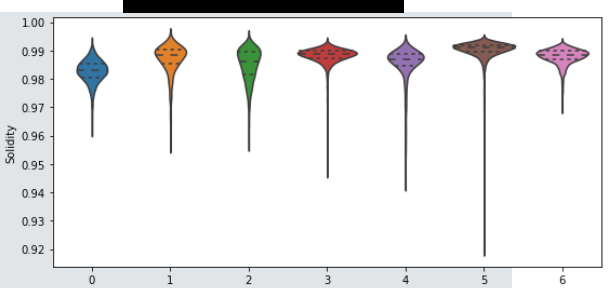
It's intriguing to see that certain characteristics show a high (linear) correlation with one another. For instance, among "compactness" and "shape factor 3," and between "convex area" and "area." This is expected given how closely connected the "area" and the "convex area" are. Although the calculation of "shape factor 3" and the other "shape factors" is not entirely obvious, it is reasonable to believe that the "compactness" of the beans had some sort of effect.

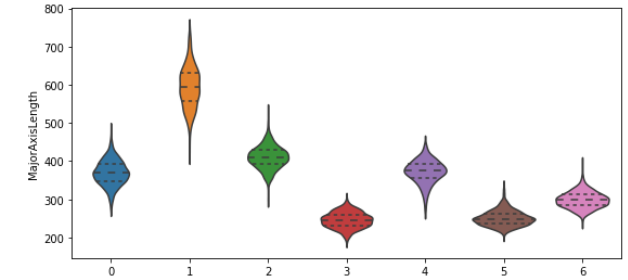
Violin Plot of Input Variables by class

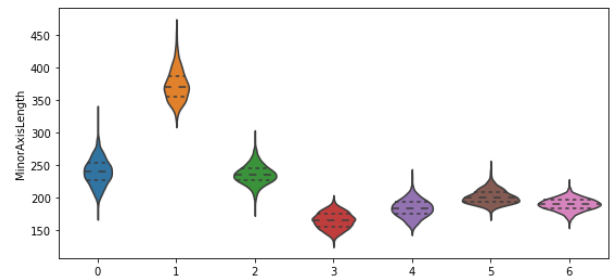


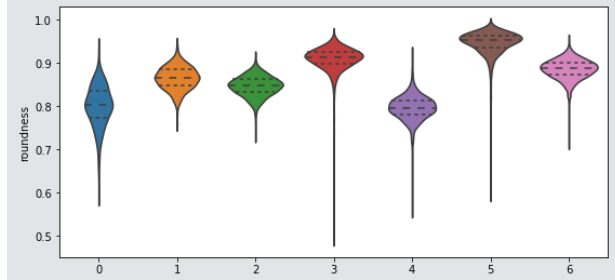


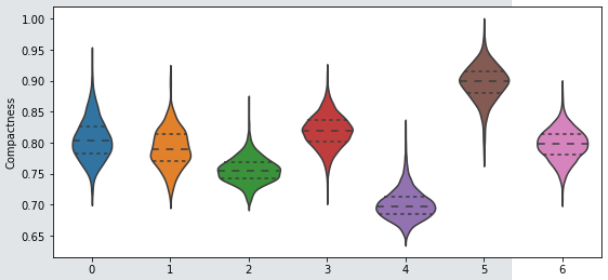


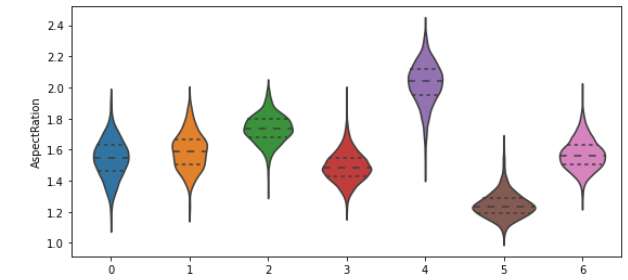


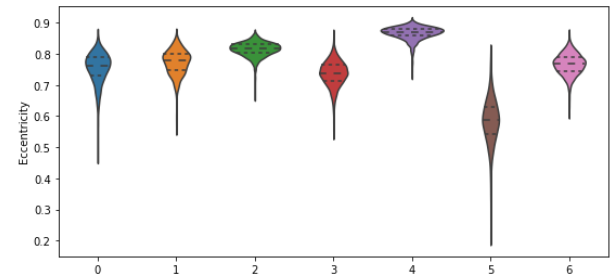


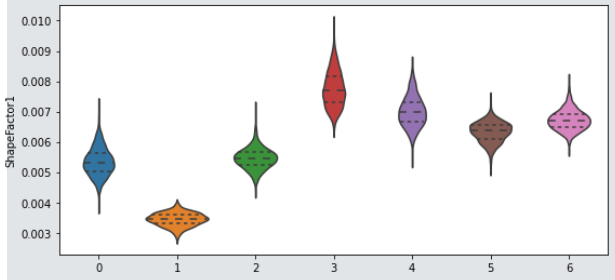


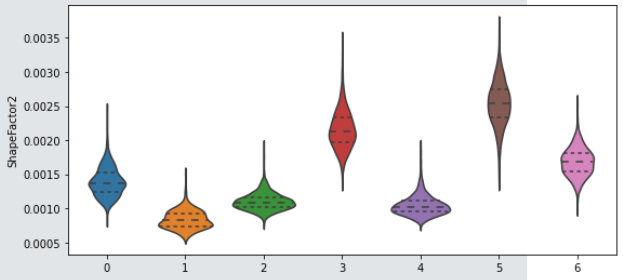


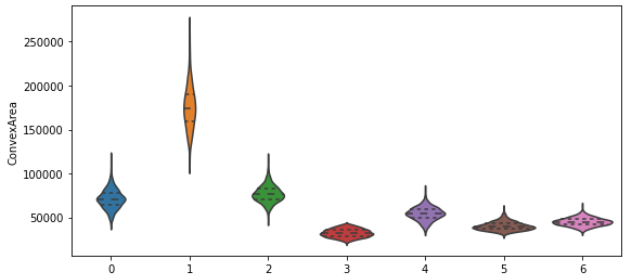


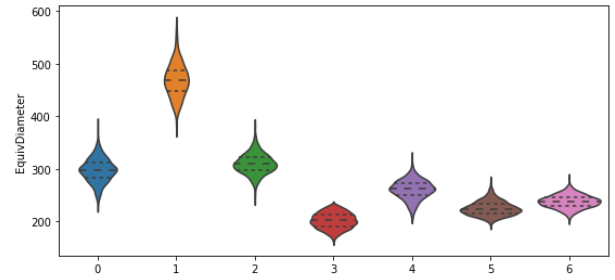


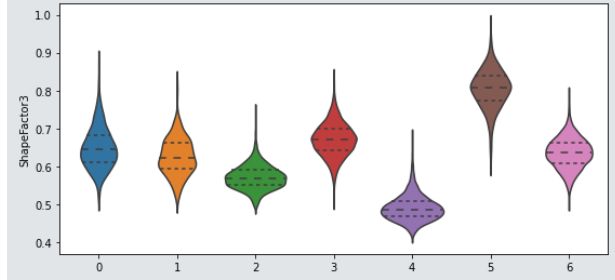


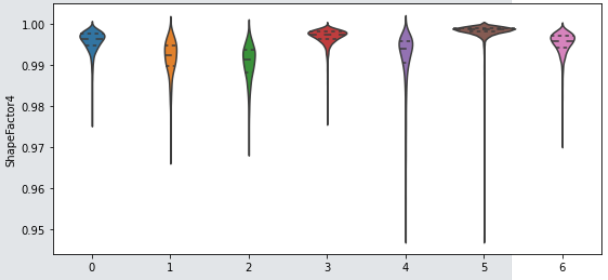












Data Preprocessing

- **Standardization:** A scaling technique where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation.

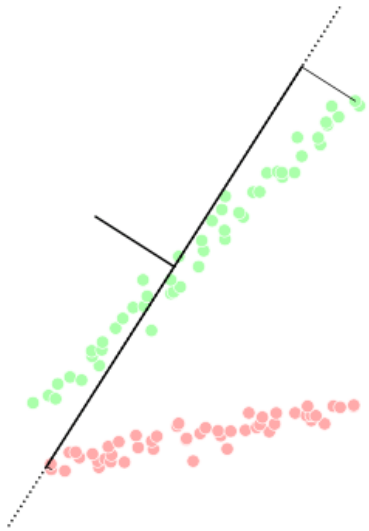
- **Label Encoding:** The target variable ‘class’ is encoded using label encoder. The names are converted to class numbers from 0 to 6 representing 7 classes

- **Test Train Split:** The dataset is split into 3 parts train, test, and valid with size 70%, 15%, 15% respectively.

PCA

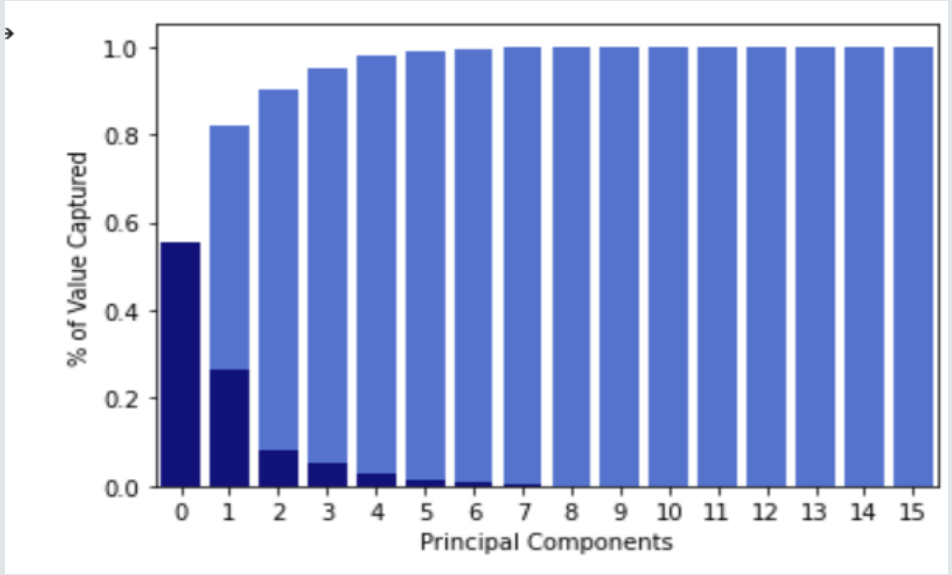
Principal component analysis (PCA) is a technique that transforms high-dimensions data into lower-dimensions while retaining as much information as possible.

Principal component analysis proceeds in this fashion, with each new component accounting for progressively smaller and smaller amounts of variance (this is why only the first few components are usually retained and interpreted).



PCA

– This is accomplished by producing fresh, uncorrelated variables that maximize variance one after the other. The components are created as linear mixtures or combinations of the basic variables, acting as new variables. The top four components account for more than 95% of the variance in the dataset, according to the principal component analysis shown below. Nearly all the volatility in the data can be explained by the first four components.



Explained Variance Captured

PC0 – 55%
PC1 – 26%
PC2 – 8%
PC3 – 5%

Final Dataset Variations

– **Baseline Dataset**

No PCA, Standardized, No Columns Dropped

- **PCA Dataset**

First 4 Principal components (95% explained variance)

Skipped Steps

- No outlier removal
- No Feature dropped in Baseline dataset

Models

Following are the different Machine Learning Models we Implemented For Classification:

1. Logistic regression

2. Neural Networks

3. Gaussian Naïve Bayes

Logistic Regression

- It's a form of statistical software that analyses the association between a dependent variable and one or more independent variables by estimating probabilities using a logistic regression equation.
- This analysis can help anticipate the likelihood of an event or a choice occurring. The output obtained in the case of logistic regression is always between (0 and 1), which is suitable for a binary classification task.
- The higher the value, the higher the probability that the current sample is classified as class= 1, and vice versa. It takes both continuous and discrete variables as input. The data fed into the model is upscaled on the minority class.

Results

- 7 Binary Classifiers.
- In case of multiple classes having output 1, the class with highest probability is chosen

– With PCA

Train Accuracy: 62.98%
Test Accuracy: 61.9%

	precision	recall	f1-score	support
0	0.57	0.17	0.26	197
1	0.32	1.00	0.49	83
2	0.66	0.37	0.48	240
3	0.68	0.94	0.79	534
4	0.60	0.94	0.73	287
5	0.71	0.97	0.82	296
6	0.00	0.00	0.00	405
accuracy			0.62	2042
macro avg	0.51	0.63	0.51	2042
weighted avg	0.51	0.62	0.53	2042

– Without PCA

Train Accuracy: 73.2%

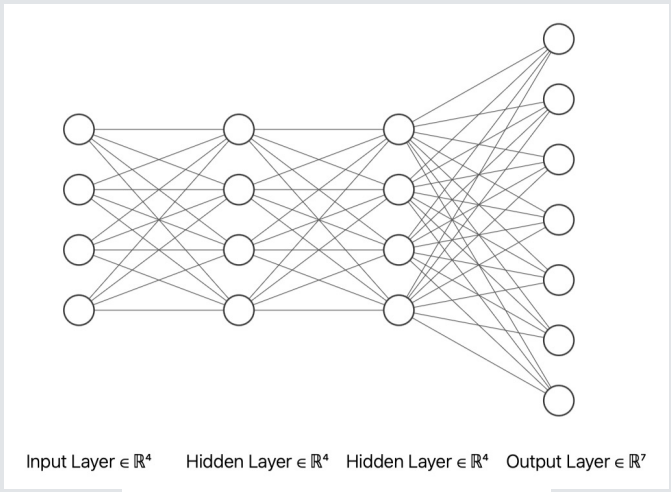
Test Accuracy: 73.51%

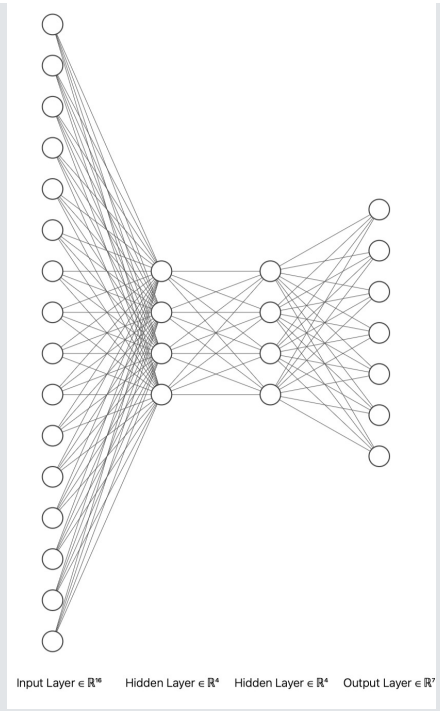
	precision	recall	f1-score	support
0	0.88	0.84	0.86	197
1	0.59	1.00	0.74	83
2	0.71	0.58	0.64	240
3	0.71	0.94	0.81	534
4	0.69	0.93	0.79	287
5	0.77	0.97	0.86	296
6	0.98	0.14	0.24	405
accuracy			0.74	2042
macro avg	0.76	0.77	0.71	2042
weighted avg	0.78	0.74	0.68	2042

Neural Network

- A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates.

- Here, we have used fully connected neural networks with 1 input layer, 2 hidden layers, and a final output layer when the softmax function is applied to give the target variable.





Results

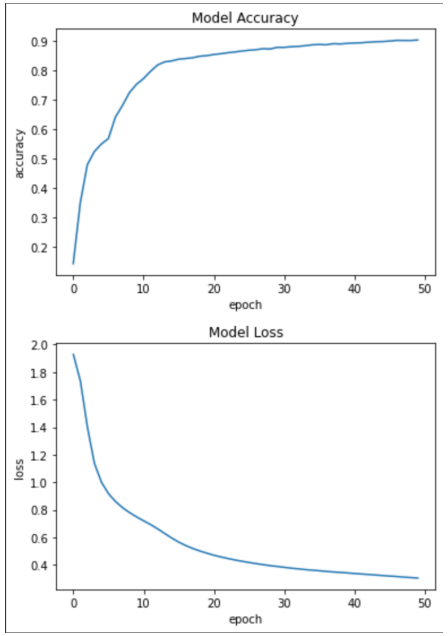
- 2 Hidden Layers with 4 nodes each.
- Trained for 50 epochs

– With PCA

Train Accuracy: 88.91%

Test Accuracy: 88.0%

	precision	recall	f1-score	support
0	0.70	0.68	0.69	197
1	1.00	1.00	1.00	83
2	0.77	0.78	0.77	240
3	0.92	0.91	0.92	534
4	0.96	0.95	0.96	287
5	0.96	0.93	0.94	296
6	0.83	0.89	0.86	405
accuracy			0.88	2042
macro avg	0.88	0.88	0.88	2042
weighted avg	0.88	0.88	0.88	2042

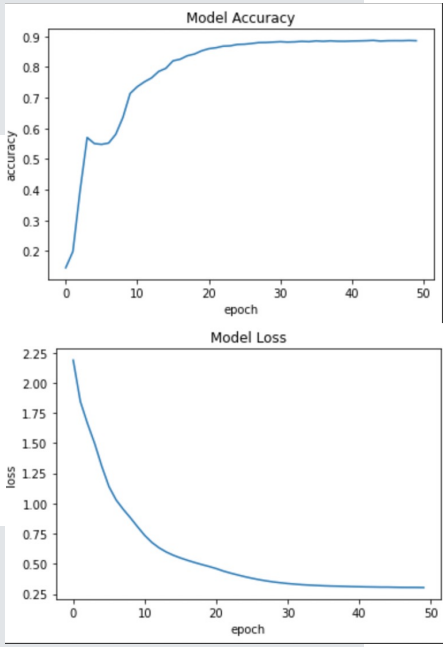


– Without PCA

Train Accuracy: 91.2%

Test Accuracy: 91.51%

	precision	recall	f1-score	support
0	0.88	0.87	0.88	197
1	0.99	0.99	0.99	83
2	0.91	0.89	0.90	240
3	0.94	0.89	0.91	534
4	0.95	0.94	0.95	287
5	0.95	0.94	0.95	296
6	0.83	0.91	0.87	405
accuracy			0.91	2042
macro avg	0.92	0.92	0.92	2042
weighted avg	0.91	0.91	0.91	2042



Gaussian Naïve Bayes

– When working with continuous data, an assumption often taken is that the continuous values associated with each class are distributed according to a normal (or Gaussian) distribution. The likelihood of the features is assumed to be

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Results

– With PCA

Train Accuracy: 86.23%

Test Accuracy: 87.27%

	precision	recall	f1-score	support
0	0.77	0.63	0.70	197
1	1.00	1.00	1.00	83
2	0.76	0.81	0.79	240
3	0.90	0.90	0.90	534
4	0.93	0.96	0.94	287
5	0.96	0.93	0.94	296
6	0.82	0.87	0.85	405
accuracy			0.87	2042
macro avg	0.88	0.87	0.87	2042
weighted avg	0.87	0.87	0.87	2042

– Without PCA

Train Accuracy: 89.55%

Test Accuracy: 89.81%

	precision	recall	f1-score	support
0	0.81	0.78	0.80	197
1	1.00	1.00	1.00	83
2	0.87	0.86	0.86	240
3	0.94	0.87	0.90	534
4	0.96	0.97	0.96	287
5	0.92	0.95	0.94	296
6	0.83	0.90	0.87	405
accuracy			0.90	2042
macro avg	0.90	0.91	0.90	2042
weighted avg	0.90	0.90	0.90	2042

Model Selection

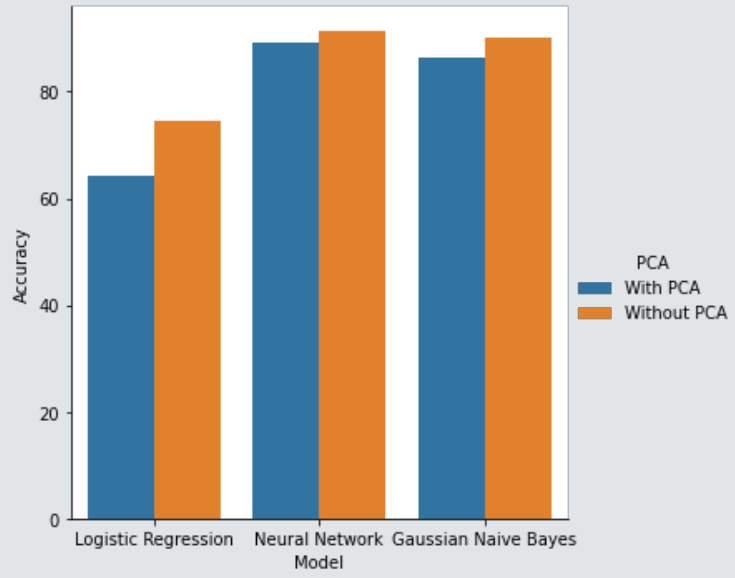
Neural Network model with standardized dataset (without PCA) has the best accuracy, f1 score, precision, and recall. The possible reasons behind it are:

- PCA doesn't takes into account the target variable in consideration and might drop features that are important that accounts for the drop of 2-10% in accuracy across all models.

- Imbalanced dataset might be the reason behind low f1-score, and precision scores.

- Even though our dataset is mostly distributed normally, few features had minute bi model bells

	Model	With PCA	Without PCA
0	Logistic Regression	64.10	74.39
1	Neural Network	89.23	91.38
2	Gaussian Naive Bayes	86.53	90.16

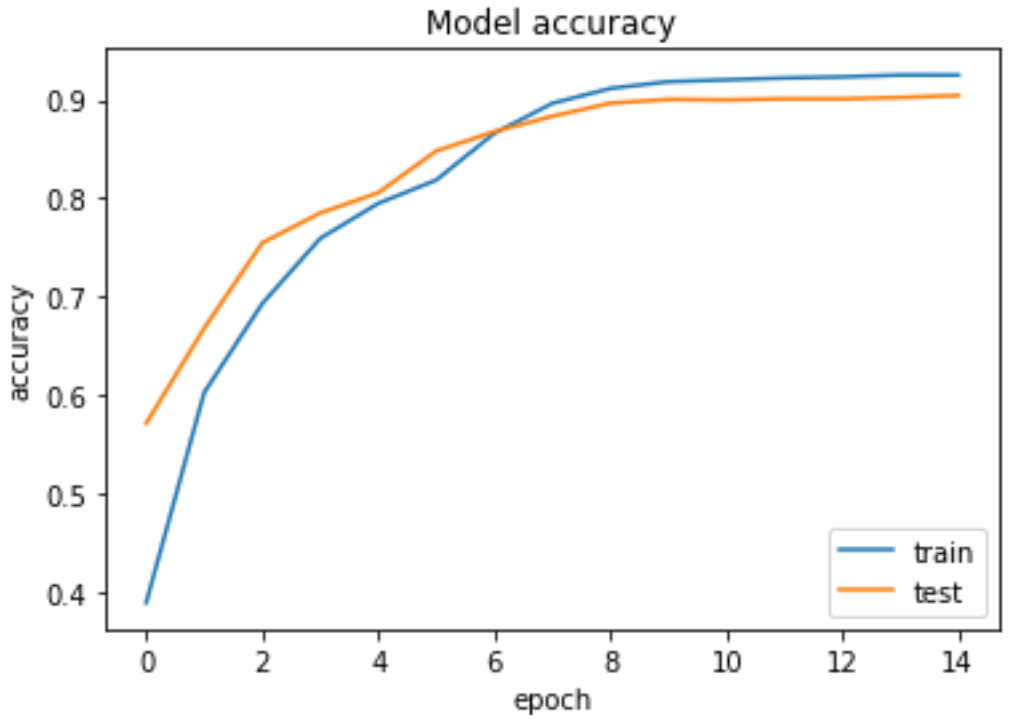


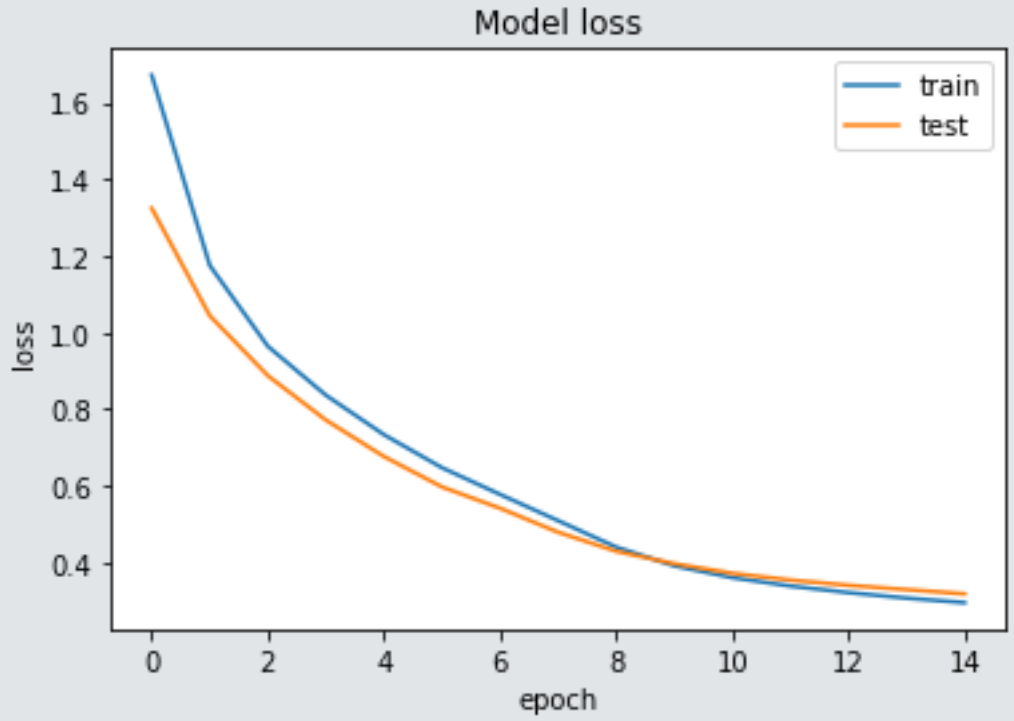
Model Tuning

Retrained the model with

– Balanced Dataset by Oversampling using SMOTE

– Reduced number of epochs from 50 to 15.





Metrics of Retrained Model

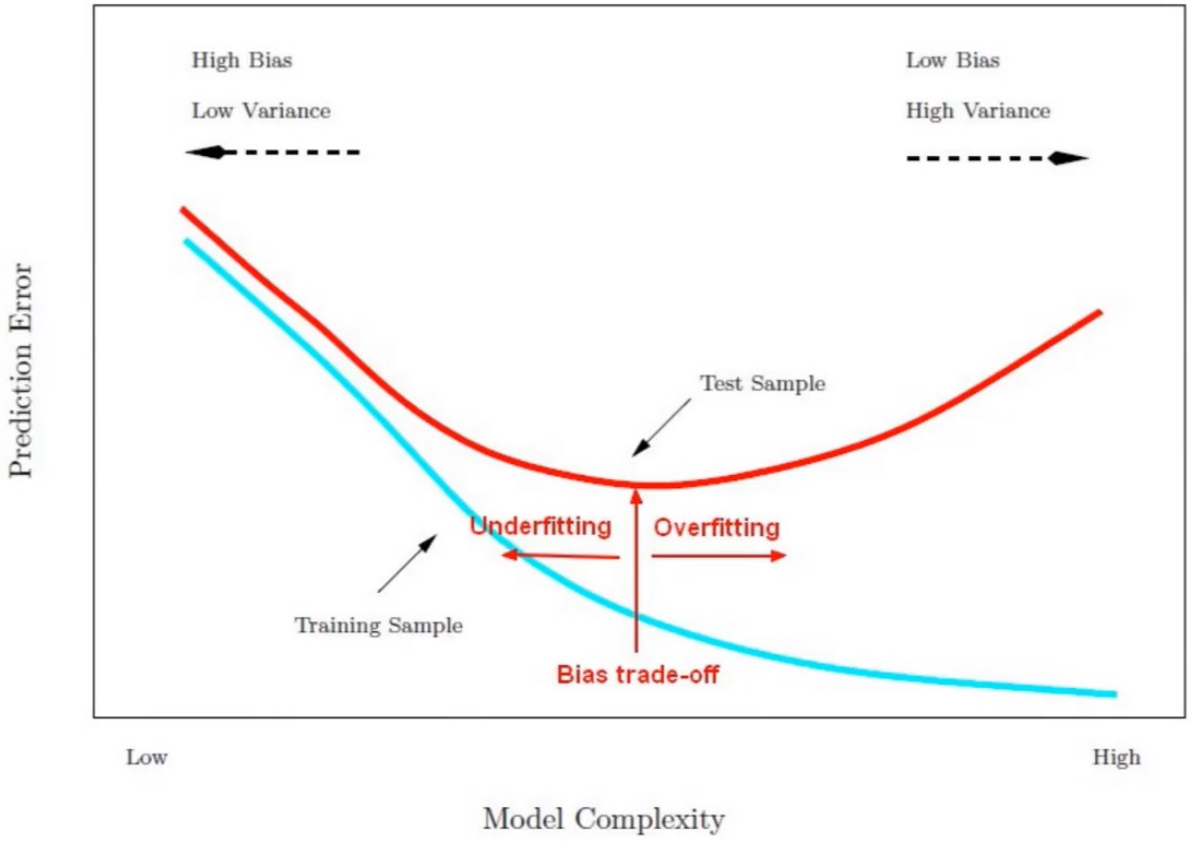
Training

	precision	recall	f1-score	support
0	0.94	0.92	0.93	2503
1	0.98	0.99	0.99	2503
2	0.93	0.93	0.93	2503
3	0.90	0.90	0.90	2503
4	0.94	0.94	0.94	2503
5	0.96	0.94	0.95	2503
6	0.84	0.85	0.84	2503
accuracy			0.93	17521
macro avg	0.93	0.93	0.93	17521
weighted avg	0.93	0.93	0.93	17521

Validation Set

	precision	recall	f1-score	support
0	0.92	0.92	0.92	198
1	0.95	1.00	0.97	78
2	0.91	0.92	0.91	239
3	0.93	0.88	0.90	509
4	0.93	0.93	0.93	301
5	0.95	0.94	0.95	323
6	0.82	0.88	0.85	394
accuracy			0.91	2042
macro avg	0.92	0.92	0.92	2042
weighted avg	0.91	0.91	0.91	2042

Fit of Model



Note: Ref graph, not generated from the model

(Using Zero-One Loss)

Train Error = 7.3%
Val Error = 9.2%