

MLP

```
In [1]: import pandas as pd
import numpy as np
from sklearn import metrics
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import PowerTransformer
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import classification_report, confusion_matrix
```

```
In [2]: df = pd.read_excel("Dry_Bean_Dataset.xlsx")
df = df.drop_duplicates()
X = df.iloc[:, :16]
y = df.iloc[:, 16:]
y = y.reset_index().drop(columns = "index")
scaler = StandardScaler()
X = pd.DataFrame(scaler.fit_transform(X), columns = X.columns)
pt = PowerTransformer(method = "yeo-johnson")
X = pd.DataFrame(pt.fit_transform(X), columns = X.columns)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 101)
```

```
In [5]: MLP = MLPClassifier(hidden_layer_sizes=(100,), random_state=101, early_stopping=True, verbose=0)
MLP.fit(X_train, np.ravel(y_train))
y_pred = MLP.predict(X_test)
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred, target_names=y.Class.unique()))
```

[243	0	9	0	0	4	4]				
[0	115	0	0	0	0	0]				
[13	0	319	0	1	2	4]				
[0	0	0	667	1	8	37]				
[0	0	5	3	337	0	3]				
[2	0	0	9	0	389	9]				
[0	0	4	50	14	7	450]]				
			precision	recall	f1-score	support				
	SEKER		0.94	0.93	0.94	260				
	BARBUNYA		1.00	1.00	1.00	115				
	BOMBAY		0.95	0.94	0.94	339				
	CALI		0.91	0.94	0.93	713				
	HOROZ		0.95	0.97	0.96	348				
	SIRA		0.95	0.95	0.95	409				
	DERMASON		0.89	0.86	0.87	525				
	accuracy				0.93	2709				
	macro avg		0.94	0.94	0.94	2709				
	weighted avg		0.93	0.93	0.93	2709				

```
In [6]: metrics.accuracy_score(y_test, y_pred)
```

Out[6]: 0.9302325581395349