

## ASSIGNMENT

- Identify your **problem statement** - **Predict the Chronic Kidney Disease (CKD)** based on the several parameters given
- **Basic info about the dataset** – Total number of **rows** is **399**, – Total number of **columns** is **25**
- **Pre-Processing Method** – Added **dummies to avoid nominal values** and also done some **standardization with sklearn preprocessing library**

Models are created with many algorithms with many **hyper tuning parameters**

With **RandomForestClassifier**, I got following result and **Weighted F1-score is 0.975** with hyper tuning parameters as

**bootstrap is True, criterion is gini, max\_depth is None, max\_features is sqrt, n\_estimators is 100**

Fitting 5 folds for each of 192 candidates, totalling 960 fits

```
[[49  1]
 [ 2 68]]
```

	precision	recall	f1-score	support
0.0	0.96	0.98	0.97	50
1.0	0.99	0.97	0.98	70
accuracy			0.97	120
macro avg	0.97	0.98	0.97	120
weighted avg	0.98	0.97	0.98	120

```
{'bootstrap': True, 'criterion': 'gini', 'max_depth': None, 'max_features': 'sqrt', 'n_estimators': 100}
```

Weighted F1-score: 0.9750338343186836

With **DecisionTreeClassifier**, I got following result and **Weighted F1-score is 0.95** with hyper tuning parameters as

**class\_weight is None, criterion is gini,max\_depth is None, max\_features is log2, splitter is random**

Fitting 5 folds for each of 216 candidates, totalling 1080 fits

```
[[49  1]
 [ 5 65]]
```

	precision	recall	f1-score	support
0.0	0.91	0.98	0.94	50
1.0	0.98	0.93	0.96	70
accuracy			0.95	120
macro avg	0.95	0.95	0.95	120
weighted avg	0.95	0.95	0.95	120

```
{'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': 'log2', 'splitter': 'random'}
```

Weighted F1-score: 0.9502262443438915

With **KNeighborsClassifier**, I got following result and **Weighted F1-score is 0.73** with hyper tuning parameters as

**Algorithm is auto, metric is manhattan, n\_jobs is -1, n\_neighbors is 7**

Fitting 5 folds for each of 360 candidates, totalling 1800 fits

Collapse Output .k\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy\ma\core.py:2820: RuntimeWarning: invalid value encountered in cast  
\_data = np.array(data, dtype=dtype, copy=copy,

```
[[39 11]
 [22 48]]
```

	precision	recall	f1-score	support
0.0	0.64	0.78	0.70	50
1.0	0.81	0.69	0.74	70
accuracy			0.72	120
macro avg	0.73	0.73	0.72	120
weighted avg	0.74	0.72	0.73	120

```
{'algorithm': 'auto', 'metric': 'manhattan', 'n_jobs': -1, 'n_neighbors': 7}
```

Weighted F1-score: 0.7269013199245757

With **Support Vector Machine**, I got following result and **Weighted F1-score is 0.975** with hyper tuning parameters as

**C is 1, gamma is scale, kernel is linear**

Fitting 5 folds for each of 32 candidates, totalling 160 fits

```
[[49  1]
 [ 2 68]]
```

	precision	recall	f1-score	support
0.0	0.96	0.98	0.97	50
1.0	0.99	0.97	0.98	70
accuracy			0.97	120
macro avg	0.97	0.98	0.97	120
weighted avg	0.98	0.97	0.98	120

```
{'C': 1, 'gamma': 'scale', 'kernel': 'linear'}
```

Weighted F1-score: 0.9750338343186836

With **Logistic Regression**, I got following result and **Weighted F1-score is 0.975** with hyper tuning parameters as

**Penalty is l2, solver is liblinear**

Fitting 5 folds for each of 6 candidates, totalling 30 fits

```
[[49  1]
 [ 2 68]]
```

	precision	recall	f1-score	support
0.0	0.96	0.98	0.97	50
1.0	0.99	0.97	0.98	70
accuracy			0.97	120
macro avg	0.97	0.98	0.97	120
weighted avg	0.98	0.97	0.98	120

```
{'penalty': 'l2', 'solver': 'liblinear'}
```

Weighted F1-score: 0.9750338343186836

With **Gaussian Naive Bayes**, I got following result and **Weighted F1-score is 0.93** with hyper tuning parameters as

**var\_smoothing is 2.848035868435805e-09**

```
Fitting 5 folds for each of 100 candidates, totalling 500 fits
```

```
[[49  1]
 [ 7 63]]
```

	precision	recall	f1-score	support
0.0	0.88	0.98	0.92	50
1.0	0.98	0.90	0.94	70
accuracy			0.93	120
macro avg	0.93	0.94	0.93	120
weighted avg	0.94	0.93	0.93	120

```
{'var_smoothing': 2.848035868435805e-09}
```

```
Weighted F1-score: 0.9337275884727307
```

With **Multinomial Naive Bayes**, I got following result and **Weighted F1-score is 0.93** with hyper tuning parameters as

**Alpha is 0.1, fit\_prior is True**

```
Fitting 5 folds for each of 20 candidates, totalling 100 fits
```

```
[[50  0]
 [15 55]]
```

	precision	recall	f1-score	support
0.0	0.77	1.00	0.87	50
1.0	1.00	0.79	0.88	70
accuracy			0.88	120
macro avg	0.88	0.89	0.87	120
weighted avg	0.90	0.88	0.88	120

```
{'alpha': 0.1, 'fit_prior': True}
```

```
Weighted F1-score: 0.8756521739130434
```

With **Bernoulli Naive Bayes**, I got following result and **Weighted F1-score is 0.97** with hyper tuning parameters as

**Alpha is 0.1, fit\_prior is True, binarize is 0.0**

Fitting 5 folds for each of 60 candidates, totalling 300 fits

```
[[49  1]
 [ 3 67]]
```

	precision	recall	f1-score	support
0.0	0.94	0.98	0.96	50
1.0	0.99	0.96	0.97	70
accuracy			0.97	120
macro avg	0.96	0.97	0.97	120
weighted avg	0.97	0.97	0.97	120

```
{'alpha': 0.1, 'binarize': 0.0, 'fit_prior': True}
```

Weighted F1-score: 0.9667519181585678

With **Complement Naive Bayes**, I got following result and **Weighted F1-score is 0.875** with hyper tuning parameters as

**Alpha is 0.1, fit\_norm is False**

Fitting 5 folds for each of 20 candidates, totalling 100 fits

```
[[50  0]
 [15 55]]
```

	precision	recall	f1-score	support
0.0	0.77	1.00	0.87	50
1.0	1.00	0.79	0.88	70
accuracy			0.88	120
macro avg	0.88	0.89	0.87	120
weighted avg	0.90	0.88	0.88	120

```
{'alpha': 0.1, 'norm': False}
```

Weighted F1-score: 0.8756521739130434

According to this Random Forest Classifier have most Weighted F1 score with most fits

So we can choose the model with **Random Forest Classifier** with hyper parameters as **bootstrap is True, criterion is gini, max\_depth is None, max\_features is sqrt, n\_estimators is 100**