T.R. Dhuwarakesh

Chronic Kidney Disease (CKD) – Assignment – Classification – ML

- **Problem Statement**

  We have 3 stages in problem statement identification.

  1. Stage 1: **Domain Selection**
  2. Stage 2: **Learning Selection**
  3. Stage 3: **Classification/Regression**

  Stage 1 (Domain Selection): Machine Learning the datasets in excel are mostly having numeric values has input.

  Stage 2 (Learning Selection): Supervised Learning

  - Requirements are clear.
  - Input and Output data are well defined.

  Stage 3 (Classification/Regression): It is a "**Classification**". All the output data are in "**Categorical**" values.

- **Basic information about the dataset**

  Total Columns – 25

  Total Rows – 399

  Input Values – 24 columns

  Output Value – 1 column

- **About Pre-Processing Method (Categorical Column Data)**

  Nominal Data – We have inputs columns with string values. So, we need to change the string data into numeric values (Binary digits – Nominal Data (One Hot Encoding)) for further processing.

- **Developing the "Best Model" below using the Classification – Machine Learning Algorithms. All the models with best values using Data Pre-Processing, GridSearchCV & (Hyper Tunning Parameters).**

- **All the research values are below documented.**

Classification Algorithms – Machine Learning Process with best values:

1. **SVM – Support Vector Machine – Classification – Pre-Processing & GridSearchCV:**

   confusion_matrix & classification_report values are below:

```
[[73  2]
 [ 0 45]]
```

	precision	recall	f1-score	support
0	1.000000	0.973333	0.986486	75.000000
1	0.957447	1.000000	0.978261	45.000000
accuracy	0.983333	0.983333	0.983333	0.983333
macro avg	0.978723	0.986667	0.982374	120.000000
weighted avg	0.984043	0.983333	0.983402	120.000000

   The best **f_score** value from parameter **{'C': 10, 'gamma': 'scale', 'kernel': 'sigmoid'}: 0.982373678025852**

   roc_auc_score à np.float64(0.9997037037037038)

2. **DT – Decision Tree – Classification – Pre-Processing & GridSearchCV:**

   confusion_matrix & classification_report values are below:

```
[[74  1]
 [ 1 44]]
```

	precision	recall	f1-score	support
0	1.0	1.0	1.0	75.0
1	1.0	1.0	1.0	45.0
accuracy	1.0	1.0	1.0	1.0
macro avg	1.0	1.0	1.0	120.0
weighted avg	1.0	1.0	1.0	120.0

The best **f_score** value from parameter **{'criterion': 'entropy', 'max_features': 'sqrt', 'splitter': 'best'}: 0.9833333333333333**

roc_auc_score à np.float64(0.9822222222222222)

3. **RF – Random Forest – Classification – Pre-Processing & GridSearchCV:**

confusion_matrix & classification_report values are below:

```
[[74  1]
 [ 0 45]]
```

	precision	recall	f1-score	support
0	1.000000	0.986667	0.993289	75.000000
1	0.978261	1.000000	0.989011	45.000000
accuracy	0.991667	0.991667	0.991667	0.991667
macro avg	0.989130	0.993333	0.991150	120.000000
weighted avg	0.991848	0.991667	0.991684	120.000000

The best **f_score** value from parameter **{'criterion': 'log_loss', 'max_features': 'log2', 'n_estimators': 100}: 0.9911497898075079**

roc_auc_score à np.float64(0.9997037037037038)

4. **Logistic Regression – Classification – Pre-Processing & GridSearchCV:**

confusion_matrix & classification_report values are below:

```
[[74  1]
 [ 0 45]]
```

	precision	recall	f1-score	support
0	1.000000	0.986667	0.993289	75.000000
1	0.978261	1.000000	0.989011	45.000000
accuracy	0.991667	0.991667	0.991667	0.991667
macro avg	0.989130	0.993333	0.991150	120.000000
weighted avg	0.991848	0.991667	0.991684	120.000000

The best **f_score** value from the parameter **{'penalty': 'l2', 'solver': 'lbfgs'}: 0.9916844900066377**

roc_auc_score à np.float64(1.0)

5. **K Nearest Neighbors (KNN) – Classification – Pre-Processing & GridSearchCV:**

   confusion_matrix & classification_report values are below:

```
[[69  6]
 [ 0 45]]
```

	precision	recall	f1-score	support
0	1.000000	0.92	0.958333	75.00
1	0.882353	1.00	0.937500	45.00
accuracy	0.950000	0.95	0.950000	0.95
macro avg	0.941176	0.96	0.947917	120.00
weighted avg	0.955882	0.95	0.950521	120.00

The best **f_score** value from parameter **{'algorithm': 'auto', 'metric': 'minkowski', 'n_neighbors': 5, 'p': 2, 'weights': 'distance'}: 0.9505208333333334**

roc_auc_score à np.float64(1.0)

6. **Naive_Bayes (NB) – BernoulliNB – Classification – Pre-Processing & GridSearchCV:**

   confusion_matrix & classification_report values are below:

```
[[72  3]
 [ 0 45]]
```

	precision	recall	f1-score	support
0	1.000000	0.960	0.979592	75.000
1	0.937500	1.000	0.967742	45.000
accuracy	0.975000	0.975	0.975000	0.975
macro avg	0.968750	0.980	0.973667	120.000
weighted avg	0.976562	0.975	0.975148	120.000

The best **f_score** value from parameter **{}: 0.9736668861092824**

roc_auc_score à np.float64(1.0)

➢ Final Model – Take Away:

Based on my research & development process, I got the best model using "Logistic Regression" Classification Algorithm – Machine Learning.

The confusion_matrix output is well defined and constant while running the process several times and gave me the same output without any changes in every run.

The classification_report with 0.99 accuracy.

The f1_score value is also 0.99 f1_score.

The roc_auc_score value is 1.0 score.