ML: Classification – Assignment

1. To predict the CKC based on given parameters.

Stages: Machine Learning -> Supervised Learning as the input and output are clear -> Classification because the expected output is yes/no.

2. Total no of Rows: 399 Columns -25
3. Rbc, pc, pcc, ba, htm, da, cad, appet, pe, ane and classification columns are string so we need to convert them as numeric.
4. Finding best algorithm using classification in Machine Learning:

I. Logistic Regression –

```
The f1_macro value for best parameter{'penalty': 'l2', 'solver': 'newton-cg'}: 0.9916844900066377
```

```
print("The confusion matrix:\n",cm)
```

```
The confusion matrix:
 [[45  0]
 [ 1 74]]
```

```
print("The report:\n",clf_report)
```

```
The report:
              precision    recall  f1-score   support

           0       0.98      1.00      0.99        45
           1       1.00      0.99      0.99        75

    accuracy                           0.99       120
   macro avg       0.99      0.99      0.99       120
weighted avg       0.99      0.99      0.99       120
```

II. Support Vector Machine:

```
The f1_macro value for best parameter{'C': 10, 'gamma': 'scale', 'kernel': 'sigmoid'}: 0.9834018801410106
```

```
print("The confusion matrix:\n",cm)
```

```
The confusion matrix:
 [[45  0]
 [ 2 73]]
```

```
print("The report:\n",clf_report)
```

```
The report:
              precision    recall  f1-score   support

           0       0.96      1.00      0.98        45
           1       1.00      0.97      0.99        75

    accuracy                           0.98       120
   macro avg       0.98      0.99      0.98       120
weighted avg       0.98      0.98      0.98       120
```

III.   Decision Tree:

```
The f1_macro value for best parameter{'criterion': 'entropy', 'max_features': 'log2', 'splitter': 'random'}: 0.975053470019913
```

```
print("The confusion matrix:\n",cm)
```

```
The confusion matrix:
 [[44  1]
 [ 2 73]]
```

```
print("The report:\n",clf_report)
```

```
The report:
              precision    recall  f1-score   support

           0       0.96      0.98      0.97        45
           1       0.99      0.97      0.98        75

    accuracy                           0.97       120
   macro avg       0.97      0.98      0.97       120
weighted avg       0.98      0.97      0.98       120
```

IV.   Random Forest:

```
Best params:
 {'criterion': 'gini', 'max_features': 'log2', 'n_estimators': 100}
```

```
print("The confusion matrix:\n",cm)
```

```
The confusion matrix:
 [[44  1]
 [ 1 74]]
```

```
print("The report:\n",clf_report)
```

```
The report:
              precision    recall  f1-score   support

           0       0.98      0.98      0.98        45
           1       0.99      0.99      0.99        75

    accuracy                           0.98       120
   macro avg       0.98      0.98      0.98       120
weighted avg       0.98      0.98      0.98       120
```

5. Uploaded code in repository.
6. Th best Model is Logistic Regression with 0.99 accuracy.