

ASSIGNMENT ON GRID CLASSIFICATION

PROBLEM STATEMENT:

1. Predict the Chronic Kidney Disease
2. Provided dataset: 399 rows 29 Columns [CKD.xls]
3. Pre-processing Method: (Below mentioned Columns)

```
dataset=pd.get_dummies(dataset,dtype=int,drop_first=True)
```

pc_normal pcc_present ba_present htn_yes dm_yes cad_yes appet_yes pe_yes ane_yes classification_yes

USED CLASSIFICATION MODEL:

- Logistic Regression
- Support Vector Machine
- Decision Tree
- Random Forest

LOGISTIC REGRESSION

ALGORITHM:

- Precision Score: The best parameters given the 99%

```
from sklearn.metrics import precision_score
precision=precision_score(y_test,grid_predictions,average='micro')
print("The Precision value for best parameter {}".format(grid.best_params_),precision)

The Precision value for best parameter {'penalty': 'l2', 'solver': 'newton-cg'}: 0.9924812030075187

print("The confusion Matrix:\n",cm)

The confusion Matrix:
[[51  0]
 [ 1 81]]
```

- F1_Score and ROC_AUC_Score: The best parameters given the 99%

Note: roc_auc_score will be reviewed

```
from sklearn.metrics import f1_score
f1_macro=f1_score(y_test,grid_predictions,average='weighted')
print("The f1_macro value for best parameter {}".format(grid.best_params_),f1_macro)

The f1_macro value for best parameter {'penalty': 'l2', 'solver': 'newton-cg'}: 0.9924946382275899

from sklearn.metrics import roc_auc_score
roc_auc_score(y_test,grid.predict_proba(X_test)[:,:1])

1.0
```

CLASSIFICATION REPORT: (CLF)

LOGISTIC REGRESSION

The report:					
	precision	recall	f1-score	support	
0	0.98	1.00	0.99	51	
1	1.00	0.99	0.99	82	
accuracy			0.99	133	
macro avg	0.99	0.99	0.99	133	
weighted avg	0.99	0.99	0.99	133	

SUPPORT VECTOR MACHINE CLASSIFICATION

ALGORITHM:

- F1_Score: The best parameters given the 99%

Note: roc_auc_score will NOT be attributed

```
from sklearn.metrics import f1_score
f1_macro=f1_score(y_test,grid_predictions,average='weighted')
print("The f1_macro value for best parameter {}".format(grid.best_params_),f1_macro)
```

The f1_macro value for best parameter {'C': 10, 'gamma': 'auto', 'kernel': 'sigmoid'}: 0.9924946382275899

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

The confusion Matrix:

```
[[51  0]
 [ 1 81]]
```

CLASSIFICATION REPORT: (CLF)

SUPPORT VECTOR MACHINE

Note: r2_score value evaluated

The report:					
	precision	recall	f1-score	support	
0	0.98	1.00	0.99	51	
1	1.00	0.99	0.99	82	
accuracy			0.99	133	
macro avg	0.99	0.99	0.99	133	
weighted avg	0.99	0.99	0.99	133	

from sklearn.metrics import r2_score					
r2_score(y_test,grid.predict(X_test))					
0.9681970349115256					

DECISION TREE CLASSIFICATION

ALGORITHM:

- F1_Score and ROC_AUC_Score: The best parameters given the 100%

Note: Confusion Matrix will be reviewed

```
from sklearn.metrics import f1_score
f1_macro=f1_score(y_test,grid_predictions,average='weighted')
print("The f1_macro value for best parameter {}".format(grid.best_params_),f1_macro)

The f1_macro value for best parameter {'criterion': 'entropy', 'max_features': 'sqrt', 'splitter': 'random'}: 1.0

print("The confusion Matrix:\n",cm)

The confusion Matrix:
[[51  0]
 [ 0 82]]
```

CLASSIFICATION REPORT: (CLF)

DECISION TREE CLASSIFICATION

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

     0       1.00      1.00      1.00        51
     1       1.00      1.00      1.00        82

 accuracy          1.00          1.00          1.00       133
 macro avg          1.00          1.00          1.00       133
weighted avg          1.00          1.00          1.00       133

from sklearn.metrics import roc_auc_score
roc_auc_score(y_test,grid.predict_proba(X_test)[:,:1])

1.0
```

RANDOM FOREST CLASSIFICATION

ALGORITHM:

- F1_Score and ROC_AUC_Score: The best parameters given the 99% & 98% respectively

Note: Confusion Matrix will be reviewed

```
from sklearn.metrics import f1_score
f1_macro=f1_score(y_test,grid_predictions,average='weighted')
print("The f1_macro value for best parameter {}".format(grid.best_params_),f1_macro)

The f1_macro value for best parameter {'criterion': 'entropy', 'max_features': 'log2', 'n_estimators': 100}: 0.9924946382275899

print("The confusion Matrix:\n",cm)

The confusion Matrix:
[[51  0]
 [ 1 81]]
```

CLASSIFICATION REPORT: (CLF)

RANDOM FOREST CLASSIFICATION

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

         0       0.98        1.00        0.99         51
         1       1.00        0.99        0.99         82

 accuracy          0.99
 macro avg         0.99
weighted avg         0.99

from sklearn.metrics import roc_auc_score
roc_auc_score(y_test,grid.predict_proba(X_test)[:,:1])

C:\Users\ANIRUDH\anaconda3\Lib\site-packages\sklearn\base.py:45
was fitted without feature names
warnings.warn(
0.9866092778574844
```