

Classification Assignment

1.Problem Statement:-

Predict Chronic Kidney Disease (CKD)

2.Data set basic info:-

Total no.of Rows = 400

Total no.of Columns = 25

3.Pre-processing method:-

Out of twenty five, twelve columns have nominal data(categorical) data. So, it needs to be converted to numerical data by using “get_dummies()” function

4. Model Development:-

Decision Tree, Random Forest and Logistic Regression algorithms have been used to develop a model with Confusion Matrix.

5.Research values of each algorithm:-

a) Decision Tree Confusion Matrix:-

```
param_grid = {'criterion':['gini', 'entropy'], 'max_features':['auto', 'sqrt', 'log2'], 'splitter':['best', 'random']}  
grid = GridSearchCV(DecisionTreeClassifier(), param_grid, refit=True, verbose=3, n_jobs=-1, scoring='f1_weighted')
```

Jupyter Classification Assignment-DT Model Creation Last Checkpoint: 3 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use 'zero_division' parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use 'zero_division' parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use 'zero_division' parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```

In [15]: `print(clf_report)`

	precision	recall	f1-score	support
0	0.00	0.00	0.00	51
1	0.62	1.00	0.76	82
accuracy			0.62	133
macro avg	0.31	0.50	0.38	133
weighted avg	0.38	0.62	0.47	133

In [16]: `#To know the f1_score value individually
from sklearn.metrics import f1_score
f1 = f1_score(Y_test,y_pred,average='weighted')
print("The f1_score value for best parameter {}:".format(grid.best_params_),f1)`

The f1_score value for best parameter {'criterion': 'gini', 'max_features': 'sqrt', 'splitter': 'random'}: 0.47029200909249863

Activate Windows
Go to Settings to activate

b)Random Forest Confusion Matrix:-

```
param_grid = {'criterion':['gini', 'entropy'],'max_features':['auto', 'sqrt', 'log2'], 'n_estimators':[10, 100]}
grid = GridSearchCV(RandomForestClassifier(), param_grid, refit=True, verbose=3, n_jobs=-1,
scoring='f1')
```

Jupyter Classification Assignment-RF Model Creation Last Checkpoint: 3 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use 'zero_division' parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use 'zero_division' parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use 'zero_division' parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```

In [40]: `from sklearn.metrics import classification_report
clf_report=classification_report(Y_test,y_pred)`

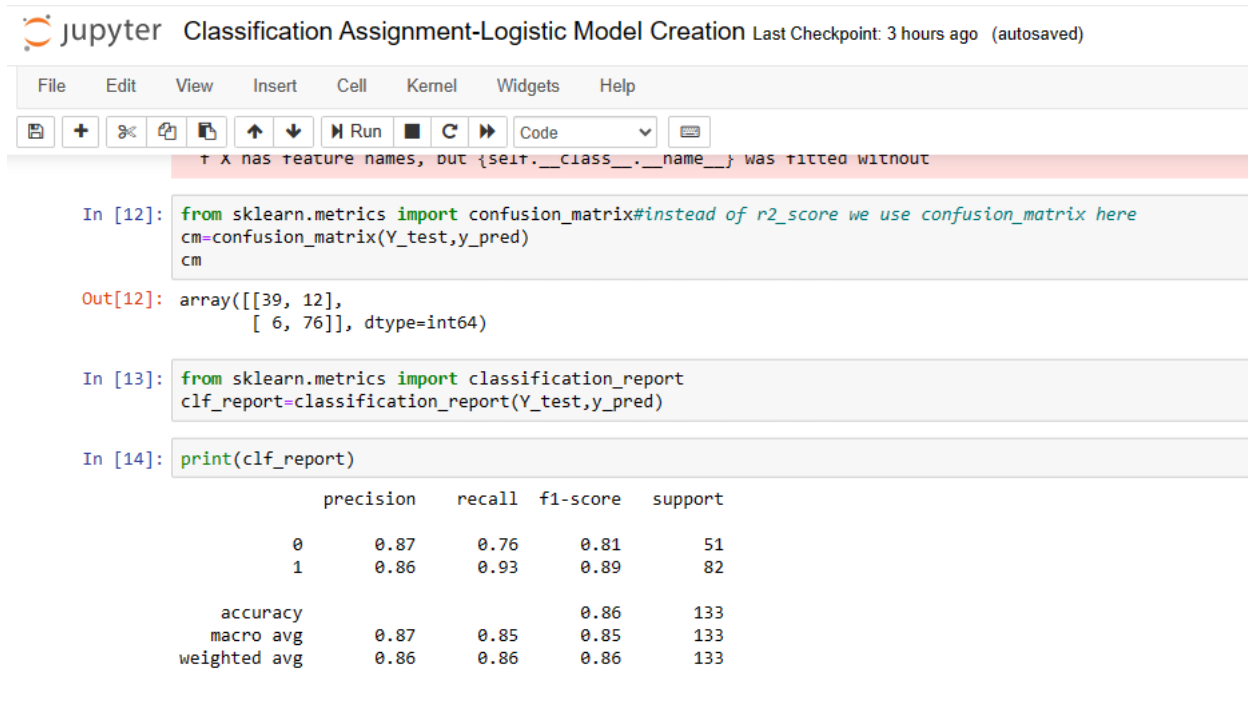
In [41]: `print(clf_report)`

	precision	recall	f1-score	support
0	0.00	0.00	0.00	51
1	0.62	1.00	0.76	82
accuracy			0.62	133
macro avg	0.31	0.50	0.38	133
weighted avg	0.38	0.62	0.47	133

Activate Windows
Go to Settings to activate

c) Logistic Regression Confusion Matrix:-

```
param_grid = {'solver':['newton-cg', 'lbfgs', 'saga', 'liblinear'],'penalty':['l2']}  
grid = GridSearchCV(LogisticRegression(), param_grid, refit=True, verbose=3, n_jobs=-1,  
scoring='f1_weighted')
```



The screenshot shows a Jupyter Notebook interface with the title "Classification Assignment-Logistic Model Creation" and a status bar indicating "Last Checkpoint: 3 hours ago (autosaved)". The notebook has a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, zooming, and running code. A red error message is visible at the top: "x has feature names, but {'self.__class__.__name__'} was fitted without". The notebook contains four code cells:

```
In [12]: from sklearn.metrics import confusion_matrix#instead of r2_score we use confusion_matrix here  
cm=confusion_matrix(Y_test,y_pred)  
cm
```

Out[12]: array([[39, 12],
[6, 76]], dtype=int64)

```
In [13]: from sklearn.metrics import classification_report  
clf_report=classification_report(Y_test,y_pred)
```

```
In [14]: print(clf_report)
```

	precision	recall	f1-score	support
0	0.87	0.76	0.81	51
1	0.86	0.93	0.89	82
accuracy			0.86	133
macro avg	0.87	0.85	0.85	133
weighted avg	0.86	0.86	0.86	133

6.Final Model:-

Logistic Regression has been considered as final model. This model provided highest accuracy as 0.86 among all other algorithms.