

Project Development Phase Model Performance Test

Date	10 November 2022
Team ID	PNT2022TMID08664
Project Name	Early Detection of Chronic Kidney Disease using Machine Learning
Maximum Marks	10 Marks

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot																														
1.	Metrics	<div>Classification Model: Confusion Matrix – [[54,4], [0,62]], Accuracy Score- 96.6666% Classification Report – <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>1.00</td><td>0.93</td><td>0.96</td><td>58</td></tr><tr><td>1</td><td>0.94</td><td>1.00</td><td>0.97</td><td>62</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.97</td><td>120</td></tr><tr><td>macro avg</td><td>0.97</td><td>0.97</td><td>0.97</td><td>120</td></tr><tr><td>weighted avg</td><td>0.97</td><td>0.97</td><td>0.97</td><td>120</td></tr></tbody></table></div>		precision	recall	f1-score	support	0	1.00	0.93	0.96	58	1	0.94	1.00	0.97	62	accuracy			0.97	120	macro avg	0.97	0.97	0.97	120	weighted avg	0.97	0.97	0.97	120	
	precision	recall	f1-score	support																													
0	1.00	0.93	0.96	58																													
1	0.94	1.00	0.97	62																													
accuracy			0.97	120																													
macro avg	0.97	0.97	0.97	120																													
weighted avg	0.97	0.97	0.97	120																													
2.	Tune the Model	Hyperparameter Tuning - Validation Method -																															

Screenshots:

ACCURACY SCORE,CONFUSION MATRIX AND CLASSIFICATION REPORT SCREENSHOTS:

▼ ACCURACY SCORE

```
[ ] print(accuracy_score(y_pred,y_test))

0.9666666666666667
```

AS WE MOVED WITH RANDOM FOREST CLASSIFIER, WE GOT ACCURACY OF 96.66666%

▼ CONFUSION MATRIX

```
[ ] conf_mat=confusion_matrix(y_test,y_pred)
conf_mat

array([[54,  4],
       [ 0, 62]])
```

▼ CLASSIFICATION REPORT

```
[ ] print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	1.00	0.93	0.96	58
1	0.94	1.00	0.97	62
accuracy			0.97	120
macro avg	0.97	0.97	0.97	120
weighted avg	0.97	0.97	0.97	120

HYPERPARAMETER TUNING:

▼ MODEL BUILDING AND FINDING BEST MODEL ACCURACY AMONG FOUR MODELS

HYPERPARAMETER TUNING

```
from sklearn.model_selection import ShuffleSplit,GridSearchCV,StratifiedKFold
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
def find_best_model(x,y):
    models={'logistic_regression':{'model':LogisticRegression(solver='liblinear',penalty='l2',multi_class='auto'),'parameter':{'C':[1,4,8]}},
            'decision_tree':{'model':DecisionTreeClassifier(splitter='best'),'parameter':{'criterion':['gini','entropy'],'max_depth':[5,7,13,15]}},
            'svm':{'model':SVC(gamma='auto'),'parameter':{'kernel':['sigmoid','linear'],'C':[1,5,10,15]}},
            'random_forest':{'model':RandomForestClassifier(criterion='gini'),'parameter':{'max_depth':[5,10,15],'n_estimators':[1,3,5]}}}
    scores=[]
    cv_shuffle=StratifiedKFold(n_splits=10)
    for model_name,model_params in models.items():
        gs=GridSearchCV(model_params['model'],model_params['parameter'],cv=cv_shuffle,return_train_score=False)
        gs.fit(x,y)
        scores.append({'model':model_name,'best_parameters':gs.best_params_,'score':gs.best_score_})
    return pd.DataFrame(scores,columns=['model','best_parameters','score'])
find_best_model(X_train,y_train)
```

	model	best_parameters	score
0	Logistic_regression	{'C': 1}	0.985714
1	decision_tree	{'criterion': 'gini', 'max_depth': 15}	0.985714
2	svm	{'C': 1, 'kernel': 'linear'}	0.978571
3	random_forest	{'max_depth': 10, 'n_estimators': 5}	0.996429