**Project Development Phase**
**Model Performance Test**

| | |
|---|---|
| Date | 19 November 2022 |
| Team ID | PNT2022TMID32547 |
| Project Name | Project – 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 | **Regression Model:**<br>MAE - , MSE - , RMSE - , R2 score -<br><br>**Classification Model:**<br>Confusion Matrix - ,<br>Accuracy Score- &<br>Classification Report - | Classification Model<br>**Accuracy**<br>`[ ] from sklearn.metrics import get_scorer`<br>`ypred = pipe.predict(xtrain.values)`<br>`scorer = get_scorer('accuracy')`<br>`score = scorer(pipe, xtest.values, ytest.values)`<br>`print("Accuracy for training data:",accuracy_score(ytrain,ypred))`<br>`print("Accuracy for test data:",score)`<br>`Accuracy for training data: 1.0`<br>`Accuracy for test data: 1.0`<br>**Confusion Matrix**<br>`[ ] ypred = pipe.predict(xtest.values)`<br>`cm = confusion_matrix(ytest, ypred)`<br>`print(cm)`<br>`[[30 0]`<br>`[ 0 12]]`<br>**Classification Report**<br>`[ ] print(classification_report(ytest, ypred))`<br>`precision recall f1-score support`<br>`ckd 1.00 1.00 1.00 30`<br>`notckd 1.00 1.00 1.00 12`<br>`accuracy 1.00 42`<br>`macro avg 1.00 1.00 1.00 42`<br>`weighted avg 1.00 1.00 1.00 42` |

| | | | |
|---|---|---|---|
| 2. | Tune the Model | Hyperparameter Tuning - Validation Method - | **Hyperparameter Tuning**<br><br>`[ ]`<br>`#n_estimators`<br>`parameters = {`<br>`'n_estimators': [10, 20, 50, 100, 200, 500, 1000, 1200, 1500, 1800, 1900, 2000, 2100, 3000]`<br>`}`<br>`clf = GridSearchCV(extra_trees_classifier, parameters, cv=5, verbose=3, n_jobs=-1)`<br>`pipe = make_pipeline(column_trans, clf)`<br><br>`pipe.fit(xtrain.values, ytrain.values.ravel());`<br><br>`pred=pipe.predict(xtest.values)`<br>`print(classification_report(ytest, pred))`<br><br>`Fitting 5 folds for each of 14 candidates, totalling 70 fits`<br>`precision recall f1-score support`<br><br>`ckd 1.00 1.00 1.00 29`<br>`notckd 1.00 1.00 1.00 13`<br><br>`accuracy 1.00 42`<br>`macro avg 1.00 1.00 1.00 42`<br>`weighted avg 1.00 1.00 1.00 42`<br><br>`[ ]`<br>`#criterion`<br>`parameters = {`<br>`'criterion': ['gini', 'entropy']`<br>`}`<br>`clf = GridSearchCV(extra_trees_classifier, parameters, cv=5, verbose=3, n_jobs=-1)`<br>`pipe = make_pipeline(column_trans, clf)`<br><br>`pipe.fit(xtrain.values, ytrain.values.ravel());`<br><br>`pred=pipe.predict(xtest.values)`<br>`print(classification_report(ytest, pred))`<br><br>`Fitting 5 folds for each of 2 candidates, totalling 10 fits`<br>`precision recall f1-score support`<br><br>`ckd 1.00 1.00 1.00 29`<br>`notckd 1.00 1.00 1.00 13`<br><br>`accuracy 1.00 42`<br>`macro avg 1.00 1.00 1.00 42`<br>`weighted avg 1.00 1.00 1.00 42`<br><br>`[ ]`<br>`#max_depth`<br>`parameters = {`<br>`'max_depth': [1, 2, 5, 8, 13, 21, 34, 53, 54, 55, 89, None]`<br>`}`<br>`clf = GridSearchCV(extra_trees_classifier, parameters, cv=5, verbose=3, n_jobs=-1)`<br>`pipe = make_pipeline(column_trans, clf)`<br><br>`pipe.fit(xtrain.values, ytrain.values.ravel());`<br><br>`pred=pipe.predict(xtest.values)`<br>`print(classification_report(ytest, pred))`<br><br>`Fitting 5 folds for each of 12 candidates, totalling 60 fits`<br>`precision recall f1-score support`<br><br>`ckd 1.00 1.00 1.00 29`<br>`notckd 1.00 1.00 1.00 13`<br><br>`accuracy 1.00 42`<br>`macro avg 1.00 1.00 1.00 42`<br>`weighted avg 1.00 1.00 1.00 42`<br><br>`[ ]`<br>`#min_samples_leaf`<br>`parameters = {`<br>`'min_samples_leaf': [1, 2, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377]`<br>`}`<br>`clf = GridSearchCV(extra_trees_classifier, parameters, cv=5, verbose=3, n_jobs=-1)`<br>`pipe = make_pipeline(column_trans, clf)`<br><br>`pipe.fit(xtrain.values, ytrain.values.ravel());`<br><br>`pred=pipe.predict(xtest.values)`<br>`print(classification_report(ytest, pred))`<br><br>`Fitting 5 folds for each of 12 candidates, totalling 60 fits`<br>`precision recall f1-score support`<br><br>`ckd 1.00 1.00 1.00 29`<br>`notckd 1.00 1.00 1.00 13`<br><br>`accuracy 1.00 42`<br>`macro avg 1.00 1.00 1.00 42`<br>`weighted avg 1.00 1.00 1.00 42` |

```
[ ]  #max_leaf_nodes
     parameters = {
         'max_leaf_nodes': [2, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, None]
     }
     clf = GridSearchCV(extra_trees_classifier, parameters, cv=5, verbose=3, n_jobs=-1)
     pipe = make_pipeline(column_trans, clf)

     pipe.fit(xtrain.values, ytrain.values.ravel());

     pred=pipe.predict(xtest.values)
     print(classification_report(ytest, pred))
```

```
Fitting 5 folds for each of 12 candidates, totalling 60 fits
              precision    recall  f1-score   support

         ckd       1.00      1.00      1.00        29
      notckd       1.00      1.00      1.00        13

    accuracy                           1.00        42
   macro avg       1.00      1.00      1.00        42
weighted avg       1.00      1.00      1.00        42
```

**Validation Method**

```
[ ]  cv_scores = cross_val_score(pipe, xtrain, ytrain, cv=5 )
     print("Cross Validation average score: %.2f" % cv_scores.mean())
```

```
Fitting 5 folds for each of 14 candidates, totalling 70 fits
Fitting 5 folds for each of 14 candidates, totalling 70 fits
Fitting 5 folds for each of 14 candidates, totalling 70 fits
Fitting 5 folds for each of 14 candidates, totalling 70 fits
Fitting 5 folds for each of 14 candidates, totalling 70 fits
Cross Validation average score: 1.00
```