Project Development Phase
Model Performance Test

Date	09 November 2023
Team ID	Team-592661
Project Name	AI-driven resource 5G optimization
Maximum Marks	10 Marks

Team members :

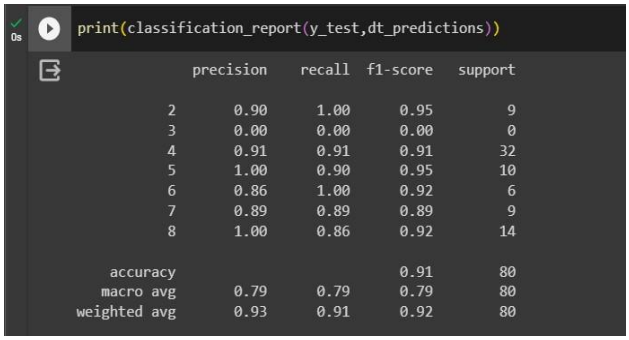
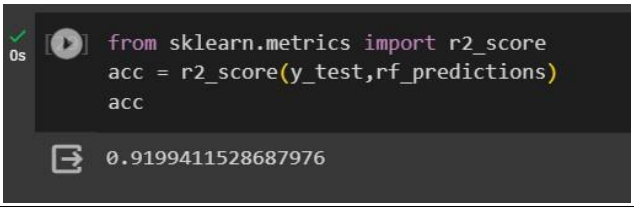
R. Sai Akshith – 21BAI1729 (Chennai) – team lead

P. Visweswar Reddy – 21BAI1602 (Chennai)

R. Jaswanth Sai – 21BAI1580 (Chennai)

Model Performance Testing:

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

S.No	Parameter	Values	Screenshot
1.	Model Summary	—	—
2.	Accuracy	Training Accuracy - 91 Validation Accuracy – 91.994	 
3.	Confidence Score (Only Yolo Projects)	Class Detected - NA Confidence Score - NA	Not Applicable

```
[ ] from sklearn.ensemble import RandomForestRegressor
    from sklearn.metrics import mean_squared_error
    rf_regressor = RandomForestRegressor(random_state=42)
    rf_regressor.fit(X_train, y_train)
    rf_predictions = rf_regressor.predict(X_test)
    rf_mse = mean_squared_error(y_test, rf_predictions)
    print("Random Forest Regressor MSE:", rf_mse)
```

```
Random Forest Regressor MSE: 0.2805937500000001
```

```
[ ] from sklearn.linear_model import LinearRegression
    linear_regressor = LinearRegression()
    linear_regressor.fit(X_train, y_train)
    linear_predictions = linear_regressor.predict(X_test)
    linear_mse = mean_squared_error(y_test, linear_predictions)
    print("Linear Regressor MSE:", linear_mse)
```

```
Linear Regressor MSE: 2.8677200653654302
```

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
dt_classifier = DecisionTreeClassifier(random_state=42)
dt_classifier.fit(X_train, y_train)
dt_predictions = dt_classifier.predict(X_test)
dt_accuracy = accuracy_score(y_test, dt_predictions)
print("Decision Tree Classifier Accuracy:", dt_accuracy)
```

```
Decision Tree Classifier Accuracy: 0.9125
```

```
[ ] from sklearn.svm import SVC
    svm_classifier = SVC(random_state=42)
    svm_classifier.fit(X_train, y_train)
    svm_predictions = svm_classifier.predict(X_test)
    svm_accuracy = accuracy_score(y_test, svm_predictions)
    print("SVM Classifier Accuracy:", svm_accuracy)
```

```
SVM Classifier Accuracy: 0.8375
```

```
from sklearn.neighbors import KNeighborsClassifier
knn_classifier = KNeighborsClassifier(n_neighbors=5)
knn_classifier.fit(X_train, y_train)
knn_predictions = knn_classifier.predict(X_test)
knn_accuracy = accuracy_score(y_test, knn_predictions)
print("KNN Classifier Accuracy:", knn_accuracy)
```

```
KNN Classifier Accuracy: 0.8625
```

```
[ ] from sklearn.naive_bayes import GaussianNB
    nb_classifier = GaussianNB()
    nb_classifier.fit(X_train, y_train)
    nb_predictions = nb_classifier.predict(X_test)
    nb_accuracy = accuracy_score(y_test, nb_predictions)
    print("Naive Bayes Classifier Accuracy:", nb_accuracy)
```

```
Naive Bayes Classifier Accuracy: 0.7875
```

```
[ ] from sklearn.metrics import r2_score
    acc = r2_score(y_test,rf_predictions)
    acc
```

```
0.9199411528687976
```

```
from sklearn.metrics import r2_score
acc = r2_score(y_test,dt_predictions)
acc
```

```
0.8430743167937229
```

```
[ ] from sklearn.metrics import r2_score
    acc = r2_score(y_test,linear_predictions)
    acc
```

```
0.18178376272396435
```

```
[ ] from sklearn.metrics import r2_score
    acc = r2_score(y_test,svm_predictions)
    acc
```

```
0.5791538495831661
```

```
from sklearn.metrics import r2_score
acc = r2_score(y_test,knn_predictions)
acc
```

```
0.7325130399893005
```

```
print(classification_report(y_test,dt_predictions))
```

```
              precision    recall  f1-score   support

           2       0.90      1.00      0.95         9
           3       0.00      0.00      0.00         0
           4       0.91      0.91      0.91        32
           5       1.00      0.90      0.95        10
           6       0.86      1.00      0.92         6
           7       0.89      0.89      0.89         9
           8       1.00      0.86      0.92        14

    accuracy                           0.91        80
   macro avg       0.79      0.79      0.79        80
weighted avg       0.93      0.91      0.92        80

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Recall and F-score are ill-defined and being s
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Recall and F-score are ill-defined and being s
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Recall and F-score are ill-defined and being s
  _warn_prf(average, modifier, msg_start, len(result))
```