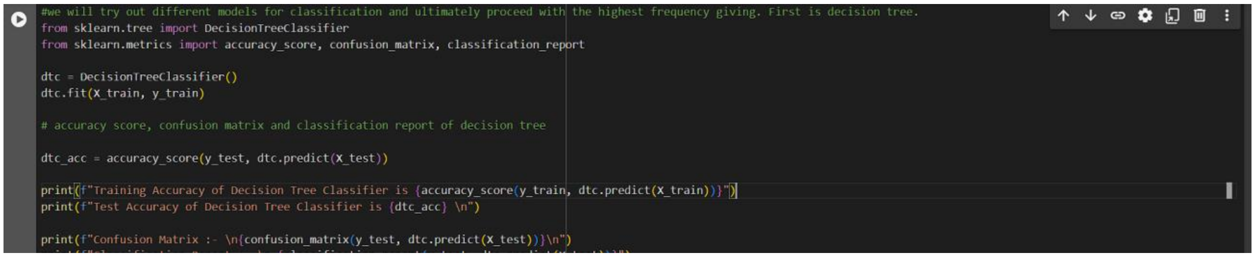
Model Development Phase Template

Date	16 JULY 2024
Team ID	SWTID1720075199
Project Title	Early Prediction of Chronic Kidney Disease Using Machine Learning
Maximum Marks	4 Marks

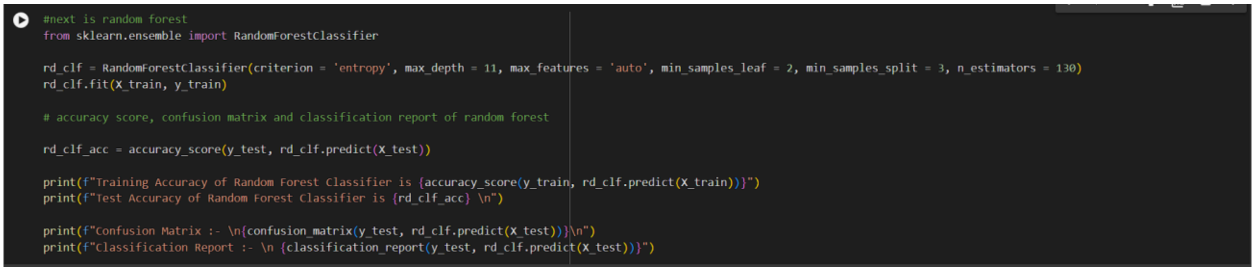
Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

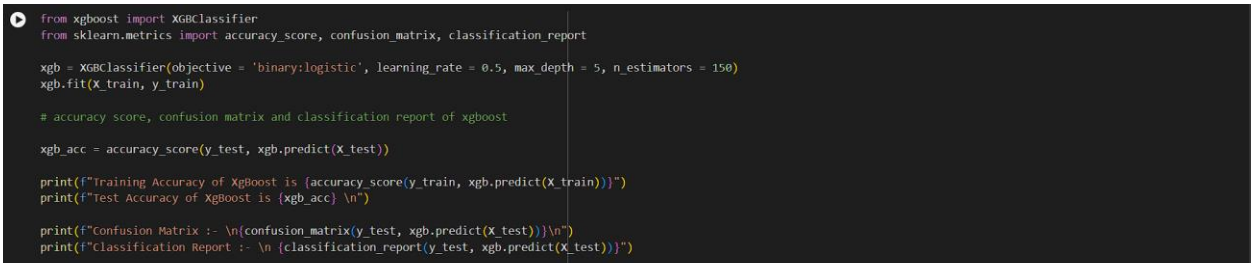
Initial Model Training Code:



```
#we will try out different models for classification and ultimately proceed with the highest frequency giving. First is decision tree.
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

dtc = DecisionTreeClassifier()
dtc.fit(X_train, y_train)

# accuracy score, confusion matrix and classification report of decision tree

dtc_acc = accuracy_score(y_test, dtc.predict(X_test))

print(f"Training Accuracy of Decision Tree Classifier is {accuracy_score(y_train, dtc.predict(X_train))}")
print(f"Test Accuracy of Decision Tree Classifier is {dtc_acc} \n")

print(f"Confusion Matrix :- \n{confusion_matrix(y_test, dtc.predict(X_test))}\n")
```

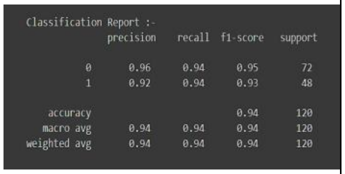
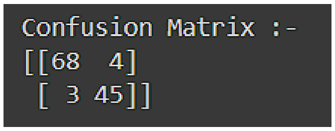
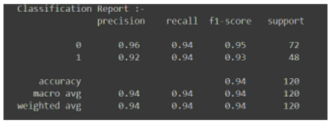
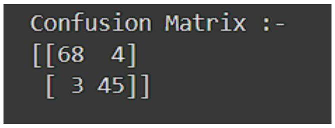
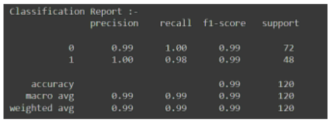
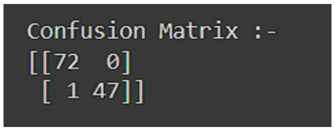


```
#next is random forest
from sklearn.ensemble import RandomForestClassifier

rd_clf = RandomForestClassifier(criterion = 'entropy', max_depth = 11, max_features = 'auto', min_samples_leaf = 2, min_samples_split = 3, n_estimators = 130)
rd_clf.fit(X_train, y_train)

# accuracy score, confusion matrix and classification report of random forest

rd_clf_acc = accuracy_score(y_test, rd_clf.predict(X_test))

print(f"Training Accuracy of Random Forest Classifier is {accuracy_score(y_train, rd_clf.predict(X_train))}")
print(f"Test Accuracy of Random Forest Classifier is {rd_clf_acc} \n")

print(f"Confusion Matrix :- \n{confusion_matrix(y_test, rd_clf.predict(X_test))}\n")
print(f"Classification Report :- \n {classification_report(y_test, rd_clf.predict(X_test))}")
```



```
from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

xgb = XGBClassifier(objective = 'binary:logistic', learning_rate = 0.5, max_depth = 5, n_estimators = 150)
xgb.fit(X_train, y_train)

# accuracy score, confusion matrix and classification report of xgboost

xgb_acc = accuracy_score(y_test, xgb.predict(X_test))

print(f"Training Accuracy of XgBoost is {accuracy_score(y_train, xgb.predict(X_train))}")
print(f"Test Accuracy of XgBoost is {xgb_acc} \n")

print(f"Confusion Matrix :- \n{confusion_matrix(y_test, xgb.predict(X_test))}\n")
print(f"Classification Report :- \n {classification_report(y_test, xgb.predict(X_test))}")
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Decision Tree	 Classification Report :- precision recall f1-score support 0 0.96 0.94 0.95 72 1 0.92 0.94 0.93 48 accuracy 0.94 120 macro avg 0.94 0.94 0.94 120 weighted avg 0.94 0.94 0.94 120	94 %	 Confusion Matrix :- [[68 4] [3 45]]
Random Forest	 Classification Report :- precision recall f1-score support 0 0.96 0.94 0.95 72 1 0.92 0.94 0.93 48 accuracy 0.94 120 macro avg 0.94 0.94 0.94 120 weighted avg 0.94 0.94 0.94 120	97%	 Confusion Matrix :- [[68 4] [3 45]]
Gradient Boosting	 Classification Report :- precision recall f1-score support 0 0.99 1.00 0.99 72 1 1.00 0.98 0.99 48 accuracy 0.99 120 macro avg 0.99 0.99 0.99 120 weighted avg 0.99 0.99 0.99 120	99%	 Confusion Matrix :- [[72 0] [1 47]]