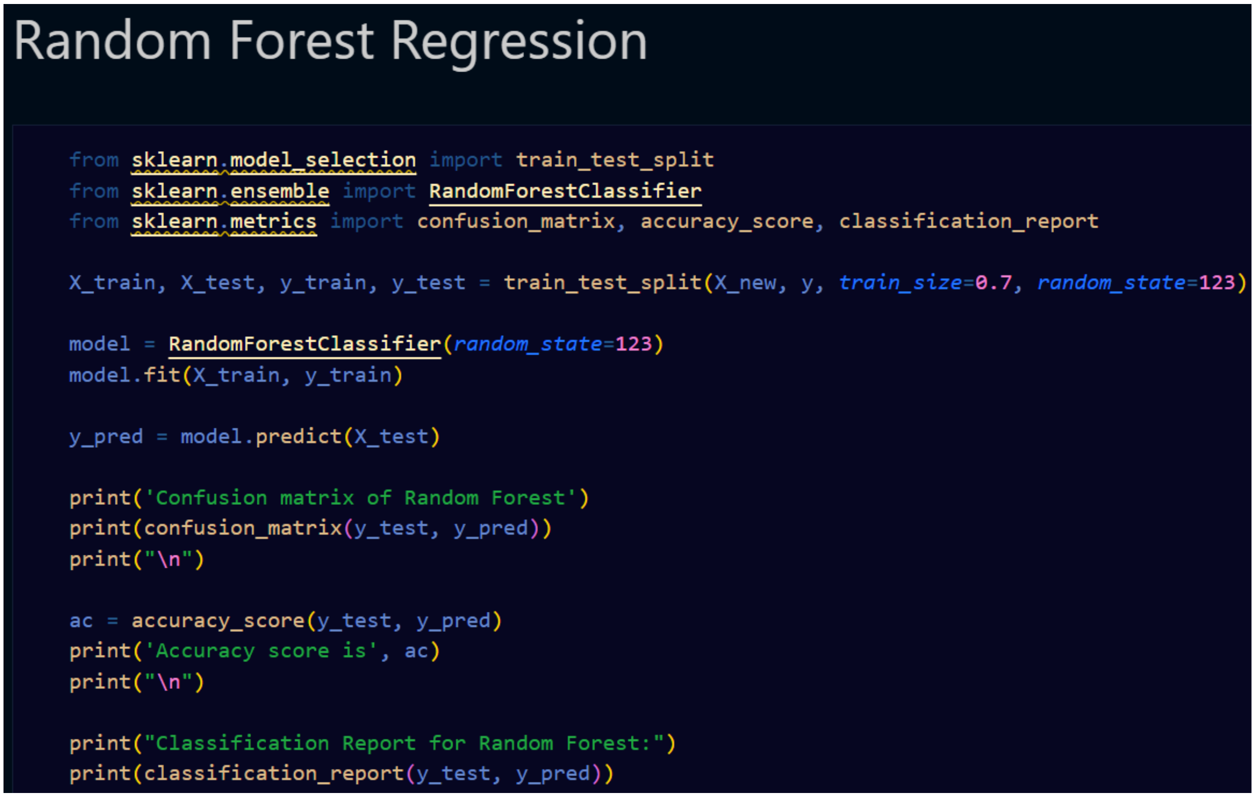
Model Development Phase Template

Date	6 July 2024
Team ID	SWTID1720082525
Project Title	Early Prediction of Chronic Kidney Disease Using Machine Learning
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

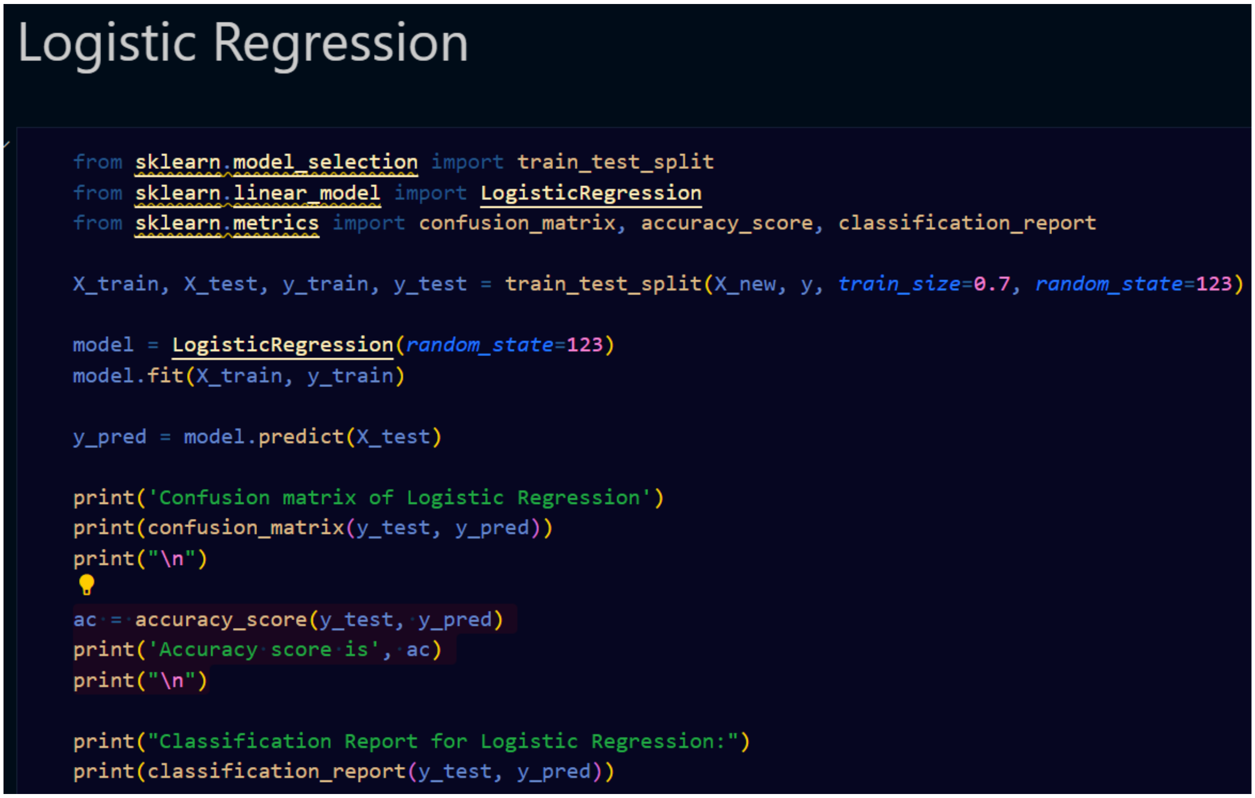
The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model Training Code:



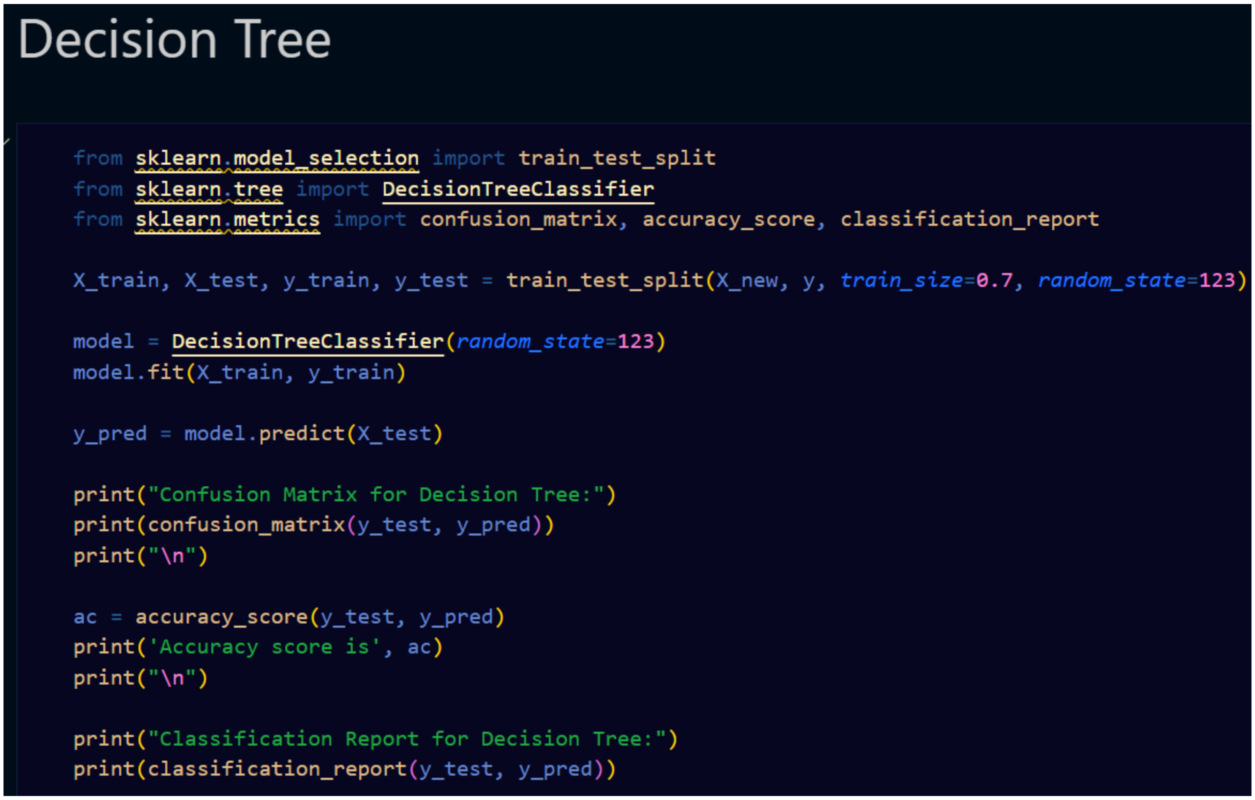
Random Forest Regression

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report

X_train, X_test, y_train, y_test = train_test_split(X_new, y, train_size=0.7, random_state=123)

model = RandomForestClassifier(random_state=123)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

print('Confusion matrix of Random Forest')
print(confusion_matrix(y_test, y_pred))
print("\n")

ac = accuracy_score(y_test, y_pred)
print('Accuracy score is', ac)
print("\n")

print("Classification Report for Random Forest:")
print(classification_report(y_test, y_pred))
```



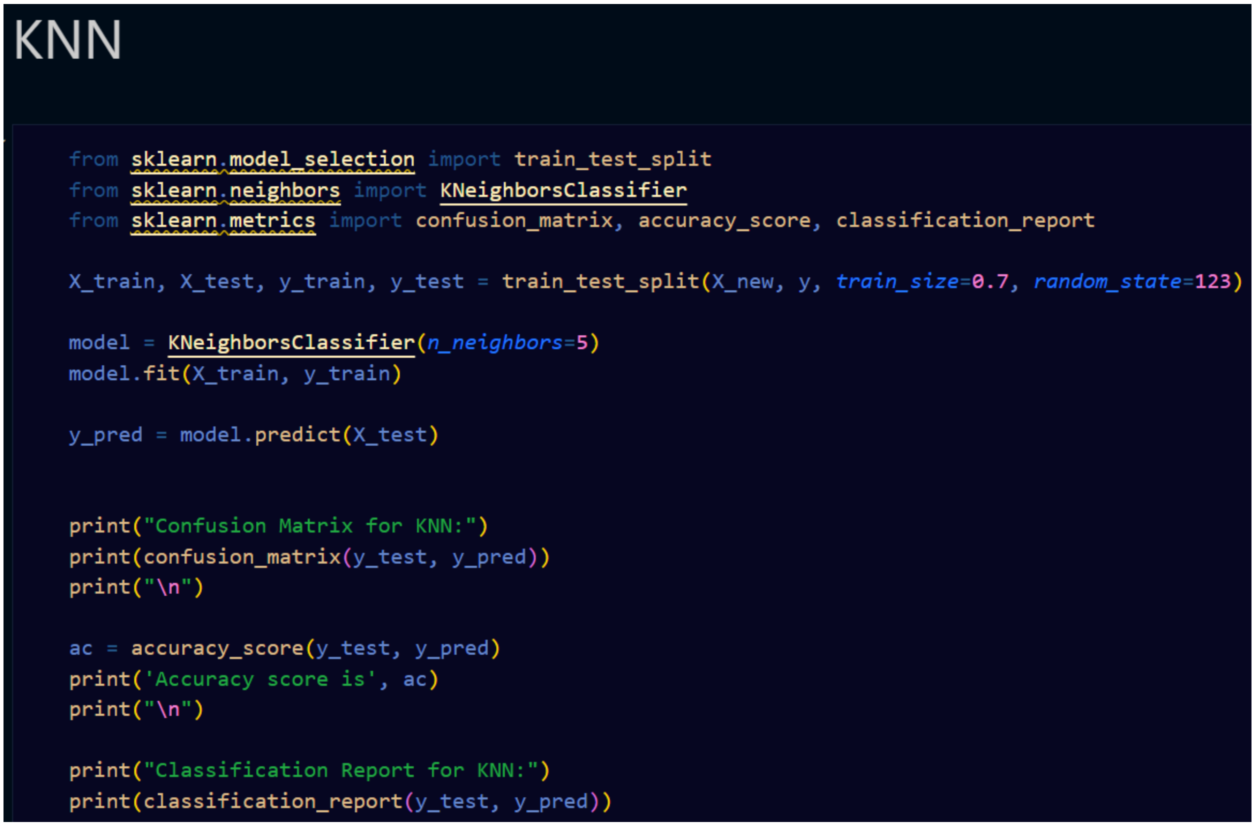
Logistic Regression

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report

X_train, X_test, y_train, y_test = train_test_split(X_new, y, train_size=0.7, random_state=123)

model = LogisticRegression(random_state=123)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

print('Confusion matrix of Logistic Regression')
print(confusion_matrix(y_test, y_pred))
print("\n")

ac = accuracy_score(y_test, y_pred)
print('Accuracy score is', ac)
print("\n")

print("Classification Report for Logistic Regression:")
print(classification_report(y_test, y_pred))
```



Decision Tree

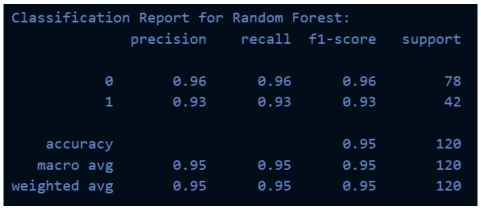
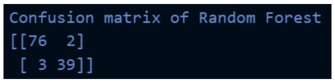
```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report

X_train, X_test, y_train, y_test = train_test_split(X_new, y, train_size=0.7, random_state=123)

model = DecisionTreeClassifier(random_state=123)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

print("Confusion Matrix for Decision Tree:")
print(confusion_matrix(y_test, y_pred))
print("\n")

ac = accuracy_score(y_test, y_pred)
print('Accuracy score is', ac)
print("\n")

print("Classification Report for Decision Tree:")
print(classification_report(y_test, y_pred))
```

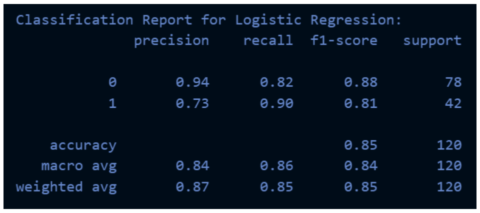
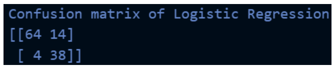
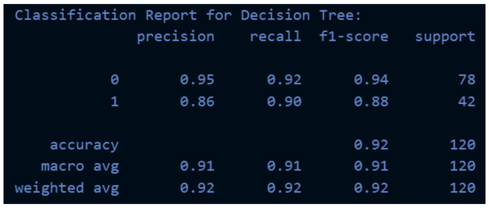
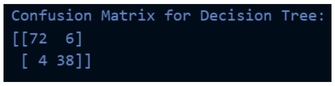
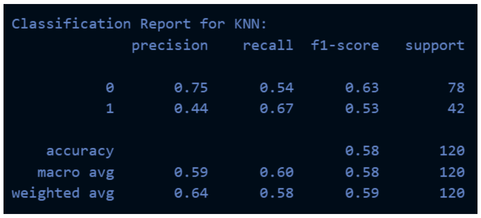
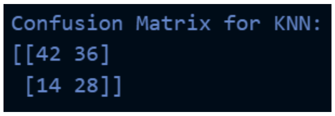


KNN

```
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report

X_train, X_test, y_train, y_test = train_test_split(X_new, y, train_size=0.7, random_state=123)

model = KNeighborsClassifier(n_neighbors=5)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)


print("Confusion Matrix for KNN:")
print(confusion_matrix(y_test, y_pred))
print("\n")

ac = accuracy_score(y_test, y_pred)
print('Accuracy score is', ac)
print("\n")

print("Classification Report for KNN:")
print(classification_report(y_test, y_pred))
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Random Forest	 Classification Report for Random Forest: precision recall f1-score support 0 0.96 0.96 0.96 78 1 0.93 0.93 0.93 42 accuracy 0.95 120 macro avg 0.95 0.95 0.95 120 weighted avg 0.95 0.95 0.95 120	95.8%	 Confusion matrix of Random Forest [[76 2] [3 39]]

Logistic Regression	 Classification Report for Logistic Regression: precision recall f1-score support 0 0.94 0.82 0.88 78 1 0.73 0.90 0.81 42 accuracy 0.85 120 macro avg 0.84 0.86 0.84 120 weighted avg 0.87 0.85 0.85 120	85%	 Confusion matrix of Logistic Regression [[64 14] [4 38]]
Decision Tree	 Classification Report for Decision Tree: precision recall f1-score support 0 0.95 0.92 0.94 78 1 0.86 0.90 0.88 42 accuracy 0.92 120 macro avg 0.91 0.91 0.91 120 weighted avg 0.92 0.92 0.92 120	92%	 Confusion Matrix for Decision Tree: [[72 6] [4 38]]
KNN	 Classification Report for KNN: precision recall f1-score support 0 0.75 0.54 0.63 78 1 0.44 0.67 0.53 42 accuracy 0.58 120 macro avg 0.59 0.60 0.58 120 weighted avg 0.64 0.58 0.59 120	58%	 Confusion Matrix for KNN: [[42 36] [14 28]]