Model Development Phase Template

Date	07 July 2024
Team ID	SWTID1720082372
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 Forset Regression

```
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))
ac = accuracy_score(y_test, y_pred)
print('Accuracy score is ',ac)
accuracy.append(ac)
print('='*50)
```

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import 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("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

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))

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

print('='*50)
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, 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 for Logistic Regression:")
print(confusion_matrix(y_test, y_pred))
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, 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")

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, 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")

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 0.99 0.97 78 1 0.97 0.93 0.95 42 accuracy 0.97 120 macro avg 0.97 0.96 0.96 120 weighted avg 0.97 0.97 0.97 120	97%	Confusion matrix of Random Forest [[77 1] [3 39]]

Logistic Regression	Classification Report for Logistic Regression: precision recall f1-score support 0 0.96 0.88 0.92 78 1 0.81 0.93 0.87 42 accuracy 0.90 120 macro avg 0.89 0.91 0.89 120 weighted avg 0.91 0.90 0.90 120	90%	Confusion matrix of Logistic Regression [[69 9] [3 39]]
Decision Tree	Classification Report for Decision Tree: precision recall f1-score support 0 0.96 0.95 0.95 78 1 0.91 0.93 0.92 42 accuracy 0.94 120 macro avg 0.93 0.94 0.94 120 weighted avg 0.94 0.94 0.94 120	94%	Confusion Matrix for Decision Tree: [[74 4] [3 39]]
KNN	Classification Report for KNN: precision recall f1-score support 0 0.70 0.50 0.58 78 1 0.39 0.60 0.47 42 accuracy 0.53 120 macro avg 0.54 0.55 0.53 120 weighted avg 0.59 0.53 0.54 120	53%	Confusion Matrix for KNN: [[39 39] [17 25]]