Model Development Phase Template

Date	15 MARCH 2024
Team ID	LTVIP2024TMID25011
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

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

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize and train the model
model = RandomForestClassifier(n_estimators=20)
model.fit(X_train, y_train)

# Predict and calculate metrics
y_pred = model.predict(X_test)
cm = confusion_matrix(y_test, y_pred)

# Print the confusion matrix and accuracy
print("Confusion Matrix:")
print(cm)
print(f"Accuracy is {round(accuracy_score(y_test, y_pred) * 100, 2)}%")

# Print the classification report
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

```python
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Print the accuracy score
print(f"Accuracy is {round(accuracy_score(y_test, model.predict(X_test)) * 100, 2)}%")

# Print the confusion matrix
print("Confusion Matrix")
print(confusion_matrix(y_test, model.predict(X_test)))

# Print the classification report
print("Classification Report")
print(classification_report(y_test, model.predict(X_test)))
```

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

# Assuming X_train, X_test, y_train, y_test are already defined
model = LogisticRegression()
model.fit(X_train, y_train)
pred = model.predict(X_test)

# Calculate the accuracy
accuracy = accuracy_score(y_test, pred)
print(f"Accuracy: {accuracy * 100:.2f}%")

# Print the confusion matrix
print("Confusion Matrix:")
print(confusion_matrix(y_test, pred))

# Print the classification report
print("\nClassification Report:")
print(classification_report(y_test, pred))
```

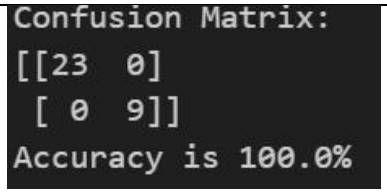
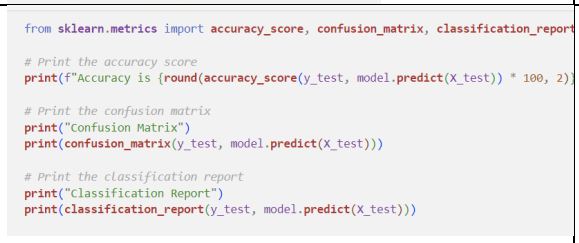
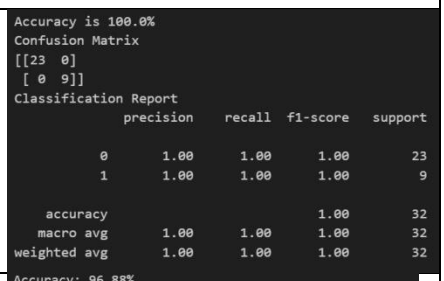
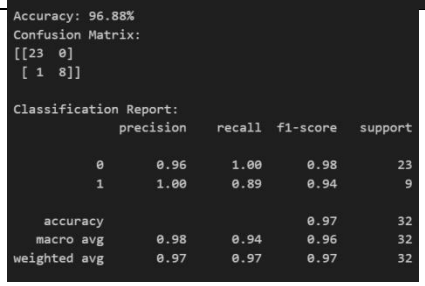
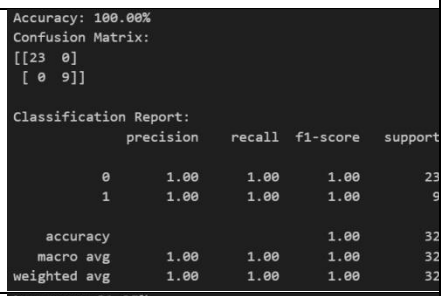
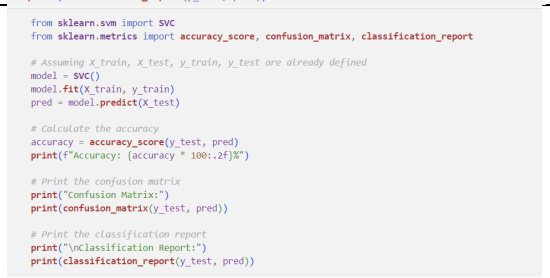
```python
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Assuming X_train, X_test, y_train, y_test are already defined
model = GaussianNB()
model.fit(X_train, y_train)
pred = model.predict(X_test)

# Calculate the accuracy
accuracy = accuracy_score(y_test, pred)
print(f"Accuracy: {accuracy * 100:.2f}%")

# Print the confusion matrix
print("Confusion Matrix:")
print(confusion_matrix(y_test, pred))

# Print the classification report
print("\nClassification Report:")
print(classification_report(y_test, pred))
```

```python
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Assuming X_train, X_test, y_train, y_test are already defined
model = SVC()
model.fit(X_train, y_train)
pred = model.predict(X_test)

# Calculate the accuracy
accuracy = accuracy_score(y_test, pred)
print(f"Accuracy: {accuracy * 100:.2f}%")

# Print the confusion matrix
print("Confusion Matrix:")
print(confusion_matrix(y_test, pred))

# Print the classification report
print("\nClassification Report:")
print(classification_report(y_test, pred))
```

```python
import pickle
pickle.dump(model, open('kidney.pkl', 'wb'))
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix

Random Forest Classifier		100.0%	
KNN		100.0%	
Logistic Regression		96.88%	
Naïve Baye's		100.0%	
SVM		81.25%