# Model Development Phase Template

| | |
|---|---|
| Date | 12 July 2024 |
| Team ID | SWTID1720083491 |
| 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.

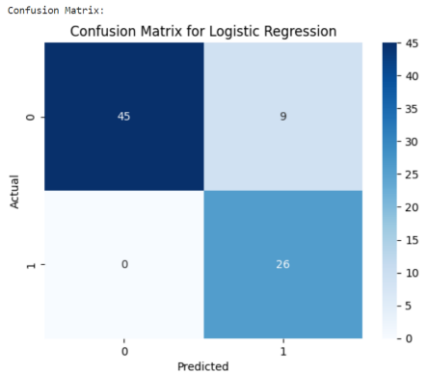
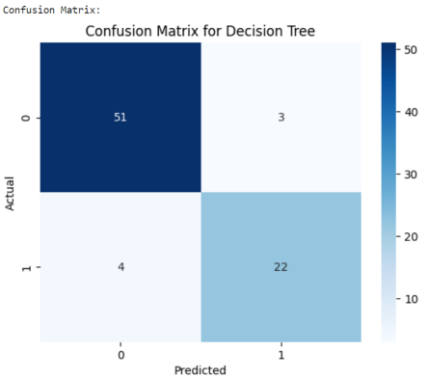
### Initial Model Training Code:

```
# Initialize the models
logreg = LogisticRegression(max_iter=1000)
dtree = DecisionTreeClassifier()
rforest = RandomForestClassifier()
svm = SVC()

# Train the models
logreg.fit(X_train, y_train)
dtree.fit(X_train, y_train)
rforest.fit(X_train, y_train)
svm.fit(X_train, y_train)

# Predict on the test set
y_pred_logreg = logreg.predict(X_test)
y_pred_dtree = dtree.predict(X_test)
y_pred_rforest = rforest.predict(X_test)
y_pred_svm = svm.predict(X_test)
```

## Model Validation and Evaluation Report:

| Model | Classification Report | Accuracy | Confusion Matrix |
|---|---|---|---|
| Logistic Regression | `print("Classification Report:")`<br>`print(classification_report(y_test, y_pred))`<br><br>Classification Report:<br>precision recall f1-score support<br><br>0 1.00 0.83 0.91 54<br>1 0.74 1.00 0.85 26<br><br>accuracy 0.89 80<br>macro avg 0.87 0.92 0.88 80<br>weighted avg 0.92 0.89 0.89 80 | 88% | `print("Confusion Matrix:")`<br>`cm = confusion_matrix(y_test, y_pred)`<br>`sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')`<br>`plt.title(f"Confusion Matrix for {model}")`<br><br>Confusion Matrix:<br>  |
| Decision Tree | `print("Classification Report:")`<br>`print(classification_report(y_test, y_pred))`<br><br>Classification Report:<br>precision recall f1-score support<br><br>0 0.93 0.94 0.94 54<br>1 0.88 0.85 0.86 26<br><br>accuracy 0.91 80<br>macro avg 0.90 0.90 0.90 80<br>weighted avg 0.91 0.91 0.91 80 | 91% | `print("Confusion Matrix:")`<br>`cm = confusion_matrix(y_test, y_pred)`<br>`sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')`<br>`plt.title(f"Confusion Matrix for {model}")`<br><br>Confusion Matrix:<br>  |

<table>
<tr>
<td>Random Forest</td>
<td>

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

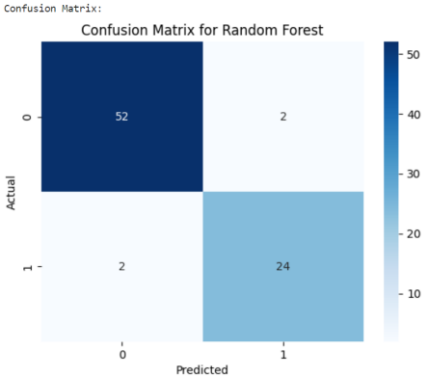
```
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.96      0.96        54
           1       0.92      0.92      0.92        26

    accuracy                           0.95        80
   macro avg       0.94      0.94      0.94        80
weighted avg       0.95      0.95      0.95        80
```

</td>
<td>95%</td>
<td>

```
print("Confusion Matrix:")
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title(f"Confusion Matrix for {model}")
```




</td>
</tr>
<tr>
<td>Support Vector Machine<br>(SVM)</td>
<td>

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

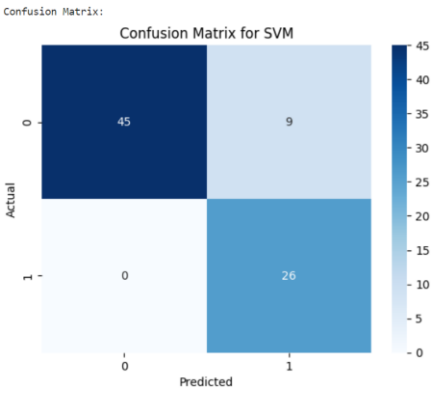
```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.83      0.91        54
           1       0.74      1.00      0.85        26

    accuracy                           0.89        80
   macro avg       0.87      0.92      0.88        80
weighted avg       0.92      0.89      0.89        80
```

</td>
<td>88%</td>
<td>

```
print("Confusion Matrix:")
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title(f"Confusion Matrix for {model}")
```




</td>
</tr>
</table>