Model Optimization and Tuning Phase Template

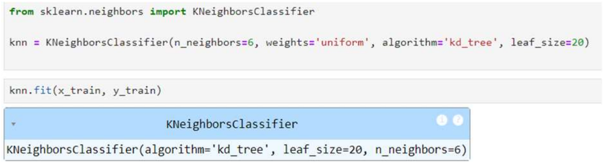
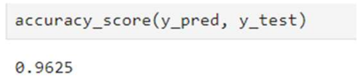
Date	14 July 2024
Team ID	SWTID1720190579
Project Title	Early Prediction of Chronic Kidney Disease Using Machine Learning
Maximum Marks	10 Marks

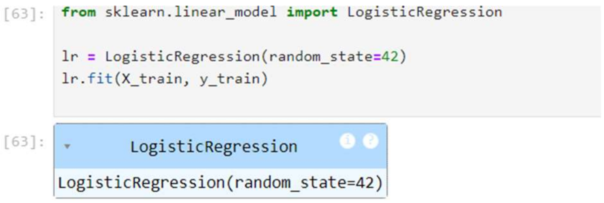
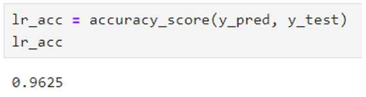
Model Optimization and Tuning Phase

The Model Optimization and Tuning Phase involves refining machine learning models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

Hyperparameter Tuning Documentation (6 Marks):

Model	Tuned Hyperparameters	Optimal Values

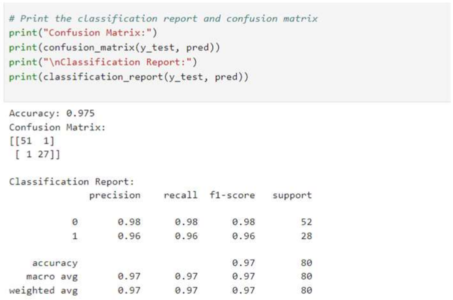
KNN	
	`from sklearn.neighbors import KNeighborsClassifier`
	`knn = KNeighborsClassifier(n_neighbors=6, weights='uniform', algorithm='kd_tree', leaf_size=20)`
	`knn.fit(x_train, y_train)`
	KNeighborsClassifier
	`KNeighborsClassifier(algorithm='kd_tree', leaf_size=20, n_neighbors=6)`
		
		`accuracy_score(y_pred, y_test)`
		`0.9625`

Logistic Regression	
	`[63]: from sklearn.linear_model import LogisticRegression`
	`lr = LogisticRegression(random_state=42)`
	`lr.fit(X_train, y_train)`
	`[63]:` LogisticRegression
	`LogisticRegression(random_state=42)`
		
		`lr_acc = accuracy_score(y_pred, y_test)`
		`lr_acc`
		`0.9625`

Performance Metrics Comparison Report (2 Marks):

Model	Base line Metric/Optimized Metric

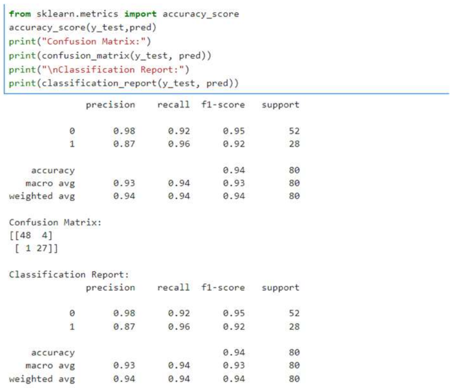
<table>
<tr>
<td>KNN</td>
<td>



```
# Print the classification report and confusion matrix
print("Confusion Matrix:")
print(confusion_matrix(y_test, pred))
print("\nClassification Report:")
print(classification_report(y_test, pred))
```

```
Accuracy: 0.975
Confusion Matrix:
[[51  1]
 [ 1 27]]

Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.98      0.98        52
           1       0.96      0.96      0.96        28

    accuracy                           0.97        80
   macro avg       0.97      0.97      0.97        80
weighted avg       0.97      0.97      0.97        80
```

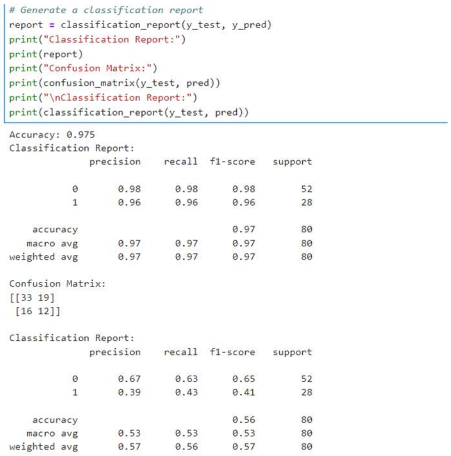
</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>



```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,pred)
print("Confusion Matrix:")
print(confusion_matrix(y_test, pred))
print("\nClassification Report:")
print(classification_report(y_test, pred))
```

```
              precision    recall  f1-score   support

           0       0.98      0.92      0.95        52
           1       0.87      0.96      0.92        28

    accuracy                           0.94        80
   macro avg       0.93      0.94      0.93        80
weighted avg       0.94      0.94      0.94        80

Confusion Matrix:
[[48  4]
 [ 1 27]]

Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.92      0.95        52
           1       0.87      0.96      0.92        28

    accuracy                           0.94        80
   macro avg       0.93      0.94      0.93        80
weighted avg       0.94      0.94      0.94        80
```

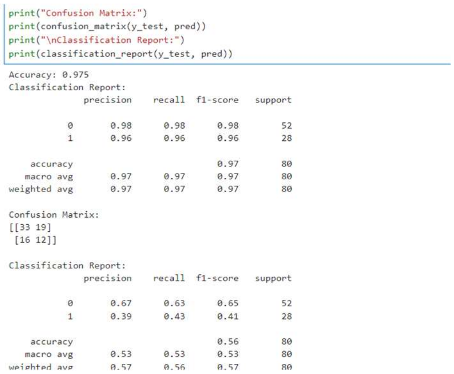
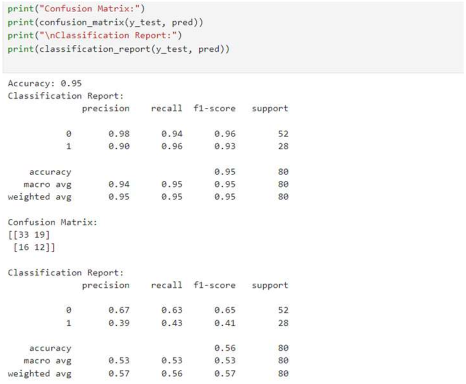
</td>
</tr>
<tr>
<td>SVM</td>
<td>



```
# Generate a classification report
report = classification_report(y_test, y_pred)
print("Classification Report:")
print(report)
print("Confusion Matrix:")
print(confusion_matrix(y_test, pred))
print("\nClassification Report:")
print(classification_report(y_test, pred))
```

```
Accuracy: 0.975
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.98      0.98        52
           1       0.96      0.96      0.96        28

    accuracy                           0.97        80
   macro avg       0.97      0.97      0.97        80
weighted avg       0.97      0.97      0.97        80

Confusion Matrix:
[[33 19]
 [16 12]]

Classification Report:
              precision    recall  f1-score   support

           0       0.67      0.63      0.65        52
           1       0.39      0.43      0.41        28

    accuracy                           0.56        80
   macro avg       0.53      0.53      0.53        80
weighted avg       0.57      0.56      0.57        80
```

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

Logistic Regression	
Decision Tree	

```
print("Confusion Matrix:")
print(confusion_matrix(y_test, pred))
print("\nClassification Report:")
print(classification_report(y_test, pred))
```

```
Accuracy: 0.975
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.98      0.98        52
           1       0.96      0.96      0.96        28

    accuracy                           0.97        80
   macro avg       0.97      0.97      0.97        80
weighted avg       0.97      0.97      0.97        80

Confusion Matrix:
[[33 19]
 [16 12]]

Classification Report:
              precision    recall  f1-score   support

           0       0.67      0.63      0.65        52
           1       0.39      0.43      0.41        28

    accuracy                           0.56        80
   macro avg       0.53      0.53      0.53        80
weighted avg       0.57      0.56      0.57        80
```

```
print("Confusion Matrix:")
print(confusion_matrix(y_test, pred))
print("\nClassification Report:")
print(classification_report(y_test, pred))
```

```
Accuracy: 0.95
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.94      0.96        52
           1       0.90      0.96      0.93        28

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

Confusion Matrix:
[[33 19]
 [16 12]]

Classification Report:
              precision    recall  f1-score   support

           0       0.67      0.63      0.65        52
           1       0.39      0.43      0.41        28

    accuracy                           0.56        80
   macro avg       0.53      0.53      0.53        80
weighted avg       0.57      0.56      0.57        80
```

Final Model Selection Justification (2 Marks):

Final Model	**Reasoning**
Logistic Regression	The logistic regression model was selected for its superior performance, exhibiting high accuracy during hyperparameter tuning. Its ability to handle complex relationships, minimize overfitting, and optimize predictive accuracy aligns with project objectives, justifying its selection as the final model. The Logistic Regression model provides detailed metrics showing an accuracy of 0.66 and a weighted average F1-score of 0.61.