Model Development Phase Template

Date	23 September 2024
Team ID	LTVIP2024TMID25021
Project Title	Prediction and Analysis of liver patient data 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:

```
def random_forest(xtrain, xtest, ytrain, ytest):
    rf = RandomForestClassifier()
    rf.fit(xtrain, ytrain)
    RFpred = rf.predict(xtest)
    RFaccuracy = accuracy_score(ytest, RFpred)
    print("Random Forest Accuracy Score: {}".format(RFaccuracy))
    print("Classification Report:\n", classification_report(ytest, RFpred))
    print("Confusion Matrix:\n", confusion_matrix(ytest, RFpred))
random_forest(xtrain, xtest, ytrain, ytest)
```

```
def decision_tree(X_train, X_test, y_train, y_test):
    dt = DecisionTreeClassifier()
    dt.fit(X_train, y_train)
    y_pred = dt.predict(X_test)
    print("Decision Tree Accuracy Score: {}".format(accuracy_score(y_test, y_pred)))
    print("Classification Report:\n", classification_report(y_test, y_pred))
    print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
decision_tree(xtrain, xtest, ytrain, ytest)
```

```
def logistic_regression(X_train, X_test, y_train, y_test):
    lr = LogisticRegression(max_iter=1000)
    lr.fit(X_train, y_train)
    LRpred = lr.predict(X_test)
    LRaccuracy = accuracy_score(y_test, LRpred)
    print("Logistic Regression Accuracy Score: {}".format(LRaccuracy))
    print("Classification Report:\n", classification_report(y_test, LRpred))
    print("Confusion Matrix:\n", confusion_matrix(y_test, LRpred))
logistic_regression(xtrain, xtest, ytrain, ytest)
```

```
def knn(X_train, X_test, y_train, y_test):
    knn_model = KNeighborsClassifier()
    knn_model.fit(X_train, y_train)
    KNNpred = knn_model.predict(X_test)
    KNNaccuracy = accuracy_score(y_test, KNNpred)
    print("KNN Accuracy Score: {}".format(KNNaccuracy))
    print("Classification Report:\n", classification_report(y_test, KNNpred))
    print("Confusion Matrix:\n", confusion_matrix(y_test, KNNpred))
knn(xtrain, xtest, ytrain, ytest)
```

```
def svm_model(X_train, X_test, y_train, y_test):
    svm = SVC()
    svm.fit(X_train, y_train)
    SVMpred = svm.predict(X_test)
    SVMaccuracy = accuracy_score(y_test, SVMpred)
    print("SVM Accuracy Score: {}".format(SVMaccuracy))
    print("Classification Report:\n", classification_report(y_test, SVMpred))
    print("Confusion Matrix:\n", confusion_matrix(y_test, SVMpred))
svm_model(xtrain, xtest, ytrain, ytest)
```

Model Validation and Evaluation Report:

Model	Classification Report	F1 Score	Confusion Matrix
Random Forest	Random Forest Accuracy Score: 0.8703703703703703 Classification Report: precision recall f1-score support 1 0.83 0.87 0.85 46 2 0.90 0.87 0.89 62 accuracy 0.87 108 macro avg 0.87 0.87 0.87 108 weighted avg 0.87 0.87 0.87 108	87%	Confusion Matrix: [[40 6] [8 54]]

Decision Tree	Decision Tree Accuracy Score: 0.7777777777777778 Classification Report: precision recall f1-score support 1 0.72 0.78 0.75 46 2 0.83 0.77 0.80 62 accuracy 0.78 108 macro avg 0.77 0.78 0.78 108 weighted avg 0.78 0.78 0.78 108	78%	Confusion Matrix: [[36 10] [14 48]]
Logistic Regression	Logistic Regression Accuracy Score: 0.7962962962962963 Classification Report: precision recall f1-score support 1 0.75 0.78 0.77 46 2 0.83 0.81 0.82 62 accuracy 0.80 108 macro avg 0.79 0.79 0.79 108 weighted avg 0.80 0.80 0.80 108	80%	Confusion Matrix: [[36 10] [12 50]]
KNN	KNN Accuracy Score: 0.75 Classification Report: precision recall f1-score support 1 0.70 0.72 0.71 46 2 0.79 0.77 0.78 62 accuracy 0.75 108 macro avg 0.74 0.75 0.75 108 weighted avg 0.75 0.75 0.75 108	75%	Confusion Matrix: [[33 13] [14 48]]
Support Vector machine	SVM Accuracy Score: 0.7037037037037037 Classification Report: precision recall f1-score support 1 0.85 0.45 0.59 51 2 0.65 0.93 0.77 57 accuracy 0.70 108 macro avg 0.75 0.69 0.68 108 weighted avg 0.75 0.70 0.68 108	70%	Confusion Matrix: [[23 28] [4 53]]