Model Development Phase Template

Date	23 September 2024
Team ID	LTVIP2024TMID24967
Project Title	SmartLender - Applicant Credibility Prediction for Loan Approval
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:

```
#importing and building the random forest model
def RandomForest(X_tarin,X_test,y_train,y_test):
    model = RandomForestClassifier()
    model.fit(X_train,y_train)
    y_tr = model.predict(X_train)
    print(accuracy_score(y_tr,y_train))
    yPred = model.predict(X_test)
    print(accuracy_score(yPred,y_test))
```

```
#printing the train accuracy and test accuracy respectively
RandomForest(X_train,X_test,y_train,y_test)
```

```
#importing and building the Decision tree model
def decisionTree(X_train,X_test,y_train,y_test):
    model = DecisionTreeClassifier()
    model.fit(X_train,y_train)
    y_tr = model.predict(X_train)
    print(accuracy_score(y_tr,y_train))
    yPred = model.predict(X_test)
    print(accuracy_score(yPred,y_test))
```

```
#printing the train accuracy and test accuracy respectively
decisionTree(X_train,X_test,y_train,y_test)
```

```
#importing and building the KNN model
def KNN(X_train,X_test,y_train,y_test):
    model = KNeighborsClassifier()
    model.fit(X_train,y_train)
    y_tr = model.predict(X_train)
    print(accuracy_score(y_tr,y_train))
    yPred = model.predict(X_test)
    print(accuracy_score(yPred,y_test))
```

```
#printing the train accuracy and test accuracy respectively
KNN(X_train,X_test,y_train,y_test)
```

```
#importing and building the Xg boost model
def XGB(X_train,X_test,y_train,y_test):
    model = GradientBoostingClassifier()
    model.fit(X_train,y_train)
    y_tr = model.predict(X_train)
    print(accuracy_score(y_tr,y_train))
    yPred = model.predict(X_test)
    print(accuracy_score(yPred,y_test))
```

```
#printing the train accuracy and test accuracy respectively
XGB(X_train,X_test,y_train,y_test)
```

Model Validation and Evaluation Report:

Model	Classification Report	F1 Score	Confusion Matrix
Random Forest	Classification report: precision recall f1-score support 0 0.92 0.73 0.81 74 1 0.76 0.93 0.84 70 accuracy 0.83 144 macro avg 0.84 0.83 0.83 144 weighted avg 0.84 0.83 0.82 144	82%	Confusion matrix: [[54 20] [5 65]]

Decision Tree	Classification report precision recall f1-score support 0 0.81 0.73 0.77 74 1 0.74 0.81 0.78 70 accuracy 0.77 144 macro avg 0.77 0.77 0.77 144 weighted avg 0.77 0.77 0.77 144	77%	Confusion matrix [[54 20] [13 57]]
KNN	Classification report: precision recall f1-score support 0 0.87 0.61 0.71 74 1 0.68 0.90 0.78 70 accuracy 0.75 144 macro avg 0.78 0.75 0.75 144 weighted avg 0.78 0.75 0.75 144	75%	Confusion matrix: [[45 29] [7 63]]
Gradient Boosting	Classification report: precision recall f1-score support 0 0.90 0.59 0.72 74 1 0.68 0.93 0.79 70 accuracy 0.76 144 macro avg 0.79 0.76 0.75 144 weighted avg 0.79 0.76 0.75 144	75%	Confusion matrix: [[44 30] [5 65]]