

Model Development Phase Template

Date	10 July 2024
Team ID	740092
Project Title	Credit card approval prediction by using ML
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:

```
#LOGISTIC REGRESSION
def logistic_reg(xtrain,xtest, ytrain, ytest):
    lr=LogisticRegression(solver="liblinear")
    lr.fit(xtrain, ytrain)
    ypred=lr.predict(xtest)
    print("*****LogisticRegression*****")
    print("Confusion matrix")
    print(confusion_matrix(ytest,ypred))
    print("Classification report")
    print(classification_report(ytest, ypred))
```

```
#RANDOM FOREST
def random_forest (xtrain,xtest, ytrain, ytest):
    rf=RandomForestClassifier()
    rf.fit(xtrain, ytrain)
    ypred=rf.predict(xtest)
    print("*****Random ForestClassifier*****")
    print("Confusion matrix")
    print(confusion_matrix(ytest,ypred))
    print("Classification report")
    print(classification_report(ytest,ypred))
```

```
#DECISION TREE
def d_tree (xtrain, xtest, ytrain, ytest):
    dt=DecisionTreeClassifier()
    dt.fit(xtrain, ytrain)
    ypred=dt.predict(xtest)
    print("***DecisionTreeClassifier***")
    print('Confusion matrix')
    print(confusion_matrix(ytest,ypred))
    print("Classification report")
    print(classification_report (ytest, ypred))
```

```
#GRADIENT BOOSTING
def g_boosting(xtrain, xtest, ytrain, ytest):
    gb=GradientBoostingClassifier()
    gb.fit(xtrain, ytrain)
    ypred=gb.predict(xtest)
    print("***GradientBoostingClassifier***")
    print("Confusion matrix")
    print(confusion_matrix(ytest, ypred))
    print("Classification report")
    print(classification_report(ytest,ypred))
```

Model	Classification Report	F1 Score	Confusion Matrix
-------	-----------------------	----------	------------------

Random Forest	<pre> precision recall f1-score support Not Approved 0.80 0.85 0.82 500 Approved 0.83 0.78 0.80 500 accuracy 0.81 1000 macro avg 0.81 0.81 0.81 1000 weighted avg 0.81 0.81 0.81 1000 </pre>	81%	<pre> Confusion matrix [[2617 75] [199 2136]] print(confusion_matrix(ytest,ypred)) </pre>
---------------	---	-----	---

Model Validation and Evaluation Report:

Decision Tree	<pre> print(classification_report (ytest, ypred)) </pre> <pre> precision recall f1-score support 0 0.99 1.00 1.00 2692 1 1.00 0.99 1.00 2335 accuracy 1.00 5027 macro avg 1.00 1.00 1.00 5027 weighted avg 1.00 1.00 1.00 5027 </pre>	79%	<pre> print("Classification report") Confusion matrix [[2685 7] [15 2320]] </pre>
Logistic Regression	<pre> Classification report precision recall f1-score support 0 0.93 0.97 0.95 2692 1 0.97 0.91 0.94 2335 accuracy 0.95 5027 macro avg 0.95 0.94 0.94 5027 weighted avg 0.95 0.95 0.95 5027 </pre> <pre> print(classification_report(ytest, ypred)) </pre>	64%	<pre> confusion_matrix(y_test,ypred) array([[43, 32], [29, 65]]) </pre>

Gradient Boosting	<pre>print(classification_report(ytest,ypred))</pre> <pre> Classification report precision recall f1-score support 0 1.00 1.00 1.00 2692 1 1.00 1.00 1.00 2335 accuracy macro avg 1.00 1.00 1.00 5027 weighted avg 1.00 1.00 1.00 5027 </pre>	78%	<pre>confusion_matrix(y_test,ypred)</pre> <pre> array([[63, 12], [26, 68]]) </pre>
-------------------	--	-----	---