Model Development Phase Template

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 Model
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report

logistic_regression = LogisticRegression()
logistic_regression.fit(x_train, y_train)
y_pred = logistic_regression.predict(x_test)

acc_lr = accuracy_score(y_test, y_pred)
c_lr = classification_report(y_test, y_pred)

print('Accuracy score: ', acc_lr)
print(c_lr)
```

```
#Random Forest Model
from sklearn.ensemble import RandomForestClassifier

random_forest = RandomForestClassifier()
random_forest.fit(x_train, y_train)
y_pred = random_forest.predict(x_test)

acc_rf = accuracy_score(y_test, y_pred)
c_rf = classification_report(y_test, y_pred)

print('Accuracy score: ', acc_rf)
print(c_rf)
```

```python
#Decision Tree Model
from sklearn.tree import DecisionTreeClassifier

decision_tree_model = DecisionTreeClassifier()
decision_tree_model.fit(x_train, y_train)
y_pred = decision_tree_model.predict(x_test)

acc_dt = accuracy_score(y_test, y_pred)
c_dt = classification_report(y_test, y_pred)

print('Accuracy score: ', acc_dt)
print(c_dt)
```

```python
# Create an XGBoost classifier
xgboost_model = xgb.XGBClassifier(use_label_encoder=False, eval_metric='logloss')
xgboost_model.fit(x_train, y_train)
y_pred = xgboost_model.predict(x_test)

acc_xgb = accuracy_score(y_test, y_pred)
c_xgb = classification_report(y_test, y_pred)

print('Accuracy score: ', acc_xgb)
print(c_xgb)
```

```python
#Gaussian Navies Bayes
from sklearn.naive_bayes import GaussianNB

NB = GaussianNB()
NB.fit(x_train,y_train)
y_pred = NB.predict(x_test)

acc_nb = accuracy_score(y_test,y_pred)
c_nb = classification_report(y_test,y_pred)

print('Accuracy score: ', acc_nb)
print(c_nb)
```

```
#Gradient Boosting Classifier
from sklearn.ensemble import GradientBoostingClassifier

GCB = GradientBoostingClassifier()
GCB.fit(x_train,y_train)
y_pred = GCB.predict(x_test)

acc_gcb = accuracy_score(y_test,y_pred)
c_gcb = classification_report(y_test,y_pred)

print('Accuracy Score: ', acc_gcb)
print(c_gcb)
```

Model Validation And Evaluation Report:

	Model	score
0	Logistic Regression	0.991935
1	Decision Tree Classifier	1.000000
2	Random Forest Classifier	1.000000
3	Gaussian Naive Bayes	0.979839
4	Support Vector Classifier	0.939516
5	Gradient Boost Classifier	1.000000
6	XGboost	1.000000