Model Development Phase Template

Date	15 March 2024
Team ID	740012
Project Title	**Predicting IMF-Based Exchange Rates: Leveraging Economic Indicators for Accurate Regression Modeling**
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:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
from sklearn.linear_model import Lasso
import xgboost as xgb
```

> Random Forest Regressor

```
model4=RandomForestRegressor(n_estimators=100, random_state=42)
model4.fit(X_train,y_train)
```

```
y_pred=model4.predict(X_test)
mse=mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test,y_pred)
print("Mean Squared Error:", mse)
print("R-squared Score:", r2)
```

```
Mean Squared Error: 43.42457074286648
R-squared Score: 0.9796780547753188
```

```
kmeans = KMeans(n_clusters=3)

kmeans.fit(df)
centroids = kmeans.cluster_centers_
labels = kmeans.labels_

plt.figure(figsize=(8, 6))
plt.scatter(centroids[:, 0], centroids[:, 1], c='red', s=200, marker='x', label='Centroids')

plt.title('K-means Clustering')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.legend()
plt.grid(True)
plt.show()
```

Model Validation and Evaluation Report

Logistic Regression Model	Logistic Regression Classification_Report `print(classification_report(reg_pred , y_test))` precision recall f1-score support 0.0 0.76 0.75 0.75 402 1.0 0.75 0.75 0.75 398 accuracy 0.75 800 macro avg 0.75 0.75 0.75 800 weighted avg 0.75 0.75 0.75 800

Random forest Model	RandomForest Classification_Report
	`from sklearn.metrics import classification_report`
	`print(classification_report(forest, y_test))`
	precision recall f1-score support
	0.0 0.91 0.91 0.91 399
	1.0 0.91 0.91 0.91 401
	accuracy 0.91 800
	macro avg 0.91 0.91 0.91 800
	weighted avg 0.91 0.91 0.91 800

Decision Tree Model	DecisionTree Classification_Report `print(classification_report(dt_pred , y_test))` precision recall f1-score support 0.0 0.81 0.80 0.80 404 1.0 0.80 0.80 0.80 396 accuracy 0.80 800 macro avg 0.80 0.80 0.80 800 weighted avg 0.80 0.80 0.80 800
Gradient Boosting	XGBoost Classification_Report `print(classification_report(y_pred, y_test))` precision recall f1-score support 0 0.91 0.92 0.91 395 1 0.92 0.91 0.92 405 accuracy 0.92 800 macro avg 0.92 0.92 0.91 800 weighted avg 0.92 0.92 0.92 800