

Model Development Phase Template

Date	20 June 2024
Team ID	739946
Project Title	Determine : Loan from KIVA crowdfunding data
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

Initial model training code prepares and trains a predictive model. Model validation and evaluation report assesses performance metrics like accuracy and AUC, ensuring the model's reliability and effectiveness.

Initial Model Training Code:

Paste the screenshot of the model training code

```
# Select relevant features and target variable
features = ['funded_amount', 'clean_borrower_genders', 'term_in_months', 'lender_count', 'loan_amount_log', 'sector', 'activity', 'region', 'country', 'country_code']
target = 'loan_amount'
# Import necessary libraries
from sklearn.preprocessing import LabelEncoder

# Select relevant features and target variable
features = ['funded_amount', 'clean_borrower_genders', 'term_in_months', 'lender_count', 'loan_amount_log', 'sector', 'activity', 'region', 'country', 'country_code']
target = 'loan_amount'

# Initialize LabelEncoder
label_encoder = LabelEncoder()
df['clean_borrower_genders'] = label_encoder.fit_transform(df['clean_borrower_genders'])
df['sector'] = label_encoder.fit_transform(df['sector'])
df['activity'] = label_encoder.fit_transform(df['activity'])
df['region'] = label_encoder.fit_transform(df['region'])
df['country'] = label_encoder.fit_transform(df['country'])
df['country_code'] = label_encoder.fit_transform(df['country_code'])

# Encode the target variable if it is categorical
df[target] = df[target].astype('category').cat.codes

# Prepare the feature matrix (X) and target vector (y)
X = df[features]
y = df[target]
```

```
# Kiva (9).ipynb
File Edit View Insert Runtime Tools Help Last edited on July 11

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

# Standardize the features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Create an instance of the Logistic Regression model
logit = LogisticRegression()

# Train the model
logit.fit(X_train, y_train)


# Predict the results
y_pred = logit.predict(X_test)

# Check the accuracy on the training and testing sets
train_accuracy = sum(logit.predict(X_train) == y_train) / len(X_train)
test_accuracy = sum(logit.predict(X_test) == y_test) / len(X_test)

# Print the accuracy
print("Training set accuracy: ", train_accuracy)
print("Testing set accuracy: ", test_accuracy)

# Evaluate and print the accuracy on the testing set
accuracy = test_accuracy
print("Accuracy: ", accuracy)
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Randomforest Classifier, Decisiontree Classifier		93%	---
...