Kiva Crowdfunding: A Machine Learning Analysis

Andrea Piolini, Caroline Sklaver, Anwesha Tomar, Sandra Valdes Salas

Background

- "Kiva.org is an online crowdfunding platform to extend financial services to poor and financially excluded people around the world. Kiva lenders have provided over $1 billion dollars in loans to over 2 million people."
- Dataset of loans issued over the last two years:
    - https://www.kaggle.com/kiva/data-science-for-good-kiva-crowdfunding

Data

Column descriptions:

- idUnique - ID for loan
- funded_amount - The amount disbursed by Kiva to the field agent(USD)
- loan_amount - The amount disbursed by the field agent to the borrower(USD)
- activity - More granular category
- sector - High level category
- useExact - usage of loan amount
- country - Full country name of country in which loan was disbursed
- region - Full region name within the country
- posted_time - The time at which the loan is posted on Kiva by the field agent
- disbursed_time - The time at which the loan is disbursed by the field agent to the borrower
- funded_time - The time at which the loan posted to Kiva gets funded by lenders completely
- term_in_months - The duration for which the loan was disbursed in months
- lender_count - The total number of lenders that contributed to this loan
- borrower_genders - Comma separated M,F letters, where each instance represents a single male/female in the group
- repayment_interval – monthly, bullet,
- loan_theme_id - Unique ID for loan theme
- loan_theme_type - General description of the loan theme category

Merged MPI data

- MPI – Multidimensional Poverty Index

Goals

- Determine the most important features
- Build a model to predict the loan amount
- Build a model to predict repayment interval

Data Cleaning and Processing

Countries: grouped 87 countries into 14 groups dividing them by world region. Eventually we decided to focus on Latin America.

Activities: grouped over 160 activities into 14 groups dividing them by sector (i.e. education, healthcare).

Borrowers’ gender: the "borrowers_gender" column includes the gender of all the people who benefitted from a specific loan – hence many entries contains both males and females.

Split the column into two columns that count the number of males and females that benefitted from each loan.

Data Cleaning and Processing (continued)

Unnecessary columns: dropped 13 columns that were not relevant for the analysis.

Identifiers : dropped all the identifiers in the dataset.

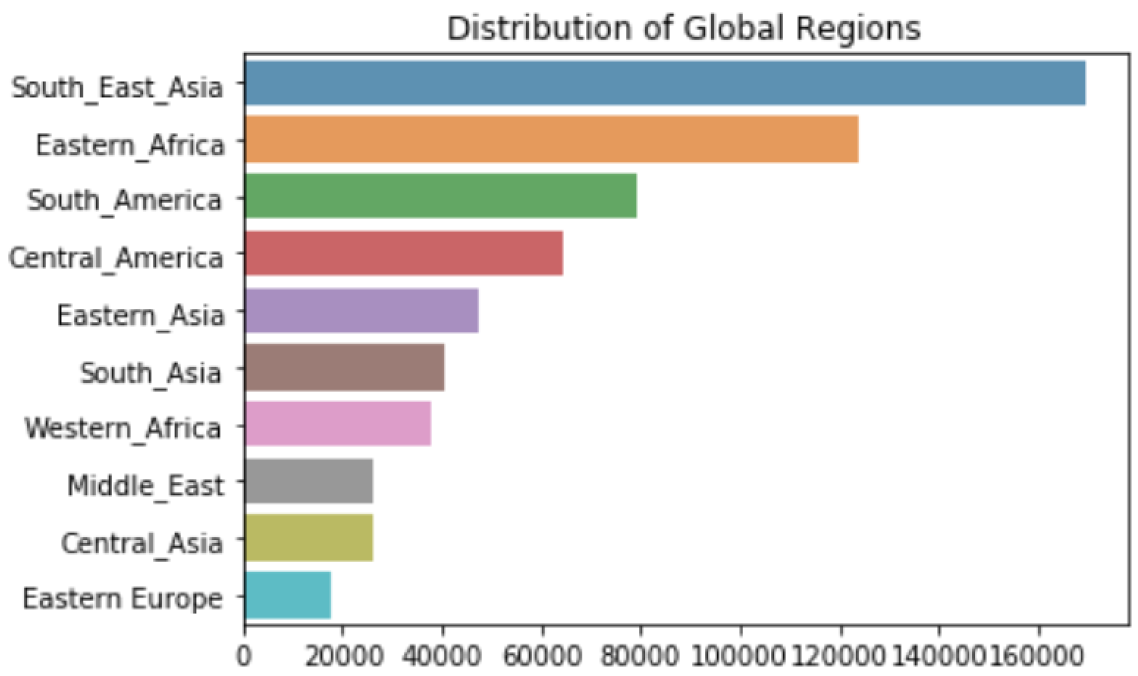
NaN

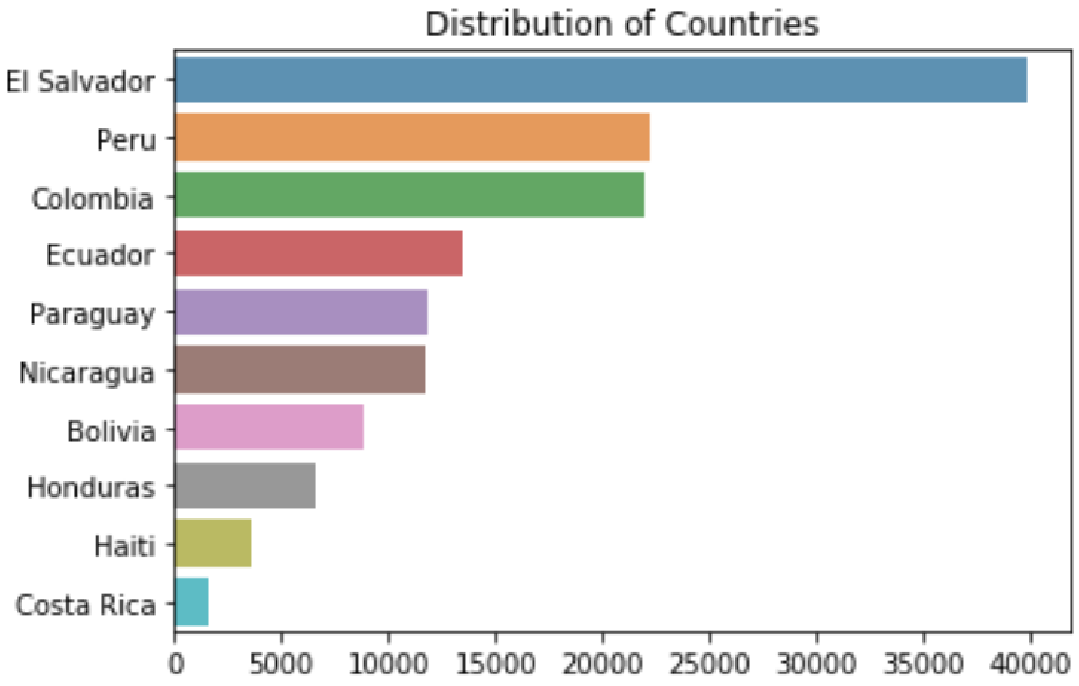
Missing data: the datasets does not have many NaNs. We dropped them for all the categorical variables.

A→0,1

Categorical variables: converted categorical data into numerical data using One-Hot-Encoder.

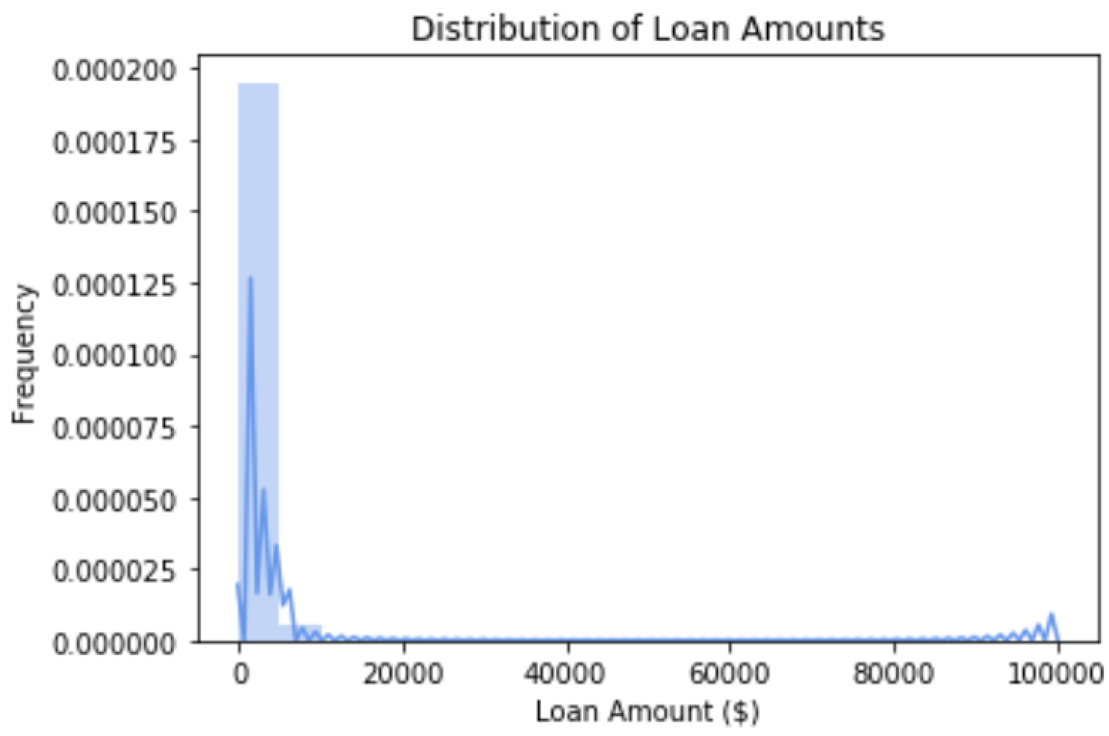
Exploratory Data Analysis

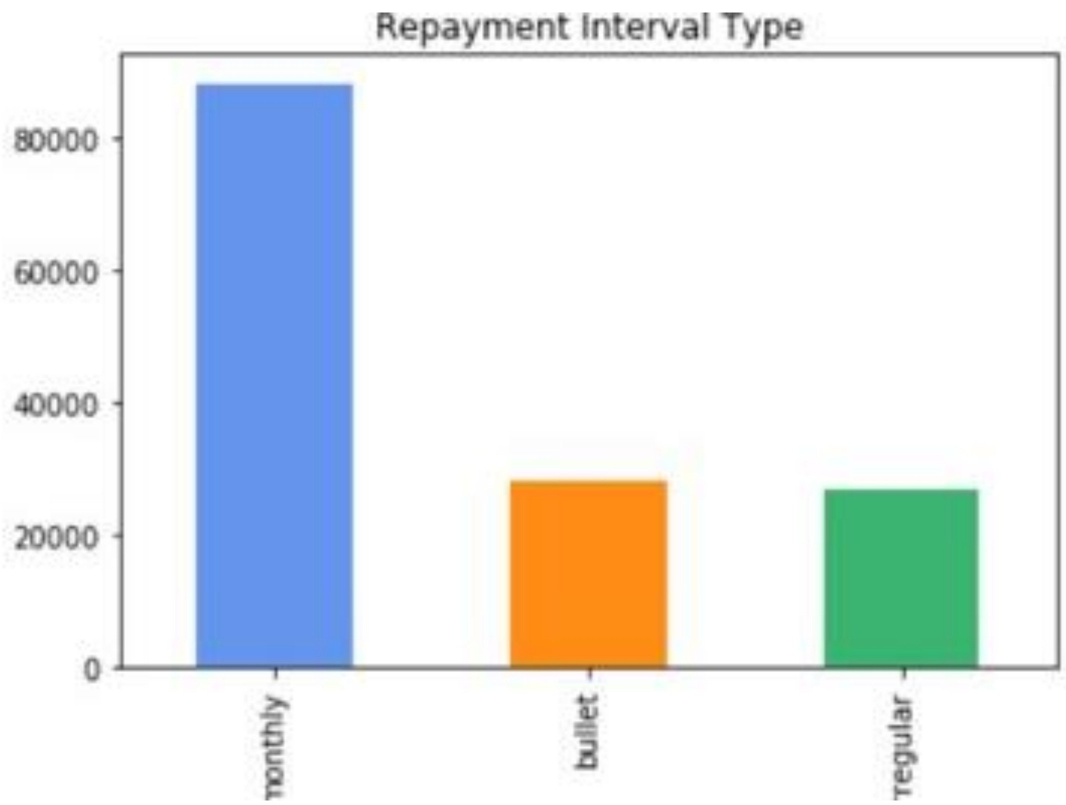


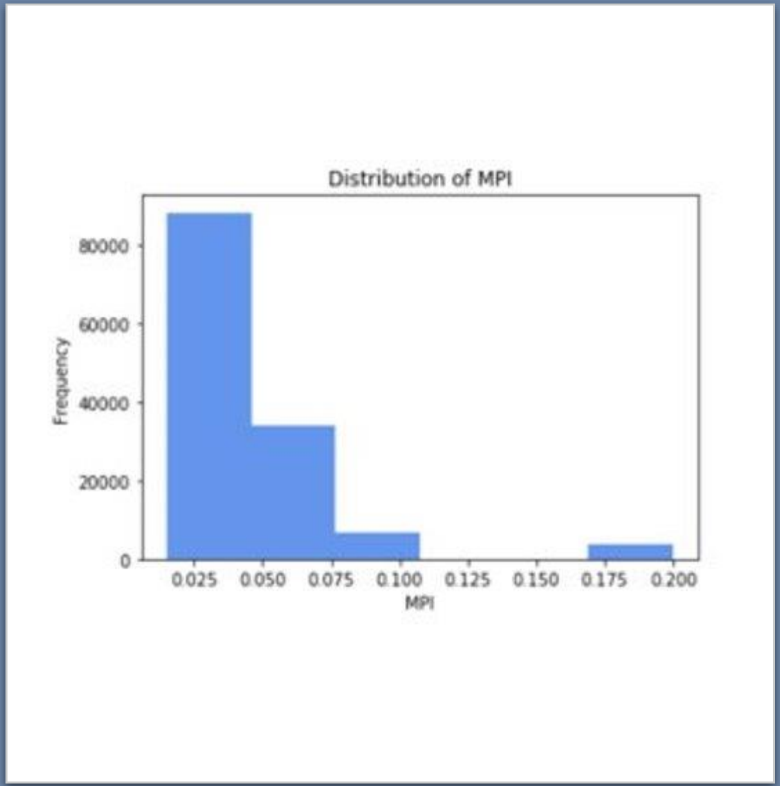


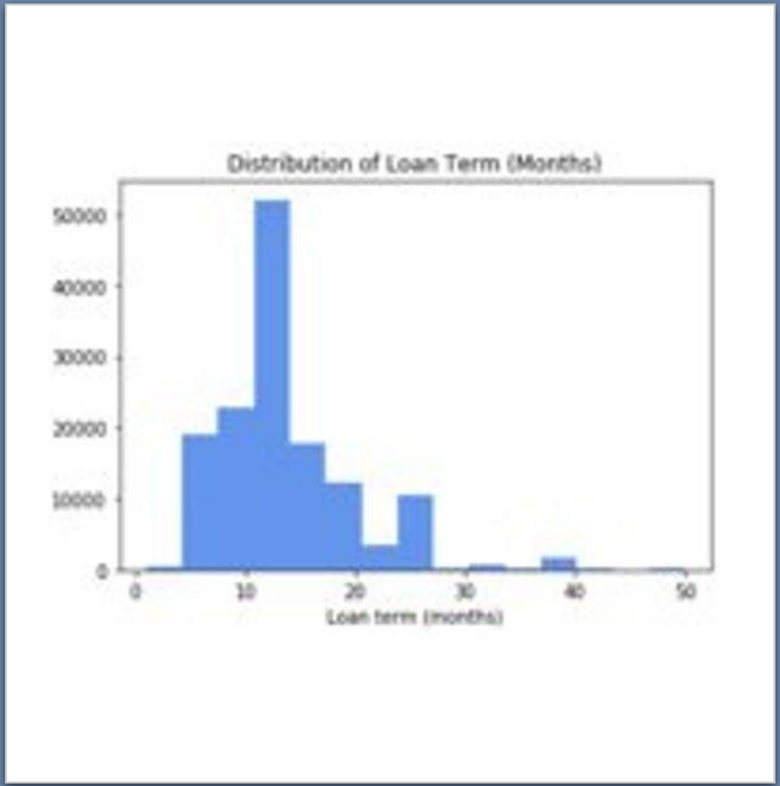
EDA

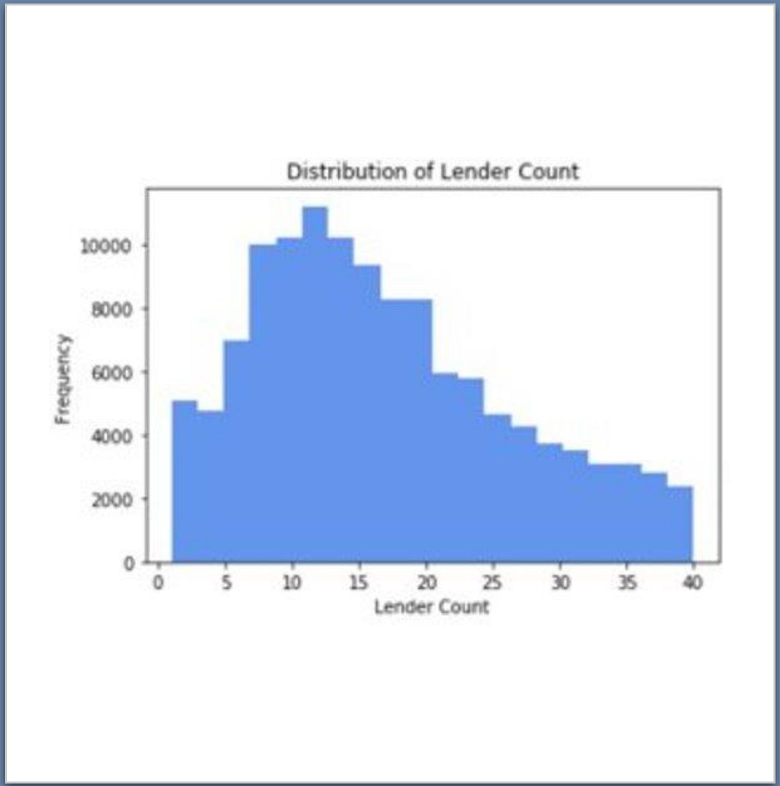
- Target Variables









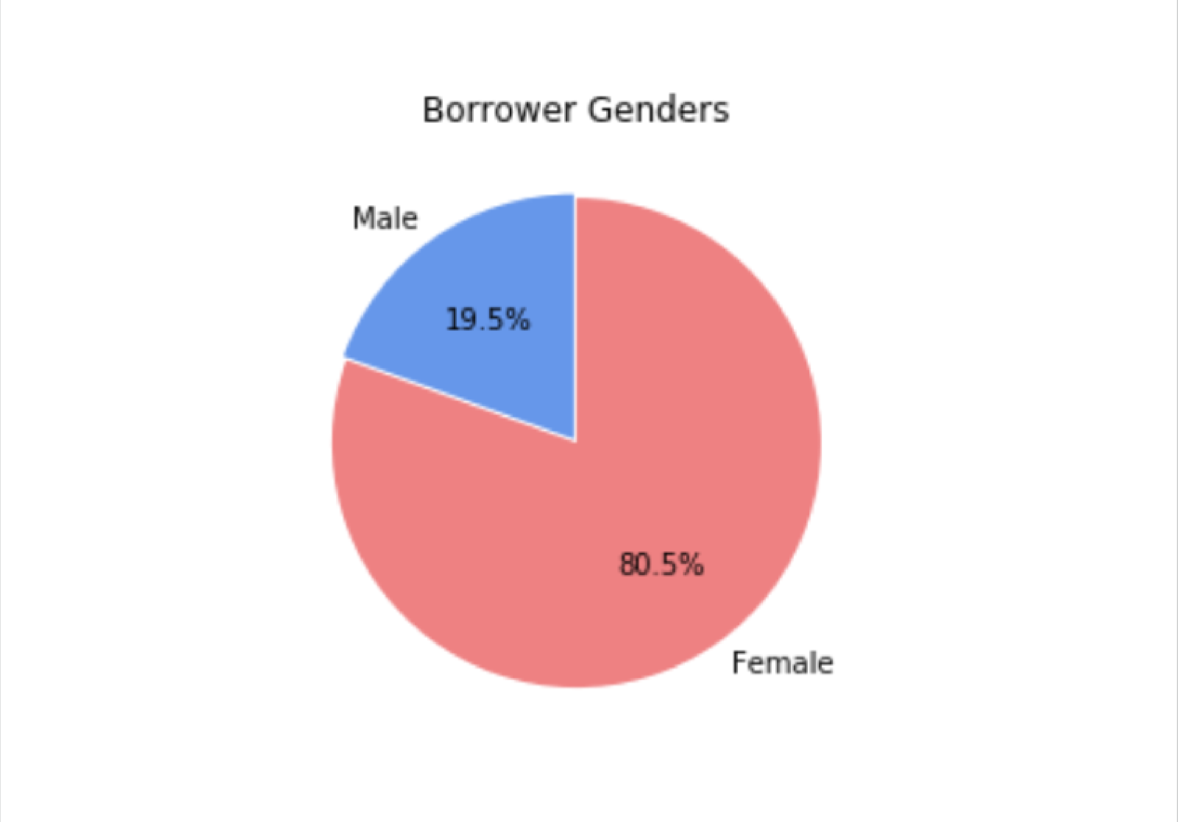


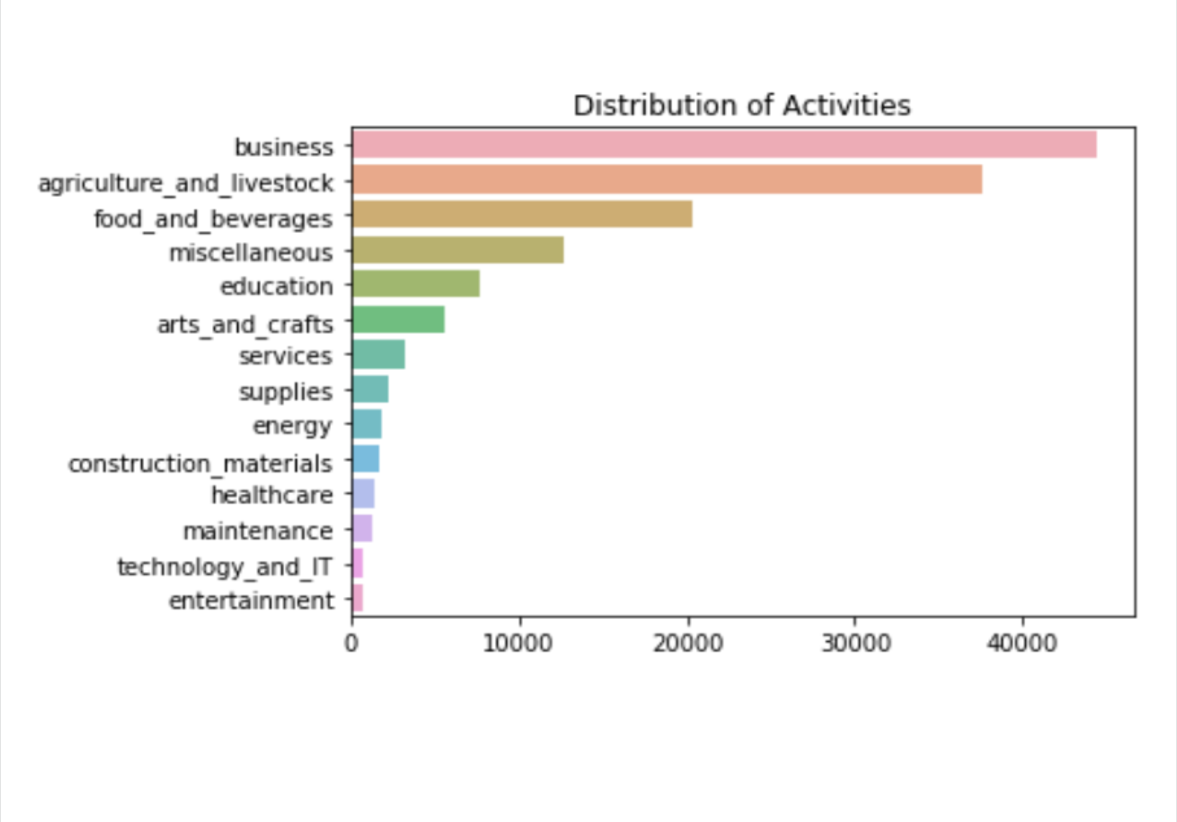
EDA

Feature Variables

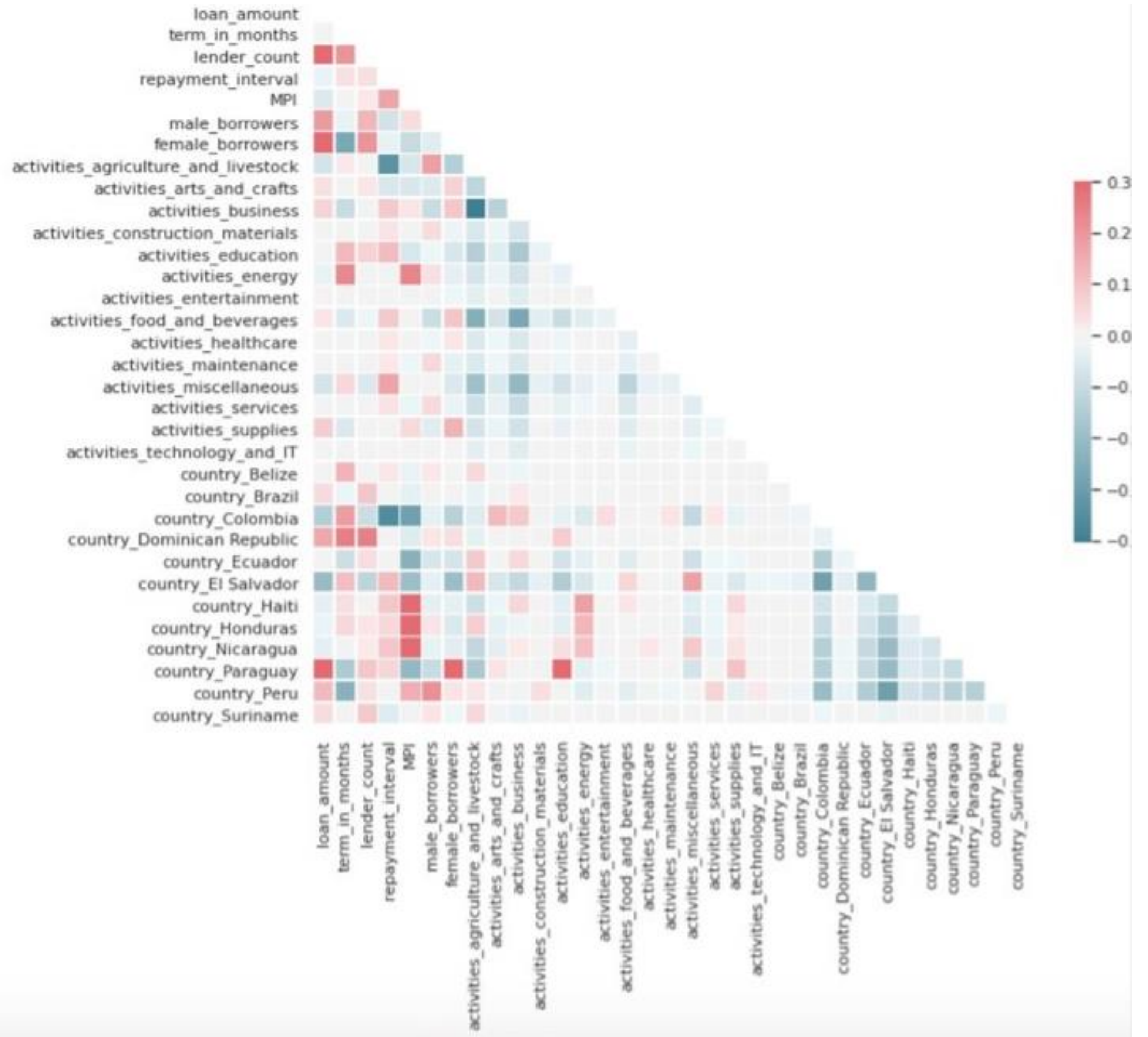
EDA

Feature Variables Cont.





Correlation Plot



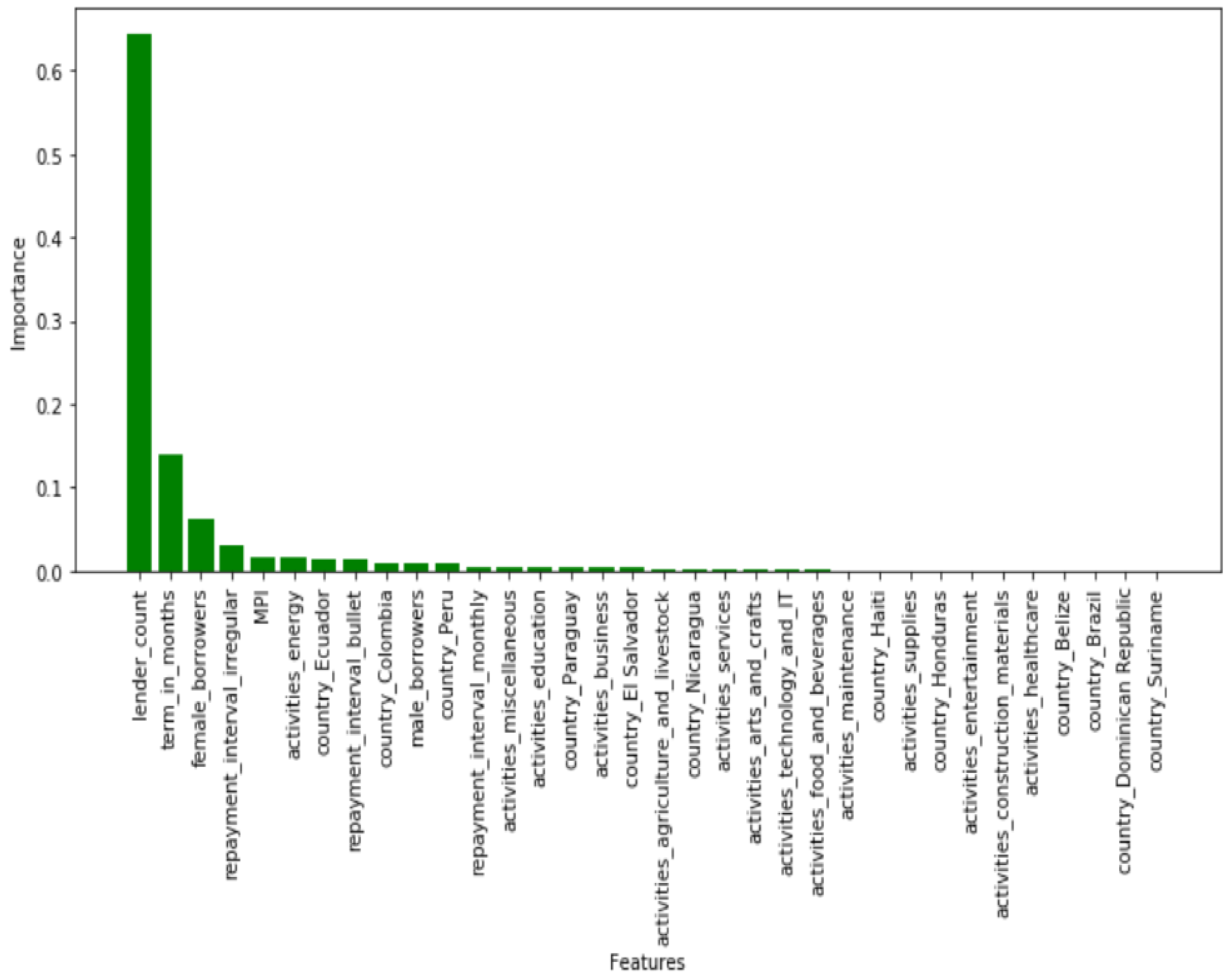
Regression

- Loan amount
- **Data preparation:** removed all the rows with outliers in the target value
- Models: Decision Tree Regressor, Random Forest Regressor and MLP Regressor
- Hyperparameter Tuning
- Calculated the Negative Mean Squared Error

Regression: Results

Best score	Best parameter	Best Estimator
-46771.35	Min samples leaf: 1 Min samples split: 20	**Random Forest Regressor**
-47037.97	Learning rate: 0.01 Alpha: 0.0001	MLP Regressor
-48428.85	Max depth: 25 Min samples leaf: 30 Min samples split: 100	Decision Tree Regressor

Regression: feature importance



Regression: Analysis of the Results

Why it did not work properly?

- The explanatory power of most of the features is low.
- The dataset might contain poor-quality data (i.e. corrupt data, inaccurate data, or incomplete data).
- Hyperparameters might not have been fine-tuned in the most efficient way.

Some Potential Solutions to Improve the Model:

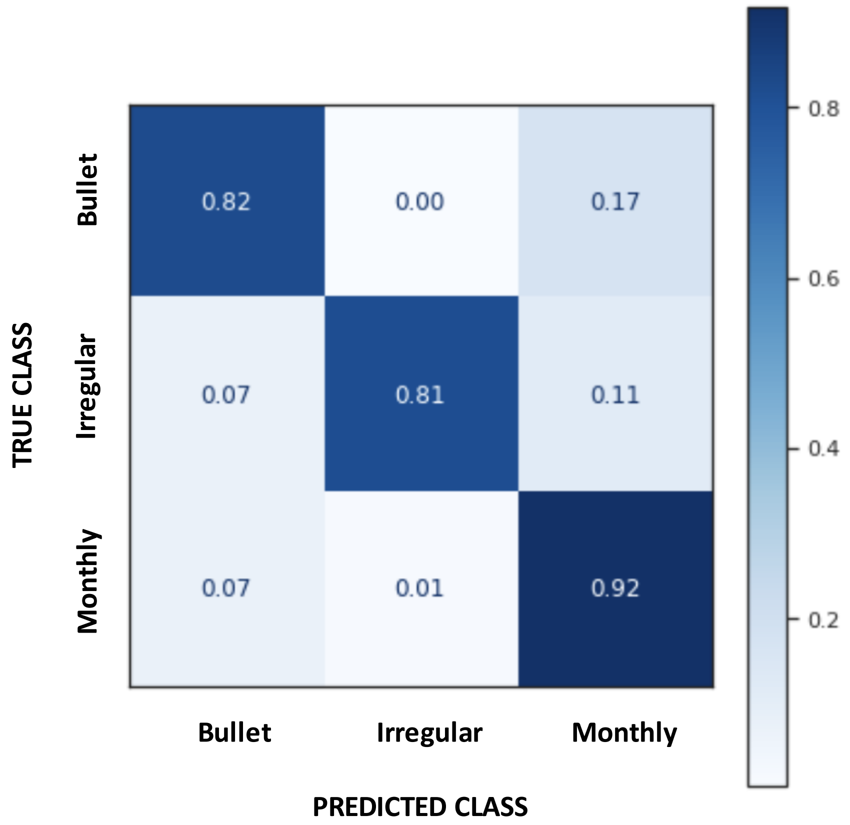
- Use the whole dataset to build the model.
- Merge the main dataset with other dataset to increase the number of features.
- Refine the fine-tuning strategies.
- Turn the model into a classification model.

Classification

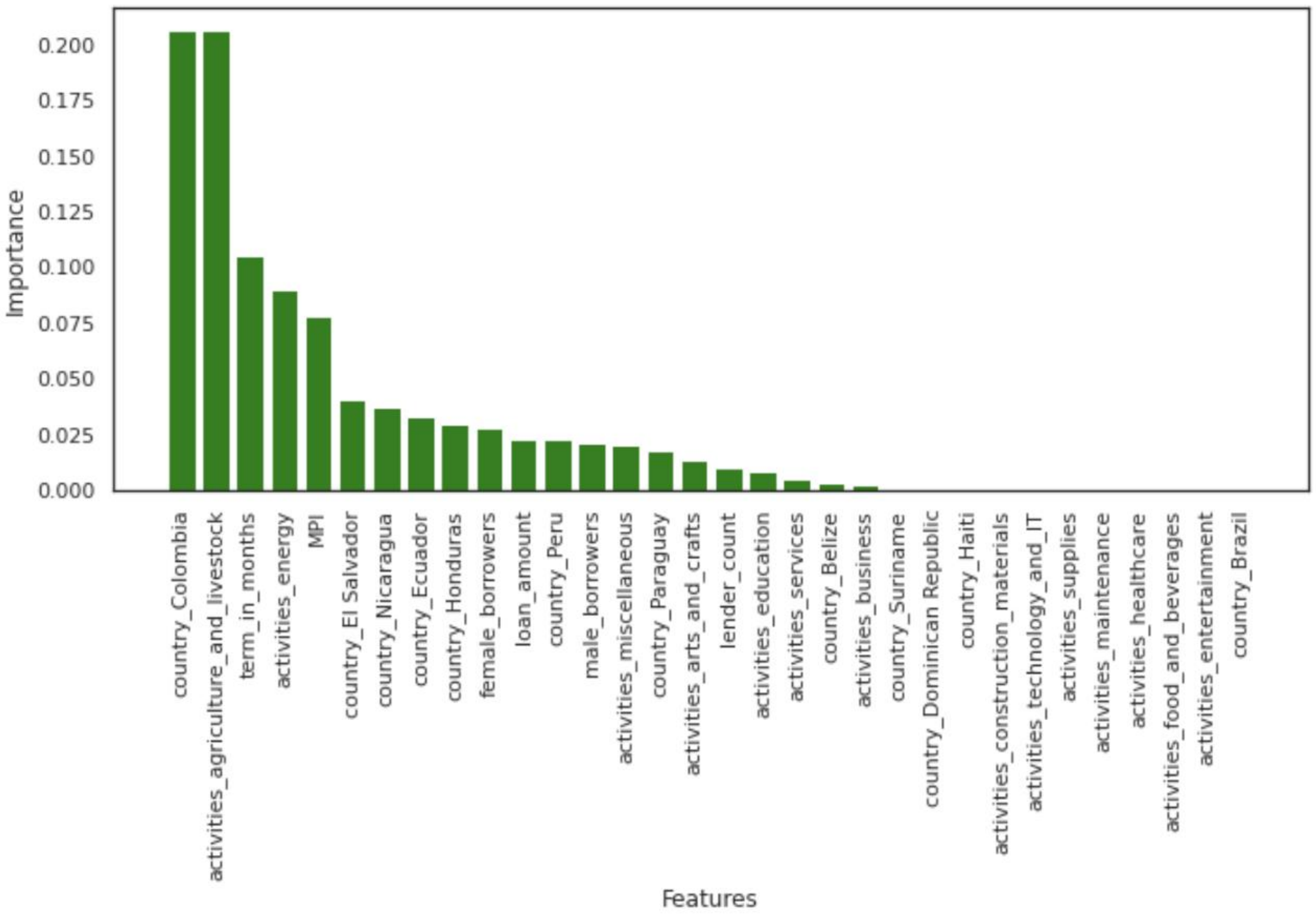
- Repayment Interval as target (3 classes: mothly, bullet, irregular)
- Data preparation: encoding and scaling
- Models: **Logistic Regression, Decision Tree, Random Forest, HGBC, XGBC, MLPC**
- Hyperparameter Tuning: GridSearchCV
- Feature Importance

Classification: Results

Best score	Best parameter	Best Estimator
0.873133	**Learning rate: 0.1 Min samples leaf: 20**	**Hist Gradient Boosting Classifier**
0.852161	Eta: 0.0001 Gamma: 10 Lambda: 0.0001	XGB Classifier
0.849966	Min samples leaf: 1 Min samples split: 20	Random Forest
0.845943	Alpha: 0.001 Learning rate: 0.01	MLP Classifier
0.840456	Max depth: 10 Min samples leaf: 1 Min samples split: 30	Decision Tree
0.706334	C grids: 10 Tol grids: 1e-6	Logistic Regression



Classification: feature importance



Multi-Target Classification and Prediction

- Selected two targets: repayment interval and activities
- Split the entire data based on features and targets
- Generated dummies for features data set
- Generated custom labels for target data set
- Implemented KNN and Random Forest
- Used user input to predict repayment interval and activities

Conclusions

We were able to successfully pre-process the data and perform EDA

Regression models had low accuracies, the reason for this may be the nature of features chosen

The classification models performed well; the best classification model was HGB with the best score of 0.874

The next best models were Random Forest and MLP

Lastly, we performed multi-target classification and performed prediction using user input

Further Steps

- Further, explore multi-target classification by splitting data into testing and training for the analysis
- Calculate the accuracies for multi-target prediction
- Explore merging additional datasets to improve results

References

- https://www.kaggle.com/kiva/data-science-for-good-kiva-crowdfunding
- http://hdr.undp.org/en/data
- https://scikit-learn.org/stable/modules/model_evaluation.html