Georgetown University Data Science Certificate Program, Cohort 13

Kiva

Fast Money Helper - Report

Team members:

Thomas O'Neil (*Team Spokesperson*)
Yamil Vargas
Nuredin Abdella
Won Choe

Table of Contents

Introduction

This project creates a model that predicts if a Kiva loan request will be fully funded within 5 days. Our group investigated data from a microlending organization called Kiva (https://www.kiva.org). A microcredit is an extension of very small loans to impoverished borrowers who typically lack collateral, steady employment, or a verifiable credit history. Microcredit, which is funded by individuals (peer to peer lending) rather than banks, is designed to support entrepreneurship and alleviate poverty. Many borrowers are illiterate, and unable to complete paperwork required to get conventional loans (*Microcredit,* Wikipedia).

Kiva is an international nonprofit, founded in 2005 and based in San Francisco, with a mission to connect people through lending to alleviate poverty. Their stated goal is to "celebrate and support people looking to create a better future for themselves, their families and their communities. Kiva operates in the microfinance space through the Internet and connects borrowers and lenders using financial institutions in the borrower's country" ("*About Kiva",* Kiva). In turn, internet users from North America, Europe, and beyond make small loans via PayPal to borrowers and borrowers pay the lenders back in about a year ("*Kiva and the Birth of Person-to-Person Microfinance",* MIT Innovations). Kiva was chosen because it provides robust financial and descriptive data such as loan amount, funded amount, borrower country, gender, sector, loan types, and loan description that can be analyzed.

Hypothesis and Application

Our original goal was to look into determinants of the loan repayment and identify factors and causes of loan default. In turn, we planned to build a data product that will predict the probability of success or failure of a loan application to help lenders make an informed decision and thereby improve repayment rates. However, after exploratory analysis of the data, we discovered that Kiva stopped publishing the status of individual loans due to privacy concerns.

As a result, we redefined the purpose of the project and focused on the borrower side to identify factors that determine faster funding of the loans. The data product is also aimed at helping borrowers predict how fast they will get funded by providing certain information.

Being funded in a timely manner is crucial for borrowers in Kiva as loan requests expire in 30 days. We explored variables that affect the time it takes for a loan to be funded. Specifically, we sought to create a model that predict if a loan request will be funded within 5 days and tease out the determinants of funding speed.

Project Pipeline

Data Ingestion

Kiva provides a wealth of information on its developer website and full datasets of the loan information (http://build.kiva.org/). We downloaded of three full datasets that were readily available in a snapshot: Loans, Lenders, and Loan To Lender Relationships. There are APIs on the website to derive data not found in the datasets, including partners information.

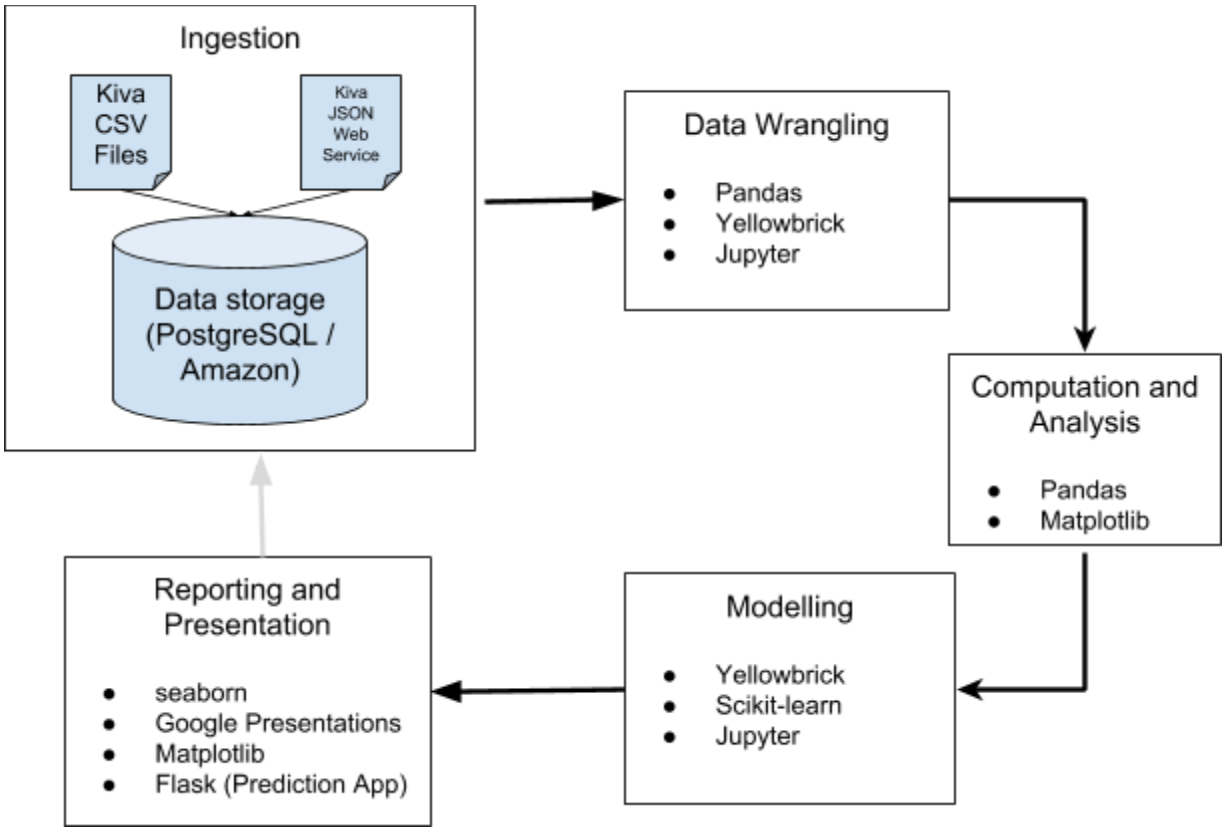
As a result, we decided to focus on three datasets: Loans, Lenders, and Lenders to Loans relationship tables. We had to derive table schemas on our own from investigating the data.

We began by building software to download and ingest data from Kiva using its snapshot data (CSV files contained in a ZIP file). We loaded this data into a PostgreSQL database stored in Amazon RDS. We used our schemas to create tables and we uploaded the data into the cloud.

Our team then decided it would be good to have partners information which was not included with the data snapshot. The partners data was pulled from Kiva's JSON APIs and converted into a SQL table. We were one of those teams who accidentally uploaded their AWS information. We have now configured a python script to load our connection string from a non-github file.

Our architecture for this project is as follows:

Figure 1. Project Pipeline



Wrangling

Our original target was helping a borrower achieve a fully funded loan. After reading the API documentation, we thought helping the lenders choose a loan that will be fully repaid would be more exciting.

During the wrangling process, we quickly discovered a flaw. Kiva stopped sending the repayment information about loans. This was our initial check of the Kiva loan statuses:

Figure 2. Loan Status

	status	count
0	expired	59081
1	funded	1350340
2	fundRaising	3608
3	refunded	6578

We were missing two key statuses (from the Kiva API documents):

Figure 3. Loan Status from Kiva

> **paid**
> The loan has been paid back in full by the borrower. The payments have been distributed back to the lenders and the loan is closed to most new activity on Kiva.
>
> **defaulted**
> Occasionally, a borrower or a field partner may fail to make payments on a loan, either to the field partner or to Kiva, respectively. Usually when this happens, a loan simply becomes delinquent and remains in the in_repayment status. When a loan remains delinquent 6 months after the end of the loan payment schedule, the loan becomes defaulted. Usually defaulted loans will never be paid back and are a financial loss to the lenders to that loan. Most loans only default in part, but it is possible for the entire amount of the loan to not be repaid.

We also noticed that most loans are fully funded. Hence, our target needed some adjustment.

We wrangled the data into another table called “loan_dates” and included a field that calculated days from posting to getting the money raised (or funded).

Exploratory Analysis

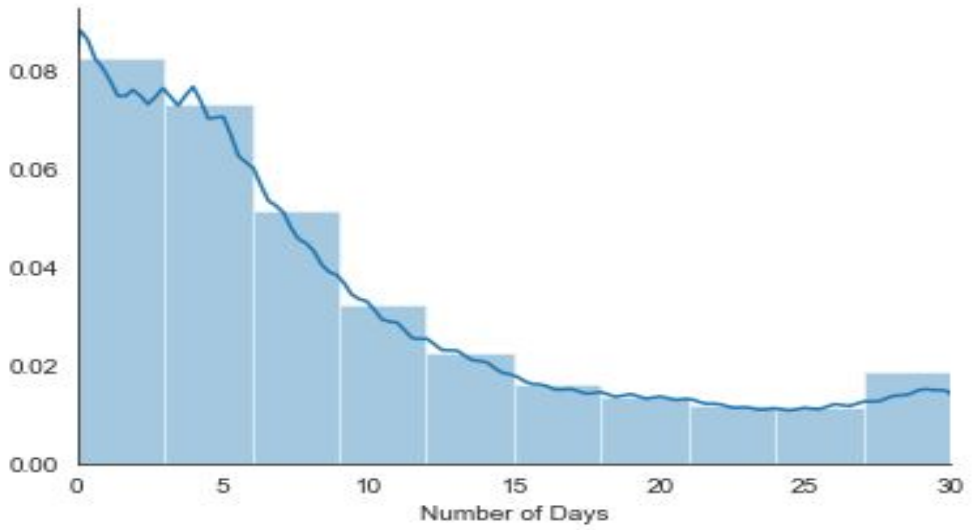
We used Jupyter Notebook, Pandas, and Matplotlib to check for data validity. These tools helped us define datasets for modeling and testing.

There were issues with the exploratory stage. We discovered our data frames were out of sync and it was hard to compare our results. We developed a class to process the sql and clean up the rest of the dataframe. The procedure would take roughly five minutes. While not ideal for most environments, it worked well for Jupyter notebooks in a shared environment.

After having a single version of the wrangled data it was easier to do our explanatory analysis. We ended up with a database containing 1,177,384 loan applications that were funded between 2006 and 2018 (12 years worth of data). The loans were originated from 96 countries and then used to finance different sectors of the economy, ranging from agriculture to manufacturing and retailing (15 sectors in total). Additionally, the data show that 93% of the funded loans were less than $2,000 dollars, with an average amount of $758 dollars.

We also used the exploratory phase to look into the target variable: number of days from the date the loan was posted in Kiva to the date it was funded. We discovered that most of the loans are funded in the first few days of being posted and the number of loans getting funded start to decline after the first five days as shown if Figure 4.

Figure 4. Number of Days from Posted to Funded



Computation and Analysis

Our group has examined the data using Pandas and Jupyter Notebook to identify trends, loan amounts, loan description language, loan description, loan description length, whether images, videos or a website was provided, genders of borrowers, and borrower sectors.

Our group has transformed categorical variables into binary variables by using one hot encoding. We also turned borrower genders into percentages of female borrowers.

Modeling

When the team reached the stage of start building a model we did not have a definitive idea in terms of which type of solution we were going to use to answer our initial question (regression vs classification). .Therefore, we had two targets: one target was the number of days from posting date to funded date. The other target was a binary variable derived from posted to raised days indicating whether or not the loan was funded in five days or less.

The first attempt was to run a regression model to test if that was the optimal solution to our problem. As it was expected, the regression model performed very bad with an R-squared of 0.16.

We focused on the classification model with the target of whether or not the loan was funded in five days or less. The target was fairly balance with 46.5% of the data falling into the five days or less (FOL) class and the rest 53.5% of data assigned to more than five days (MTF).

Once the target was identified we worked on selecting the features that we used in our model. For the feature selection process we considered the 58 features we had in our database, some of them were fields we got from the original tables retrieved from Kiva and other were engineered by the team, such as percentage of borrowers that are women and description length. Using the feature selection model from Scikit-learn and the Lasso, Ridge and ElasticNet algorithms we trimmed down the number of features to 25.

The team also used pearson correlation and the features module from Yellowbrick to make our decision for feature selection. The 25 features selected can be summarized as: language (english or not), description length, loan use length, currency (usd or not), hashtag (exist or not), percentage of borrowers that are women (a loan application could have more than one borrower), sector (14 features derived from sector), loan type (direct or field partner) and repayment schedule (3 features derived from repayment).

After figuring out our target and features that we wanted to use for our model we decided to run eight classifiers and evaluate them based on accuracy and log loss. The best two performers, based on these two metrics, were Gradient Boosting (66%, 0.62) and Logistic Regression (65%, 0.63). To further our analysis and evaluate the classifiers, we also generated a confusion matrix using the classification report visualization from Yellowbrick. The recall, precision and F1 scores were very balanced between classes in both cases and the overall scores were very similar. The final decision came down to runtime. The Logistic Regression model runtime performance was 18 times faster than that of the Gradient Boosting classifier. Since our model was built to be used in a data product runtime performance is a key component taking into account that the difference in accuracy, log loss, precision, recall and F1 score was very minimal between the two classifiers.

The Regression Model scored very balanced results and performed well in our ROC analysis with a 0.70 area under the ROC. Nevertheless, the team tried to improve the predictability power of the model by tuning the hyperparameter using GridSearch and cross validation. Additionally, we used scalers to scale and normalize the data to reduce any noise due to large variance. But none of these methods were effective.

Results

We used the Logistic Regression model fitted using 80% of the data (training data). The hyperparameters were left as default since the model achieved better or same results as tuning the hyperparameters to other values and at a faster speed. The fitted model was dumped to a pickle and embedded to our application.

Model Retraining

Our team recognizes that the model needs to be retrained. Retraining is planned to be a JSON download of the recent loans and transformed into the “X_train” and “y_train” variables to fit the model.

Application

Early on in the project we decided our data product would be a website. Later on we discovered how we could serialize a model and use it in Flask.

We used python’s built-in persistence module pickle to save our model. Then using the microframework Flask, we created a website that serves the model.

Our Flask application takes input from the user. Then, we process the input and turn it into features. The features are passed to the pickled model and a result is returned back to the user. This application is available at http://kivateam.pythonanywhere.com.

The source code for our project and the server application is available on our project’s Github repository (https://github.com/georgetown-analytics/Kiva).

Conclusion

Initially our team was interested in looking into the determinants of loan repayments i.e. identify factors/causes of loan default and build a data product. The tool will predict the probability of success/failure of a loan application to help lenders make an informed decision and thereby improve repayment rates. The team later found out that Kiva is no longer publishing the status of individual loans due to privacy concerns and the team was not able to go in that direction. The team therefore re-defined the goal of the project and focused on the borrower side in identifying the factors that determine the faster funding of the loans. The data product is also aimed at helping borrowers predict how fast they will get funded by providing certain information.

In building a model that predicts the time it takes for a loan to get funded and developing a data product, the team worked with huge amount of data presented by Kiva with over 1.4 million instances and more than 58 features.

In the process of data wrangling, the team had to deal with missing and extreme values, drop insignificant features and features that will create multicollinearity.

The biggest challenge that the team faced was the skewness of the target variable. The team couldn't proceed with a regression model as it performed poorly when tested. The team therefore resorted to a classification model.

After deciding to work on a classification model, the team first experimented with 4 bins/classes which created class imbalance and later resorted to a binary classification to overcome the class imbalance problem.

As initially hypothesized, the results of the model established the right relationship between the target variable (posted_to_raised_days) and features incorporated in the model. For example:

- The size of the loan a borrower requested is negatively related to the time it takes for a loan to get funded – the higher the loan amount requested, the longer it takes for a loan to get funded.
- The better the description of the loan purpose (description_length), the faster a loan gets funded.
- The shorter the loan term, the faster the loan gets funded and etc.

The data product built by the team, a predictor of time for a loan to get funded, is very helpful for borrowers as Kiva gives a time window of 30 days from the time the loan gets posted on Kiva website until it gets funded. With the data product the team built, borrowers can check if they can get funded in 5 days or less or if it will take more than 5 days. In a situation where it will take more than 5 days, borrowers can work on some of the features and shorten the days it will take to get funded.

Potential future work

For those interested in working on Kiva data, there is a potential to build a model and data product that would help lenders make better decision and thereby improve loan repayment rates if data on loan repayments are made available.

Bibliography

Flannery, Matt. “Kiva and the Birth of Person-to-Person Microfinance.” *Innovations: Technology, Governance, Globalization*, vol. 2, no. 1-2, 2007, pp. 31–56., doi:10.1162/itgg.2007.2.1-2.31.

“Microcredit.” *Wikipedia*, Wikimedia Foundation, 1 Oct. 2018, en.wikipedia.org/wiki/Microcredit.

“About | Kiva” *Kiva*, www.kiva.org/about.