

# Document Classification Case Study: Kiva Loans

Dr. Stephen W. Thomas, Queen's University

July 14, 2017

---



## 1 Introduction

Kiva Microfunds is a non-profit that allows people to lend money to low-income entrepreneurs and students around the world. Started in 2005, Kiva has crowd-funded millions of loans with a repayment rate of 98% to 99%.

Kiva includes both traditional demographic information, such as gender and location, as well as personal stories on each borrower because Kiva wants lenders to connect with the borrowers on a human level. An example:

*Evelyn is 40 years old and married with 3 kids. She is in the Karura Hope women group and her life has been changed by the first KIVA loan she received last year which she is completing this quarter. Before she received the loan, she used to sell 9 litres of milk daily to local residents. After receiving the loan she bought iron sheets, five cement packets, one lorry of sand, some ballast and animal feed for her cows and improved her cow shed. Today she sells a daily average of 40 litres of milk to the Kiamba Dairy cooperative society, which is affiliated to the Kenya Cooperative Creameries at a cost of USD 0.28 per litre. Her daily farming has really grown. Evelyn intends to buy another dairy cow and a tank of water for home consumption and for her cows. She intends to repay in monthly installments.*

Despite her uplifting story, and her previous successful loan, Evelyn defaulted on her next loan of 900 USD.

In this case study, we will explore past Kiva loans and build a prediction model (in particular, a decision tree classifier) to predict which future borrowers will pay back loans, and which will default. A key question we will explore is: does adding text (i.e., the personal stories) to the prediction model increase or decrease the model's prediction power?

This case study will provide lots of data, tables, and graphs, but is intentionally light on commentary, analysis, and decision making. That's your job!

### 1.1 Case Discussion Questions

At the end of this case study, we will have a group discussion around the following questions:

1. Does text data help in predicting which loan seeker will default?
2. Which words are most biased towards defaulting? Is this expected/intuitive?
3. According to the decision tree prediction models, what variables best predict a default?
4. As a decision maker, would you recommend the use of textual data in your prediction models?
5. As a lender, what other information would you like to have?

## 2 Kiva Background

The key concepts in the Kiva world are:

- **Loan.** A loan is the most important concept at Kiva. Most other concepts are in some way related to a loan.
- **Borrower.** A borrower is someone who has requested a loan. Borrowers are often referred to as **businesses' ' entrepreneurs**” in order to emphasize the entrepreneurial spirit of these individuals who work to make a difference in their lives.
- **Lender.** A lender is a user registered on the Kiva website for the purposes of lending money and participating in the community. Some lenders have public profiles, known as lender pages, on the Kiva website, where they can share details about their activities and mission. Most lenders, however, refrain from displaying their public information and are referred to as “anonymous.”
- **Partner.** A partner, or Kiva field partner, is usually a microfinance institution with which Kiva works to find and fund loans. Every loan at Kiva is offered by a partner to a borrower, and the partner works with Kiva to get funding for that loan from lenders.

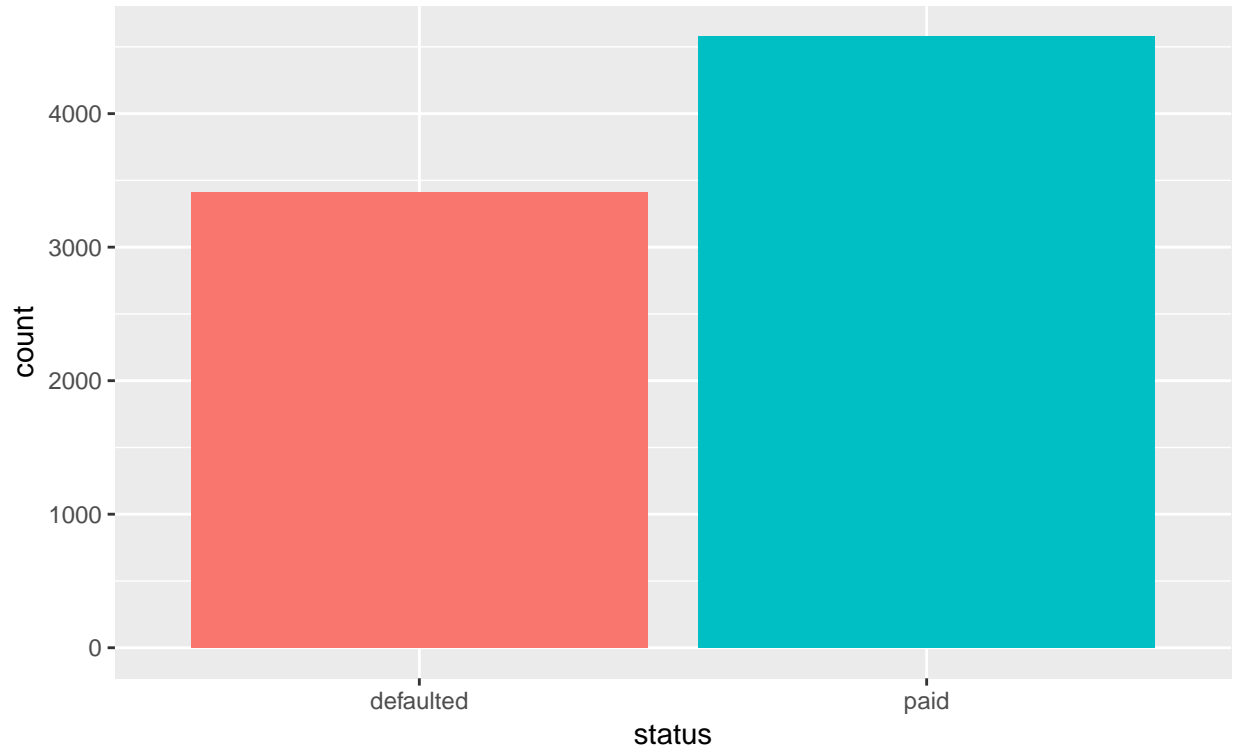
## 3 Data Description

The data in this case study was collected from <http://build.kiva.org>, Kiva’s website that provides snapshots of Kiva loan data. In the full dataset, about 98% of loans are paid and 2% defaulted. In this case study, we look at only a sample of the data, where the split between paid and defaulted is closer to 50%-50%.

Let’s look at our sample to understand the size, shape, values, and patterns in the variables. The sample includes 8 variables, named: status, sector, en, country, gender, loan\_amount, nonpayment, id. Each variable is explored in turn. The **en** variable is the text variable, i.e., the personal story of the loan seeker, and will be our main source of investigation. There are 7988 records/rows/loans in our sample.

### 3.1 Variable: status

The **status** variable indicates whether a loan was **paid** or **defaulted**. As previously described, the data has a fairly even split between these two options.

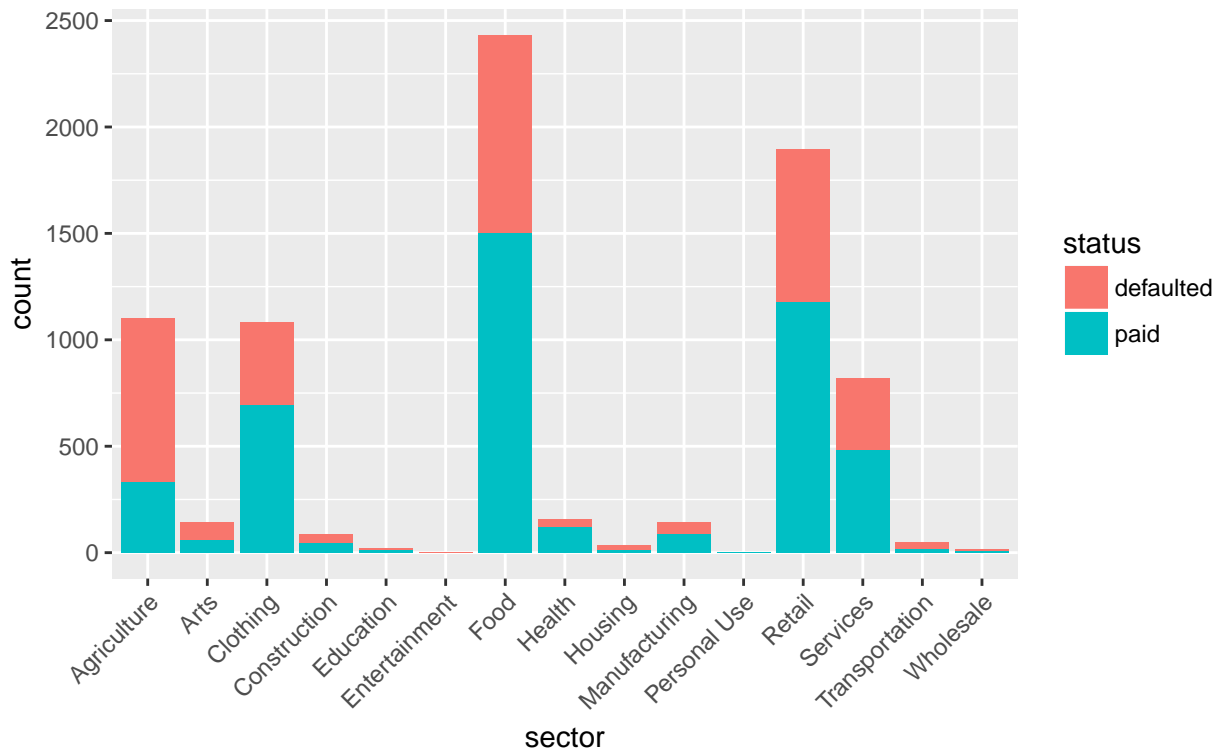


Below is a table with the actual counts.

status	count
defaulted	3411
paid	4577

### 3.2 Variable: sector

The figure below shows the number of loans in each sector, coloured by the loan's **status**.

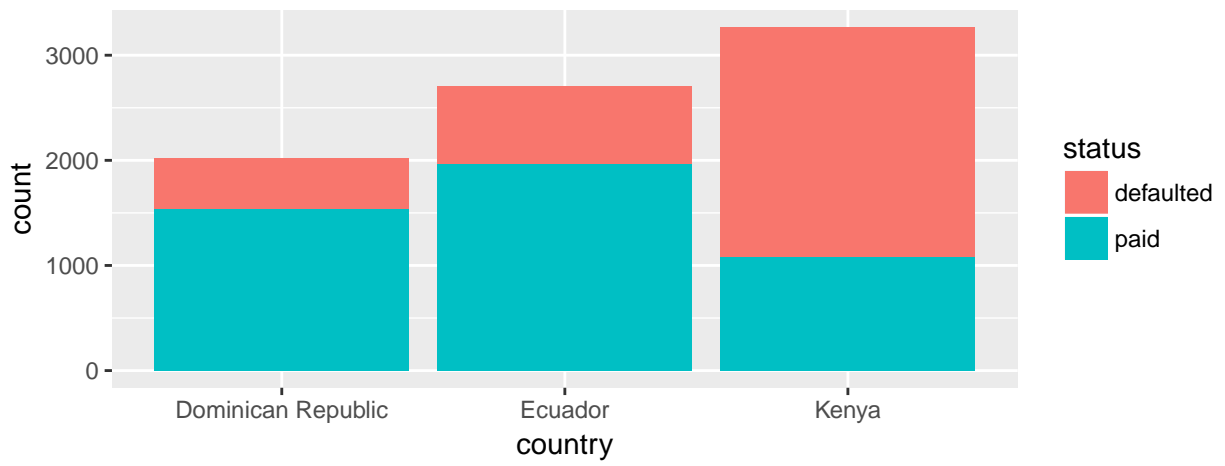


Below is a tabulation of the same data.

sector	defaulted	paid	total
Agriculture	769	335	1104
Arts	81	63	144
Clothing	386	694	1080
Construction	42	47	89
Education	5	13	18
Entertainment	2	2	4
Food	927	1505	2432
Health	33	123	156
Housing	23	12	35
Manufacturing	52	92	144
Personal Use	NA	3	NA
Retail	718	1177	1895
Services	337	485	822
Transportation	31	17	48
Wholesale	5	9	14

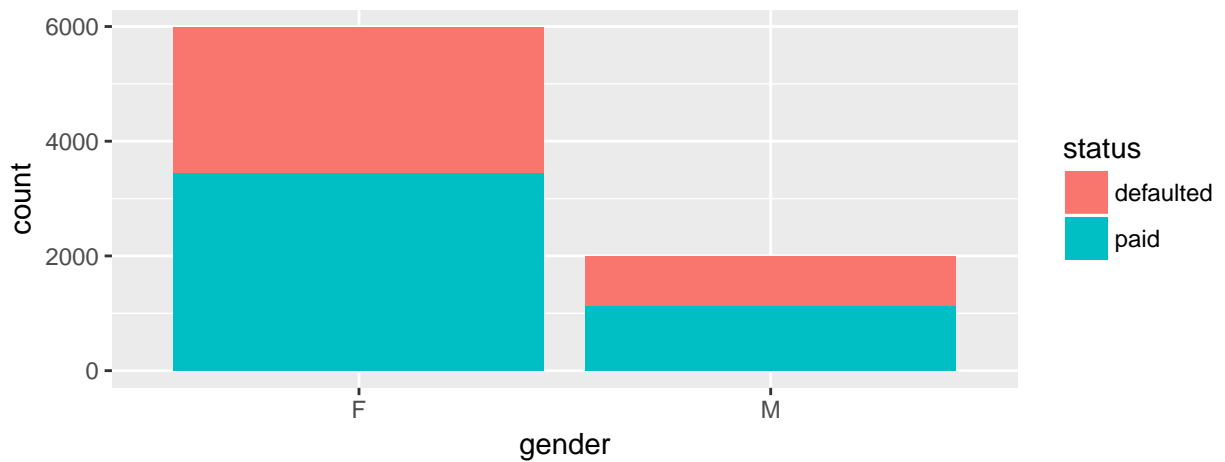
### 3.3 Variable: country

Below is the same for the `country` variable.



country	defaulted	paid	total
Dominican Republic	485	1533	2018
Ecuador	739	1963	2702
Kenya	2187	1081	3268

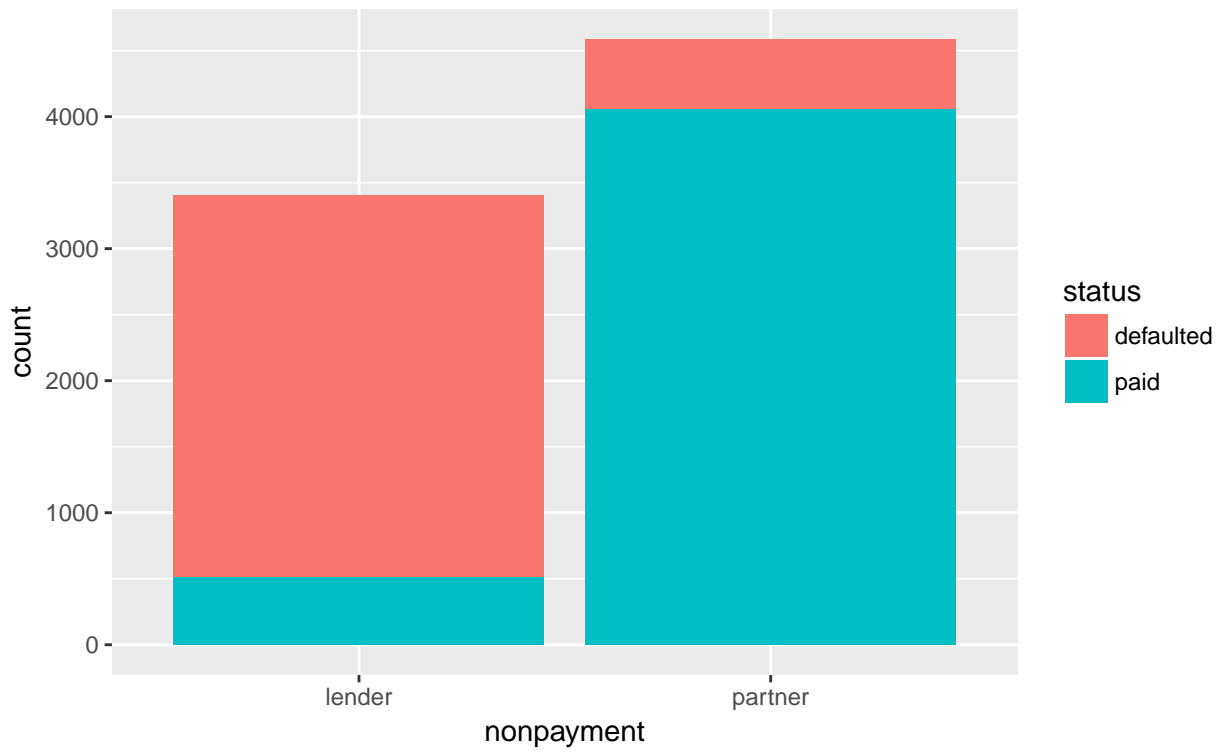
### 3.4 Variable: gender



gender	defaulted	paid	total
F	2541	3445	5986
M	870	1132	2002

### 3.5 Variable: nonpayment

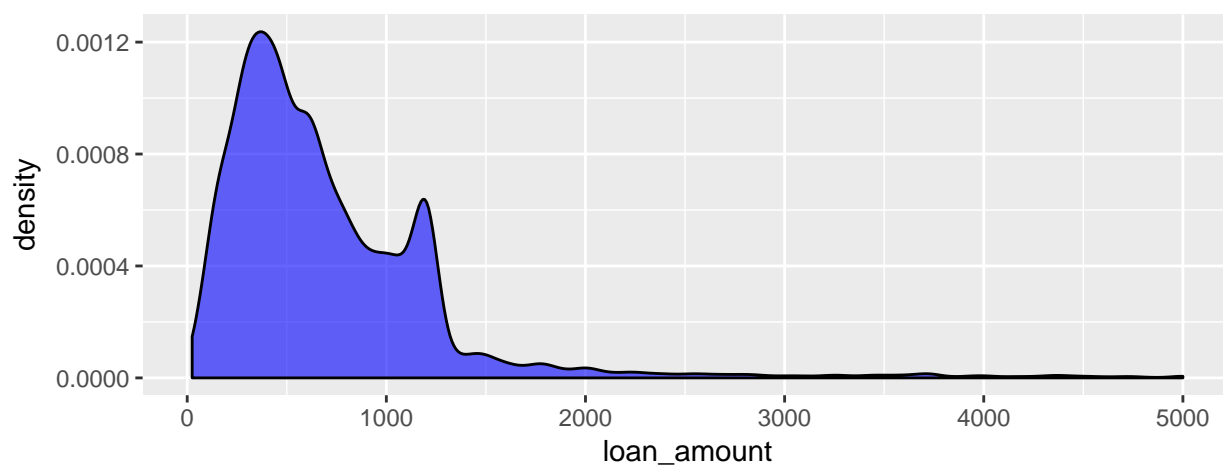
The nonpayment variable captures who is liable if a loan defaults: the lender, or the partner.



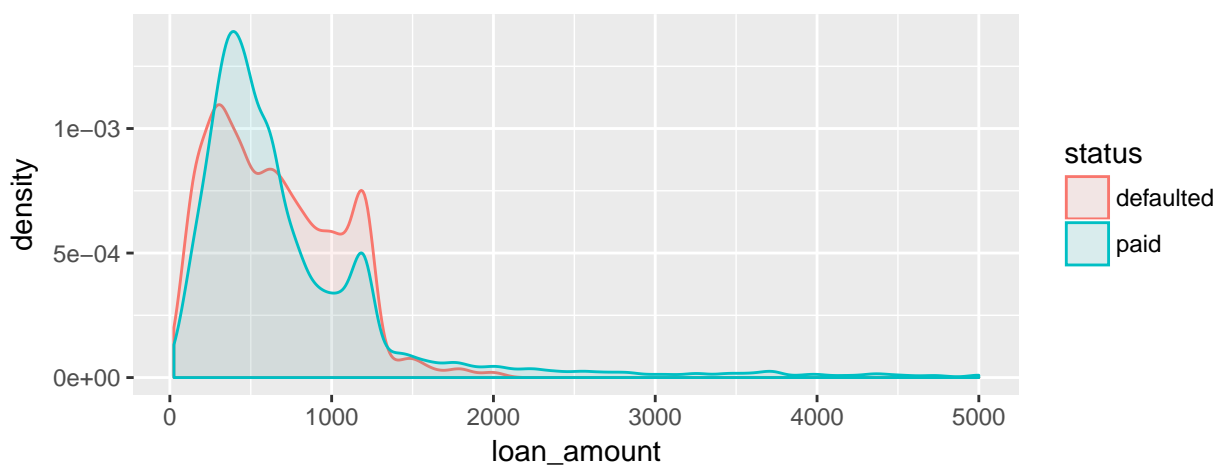
nonpayment	defaulted	paid	total
lender	2887	516	3403
partner	524	4061	4585

### 3.6 Variable: loan\_amount

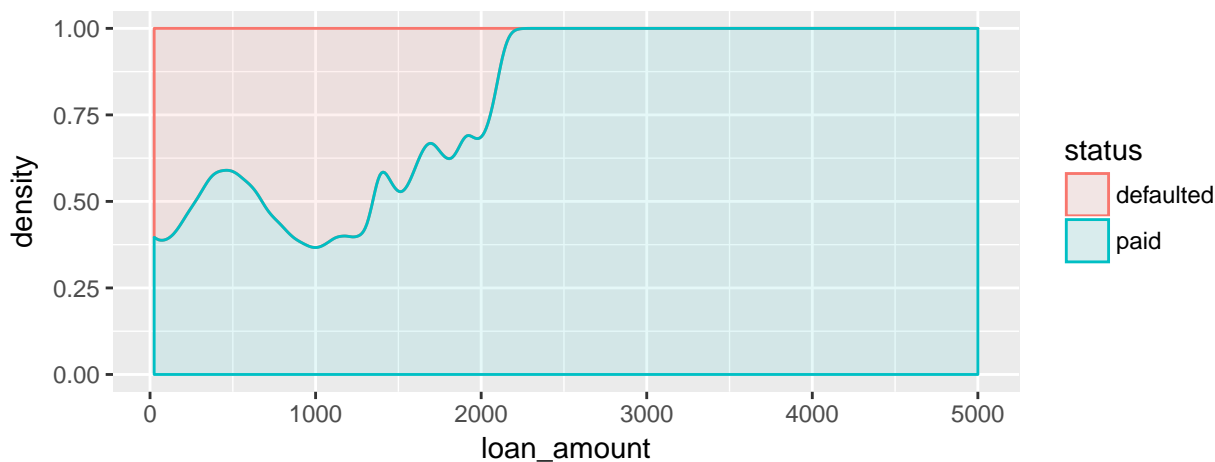
Unlike the other variables thus far, the `loan_amount` variable is numeric. Below is a density plot, which shows the most popular loan amounts.



Below are separate density curves for `status=defaulted` and `status=paid`.



Below is a filled density plot:

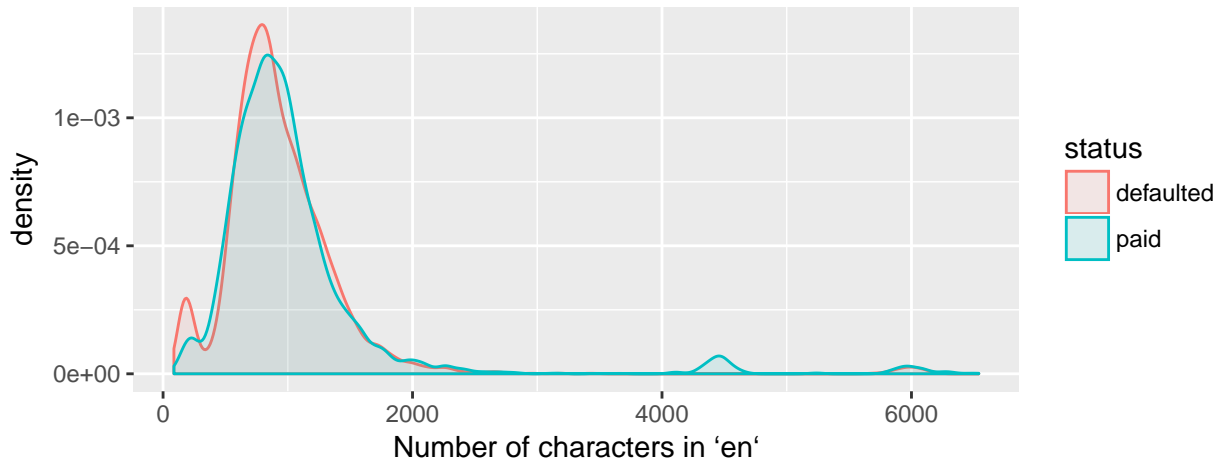


### 3.7 Variable: en

The `en` variable is raw English text, and there's lots of ways to look at it.

#### 3.7.1 Length

The figure below is a density plot of the length (number of characters/letters).



#### 3.7.2 Top Words

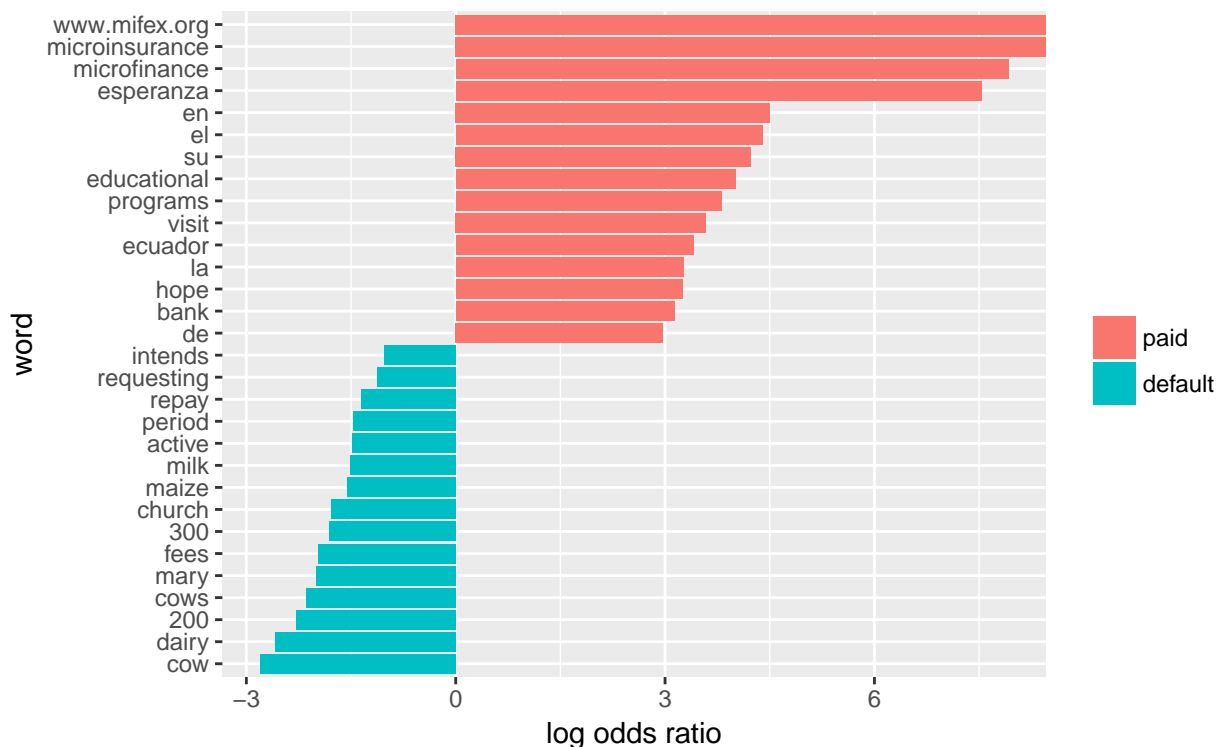
The table below shows the top (i.e, most frequently occurring) words.

word	count	freq
children	8634	0.0151617
school	5124	0.0089980
buy	4963	0.0087153
sells	3957	0.0069487
family	3764	0.0066098
products	3289	0.0057756
husband	3273	0.0057475
income	3203	0.0056246
married	3081	0.0054104
selling	2979	0.0052313
home	2954	0.0051874
community	2681	0.0047080
rice	2638	0.0046325
purchase	2605	0.0045745
started	2575	0.0045218
store	2427	0.0042619
clients	2291	0.0040231
increase	2230	0.0039160
house	2176	0.0038212
customers	2170	0.0038106



### 3.7.3 Most Biased Words

The plots below show which words are most biased towards being **paid** or **defaulted**, using the log odds ratio metric.



Below are tabular versions of the same data above, starting with the words that are biased towards **paid**:

word	defaulted	paid	total	log_ratio
microinsurance	0	603	603	Inf
www.mifex.org	0	601	601	Inf
microfinance	3	726	729	7.918863
esperanza	6	1112	1118	7.533979
en	24	544	568	4.502500
el	30	633	663	4.399171
su	41	769	810	4.229288
educational	38	613	651	4.011816
programs	55	771	826	3.809227
visit	62	746	808	3.588836
ecuador	101	1070	1171	3.405184
la	69	663	732	3.264341
hope	52	495	547	3.250845
bank	122	1071	1193	3.134005
de	193	1514	1707	2.971693
lucia	71	541	612	2.929738
access	147	1105	1252	2.910158

And those that are biased towards **default**:

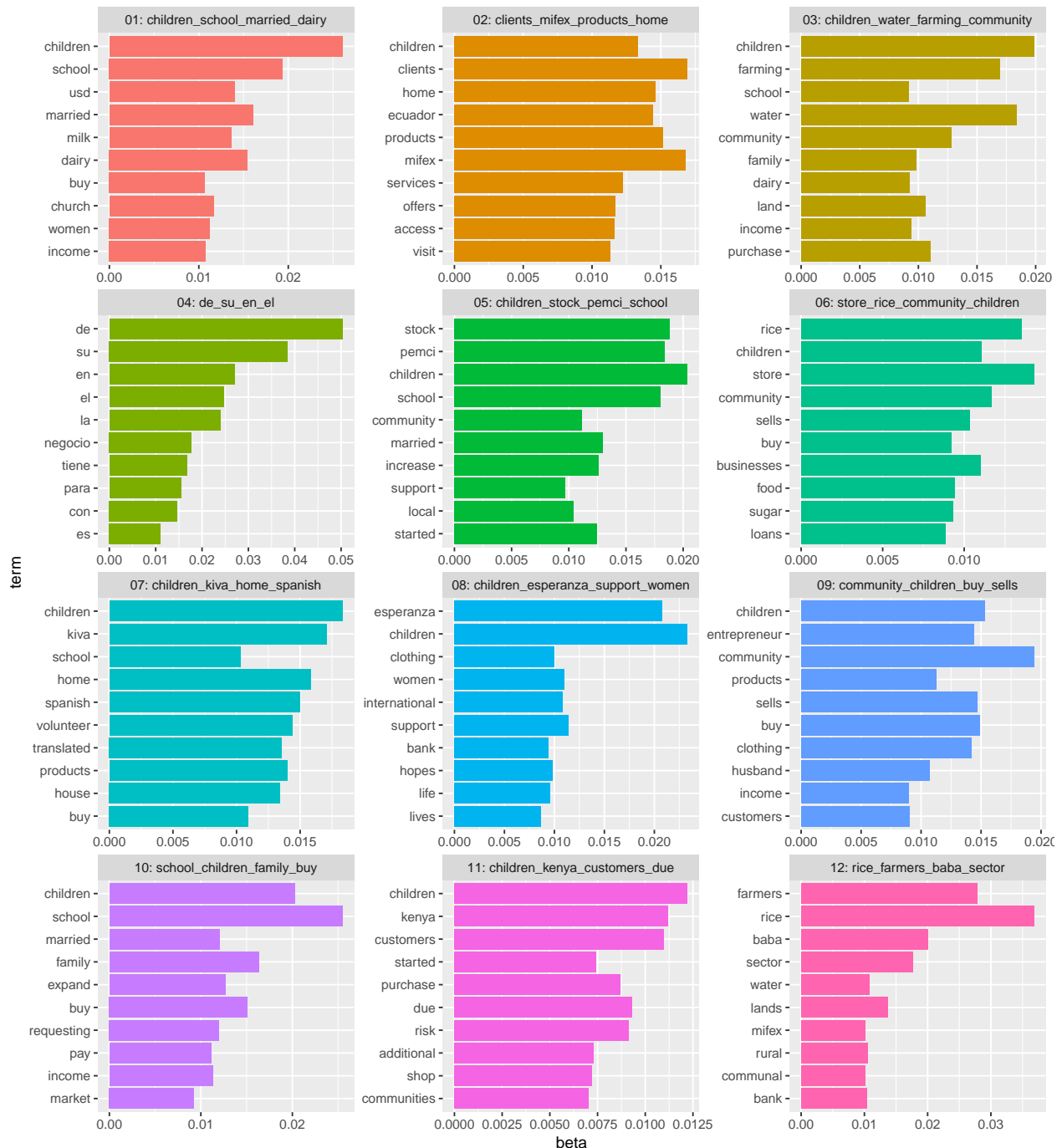
word	defaulted	paid	total	log_ratio
cow	455	65	520	-2.8073549
dairy	914	152	1066	-2.5881228
200	454	93	547	-2.2873897
cows	414	94	508	-2.1388981
mary	412	103	515	-2.0000000
fees	687	176	863	-1.9647347
300	394	112	506	-1.8146969
church	561	163	724	-1.7831288
maize	480	163	643	-1.5581624
milk	767	270	1037	-1.5062672
active	504	180	684	-1.4854268
period	404	146	550	-1.4683869
repay	1003	391	1394	-1.3590811
requesting	944	433	1377	-1.1244198
intends	456	224	680	-1.0255351
100	530	266	796	-0.9945661
personal	432	232	664	-0.8969065
farming	676	385	1061	-0.8121648
usd	623	355	978	-0.8114131
12	394	236	630	-0.7394088

## 3.8 LDA Topics

Next, we used a technique called Latent Dirichlet Allocation (LDA) to automatically extract high-level topics from the documents. We told LDA to extract the 12 most important topics; LDA will also tell us which topics are in which documents.

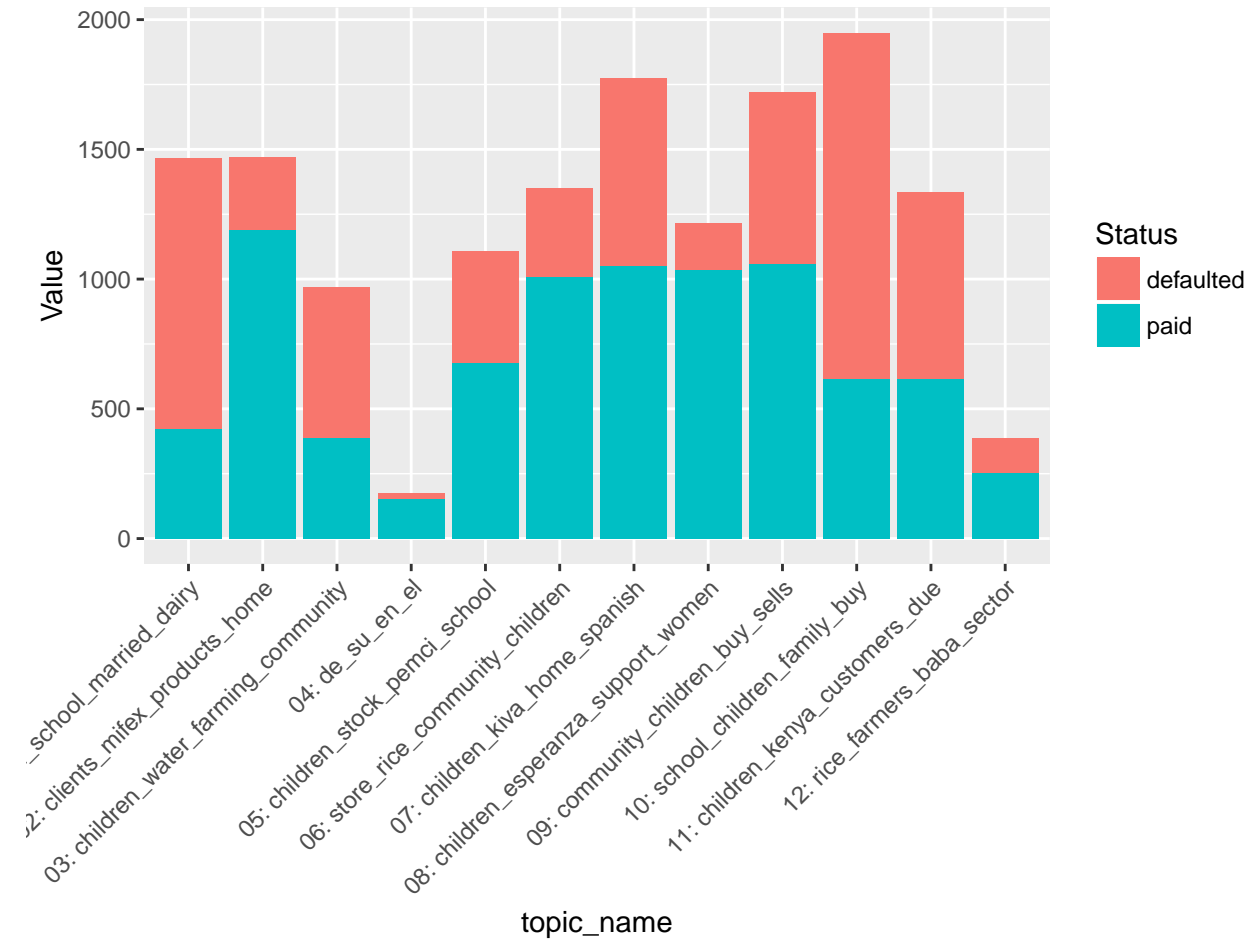
### 3.8.1 LDA Top Terms Per Topic

This figure shows the top terms (words) in each of the 12 discovered topic.



### 3.8.2 Documents per LDA Topic

The figure below shows the number of documents that contain each topic, coloured by **status**.



topic	topic_name	defaulted	paid	total
1	01: children_school_married_dairy	1041	424	1465
2	02: clients_mifex_products_home	279	1189	1468
3	03: children_water_farming_community	581	388	969
4	04: de_su_en_el	23	153	176
5	05: children_stock_pemci_school	430	676	1106
6	06: store_rice_community_children	343	1007	1350
7	07: children_kiva_home_spanish	724	1050	1774
8	08: children_esperanza_support_women	179	1038	1217
9	09: community_children_buy_sells	660	1059	1719
10	10: school_children_family_buy	1332	617	1949
11	11: children_kenya_customers_due	718	616	1334
12	12: rice_farmers_baba_sector	131	254	385

## 4 Building a Classifier Model

Now that we have explored the data, it's time to dive deeper. Which variable(s) are the biggest predictors of **status**? This is where classifier models shine. They can tell us exactly how all the variables relate to each other, and which are most important.

A decision tree is a popular classifier model in analytics. Here, the decision tree is automatically created by a machine learning algorithm as it learns simple decision rules from the data. These automatically-learned rules can then be used to both understand the variables and to predict future data. A big advantage of decision trees over other classifier models is that they are relatively simple for humans to understand and interpret.

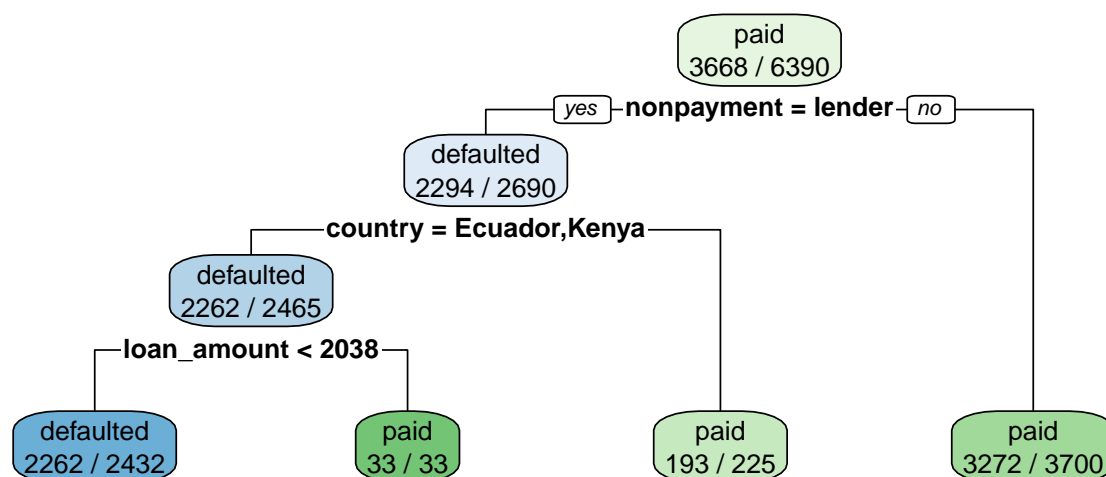
A decision tree consists of nodes. Each node splits the data according to a rule. A rule is based on a variable in the data. For example, a rule might be "Age greater than 30." In this case, the node splits the data by the age variable; those passengers that satisfy the rule (i.e., are greater than 30) follow the left path out of the node; the rest follow the right path out of the node. In this way, paths from the root node down to leaf nodes are created, describing the fate of certain types of passengers.

A decision tree path always starts with a root node (node number 1), which contains the most important splitting rule. Each subsequent node contains the next most important rule. After the decision tree is automatically created by the machine learning algorithm, one can use the decision tree to classify an individual by simply following a path: start at the root node and apply each rule to follow the appropriate path until you hit an end.

When creating a decision tree from data, the analyst can specify the number of nodes for the machine learning algorithm to create. More nodes leads to a more accurate model, at the cost of a more complicated and harder-to-interpret model. Likewise, fewer nodes usually leads to a less accurate model, but the model is easier to understand and interpret.

## 4.1 A Prediction Model without the Text

First, as a baseline, we trained a decision tree classifier model without using any of the text or topics. Below is a graphical depiction of the model after it has been trained:



To measure the prediction performance, we used some never-before-seen data (called *testing data*). We gave the testing data to the classifier, asked it to make a prediction (i.e., whether the borrower will pay or not), and then compared it to the true answer.

The following table summarizes the predictions of the classifier.

predicted	total	actual	
		defaulted	paid
defaulted	74	50	24
paid	159	39	120

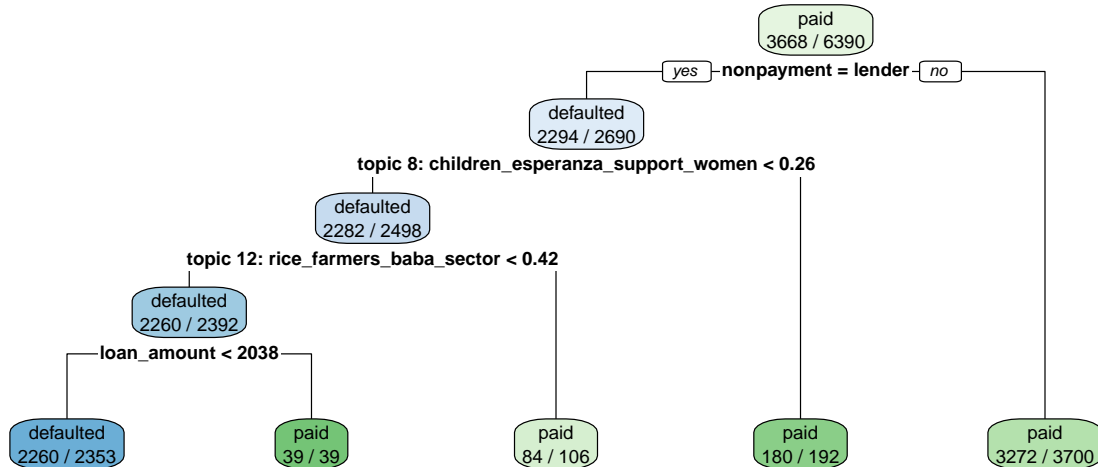
That is, the model predicted **defaulted** 74 times: 50 times correctly, and 24 times incorrectly. It predicted **paid** 159 times: 120 times correctly, and 39 times incorrectly.

Below is the accuracy and other metrics of the classifier on the testing data.

```
## [1] "Accuracy:    0.730"
## [1] "Precision:   0.676"
## [1] "Recall:      0.562"
## [1] "F1 Score:    0.613"
## [1] "Sensitivity: 0.562"
## [1] "Specificity: 0.755"
```

## 4.2 A Prediction Model with the Text

We then built the same kind of decision tree classifier model as before, except now, we included the LDA topics, which were built from the text. (Note: there are many *other* textual features we could include in this model: individual words, clusters, etc. However, we kept it simple for now.) Below is the result.



Below is a summary of its predictions:

predicted	total	actual	
		defaulted	paid
defaulted	608	573	35
paid	662	106	556

Metrics:

```
## [1] "Accuracy:    0.889"
## [1] "Precision:   0.942"
## [1] "Recall:      0.844"
## [1] "F1 Score:    0.890"
## [1] "Sensitivity: 0.844"
## [1] "Specificity: 0.840"
```

## 5 Appendix: Further Reading

- Kiva.org. Kiva's homepage.
- Build.Kiva. Kiva data dumps and data description.