Final Project: Big Data Analysis

Tat Dat Tran

Leipzig, 19.11.2021

Task

You work as a data scientist in a start-up. You opened your business a year ago and now you want to take the next step and expand your services. Your business model is to operate a platform where people who have a business idea but do not have the required money can register and collect money for their project within a given time. On the other hand, you have financiers who would like to invest their money in projects and who are looking for investments. As an intermediary, your platform brings together borrowers and lenders. You earn your money with a commission for every project that lands on your platform. Your database is the history of your platform. All projects are completed projects, i.e. the time to raise money for your project has expired. Your business model stipulates that the money collected will be paid out even if the target amount has not been reached. There are NO duplicates in the record. The split data record contains the following columns (including meaning):

- funded_amount ... amount received / paid amount in USD at the end of the crowdfunding period
- loan_amount ... Target amount (amount that you wanted to achieve with funding) in USD
- activity ... Sub-category to which the goal of crowdfunding belongs thematically
- sector ... main category in which the crowdfunding topic falls
- use ... Brief description of what the money should be used for
- country_code ... Country code according to ISO standard
- country ... country name according to ISO standard
- region ... Region
- currency ... Currency in which the funded_amount was then paid out
- term in months ... Duration over which the loan is to be paid out
- lender_count ... Lender (i.e. how many people gave money for the project)
- borrower_genders ... gender and number of borrowers, i.e. those who initiated the crowdfunding project
- repayment interval ... repayment modalities / frequency

Import libraries and data

Import libraries and data

Import libraries

We first import all libraries we need for our analysis

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from plotly.offline import init_notebook_mode, iplot
init_notebook_mode(connected=True)
```

Collect data

We import the first data .csv due to the library pandas: https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html and use .head() https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.head.html to see the first 5 rows

In [2]:
```
df1 = pd.read_csv('part1.csv')
df1.head()
```

Out[2]:

	Unnamed: 0	funded_amount	loan_amount	activity	sector	use	country_code	country	region	currency	term_in_months	lender_count	borrower_genders	repayment_interval
0	0	300.0	300.0	Fruits & Vegetables	Food	To buy seasonal, fresh fruits to sell.	PK	Pakistan	Lahore	PKR	12.0	12	female	irregular
1	1	575.0	575.0	Rickshaw	Transportation	to repair and maintain the auto rickshaw used ...	PK	Pakistan	Lahore	PKR	11.0	14	female, female	irregular
2	2	150.0	150.0	Transportation	Transportation	To repair their old cycle-van and buy another ...	IN	India	Maynaguri	INR	43.0	6	female	bullet
3	3	200.0	200.0	Embroidery	Arts	to purchase an embroidery machine and a variet...	PK	Pakistan	Lahore	PKR	11.0	8	female	irregular
4	4	400.0	400.0	Milk Sales	Food	to purchase one buffalo.	PK	Pakistan	Abdul Hakeem	PKR	14.0	16	female	monthly

We check the shape of this first data https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.shape.html

In [3]:
```
df1.shape
```

Out[3]: (335000, 14)

Now we import the second data and see the first 5 rows

In [4]:
```
df2 = pd.read_csv('part2.csv', sep='#')
df2.head()
```

Out[4]:

	Unnamed: 0	funded_amount	loan_amount	activity	sector	use	country_code	country	region	currency	term_in_months	lender_count	borrower_genders	repayment_interval
0	0	175.0	175.0	Liquor Store / Off-License	Food	to purchase additional stock of coconut wine t...	PH	Philippines	Palo, Leyte	PHP	8.0	6	female	irregular
1	1	325.0	325.0	Livestock	Agriculture	to buy 3 zebus and food to fatten them up.	MG	Madagascar	Antsirabe	MGA	12.0	13	female	monthly
2	2	550.0	550.0	Food Stall	Food	to buy ingredients for her food-vending busine...	PH	Philippines	Cordova, Cebu	PHP	5.0	6	female	irregular
3	3	1300.0	1300.0	Cattle	Agriculture	to buy one head of cattle.	EG	Egypt	Baniswef	EGP	14.0	50	male	monthly
4	4	900.0	900.0	Consumer Goods	Personal Use	to buy consumer goods amongst others.	PE	Peru	Urubamba - Urubamba - Cusco	PEN	6.0	1	female	irregular

Then we check the shape of this second data

In [5]:
```
df2.shape
```

Out[5]: (336205, 14)

Import libraries and data

We see that two datas share the same features, therefore we could combine two data into a bigger data by using concat in library pandas: https://pandas.pydata.org/docs/reference/api/pandas.concat.html and see the first 5 rows to check

In [6]:
```
df = pd.concat([df1,df2], axis=0)
df.head()
```

Out[6]:

	Unnamed: 0	funded_amount	loan_amount	activity	sector	use	country_code	country	region	currency	term_in_months	lender_count	borrower_genders	repayment_interval
0	0	300.0	300.0	Fruits & Vegetables	Food	To buy seasonal, fresh fruits to sell.	PK	Pakistan	Lahore	PKR	12.0	12	female	irregular
1	1	575.0	575.0	Rickshaw	Transportation	to repair and maintain the auto rickshaw used ...	PK	Pakistan	Lahore	PKR	11.0	14	female, female	irregular
2	2	150.0	150.0	Transportation	Transportation	To repair their old cycle-van and buy another ...	IN	India	Maynaguri	INR	43.0	6	female	bullet
3	3	200.0	200.0	Embroidery	Arts	to purchase an embroidery machine and a variet...	PK	Pakistan	Lahore	PKR	11.0	8	female	irregular
4	4	400.0	400.0	Milk Sales	Food	to purchase one buffalo.	PK	Pakistan	Abdul Hakeem	PKR	14.0	16	female	monthly

Then we check the shape of our new data which looks fine

In [7]:
```
df.shape
```

Out[7]: (671205, 14)

Data Preprocessing

Data Preprocessing

First look at the data

We first take a look at our data by using

- .nunique() to see the number of unique values in each feature: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.nunique.html
- .isnull().sum() to see how many null values in each feature: https://pandas.pydata.org/docs/reference/api/pandas.isnull.html https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.sum.html
- .describe() to generate some descriptive statistics: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.describe.html
- .info() to print a concise summary of our dataframe: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.info.html

In [8]:
```
df.nunique()
```

Out[8]:
```
Unnamed: 0            336205
funded_amount            610
loan_amount              479
activity                 163
sector                    15
use                   424912
country_code              86
country                   87
region                 12695
currency                  67
term_in_months           148
lender_count             503
borrower_genders       11298
repayment_interval         4
dtype: int64
```

In [9]:
```
df.isnull().sum()
```

Out[9]:
```
Unnamed: 0                0
funded_amount             0
loan_amount               0
activity                  0
sector                    0
use                    4232
country_code              8
country                   0
region                56800
currency                  0
term_in_months            0
lender_count              0
borrower_genders       4221
repayment_interval        0
dtype: int64
```

Import libraries and data

In [10]:
```
df.describe()
```

Out[10]:

	Unnamed: 0	funded_amount	loan_amount	term_in_months	lender_count
count	671205.000000	671205.000000	671205.000000	671205.000000	671205.000000
mean	167801.290828	785.995061	842.397107	13.739022	20.590922
std	96881.105762	1130.398941	1198.660073	8.598919	28.459551
min	0.000000	0.000000	25.000000	1.000000	0.000000
25%	83900.000000	250.000000	275.000000	8.000000	7.000000
50%	167801.000000	450.000000	500.000000	13.000000	13.000000
75%	251701.000000	900.000000	1000.000000	14.000000	24.000000
max	336204.000000	100000.000000	100000.000000	158.000000	2986.000000

In [11]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 671205 entries, 0 to 336204
Data columns (total 14 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   Unnamed: 0          671205 non-null  int64
 1   funded_amount       671205 non-null  float64
 2   loan_amount         671205 non-null  float64
 3   activity            671205 non-null  object
 4   sector              671205 non-null  object
 5   use                 666973 non-null  object
 6   country_code        671197 non-null  object
 7   country             671205 non-null  object
 8   region              614405 non-null  object
 9   currency            671205 non-null  object
 10  term_in_months      671205 non-null  float64
 11  lender_count        671205 non-null  int64
 12  borrower_genders    666984 non-null  object
 13  repayment_interval  671205 non-null  object
dtypes: float64(3), int64(2), object(9)
memory usage: 76.8+ MB
```

Comments:

- There are missing data at columns: 'use', 'country_code', 'region', 'borrower_genders'.
- Some data types could be changed to save memory such as:
  - 'funded_amount', 'loan_amount', 'term_in_months', 'lender_count' could be int32
  - 'activity', 'sector', 'use', 'country_code', 'country', 'region', 'currency', 'borrower_genders', 'repayment_interval' could be string.
- The column 'Unnamed: 0' is not important and could be dropped.

Drop unnecessary data

Drop unnecessary data

We drop the unnecessery feature 'Unnamed: 0' by using .drop: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.drop.html and see the first 5 rows as well as the shape of our new data.

In [12]:
```
df.drop(['Unnamed: 0'],axis= 1,inplace=True)
```

In [13]:
```
df.head(5)
```

Out[13]:

	funded_amount	loan_amount	activity	sector	use	country_code	country	region	currency	term_in_months	lender_count	borrower_genders	repayment_interval
0	300.0	300.0	Fruits & Vegetables	Food	To buy seasonal, fresh fruits to sell.	PK	Pakistan	Lahore	PKR	12.0	12	female	irregular
1	575.0	575.0	Rickshaw	Transportation	to repair and maintain the auto rickshaw used ...	PK	Pakistan	Lahore	PKR	11.0	14	female, female	irregular
2	150.0	150.0	Transportation	Transportation	To repair their old cycle-van and buy another ...	IN	India	Maynaguri	INR	43.0	6	female	bullet
3	200.0	200.0	Embroidery	Arts	to purchase an embroidery machine and a variet...	PK	Pakistan	Lahore	PKR	11.0	8	female	irregular
4	400.0	400.0	Milk Sales	Food	to purchase one buffalo.	PK	Pakistan	Abdul Hakeem	PKR	14.0	16	female	monthly

In [14]:
```
df.shape
```

Out[14]: (671205, 13)

Optimal in memory

Optimal in memory

We change some features from 'float64' or 'int64' into 'int32' by using .astype to save memory: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.astype.html

The memory size is reduced from 71.7MB into 61.5MB after this process.

In [15]:

```
df.memory_usage(deep=True)
```

Out[15]:

```
Index                  5369640
funded_amount          5369640
loan_amount            5369640
activity              46521289
sector                43603755
use                   79433763
country_code          39600879
country               43923974
region                46603210
currency              40272300
term_in_months         5369640
lender_count           5369640
borrower_genders      47039080
repayment_interval    43400106
dtype: int64
```

In [16]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 671205 entries, 0 to 336204
Data columns (total 13 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   funded_amount       671205 non-null  float64
 1   loan_amount         671205 non-null  float64
 2   activity            671205 non-null  object
 3   sector              671205 non-null  object
 4   use                 666973 non-null  object
 5   country_code        671197 non-null  object
 6   country             671205 non-null  object
 7   region              614405 non-null  object
 8   currency            671205 non-null  object
 9   term_in_months      671205 non-null  float64
 10  lender_count        671205 non-null  int64
 11  borrower_genders    666984 non-null  object
 12  repayment_interval  671205 non-null  object
dtypes: float64(3), int64(1), object(9)
memory usage: 71.7+ MB
```

In [17]:

```
df.funded_amount = df.funded_amount.astype('int32')
df.loan_amount = df.loan_amount.astype('int32')
df.term_in_months = df.term_in_months.astype('int32')
df.lender_count = df.lender_count.astype('int32')
# df.activity = df.activity.astype('str')
# df.sector = df.sector.astype('str')
# df.use  = df.use.astype('str')
# df.country_code = df.country_code.astype('str')
# df.country = df.country.astype('str')
# df.region = df.region.astype('str')
# df.currency = df.currency.astype('str')
# df.borrower_genders = df.borrower_genders.astype('str')
# df.repayment_interval = df.repayment_interval.astype('str')

df.memory_usage(deep=True)
```

Finding and working with the missing values

Finding and working with the missing/null values

Now we consider more details features where there are missing data.

Column 'use'

In [19]:
```
df['use'].isnull().sum()
```

Out[19]: 4232

Comment:

- There are 4232 missing data in 'use' but the information is not so important, we could ignore the missing data here.

Column 'country_code'

In [20]:
```
df['country_code'].isnull().sum()
```

Out[20]: 8

In [21]:
```
df[df['country_code'].isna()].country.unique() # Check the unique values in 'country' where 'country_code' is np.nan
```

Out[21]: array(['Namibia'], dtype=object)

In [22]:
```
# df.country_code.fillna('NA', inplace=True)
df.loc[df['country_code'].isna(),'country_code'] = 'NA' # Assign the missing values in 'country_code' by 'NA'
```

Comment:

- There are 8 missing data in 'country_code', all in country Namibia, we could fill in the missing data for 'country_code' as 'NA'.

Finding and working with the missing values

Column 'region'

In [23]:
```
df['region'].isnull().sum()
```

Out[23]:
```
56800
```

In [24]:
```
df[df['region'].isna()].country.unique()
```

Out[24]:
```
array(['Kenya', 'El Salvador', 'Senegal', 'Iraq', 'United States', 'Peru',
       'Tanzania', 'Guatemala', 'Colombia', 'Indonesia', 'Kosovo',
       'Timor-Leste', 'Turkey', 'Philippines', 'Palestine', 'Burundi',
       'Tajikistan', 'Honduras', 'Jordan', 'Mexico', 'Lebanon', 'Albania',
       'Nicaragua', 'Bolivia', 'Israel', 'Rwanda', 'Azerbaijan',
       'Ecuador', 'Mongolia', 'Haiti', 'Cambodia', 'Sierra Leone',
       'Yemen', 'Zimbabwe', 'Paraguay', 'Uganda', 'Armenia',
       'Dominican Republic', 'Benin', 'Belize', 'Ghana', 'Mozambique',
       'Zambia', 'Samoa', 'Brazil', 'Panama', 'Pakistan', 'Burkina Faso',
       'Suriname', 'Virgin Islands', 'Togo', 'South Africa', 'Malawi',
       'Nigeria', 'Liberia', 'Vietnam', 'Costa Rica', 'Guam',
       'Myanmar (Burma)', 'Mali', 'Madagascar',
       'The Democratic Republic of the Congo', 'Cameroon', 'Georgia',
       'Puerto Rico', 'South Sudan', 'Moldova', 'Chile', 'Kyrgyzstan',
       'India', 'China', 'Bhutan'], dtype=object)
```

In [25]:
```
df[df['region'].isna()].repayment_interval.unique()
```

Out[25]:
```
array(['irregular', 'monthly', 'bullet', 'weekly'], dtype=object)
```

Column 'borrower_genders'

In [26]:
```
df['borrower_genders'].isnull().sum()
```

Out[26]:
```
4221
```

In [27]:
```
df[df['borrower_genders'].isna()].repayment_interval.unique()
```

Out[27]:
```
array(['monthly', 'bullet', 'irregular'], dtype=object)
```

Comment:

- The missing values of feature 'region' and 'borrower_genders' could not be assigned reasonly if we do not have more external information. By gooling, we could find, for example, the database 'kiva.loans.csv' in https://www.kaggle.com/kiva/data-science-for-good-kiva-crowdfunding/version/5?select=kiva_loans.csv which consists of 20 features instead of our 13 features. With the feature 'tags' we could somehow assign missing values for 'region' and 'borrowe_genders' but this is out of this project.

First look

Exploratory Data Analysis

In [28]:
```
df.describe()
```

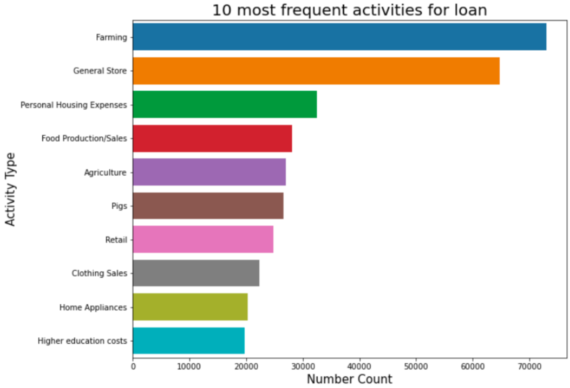
Out[28]:

	funded_amount	loan_amount	term_in_months	lender_count
count	671205.000000	671205.000000	671205.000000	671205.000000
mean	785.995061	842.397107	13.739022	20.590922
std	1130.398941	1198.660073	8.598919	28.459551
min	0.000000	25.000000	1.000000	0.000000
25%	250.000000	275.000000	8.000000	7.000000
50%	450.000000	500.000000	13.000000	13.000000
75%	900.000000	1000.000000	14.000000	24.000000
max	100000.000000	100000.000000	158.000000	2986.000000

Comments:

- The mean number of lenders is 20
- The mean number of months is 13, while minimum is 1 and maximum is 158

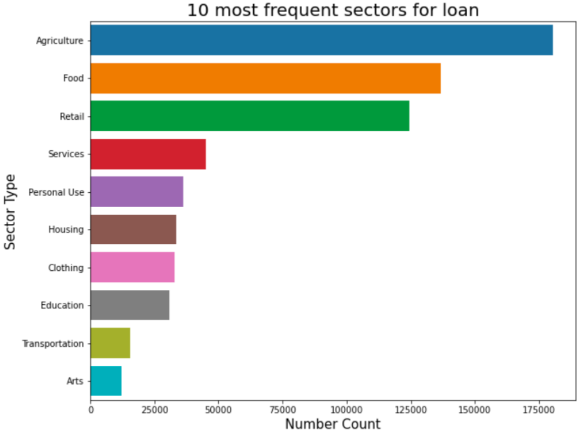
Activities for loan



Comment:

- We see that Farming is the dominant activity of all followed by General Store, Personal Housing Expenses, etc.

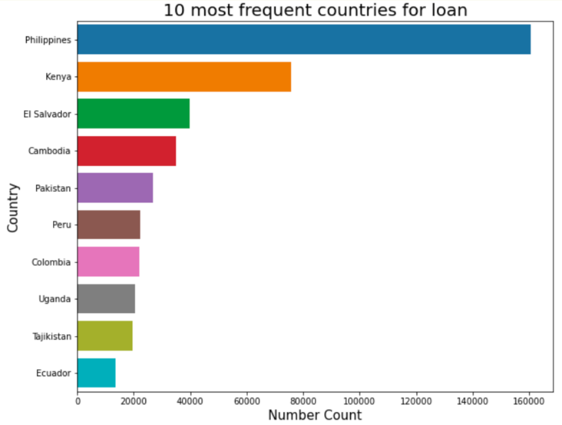
Sectors for loan



Comment:

- We see that Agriculture is the dominant sector of all followed by Food, Retail, Services etc.

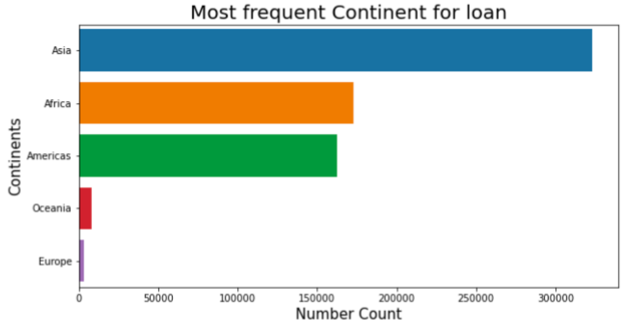
Countries for loan



Comment:

- Hence we can see that Philippines has been the main focus of the loans followed by Kenya, El Salvador, etc.

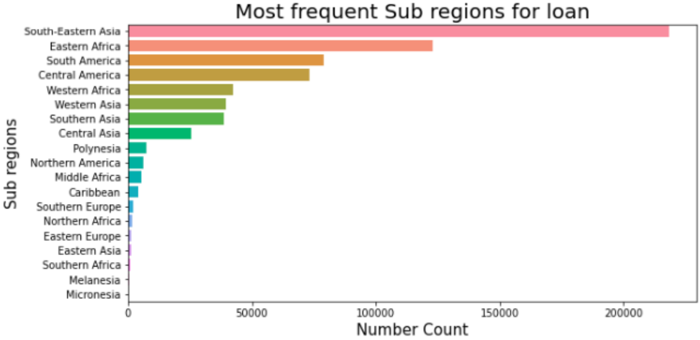
Continents for loan



Comment:

- The Asian has most active in loan, follows are Afria, Americas, Oceania and the last is Europe.

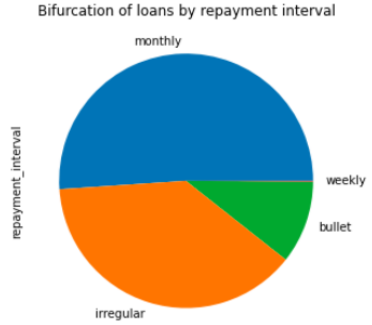
Subregions for loan



Comment:

- South East Asia is the dominant sub region. Followed by Eastern Africa,South America, Central America and Western Africa.

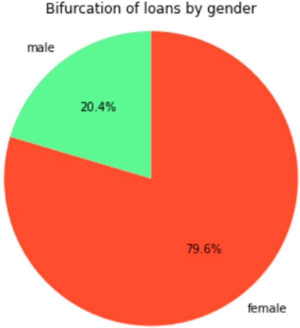
Repayment interval for loan



Comment:

- As we can see that most of the loans have repayment interval of monthly, where as it is important to notice that around 38% of loans are irregular in nature. On the other hand as we can see weekly intervals are just negligible.

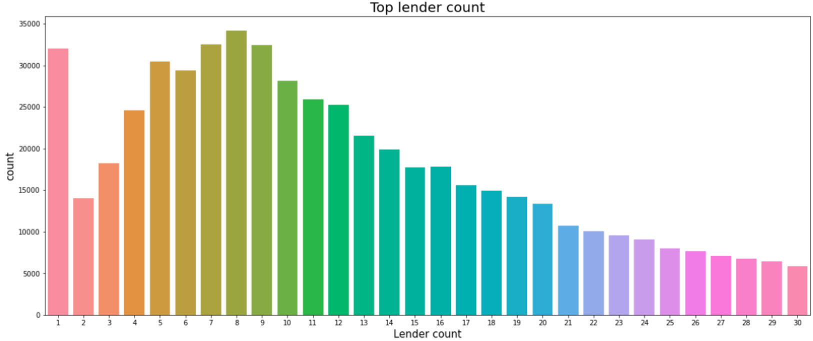
Genders vs loan



Comment:

- From the above pie chart it is evident that females are active to loans as four times as males.

Distributions of lenders



Comment:

- As we can see there is an increasing trend till highest at 8 lenders and decreasing after that. There lies another peak at solo lenders as well.

Summary

- Farming is the dominant activity of all for loan, followed by General Store, Personal Housing Expenses, etc.
- We see that Agriculture is the dominant sector of all followed by Food, Retail, Services etc.
- Philippines has been the main focus of the loans followed by Kenya, El Salvador, etc.
- Females are active to loans as four times as males
- The Asian has most active in loan, follows are Afria, Americas, Oceania and the last is Europe.
- Most of the loans have repayment interval of monthly, where as it is important to notice that around 38% of loans are irregular in nature. On the other hand as we can see weekly intervals are just negligible.
- South East Asia is the dominant sub region for loan. Followed by Eastern Africa,South America, Central America and Western Africa.
- There is an increasing trend till highest at 8 lenders and decreasing after that. There lies another peak at solo lenders as well.