



Get unlimited access

Open in app



Avengers Team

Dec 1 · 5 min read · Listen



Save



Hadoop and MapReduce Project-Social Development Bank Loans 2019



hadoop mapReduce

Content

1. *Introduction*
2. *Data Review*
3. *Data Preprocessing*
4. *MapReduce*
5. *Hadoop*
6. *Results*



[Get unlimited access](#)[Open in app](#)

Social Development Bank

About Social Development Bank:

“The Bank is considered to be one of the main government pillars for economic and social development funding to the citizens in Saudi Arabia. SDB focus’s in providing social financing products and business solutions to the low-income citizens, and creating awareness in financial planning & saving, as well as funding freelancers, micro, startup, and small businesses in way to enable them to contribute effectively to the economic growth of the country. The objectives have been verified as the following:

To provide free of interest loans for freelancers, micro, startup and small business to encourage them to run their own businesses independently.

To provide free of interest Social Loans for low-income citizens, in order to help them overcome their financial difficulties.

To encourage savings for individuals and institutions in the Kingdom, and to find the appropriate tools to achieve this goal.

The Bank has 24 branches in different regions of the Kingdom of Saudi Arabia to deliver services efficiently to its citizens.”



Vision 2030





individuals and families towards a vibrant and productive society, We found it necessary to anticipate the future value of financing loans granted to the individual, because through this it is possible to predict how the individual will become financially independent, enhance financial sufficiency and raise economic productivity.

The goal:

In order to achieve the goals of Vision 2030, we must know what is the goal of the Saudi individual in requesting loans and what the individual needs in order to achieve his successful future by knowing the most requested type of loan and the most commonly requested classification for that type of loan.

2. Data Review:

The data was obtained in the period of 2019 as described in the official website but we took our dataset from Kaggle as it was translated into English.

It contains 15 columns and 11,176 rows.

Saudi Arabia Social Development Bank Loans 2019

Saudi Arabia Social Development Bank Loans 2019 English version

www.kaggle.com

	ID	bank branch	funding type	funding classification	customer sector	financing value	installment value	cashing date	sex	age	social status	special needs	number of family members	saving loan	income
0	1.0	Tabūk	social	family	government employee	60000.0	>= 1000	2019/02	MALE	>= 30	married	No	>= 05	No	< 5000
1	2.0	Hail	project	solution	NaN	160000.0	>= 1000	2019/01	MALE	< 30	single	No	< 02	No	< 5000
2	3.0	Tabūk	social	marriage	government employee	60000.0	>= 1000	2019/02	MALE	< 30	married	No	>= 02	No	>= 7500
3	4.0	Medina	social	marriage	employee of a government company	60000.0	< 1000	2019/03	MALE	< 30	married	No	>= 10	No	>= 5000
4	5.0	Medina	social	family	private sector employee	60000.0	>= 1000	2019/02	FEMALE	>= 30	divorced	No	>= 02	No	>= 10000

A view on the database.





of Data Preprocessing.

In this chapter, we explored the database and found that it contains missing values. we have 4 columns with missing values.

```
ID          0
bank branch 0
funding type 0
funding classification 0
customer sector 3950
financing value 0
installment value 0
cashing date 0
sex 0
age 6
social status 0
special needs 0
number of family members 43
saving loan 0
income 114
dtype: int64
```

Missing values.

1. Handle missing values.

To handle missing values each column's missing values were replaced with its most frequent value.

We filled in the missing values using the most frequent strategy only because the data type of these columns is Categorical.

```
# Fill in the Missing Values using the Simple Imputer with the Most Frequent strategy
imputer = SimpleImputer(strategy='most_frequent', missing_values=np.nan)
imputer = imputer.fit(data[['customer sector', 'income', 'number of family members', 'age']])
data[['customer sector', 'income', 'number of family members', 'age']] = imputer.transform(
    data[['customer sector', 'income', 'number of family members', 'age']])
```

2. Delete unimportant columns.

We deleted some unnecessary columns to our goal



[Get unlimited access](#)[Open in app](#)

```
data.drop(['ID'], axis=1, inplace=True)
data.drop(['cashing date'], axis=1, inplace=True)
data.drop(['social status'], axis=1, inplace=True)
data.drop(['special needs'], axis=1, inplace=True)
data.head(5)
```

4. MapReduce:

Convert the dataframe into text file

```
data.to_csv("/content/SocialBankLoans.txt", index = False)
```

1st MapReduce

We create a MapReduce class to count the occurrence of each loan type in order to find the most requested type from the social development bank loans.

Create a text file containing one column from the full data text.

```
# funding Types text file to find the most requested types of loan
data['funding type'].to_csv("/content/fundingType.txt", index = False)
```

Create the Class.

```
%%file LoanTypesCount.py
# %%file is an Ipython magic function that saves the code cell as a file

from mrjob.job import MRJob # import the mrjob.job library
from mrjob.step import MRStep # import the mrjob.step library

class SA_LoanTypesCount(MRJob):
```





```
def mapper_get_loanType(self, _, loan_type):
    # output each line as a tuple of (loan_type, 1)
    yield (loan_type, 1)

# the reduce step: combine all tuples with the same key. In this case, the
# then sum all the values of the tuple, which will give the most loan type
def reducer_count_loanTypes(self, key, values):
    yield (key, sum(values))

if __name__ == "__main__":
    SA_LoanTypesCount.run()
```

```
!python LoanTypesCount.py fundingType.txt
```

```
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/LoanTypesCount.root.20221130.110600.038980
Running step 1 of 1...
job output is in /tmp/LoanTypesCount.root.20221130.110600.038980/output
Streaming final output from /tmp/LoanTypesCount.root.20221130.110600.038980/output...
"funding type" 1
"project"      656
"social"       10384
"transfer"     135
Removing temp directory /tmp/LoanTypesCount.root.20221130.110600.038980...
```

The most requested loan type is the social type.

We can conclude from these results that the most requested loan type is the social type, with 10384 occurrences.

2nd MapReduce

MapReduce is also used to find the specific loan type that is most commonly requested within social loans.

Create a text file containing one column from the full data text.

```
# funding classification text file to find the exact classification of the most
```





```
%%file ClassOfLoanCount.py
# %%file is an Ipython magic function that saves the code cell as a file

from mrjob.job import MRJob # import the mrjob.job library
from mrjob.step import MRStep # import the mrjob.step library

class SA_ClassOfLoanCount(MRJob):
    def steps(self):
        return [
            MRStep(mapper=self.mapper_get_ClassOfLoan,
                    reducer=self.reducer_count_ClassOfLoans)
        ]

    def mapper_get_ClassOfLoan(self, _, loan_class):
        # output each line as a tuple of (loan_class, 1)
        yield (loan_class, 1)

    # the reduce step: combine all tuples with the same key. In this case, the
    # then sum all the values of the tuple, which will give the most loan class
    def reducer_count_ClassOfLoans(self, key, values):
        yield (key, sum(values))

if __name__ == "__main__":
    SA_ClassOfLoanCount.run()
```

```
!python ClassOfLoanCount.py fundingClassification.txt
```

```
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/ClassOfLoanCount.root.20221130.110623.485606
Running step 1 of 1...
job output is in /tmp/ClassOfLoanCount.root.20221130.110623.485606/output
Streaming final output from /tmp/ClassOfLoanCount.root.20221130.110623.485606/output...
"emerging"      458
"excellence"    6
"family"        3273
"private"       7
"renovation"    103
"solution"      161
"taxi cab"      135
"telecom"       4
"food trucks"   7
"fresh graduate" 19
"funding classification" 1
```





Based on our results, marriage has been the most commonly requested classification for a social loan.

5. Hadoop:

Utilizing HDFS on our MapReduce program.

```
[maria_dev@sandbox-hdp avengers]$ python LoanTypesCount.py -r hadoop --hadoop-s
streaming-jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar fundi
ngType.txt
No configs found; falling back on auto-configuration
No configs specified for hadoop runner
Looking for hadoop binary in $PATH...
Found hadoop binary: /usr/bin/hadoop
Using Hadoop version 2.7.3.2.6.5.0
Creating temp directory /tmp/LoanTypesCount.maria_dev.20221130.104927.254960
uploading working dir files to hdfs:///user/maria_dev/tmp/mrjob/LoanTypesCount.m
aria_dev.20221130.104927.254960/files/wd...
Copying other local files to hdfs:///user/maria_dev/tmp/mrjob/LoanTypesCount.mar
ia_dev.20221130.104927.254960/files/
Running step 1 of 1...
packageJobJar: [] [/usr/hdp/2.6.5.0-292/hadoop-mapreduce/hadoop-streaming-2.7.3.2.6.5.0-292.jar] /tmp/streamjob590992890826632329
8.jar tmpDir=null
Connecting to ResourceManager at sandbox-hdp.hortonworks.com/172.18.0.2:8032
Connecting to Application History server at sandbox-hdp.hortonworks.com/172.18.0.2:10200
Connecting to ResourceManager at sandbox-hdp.hortonworks.com/172.18.0.2:8032
Connecting to Application History server at sandbox-hdp.hortonworks.com/172.18.0.2:10200
Total input paths to process : 1
number of splits:2
Submitting tokens for job: job_1669805390238_0001
Submitted application application_1669805390238_0001
The url to track the job: http://sandbox-hdp.hortonworks.com:8088/proxy/application_1669805390238_0001/
Running job: job_1669805390238_0001
Job job_1669805390238_0001 running in uber mode : false
  map 0% reduce 0%
  map 100% reduce 0%
  map 100% reduce 100%
Job job_1669805390238_0001 completed successfully
Output Directory: hdfs:///user/maria_dev/tmp/mrjob/LoanTypesCount.maria_dev.20221130.104927.254960/output
Counters: 49
  File Input Format Counters
    Bytes Read=135510
  File Output Format Counters
    Bytes Written=61
  File System Counters
    FILE: Number of bytes read=146226
    FILE: Number of bytes written=765341
    FILE: Number of large read operations=0
    FILE: Number of read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=135884
    HDFS: Number of bytes written=61
    HDFS: Number of large read operations=0
    HDFS: Number of read operations=9
    HDFS: Number of write operations=2
  Job Counters
    Data-local map tasks=2
    Launched map tasks=2
    Launched reduce tasks=1
    Total megabyte-milliseconds taken by all map tasks=3612000
    Total megabyte-milliseconds taken by all reduce tasks=1946250
    Total time spent by all map tasks (ms)=14448
    Total time spent by all maps in occupied slots (ms)=14448
```





```
FILE: Number of write operations=0
HDFS: Number of bytes read=135884
HDFS: Number of bytes written=61
HDFS: Number of large read operations=0
HDFS: Number of read operations=9
HDFS: Number of write operations=2
Job Counters
  Data-local map tasks=2
  Launched map tasks=2
  Launched reduce tasks=1
  Total megabyte-milliseconds taken by all map tasks=3612000
  Total megabyte-milliseconds taken by all reduce tasks=1946250
  Total time spent by all map tasks (ms)=14448
  Total time spent by all maps in occupied slots (ms)=14448
  Total time spent by all reduce tasks (ms)=7785
  Total time spent by all reduces in occupied slots (ms)=7785
  Total vcore-milliseconds taken by all map tasks=14448
  Total vcore-milliseconds taken by all reduce tasks=7785
Map-Reduce Framework
  CPU time spent (ms)=3500
  Combine input records=0
  Combine output records=0
  Failed Shuffles=0
  GC time elapsed (ms)=2020
  Input split bytes=374
  Map input records=11176
  Map output bytes=123868
  Map output materialized bytes=146232
  Map output records=11176
  Merged Map outputs=2
  Physical memory (bytes) snapshot=546082816
  Reduce input groups=4
  Reduce input records=11176
  Reduce output records=4
  Reduce shuffle bytes=146232
  Shuffled Maps =2
  Spilled Records=22352
  Total committed heap usage (bytes)=273154048
  Virtual memory (bytes) snapshot=5833482240
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
Job output is in hdfs:///user/maria_dev/tmp/mrjob/LoanTypesCount.maria_dev.20221130.104927.254960/output
Streaming final output from hdfs:///user/maria_dev/tmp/mrjob/LoanTypesCount.maria_dev.20221130.104927.254960/output...
"funding type" 1
"project" 656
"social" 10384
"transfer" 135
Removing HDFS temp directory hdfs:///user/maria_dev/tmp/mrjob/LoanTypesCount.maria_dev.20221130.104927.254960...
Removing temp directory /tmp/LoanTypesCount.maria_dev.20221130.104927.254960...
[maria_dev@sandbox-hdp avengers]$
```

```
from mrjob.job import MRJob # import the mrjob library
from mrjob.step import MRStep

class SA_LoanTypesCount(MRJob):
    def steps(self): # Create method named steps and pass the mapper and the reducer for MRStep
        return[
            MRStep(mapper=self.mapper_get_loanType,
                    reducer=self.reducer_count_loanTypes)
        ]

    def mapper_get_loanType(self, _, loan_type):
        # output each line as a tuple of (loan_type, 1)
        yield (loan_type, 1)

    # the reduce step: combine all tuples with the same key. In this case, the key is the loan
    # then sum all the values of the tuple, which will give the most loan type requested
    def reducer_count_loanTypes(self, key, values):
        yield (key, sum(values))

if __name__ == "__main__":
    SA_LoanTypesCount.run()
[maria_dev@sandbox-hdp avengers]$
```



[Get unlimited access](#)[Open in app](#)

```
Running step 1 of 1...
job output is in /tmp/LoanTypesCount.maria_dev.20221130.105727.219484/output
Streaming final output from /tmp/LoanTypesCount.maria_dev.20221130.105727.219484/output...
"funding type"  1
"project"       656
"social"        10384
"transfer"      135
Removing temp directory /tmp/LoanTypesCount.maria_dev.20221130.105727.219484...
```

6. Results:

After we used the MapReduce and Hadoop Program, we found that most individuals request a loan for a social purpose, and it is often for the purpose of marriage.

