BDAD Summer 2019 Symposium

NYU Courant, Computer Science

August 8, 2019

Big Data Applications Symposium

Project Name: Fair Lending Finder

Team:

- Cody Gilbert
- Fang Han
- Jeremy Lao

Abstract: *We want to help people increase their chances of securing a mortgage related loan. Our application will ask you for your details and provide you with the lender that is most likely to lend to you. We trained our application using publicly available anonymized mortgage application information.*

Motivation

Who are the users of this application?

- General Public
- Banking Regulators

Who will benefit from this application?

- Anyone that is looking for a mortgage loan
- Low to moderate income (LMI) borrowers
- People in states with high loan denial rates

Why is this application important?

- While there have been improvements in the mortgage lending process over the last decade, unconscious bias remains a factor in provisioning credit to average income borrowers. Our application will help borrowers use that unconscious bias in their favor.

Goodness

What steps were taken to assess the "goodness" of the analytic itself?

We utilized publicly available Home Mortgage Disclosure Act (HMDA) data from 2007-2017 that contains over 207 million anonymized home mortgage application records to train a machine learning model on "approved" or "denied" mortgage applications.

We use the following features to train a Naive Bayes model:

- Loan Amount
- Applicant Income
- Race
- Gender
- Lender
- State

Goodness (contd.)

$$\begin{aligned} P(y=k) &= \beta_0 loanAmt_{obs} + \beta_1 applicantIncome_{obs} \\ &+ \delta_0 race + \delta_1 ethnicity + \delta_2 gender \\ &+ \delta_3 lender + \delta_4 state + \delta_5 year \end{aligned}$$

Where δ are dummy variables for the categorical variables and β are coefficients. The outcome (k), approve or deny, $k \in 0, 1$

MLModel	Training/Test	AUC
Logistic Regression	80/20	60%
SVM	80/20	59%
Naive Bayes	80/20	79%

Table 1: Model Evaluation

Goodness (contd.)

Naive Bayes

$$P(y = k \mid loanAmt, applicantIncome$$
$$race, ethnicity, gender$$
$$lender, state, year)$$

Actuation/Remediation

What actuation or remediation actions are/could be performed by this application?

- The loan applicant will use this application to determine the lender that will most likely extend credit, and the applicant can apply directly to that lender.
- A banking regulator can use this to determine the lenders that are least likely to extend credit to LMI and minority communities

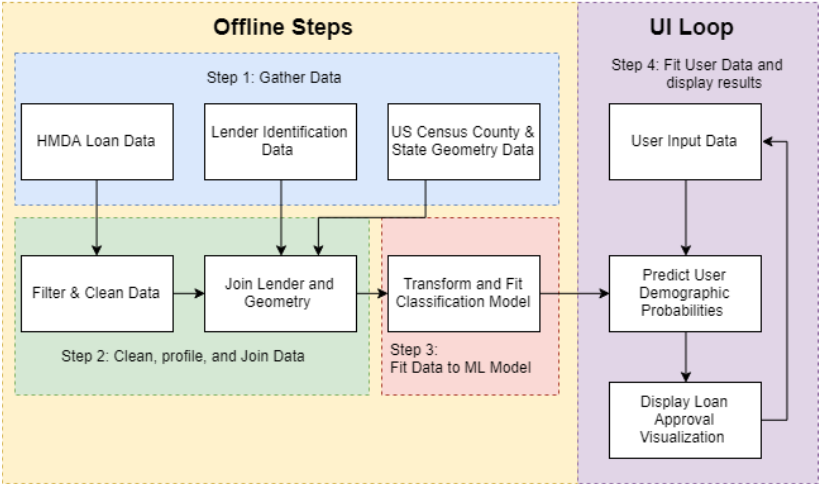
Data Sources

Name	HMDA Data Set
Description	Anonymized mortgage loan application information
Size of data	$>$ 120 GB

Name	Geospatial Data
Description	Latitude and Longitude of States and Counties
Size of data	$>$ 100 MB

Name	HMDA Panel Information
Description	Lender metadata, such as parent ID and head office
Size of data	$>$ 100 MB

Design Diagram



Platform(s) on which the application runs:

NYU HPC Cluster (DUMBO)

Code Walkthrough

HMDA

For data profiling, we originally ingested the data into a dataframe. The entire profiling exercise would take 5-7 hours.

```
val dataForAnalysis =
    spark.read.format("csv").option("header", "true").
                option("inferSchema",
                    "true").load(hdfsPath).
                select("loan_amount_000s",...)
```

We changed our strategy and leveraged Spark Context RDDs to profile the data, reducing run-time to 1.5 hour:

```
val dataForAnalysis = sc.textFile(hdfsPath)
val reducedLoanAmtData =
    mapReduceFunc(dataForAnalysis, 7)
```

Code Walkthrough (contd.)

HMDA

While dataframes have the .count() function, we had to write a custom function to perform count():

```
def mapReduceFunc(dataForAnalysis : RDD[String], colNum :
    Integer) : RDD[String] = {
  val firstLine = dataForAnalysis.first()
  val data = dataForAnalysis.filter(row => row !=
     firstLine)
  val keyAmt = data.map(_.split(",")). map(c =>
     (c(colNum),1)). reduceByKey((x,y) => x+y)
  val mrAmt = keyAmt.map(x =>
     x._1.stripPrefix("\"").stripSuffix("\"") + "," +
     x._2)
  mrAmt
}
```

Code Walkthrough (contd.)

HMDA

While dataframes preserve column names, you have to manually incorporate them in the RDD before saving as a .csv file:

```
val header: RDD[String]=
    sc.parallelize(List("loan_amount,frequency"))
 header.union(reducedLoanAmtData).saveAsTextFile(<path>)
```

Code Walkthrough (contd.)

HMDA - MLLib

We developed the model using MLLib and saved the model to HDFS for our interactive application to use:

```
val indexer1 = new StringIndexer().
..
val encoder1 = new OneHotEncoder().
..
val assembler = new VectorAssembler(). //feature matrix
..
val pipeline = new
    Pipeline().setStages(Array(indexer,..,assembler,NaiveBayes))
...
val nbFinalModel = cv.fit(hmdaInstitutionsBucketed)
nbFinalModel .save("<path>/HMDAModel")
```

Code Walkthrough (contd.)

HMDA - Visualization

We had to use subplots in order to slice the data (gender, state, etc.)

```
fig = make_subplots(
    rows=3, cols=2,
    column_widths=[0.5, 0.5], # corresponding to each row!
    row_heights=[0.25, 0.30, 0.45], # corresponding to
        each column!
    specs=[[{"type": "scatter", "rowspan": 2}, {"type":
        "scatter", "rowspan": 2}],[None ,None],
          [{"type": "scatter"}, {"type": "scatter"}]],
    subplot_titles=("Denial Rate Per Race", "Denial Rate
        Per Income Percentile",
                    "Denial Rate Per Ethnicity", "Denial
                        Rate Per Gender")
)
```

Insights

1. Loan Application amounts (sampled 2013 data) appear normally distributed between $10,000 to $500,000
2. Naive Bayes had the best AUC and the fastest performance time despite poor accuracy (79% AUC)
3. Poor modeling results indicate that loans are not first-order dependent on applicant race, gender, or ethnicity

Obstacles

1. Relatively large dataset, we needed to find ways to work around the speed of sparkSQL dataframes
2. The panel data was not as clean as we hoped - lenders are subsidiaries of bank holding companies (i.e. parent lender)
   - Panel data information for some lenders was not uniformly entered from year to year
   - Respondent IDs were not unique across regulating agencies (i.e. FDIC respondent 1 is not the same as Fed's respondent 1)
3. Proper model iteration hindered by data size and relatively few available models

Summary

This application will help you find the lender that will most likely extend credit based on your metadata. It leverages historical information to learn lender patterns and bias.

Acknowledgements

- NYU HPC
- CFPB for making the data publicly available
- The Federal Reserve's and CFBP's data aggregators and collectors
- Professor McIntosh!

References

- S Frumkin. "HMDA Data: Identifying and Analyzing Outliers". *Supervisory Insights, Winter 2007*. Federal Deposit Insurance Corporation.
- R. B. Avery, K. P. Brevoort, G. B. Canner. "Opportunities and Issues in Using HMDA Data". *Journal of Real Estate Research, American Real Estate Society*. Vol. 29(4), pages 351-380. 2007.
- United States Census Bureau, Department of Commerce. TIGER/Line Shapefile, 2017, nation, U.S., Current County and Equivalent National Shapefile. Data.Gov. Updated June 2019. Retrieved July 2019.
- United States Census Bureau, Department of Commerce. TIGER/Line Shapefile, 2017, nation, U.S., Current State and Equivalent National. Data.Gov. Updated February 2019. Retrieved July 2019.
- Bissmark, Johan and Warnling, Oscar. The Sparse Data Problem Within Classification Algorithms: The Effect of Sparse Data on the Naïve Bayes Algorithm. KTH, Datavetenskap, June 2017.
- HMDA: Consumer Financial Protection Bureau public website of HMDA data

Demo!

DEMO

Thanks

Thank you!!