

Consumer Financial Complaint Analysis

Ishaan Thakker
iht2086@rit.edu

Vaibhav Shah
vsb3841@rit.edu

Amol Gaikwad
asg9751@rit.edu

Abstract

The intention of this project is to analyze the United States' Consumer Financial Protection Bureau data regarding consumer complaints from 2011 to present. This dataset gives insight into consumer complaints about private financial services, including nature of the complaint, date of the complaint, company name, location of the consumer, and demographic of the consumer. By analyzing this data pertaining to complaints about loan providers, banking services, identity theft and fraud, and other financial consumer issues, the authors' goal is to discover relationships between complaints within a company or industry, as well as demographics and locations of consumers making those complaints. This dataset can be found at:

https://catalog.data.gov/dataset/consumer-complaint-database#topic=consumer_navigation

1. OVERVIEW

The purpose of this paper is to give us information about how the Data Set of Consumer complaints obtained from the government's website will help us analyze different issues. Section 2 deals with the tools used in this project, Section 3 details the current status of our work, and section 4 outlines our plan to achieve our final goal.

2. PROJECT MOTIVATION

Analysis of complaints is a very necessary problem in today's world as proper analysis of complaints can be helpful to resolve issues relatively quickly, it can also help us figure out the problems with the existing methods or medium or resolving those complaints and can also help us find the patterns related to resolving complaints in terms of a specific product of a specific company in a specific location. Analysis of complaints helps run the business smoothly and more efficiently as it increases customer satisfaction.

Apart from such issues related to problem solving and customer satisfaction another advantage

of complaints analysis is that if there is problem with a specific product of a company in any location which is not being resolved and customer satisfaction is very low then also it can be drawn to attention.

3. DIVISION OF PROJECT

This project we have worked on is divided in two parts on the whole. The first part is the Data Management part for which we have built a web application and the other part is the Data mining part for which we are using Classification techniques to classify the issues of the new complaints according to the narratives of the complaints.

3.1 TOOLS USED AND WHY

For the Data Management part we have decided to go with java-backend and use technologies such as Spring-MVC as a skeleton of the whole web application and JDBC in order to work with the database. The simple reason to choose Spring-MVC was the to remove the redundancy of the files using the advantage of the controller and faster execution of the queries and web pages as the back-end will totally work in core Java. Generally what happens if we use JSP and Servlets is that the JSP page gets converted into Servlet and then is executed which results in extra time taken by the server due to the conversion. What the MVC architecture does is it separates the user interface from the back-end working and both can work independently which results in faster execution. And JDBC is nothing but Java's database connectivity programming interface which allows the programmer to connect to the database and execute SQL queries.

For the data mining task from the many languages available to choose from, we decided to go with Python pandas, in order to work with CSV files because of the libraries it provides in order to efficiently work with CSV files. It also includes high performance and easy to use data structures for working with data. It is very efficient with working with

Consumer Financial Complaint Analysis

Ishaan Thakker
iht2086@rit.edu

Vaibhav Shah
vsb3841@rit.edu

Amol Gaikwad
asg9751@rit.edu

CSV files which is the format of our raw data. The main use of Python pandas here is utilizing the feature that extracts data column-by-column quickly.

Once we read data from the CSV file we are storing the data in data frames, which makes it easier to play around with the data.

Another option we had to work with was R programming. We chose python to work with over R, as after comparing both of the languages, we felt that cleaning the data was quite a lot easier using the libraries of Python. We are also all more familiar with python as opposed to R programming. For data visualization we have decided to go with matplotlib lib and tableau.

4. IMPLEMENTATION

In this section we are describing work done in each part of the project. Mainly the three parts of the project are Data Management, Data Mining and Data Visualization.

4.1 DATA MANAGEMENT:

First What we did for the Data Management part was that we took the original CSV file which had all the information related to customer's complaints and decided that which all attributes belong to the same table. After such design consideration of the database we moved forward to normalize the Database in 10 separate tables. After spending a considerable amount of time analyzing and understanding its components. We have iteratively devised relational schemas for our database in order to best serve our needs for analysis.

The current, normalized schema we have developed is as follows and the Relational diagram is shown in figure 1.:

CFPB_Complaint(complaint_ID, date_received, product_ID, subproduct_ID, issue_ID, subissue_ID, company_ID, submitted_via, complaint_narrative, tags, date_sent_to_company)

Company_Response(response_ID, complaint_ID, company_ID, company_response, response_status, timely_response, consumer_disputed)

Company(company_ID, company_name)

Company_location: (company_ID, ZIP)

Product(product_ID, product_name)

Subproduct(subproduct_ID, product_ID, subproduct_name)

Issue(issue_ID, issue_name)

Subissue(subissue_ID, issue_ID, subissue_name)

Customer: (customer_ID, customer_name, ZIP)

Location : (ZIP, state, city)

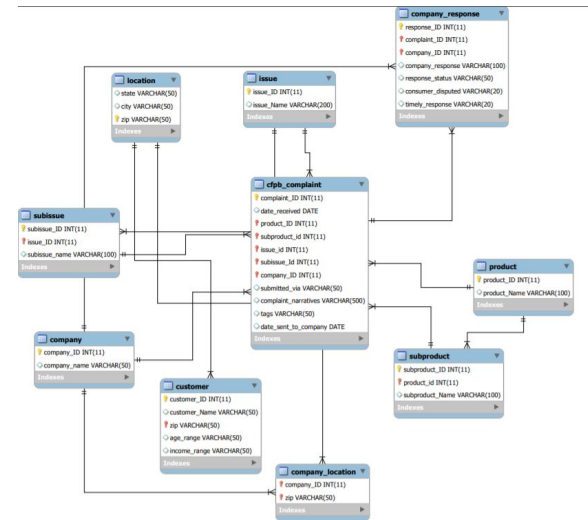


Figure 1: Relational Diagram

We then created all the tables in MySQL according to the relational schema and created sample code for loading data from CSV files(consumer_complaints.csv and free-zipcode-database-Primary.csv) to a MySQL database. We used another set of data to enter to values to ZIP as the consumer_complaints dataset does not have all the zip codes in it as it has only the zip codes from which they currently have record of the complaint. But as we wanted to enforce foreign key constraint we used a different set of data which had all the zip codes on United states which will make it easy to add new data in the future with checking the foreign key constraints.

After normalizing the tables we wrote a python script to read the data from the CSV file and load it accordingly in the normalized tables. The approach which we took for loading the database was

Consumer Financial Complaint Analysis

Ishaan Thakker
iht2086@rit.edu

Vaibhav Shah
vsb3841@rit.edu

Amol Gaikwad
asg9751@rit.edu

that we first loaded only a few values to check whether the data was loaded properly or not. Once we saw that a small set of values are being loaded properly in the database then we tried running the script on the whole dataset which comprises of approximate a million rows. Now here comes the interesting part which is that in order to load the data in the tables quickly what we did was we did not enforce any foreign key constraint on any of the tables because if the constraint was enforced then to add each row for each table it would have checked for the constraint and taken a lot of time which would have resulted in slower execution. Once all the data was loaded in all the tables we enforced foreign key constraints.

So for this time loading the data efficiently we used python dictionary data structure. The main intention to use it was to link the attributes with its foreign keys with optimized code. Here below is a sample of the data structure:

- Structure: {"table_name" : {"column in database" : "column in csv"}}
- Structure: {"table" : {"foreign key" : "primary key table"}}
- Structure: {"table_name" : {"unique in db" : "column in csv"}}

Now the added data had varchar formats for Date as it was faster for execution so what we later did was that we wrote another python script to modify the attribute values of attributes which were supposed to be date. Once that was done in order to demonstrate efficient management of data we added a few attributes by our own which were not present in the original data like customer_name, Age and Income and then fabricated data for that to query the data according to the age of the customer or income of the customer. We decided to add these attributes as we felt that the current data was missing such information. As we did not have the data from our original csv file we decided to add data to such attributes on a random basis but we kept those random parameters fairly realistic.

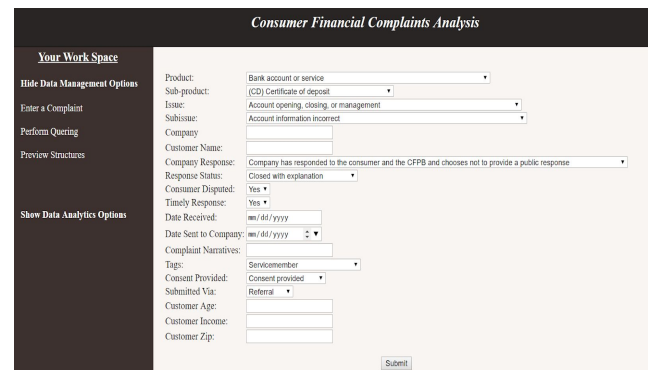


The screenshot shows a web application interface with a sidebar on the left containing navigation options like 'Your Work Space', 'Hide Data Management Options', 'Enter a Complaint', 'Perform Querying', 'Preview Structures', and 'Show Data Analytics Options'. The main content area displays a table titled 'Consumer Financial Complaints Analysis' with columns 'company_id' and 'company_name'. The table lists the following data:

company_id	company_name
4023	M&T BANK CORPORATION
4024	TRANSUNION INTERMEDIATE HOLDINGS, INC.
4025	CITIZENS FINANCIAL GROUP, INC.
4026	AMERICAN EXPRESS COMPANY
4027	CITIBANK, N.A.

Figure 1

Once the database was ready we proceeded to work on the Web portal from which user see the demonstration of our Data Management. We created front-end of the web application using HTML and CSS. The web portal allows a number of activities available to the user. The user can see the structure of the data tables available in the database and also preview their structures as shown in Figure 1. The preview of the data tables would populate the user interface with top 5 rows of each the table selected so that the user can see what kind of data to expect in what table.



The screenshot shows a web application interface with a sidebar on the left containing navigation options like 'Your Work Space', 'Hide Data Management Options', 'Enter a Complaint', 'Perform Querying', 'Preview Structures', and 'Show Data Analytics Options'. The main content area displays a form titled 'Consumer Financial Complaints Analysis' with various input fields and dropdown menus for entering complaint details. The form includes fields for Product, Sub-product, Issue, Sub-issue, Company, Customer Name, Company Response, Response Status, Consumer Disputed, Timely Response, Date Received, Date Sent to Company, Complaint Narratives, Tags, Consent Provided, Submitted Via, Customer Age, Customer Income, and Customer Zip. A 'Submit' button is located at the bottom right of the form.

Figure 2

Apart from that we also allow the user to insert a new complaint in the database which we are doing using HTML forms. The form will have a

Consumer Financial Complaint Analysis

Ishaan Thakker
iht2086@rit.edu

Vaibhav Shah
vsb3841@rit.edu

Amol Gaikwad
asg9751@rit.edu

numerous fields for the user to fill out of which a few will be populated using the current data in database like Complaint, Issues, Products, Sub-issues of issues, sub-products of products and location.

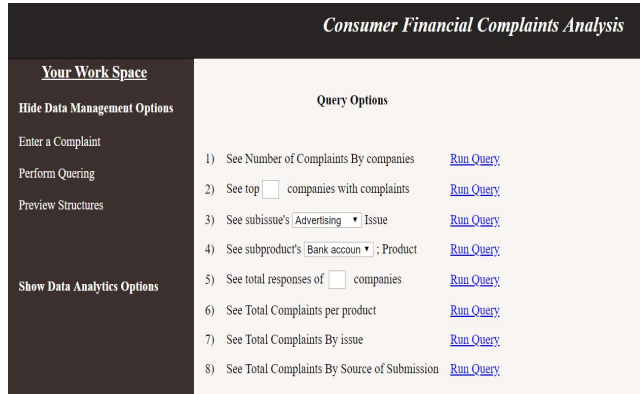


Figure 3

The most important feature provided by the web portal is shown in the Figure 3. it gives the user privilege of basic querying by taking values from the user. There are numerous queries a user can run by providing values in the input field like get number of complaints grouped by companies, see highest complaints according to location, by company, by product or by any issue. This we basic analysis can be performed and user can retrieve information. And lastly the user can also delete any complaint by it's id if that complaint is not relevant any more.

4.2 DATA MINING

We first need to clean our dataset. Certain columns, such as those pertaining to the company name and the issue type, have redundant or incomplete information that needs to be resolved before we proceed. Since there is also concern about the richness of the data set, we will also be filling in some of the null values with generated data (including location data that is missing for roughly 15,000 records). After handling the data inconsistencies we played along a number of ideas and algorithms in order to decide what kind of problem should we select

for mining. The possible insights of data mining on this data set include answers to questions such as:

- Is there a relationship between consumer demographics and the type of complaints they are submitting?
- Are there trends pertaining to the types of complaints that are submitted about individual financial institutions?
- Based on the new incoming complaint how much possible it is to classify the issue of the complaint based on the features selected.
- Is consumer location significant in the types of complaints they are submitting, or which financial institutions they are reporting?
- Is there a relationship between the company's public response and how the complaint was settled?
- Are certain companies more likely to respond in a timely manner, or have their responses to complaints disputed?
- Does a customer in a specific age range have problem with a specific product
- Does demographic location relate to customer complaints in a specific age range or a specific income range.
- What kind of customers in a specific income range complain about which product.
- Is complaint of product relatable to a specific season/month in a year or not.

From the above mentioned problem what we thought that the most interesting task to choose as a part of data mining is to classify the issue of the new complaints coming based on the complaint narratives of the customer. This is a very important issue as if implemented properly then the task of classifying the complaints manually would no longer be used and it would be of great advantage as it takes a lot of time to classify the complaints manually. It is very much possible to give a chance to the customer themselves to classify the complaint at the time of complaining but it is of very use because in many of the cases the customer is only not aware of the nature and the complexity of the problem as many issues lie in overlapping domains.

Consumer Financial Complaint Analysis

Ishaan Thakker
iht2086@rit.edu

Vaibhav Shah
vsb3841@rit.edu

Amol Gaikwad
asg9751@rit.edu

```
In [187]: predicted = classifier.predict(y_test)

In [188]: len(predicted)
Out[188]: 23497

In [189]: runfile('C:/Users/vbsha/Downloads/naive_ayes_classification.py', wdir='C:/Users/vbsha/Downloads')

In [190]: accuracy_score(y_test, predicted)
Out[190]: 0.5518151253351492

In [191]: runfile('C:/Users/vbsha/Downloads/KNN_classification.py', wdir='C:/Users/vbsha/Downloads')
```

Figure 4.1: Output for Naive Bayes Classifier

Here in this picture we have shown the output of Naive Bayes classifier with 1000 features. The libraries we have are Sklearn, Nltk, Pandas and pickle.

Originally what our data has is 166 unique issues related to products. But all of the issues do not of enough Complaint Narratives. So what we did was just chose 10 issues with highest number of narratives for classification. This issues have at least 4000 Narratives each. First what we did was clean the data, tokenized it and removed all the stop words and also the numeric values keeping only alphabetic values.

After that what we did is we generated TF-IDF sparse matrix of this data. The original unique words (columns) in TF IDF vector was 54,000 which is not possible to handle by Naive Bayes classifier so we selected only top 1000 features from it. Basically what TF-IDF vector is each row represents each document and each column represents unique word in that document so the value in each item in the matrix will represent the weight of the word in that document. After this we converted this sparse matrix in numpy array. There is also a function called clean_data which cleans the data.

After this was done what we did was split the data is split it in 80% and 20% for training and testing respectively. We labeled each issue with a number uniquely. After this we trained our Naive Bayes Classifier on the training data and later predicted the testing data over our trained classifier. After testing

the data what we found was that the accuracy of the classifier was approximately 57.10%.

In order to see if another algorithm worked more efficiently we trained and tested the data in similar way with another algorithm named KNN classifier. But what we found that the accuracy of this classifier was lower as we found just 47% accuracy with it and also it's execution took more time compared to Naive Bayes. which was the biggest downside.

```
self.items]

File "C:\ProgramData\Anaconda3\lib\site-packages\sklearn\externals\joblib\parallel.py", line 131, in <listcomp>
    return [func(*args, **kwargs) for func, args, kwargs in self.items]

KeyboardInterrupt

In [192]:

In [192]: runfile('C:/Users/vbsha/Downloads/KNN_classification.py', wdir='C:/Users/vbsha/Downloads')
0.475081925352

In [193]: with open('KNN_classifier_trained.pkl', 'wb') as f:
...:     pickle.dump(clf)
...:
```

Figure 4.2: Output for KNN classifier

Above in the figure 4.2 we can see the output for KNN classifier which is only 47% and sufficiently compared to Naive Bayes Classifier. So comparing both the algorithms we can see that in our case Naive Bayes works far better compared to KNN classifier.

Loan's contract number	0	0	0	0	0
Can't repay my loan	0	0	0	0	0
Can't stop charges to bank account	0	0	0	0	0
Cash advance	0	0	0	0	0
Cash advance fee	0	0	0	0	0
Charged bank acct wrong day or amt	0	0	0	0	0
Charged fees or interest I didn't expect	0	0	0	0	0
Closing an account	0	0	0	0	0
Closing on a mortgage	0	0	0	0	0
Closing your account	0	0	0	0	0
Closing/canceling account	0	0	0	0	0
Collection debt dispute	0	0	0	0	0
Collection practices	0	0	0	0	0
Communication tactics	0	0	11	3	0
Can't get my credit report	0	0	3	32	0
Consumer credit	0	0	0	0	0
Credit card protection / debt protection	0	0	0	0	0
Credit decision / Underwriting	0	0	0	0	0
Credit determination	0	0	0	0	0
Credit line increase/decrease	0	0	0	0	0
Credit monitoring or identity protection	0	0	0	0	0
Credit reporting	0	0	0	0	0
Credit reporting company's investigation	0	0	0	5	0
Customer service / Customer relations	0	0	0	0	0

Figure 4.3: Confusion Matrix

Consumer Financial Complaint Analysis

Ishaan Thakker
iht2086@rit.edu

Vaibhav Shah
vsb3841@rit.edu

Amol Gaikwad
asg9751@rit.edu

The above figure is the sample confusion matrix generated.

4.3 DATA VISUALIZATION

Along with Data management and Data mining we have provided visualization over several components

Number of complaints by State

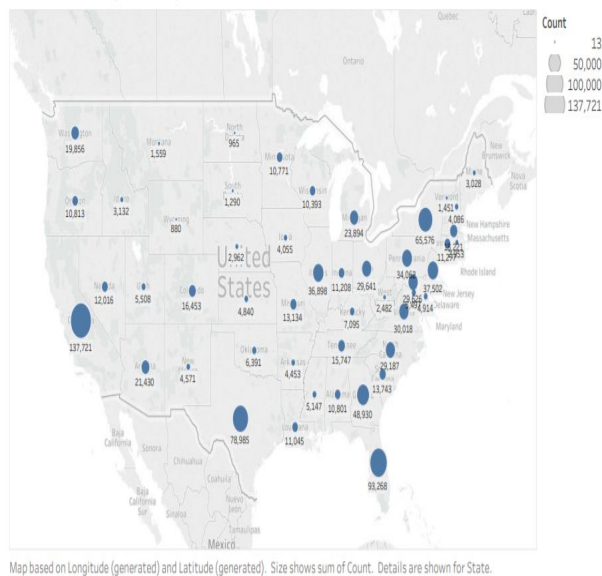


Figure 5: Representation of complaints by states

Changes in the complaints by companies

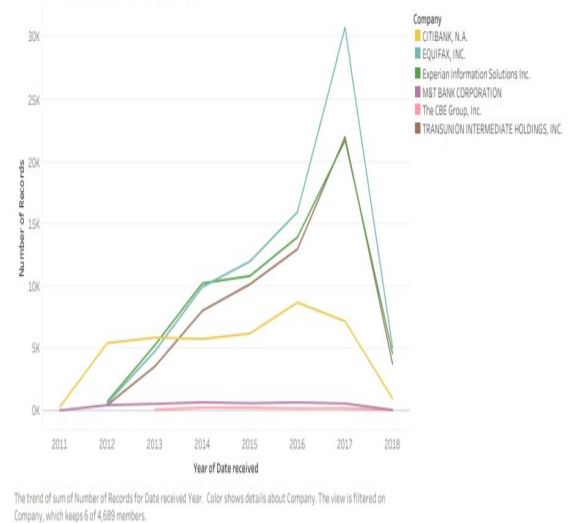


Figure 6: Representation of complaints by companies by years

Figure 4 shows representation of complaints based on their location and the level of location chosen by us is state for better representation of complaints geographically. Using this we can easily make out which areas have maximum complaints so that it can be used for further analysis.

Similarly figure 5 is the representation of complaints for each companies keeping track of the year. This is very useful information as we can know that which company faces more complaints by years or which companies have less complaints.

Consumer Financial Complaint Analysis

Ishaan Thakker

iht2086@rit.edu

Vaibhav Shah

vsb3841@rit.edu

Amol Gaikwad

asg9751@rit.edu

Top 10 companies with highest complaints

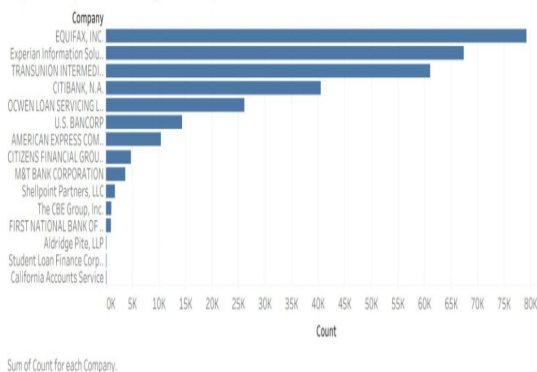


Figure 7: Representation of companies with highest complaints.

Here figure 6 above gives us information about top 10 companies which are having highest amount of complaints. Here this graph is very much useful as we can know where more work is required or what is the real problem which needs to be resolved in these companies.

Changes in complaints by products over the year

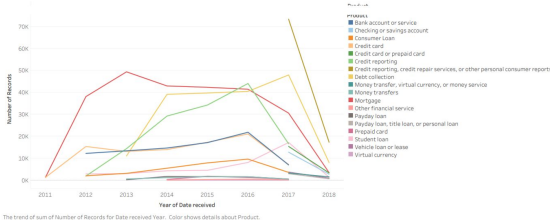


Figure 8: Representation of complaints of products by year

Here in the figure above (figure 7) we have shown the graph number of complaints for each product available according to year.

4. FUTURE WORK

There are a number of applications of this project. In the future, we plan to integrate the data mining part with the web application so that the user can perform the data mining tasks using the web portal itself. We may

plan to keep the data mining task semi dynamic which means that there are a few tasks will be given as options in the web portal and user can get just select which activity is to be performed on the data and can get the results.

Apart from the data mining tasks discussed above, we may also plan to work on some machine learning and neural network algorithms which can be also implemented in order to predict few tasks like:

- Predicting the response of a company on a certain type of a new complaint and how much time it will approximately take to resolve it.
- Predicting the complaints based on a specific demographic, age range, salary range in future.

In the future, we also plan to improve the efficiency of our algorithm by collecting more robust data which can be used for classification. This data can be obtained by gathering data from multiple websites.

5. LIMITATIONS

Currently, what our limitation is that the accuracy of our algorithm is just 57.1% which needs to be improved a lot. But it is very difficult to improve it as we do not have good data which we can take as features for classification. And also the data which we have is not quite enough as many complaints do not have proper complaint narratives.

5. References

- Albon C. 2017. Loading a CSV into pandas. Chris Albon.
- Pandas reads CSV. Python Tutorials.
2016. Importing data from a MySQL database into a Pandas data frame including column names [duplicate]. Overflow.
2017. Loading CSV data with Python in pandas. Python How
- CFPB's consumer complaints database analysis reveals valuable insights. Deloitte
- An Introduction to data cleaning with R, cran.r-project
- JDBC insert records tutorials point.
- Jason Brown lee, How to implement Naive Bayes in Python
- Jason Brown lee, How to implement KNN in Python.