**Assignment: Consumer complaints**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Name (as appears in Canvas)** | **ID NO** | **Contribution** |
| **1** | **SUMIL KHOSLA** | **2021SC04457** | **100%** |

**Problem Statement**

The banks that are receiving customer complaints filed against them will analyse the complaint data to provide results on where the most complaints are being filed, what products/ services are producing the most complaints and other useful data. These datasets fall under the complaints of Credit reporting, Mortgage, Debt Collection, Consumer Loan and Banking Accounting.

**Objective**

Build a clustering model using Python language or other suitable tools to find how many similar complaints are there in relation to the same bank or service or product. This model will be used to assist banks in identifying the location and types of errors for resolution leading to increased customer satisfaction to drive revenue and profitability.

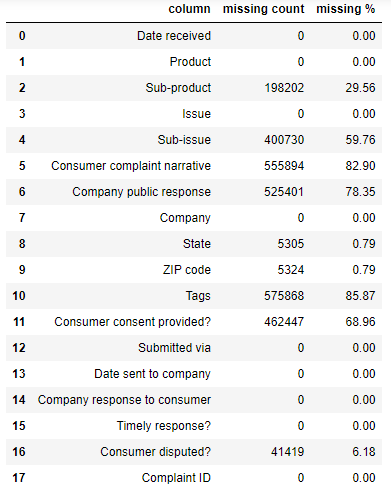
**Solution Approach:**

The solution approach of the problem can be broken down into following steps:

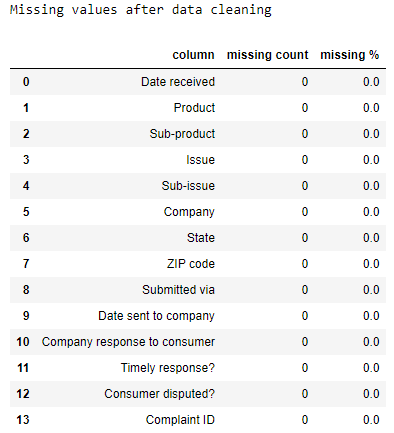
1. Data pre-processing and Exploratory Data Analysis
2. Data visualization
3. Building Model
4. Performance evaluation of Clustering models

**1. Data pre-processing and Exploratory Data Analysis:**

* Imported the dataset
* Checked information about dataset: 18 columns and 670598 records found. Datatype of “Complaint ID” is integer and remaining are object
* Identified Missing values & imputed missing values
* Dealt with columns having high Cardinality
* Created new features “Year”, “Month”, “Day” from column “Date received”
* **Imputing missing values**

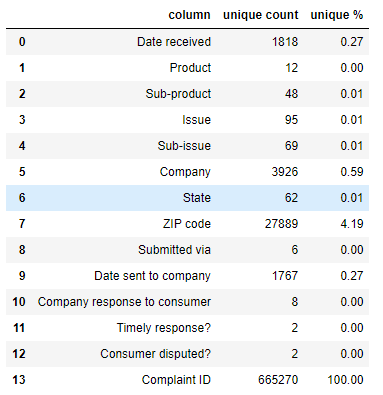


1. Dropped 4 columns (Consumer complaint narrative, Company public response, Tags, Consumer consent provided) having >60% missing values
2. “State” and “ZIP code” have only 0.79% missing values. So, removed data with missing state and ZIP code
3. “Consumer disputed?” is independent categorical feature and its missing values imputed with mode
4. “Sub-product” and “Sub-Issue” are sub-categories of “Product” and “Issue” respectively. Its missing values imputed with “Product” and “Issue” wise mode

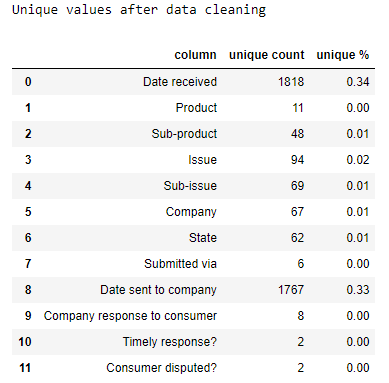


* **Dealing with columns having high Cardinality**

Dealing with features having too many unique vales is important since we are applying One-hot encoding and it will have separate columns for each unique value (indicating its presence or absence). This leads to two problems: **Space consumption** and **high dimensionality.**

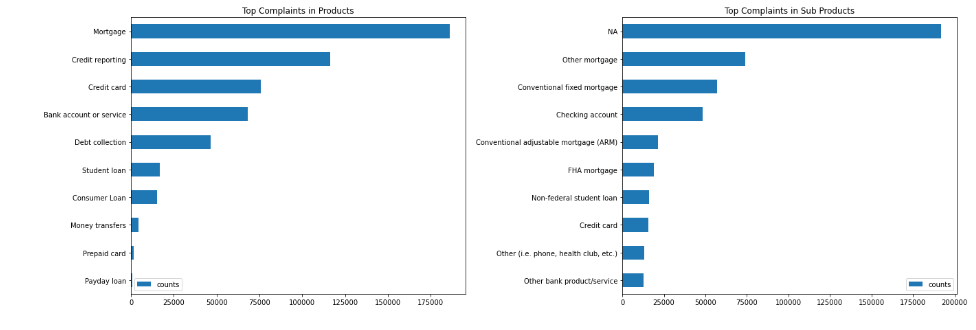


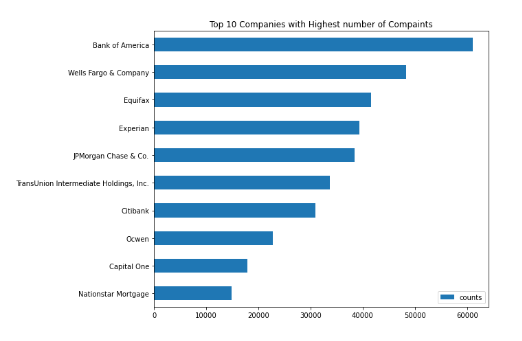
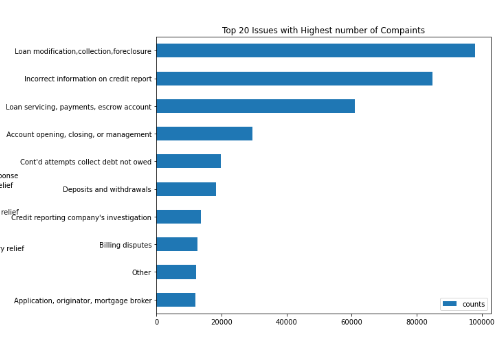
* 1. Dropped columns (ZIP code, Complaint ID) with high uniqueness percentage (>4%)
  2. “Company” column has 0.59% unique values.
  3. Out of 3926 unique values for “Company” column, 3859 companies have less than 1000 complaints, and can be dropped

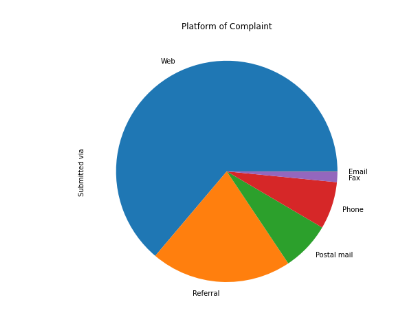
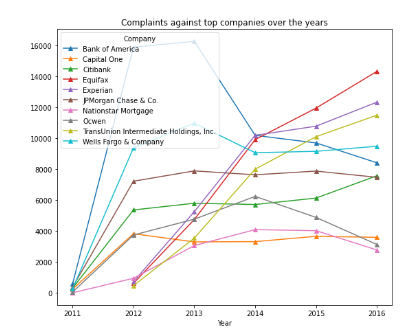


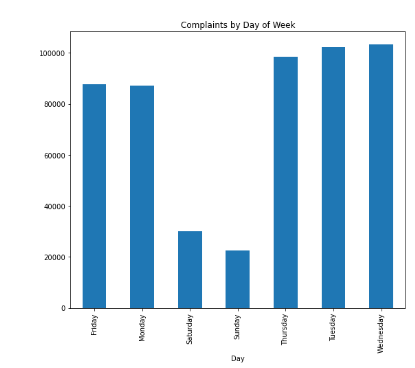
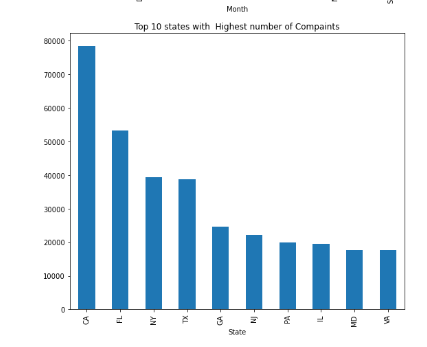
**2. Data Visualization:**

Plotting the association of complaints with various categorical features.



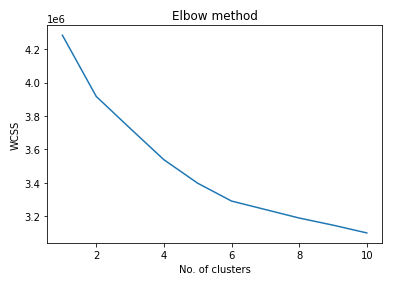
 

Observations:

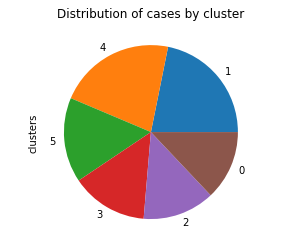
* Top complaints in "Products" are related to Mortgage, credit reporting, credit card, Bank account/service, debt collection.
* Top "Companies" with highest no. of complaints are Bank of America, Well fargo & company, Equifax, Experian, PMorgan chase & Co., TransUnion intermediate holdings, Citibank, Oswen, Capital one, Nationstar mortgage.
* Top "Issues" with highest no. of complaints are Loan-modification-collection-foreclosure, incorrect information on credit report, Loan-servicing-payments-ascrow-account, Account-opening-closing-management.
* Most of the complaints received via Web, Referral, Postal mail, phone.
* Most of the complaints received on weekdays (Monday to Friday).
* Top "States" with highest no. of complaints are CA, FL, NY, TX, GA, NJ, PA, IL, MD.

**3. Building Model:**

* Feature selection
* Out of the features available after pre-processing, we can drop “Date received” and “Date sent to company” since it has no relevance for model building
* “Year” can be dropped since it cannot be used as a categorical feature for clustering
* Therefore, selected features are “Product, Sub-product, Issue, Sub-issue, Company, State, Submitted via, Company response to consumer, Timely response?, Consumer disputed?, Month, Day”
* One-hot encoding of selected categorical features
* Using one-hot encoding, we create a new set of dummy (binary) variables that is equal to the number of categories in the categorical feature.
* The dummy (binary) variable will just take the value 0 or 1 to indicate the exclusion or inclusion of a category.
* This way, we have a separate column for each unique value of a categorical feature
* Implementing **K-Means** clustering
* This algorithm computes centroids and repeats until the optimal centroid is found. It is presumptively known how many clusters there are.
* We used Elbow method to find optimal no. of clusters: We see bend in graph when no. of clusters is 6, so we can consider 6 as the optimal no. of clusters.



* Applied K-Means clustering to form clusters

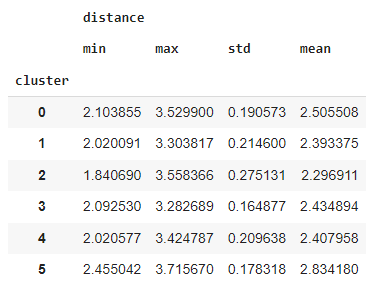
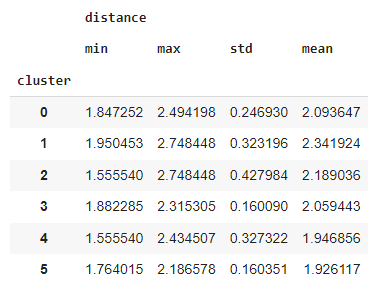


* Agglomerative Hierarchical clustering
* In this technique, initially each data point is considered as an individual cluster. At each iteration, the similar clusters merge with other clusters until one cluster or K clusters are formed.
* Hierarchical clustering is difficult to implement with large datasets due to high memory requirement. We tried for this dataset but got MemoryError (Unable to allocate 1.03 TiB for an array with shape (141382487646,) and data type float64)
* DBSCAN clustering
* Key idea is that for each point of a cluster, the neighbourhood of a given radius has to contain at least a minimum number of points.
* DBSCAN is too slow and consumes too much memory for large datasets

**Note:** We have chosen K-Mean clustering over Hierarchical clustering and DBSCAN because of these constraints.

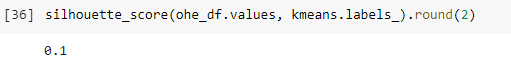
**4. Performance evaluation of Clustering models:**

* Cohesion and Separation technique:
* Cohesion - Observations within a cluster should be as close as possible
* Separation - Observations from two clusters should be far from each other
* Cohesion is measured by SSE and Separation is measured by BSS
* For cohesion, lower value is desirable, whereas for separation higher value is desirable

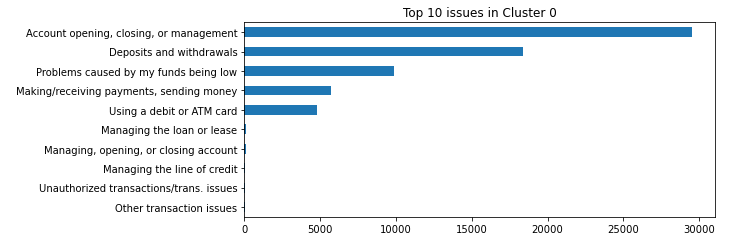
* Silhouette coefficient: The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighbouring clusters.

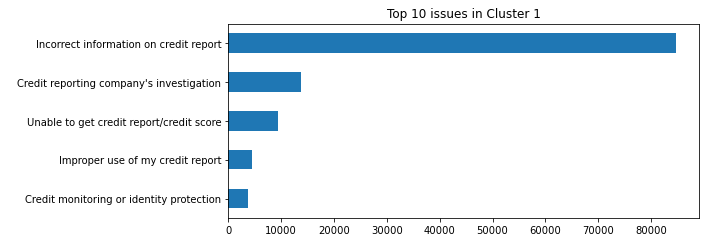


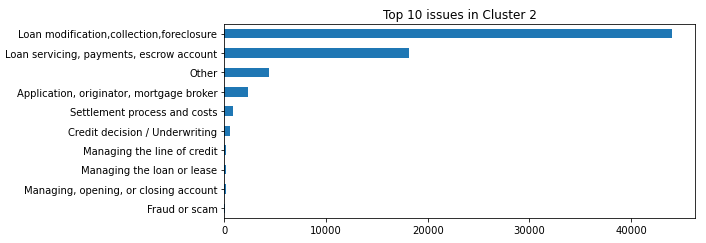
We got Silhouette score 0.1 which is good for clustering.

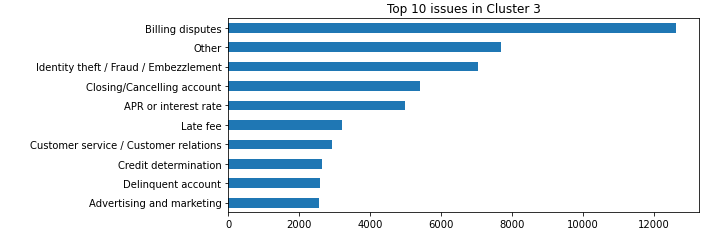
**Conclusion:**

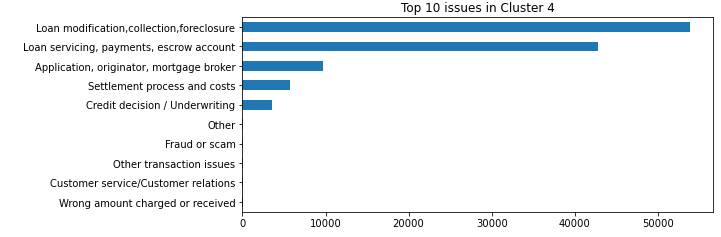
* We have got 6 optimal no. of clusters
* Cluster 1 has only 5 types of issues, whereas others clusters have 10 or more
* Top issues are not repeating across clusters. This means that the distribution of issues among clusters is good

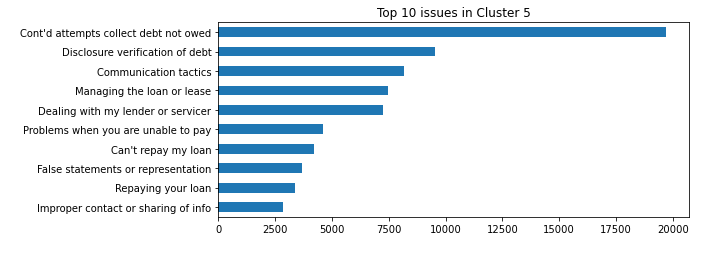












**Summary:**

* A positive silhouette score indicates that clusters are well formed
* Analysis of top issues in all clusters indicate that the distribution of issues among clusters is good
* But silhouette value being close to zero, and very close cohesion and separation values suggest that there can be some overlapping, and the model can be improved

**Suggested ways of Improvement:**

* For k-means algorithm, **initialization** can be the key to the performance
* If the center with the maximum distance gets initially chosen, we can easily select an outlier as the center.
* To alleviate this, we can try initializing the centers multiple times and selecting the initialization that has the lowest inertia
* More sophisticated initialization techniques can also the tried like **Partitioning Around Medoids**