**CAR LOAN APPLICATION WITH CAR SPECIFIC AND POPULARITY INFORMATION**

**Abstract**

The purpose of the assignment is build a conceptual schema for car loan application containing car specific information and car popularity information using real world data. The data was obtained from three sources were cleaned, reformatted and then combined to fit the conceptual database schema.

**Data Source:-**

* The Dataset was downloaded from kaggle. The Dataset contained more than twenty thousand entries and also more than thirty columns of Applicant information till August 2016. <https://www.kaggle.com/sumit9/pricing-model>
* Data was scraped from https://**autoportal**.com
* Using Twitter Web API

**Conceptual Data Schema**

The conceptual schema shows the entities and the relationship between the entities. The various entities are as follows:

1. **Car\_Loan\_Application**

Primary Key : Car\_Loan\_Application\_ID

Attributes: Requested\_Amount\_In\_INR, Age, Applicant\_Postal\_Code, Car\_Fullname, Loan\_Term, Applicant\_State\_Desc, Applicant\_City\_Desc, Car\_Loan\_Application\_Creation\_Date, Car\_Type, Cibil\_Score, Car\_Loan\_Disbursed, IRR.

1. **Car**

Primary Key : Car\_Fullname

Attributes: Car\_Price\_In\_INR, Car\_Power, Car\_Mileage, Car\_Fueltype

1. **Car\_Popularity\_Information**

Primary Key : Car\_Fullname

Attributes: Total\_Status\_Count, Total\_Favorite\_Count, Total\_Retweet\_Count

Relationship between the entities are as follows:

1. Car and Car\_Loan\_Application have one to many relationship as one car variety could have been used for multiple loan applications by multiple applicants.
2. Car and Car\_Popularity\_Information have one to one relationship as one car row from Car entity can only have one related row in Car\_Popularity\_Information entity.

**Data Reformate:-**

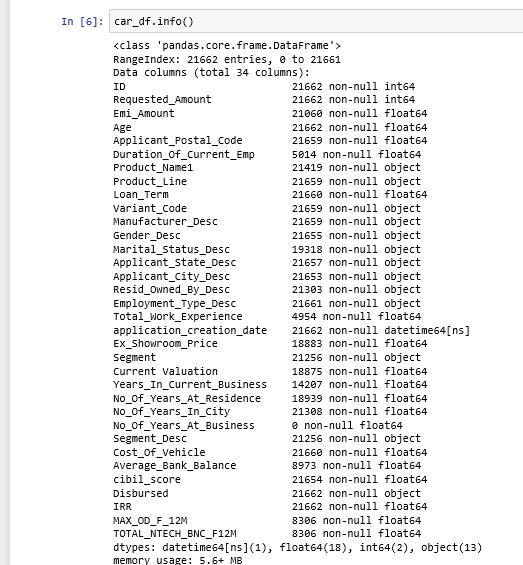
The whole dataset had more information than required for the proposed conceptual database model. Dataset was divided into two separate tables like Applicant information And Vehicle Information. The non-numeric data entries weren’t uniform (somewhere in capital), they were replaced uniformly with small letters. A few of the columns have been renamed for the purpose of making it simple and understandable.

**Data Cleaning:-**

**Data cleaning** is the process of deleting the corrupt records from a column or entry and refers to identifying null values and also incorrect entries in the table. After having a rough idea on the conceptual Database, we will also have a few column that are of no use and can be dumped using the python commands. Data cleaning is done to remove the null values and all also the discrepancies in the data.

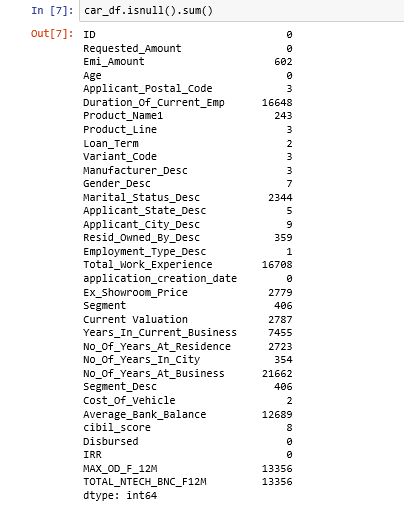
Before cleaning the data we need to find the variable types and also the number of entries in the column. So using the command

car\_df.info()



After that we check for the null values on each row using the command,

car\_df.isnull().sum ()



Once we check for the null values, we can see the count of the null values in each of the columns. To eliminate this null values we can either delete the entire column when it is of not much significance to the proposed data model or we can erase only the row containing the null value when the column is of much importance.

This table here explains the reason for deleting the columns-

|  |  |
| --- | --- |
| 'No\_Of\_Years\_At\_Business | Belongs to applicant entity /null values |
| Duration\_Of\_Current\_Emp | Belongs to applicant entity /null values |
| Total\_Work\_Experience | Belongs to applicant entity /null values |
| MAX\_OD\_F\_12M | Insignificant /null values |
| TOTAL\_NTECH\_BNC\_F12M | Insignificant /null values |
| Current Valuation | Insignificant /null values |
| Ex\_Showroom\_Price | Data obtained from web scrapping |
| Marital\_Status\_Desc | Belongs to applicant entity /null values |
| No\_Of\_Years\_At\_Residence | Belongs to applicant entity /null values |
| Average\_Bank\_Balance | Belongs to applicant entity /null values |
| Cost\_Of\_Vehicle | Redundant/null values |
| Emi\_Amount | Insignificant /null values |
| No\_Of\_Years\_In\_City | Belongs to applicant entity /null values |
| Segment | Insignificant /null values |
| Resid\_Owned\_By\_Desc | Belongs to applicant entity /null values |
| Gender\_Desc | Belongs to applicant entity /null values |
| Employment\_Type\_Desc | Belongs to applicant entity /null values |
| Product\_Name1 | Redundant/null values |
| Variant\_Code | Belongs to applicant entity /null values |
| Years\_In\_Current\_Business | Belongs to applicant entity /null values |
| Manufacturer\_Desc | Redundant/null values |
|  |  |

|  |  |
| --- | --- |
| Segment\_Desc | Nulls values in a few rows |
| cibil\_score | Nulls values in a few rows |
| Applicant\_Postal\_Code | Nulls values in a few rows |
| Loan Term | Nulls values in a few rows |
| Applicant\_City\_Desc | Nulls values in a few rows |

There was a chance to create another entity by name Applicant with applicant specific information obtained from raw CSV. Since there were no field having unique values pertaining to the applicant, the ‘Applicant’ entity couldn’t be created and its related fields were dropped

**Raw CSV**:

Data fields obtained from CSV include:

Car\_Loan\_Application\_ID: Unique ID associated with every car loan application

Requested\_Amount\_In\_INR: Amount requested as loan in INR

Age: Age of the applicant

Applicant\_Postal\_Code: Postal code of the applicant

Car\_Fullname: Make, model and variant information of the car

Loan\_Term: Term of loan in months.

Applicant\_State\_Desc: Resident state of the applicant

Applicant\_City\_Desc: Resident city of the applicant

Car\_Loan\_Application\_Creation\_Date: Creation date of the car loan application

Cibil\_Score: Cibil score of the applicant

Car\_Loan\_Disbursed: Indicator, whether the loan amount has been disbursed

IRR: Interest rate

**Web Scrapping**

The web scrapper used the beautifulsoup package and also the panda framework. This was written using the python commands. Urllib was used to access the specified webpage by the scrapper. Number of pages of the webpage were scraped to collect around 1300 records of car data. The HTML tags were also used to scrape data in various stages in the code to gather all the data from the webpage. The scraped data was stored in the pandas data frame before which each page for the defined information.

Data fields scraped using Webscraper include:

1. Car\_Fullname: Make, model and variant information of the car.
2. Car\_Price\_In\_INR: Total price of the car in Indian National Rupees(INR).
3. Car\_Power: Power of the car in rpm, bhp etc.
4. Car\_Mileage: Mileage of the car mentioned in liter/km.
5. Car\_Fueltype: Fuel type of the car(Petrol, Diesel etc)

**Web API**

The twitter API was accessed through the authentication token. The data obtained through Web scrapping was used to prepare hashtags for respective cars. The hashtags were given as input parameter to call the twitter api along with the count to restrict the number of tweets.

Only the ‘STATUSES’ portion of the result were considered for extracting the popularity of the car firstly the English language filter was applied , the overall count of favorite and the retweets were stored in respective lists . Finally a data frame was prepared using all the retrieved information and the data from the data frame was exported to a CSV file .

Web API:

Data fields obtained from Web API include:

1. Total\_Status\_Count: Total number of twitter statuses recorded against a particular car.
2. Total\_Favorite\_Count: Total number od favorites recorded against all statuses associated with the car.
3. Total\_Retweet\_Count: Total number of retweets recorded against all statuses associated with the car.

**AUDIT VALIDITY:-**

The whole dataset had more information than required for the proposed conceptual database model. To fit in to the database schema, the dataset was reformatted. Dataset was divided into two separate tables like Applicant information And Vehicle Information. The non-numeric data entries weren’t uniform (somewhere in capital), they were replaced uniformly with small letters. A few of the columns have been renamed for the purpose of making it simple and understandable.

**AUDIT COMPLETENESS**

The dataset is up to date, needs no more cleaning and matches the quality of the real world data.

**AUDIT CONSISTENCY/UNIFORMITY**

The possible range of the dataset is covered from the new resultant data set. The data does not have any null values, limitations, negative values.

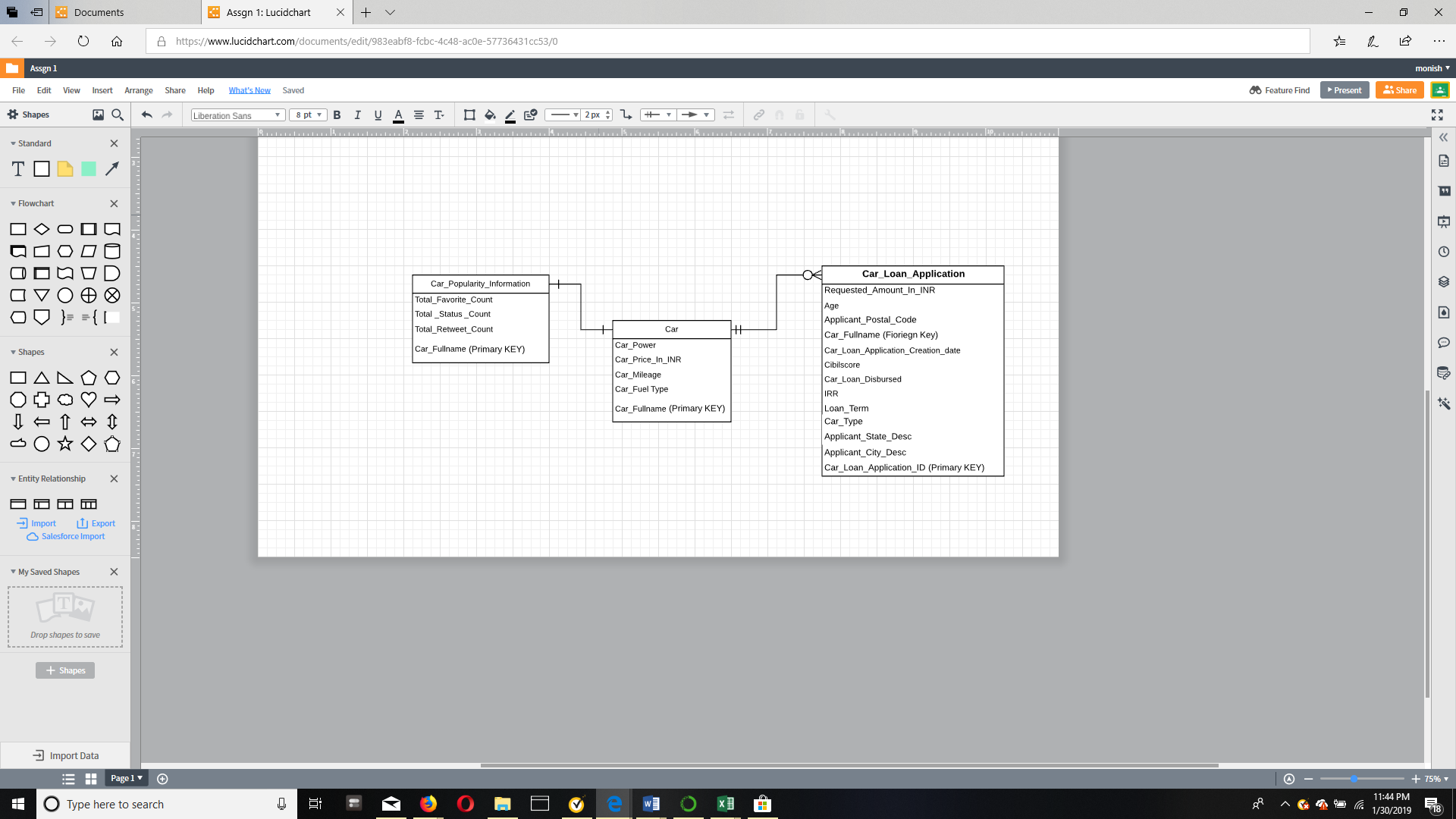
**FILE INFORMATION :-**

The following are the assignment files

1. Source\_1\_Web\_API\_Extraction.ipynb ->Jupyter file of the code.
2. Source\_2\_Web­\_scraping.ipynb
3. Source\_1\_Web\_API\_Extraction.csv –contains info gathered using API
4. Source\_2\_Web­\_scraping.csv-contains info gathered using web scraping.
5. Source\_3\_Excel\_Data\_Extraction.ipynb
6. Source\_3\_Excel\_Data\_Extraction.csv
7. Assignment 1 DMDD Report

**Conceptual Database Schema-**

Diagram:-

This diagram was created using lucid software tools

**Conclusion:**

In the assignment, all of the CSV tables were populated are populated with real-word data collected from three sources: web scraper, web API and raw CSV file. A conceptual schema is created for cap loan application with car specific and popularity information model. The collected data was reformatted, cleaned to fit the database schema.

Citations and References:

1. https://www.geeksforgeeks.org

2. https://github.com/nikbearbrown/INFO\_6210

3.google

4.kaggel <https://www.kaggle.com/sumit9/pricing-model>

5. <https://www.kaggle.com/sumit9/pricing-model>