**ETL PROJECT PROPOSAL**

**Team members:**

1. Misael Obregon

2. XuanCong Tran

Project describtion:

1. Finding Data

In this project, we propose to create a database that captures information on cars and their respective brand, model, year, and best deals currently on Edmunds.com. Firstly, we use a datafile from Kaggle.com named top\_selling\_car.csv, then based on the model and what brand on the list, we will use tools of web scraping to get the best deal price for that car in a year.

2. Data Cleanup & Analysis:

We will analyze and manipulate the data using Python and SQL. Data not found and matching between Edmunds car sale site and Kaggle “top\_selling\_car.csv” data will be deleted and not included in the final joined database.

3. Store data:

We will do the database on PostgresSQL.

**Data Cleaning and Organization of CSV Data**

The following steps were done to organize and clean data. Renaming of first column from original data set (top\_selling\_car.csv) has been renamed from “#” to “Ranking,” which represents the top selling used car (Models sold, in thousands), data was create in 2018 and accessed from Kaggle.com.

Columns E - I from original “top\_selling\_car.csv” dataset are dropped as only first four columns are needed.

|  |  |
| --- | --- |
| **Columns Needed** | **Columns Dropped** |
| A = Ranking | E = Models sold, in thousands |
| B = Car brand | F = Maximum speed, km/h |
| C = Car model | G = Percent of sells among the brand |
| D = Release year | H = Sales in 2018, in thousands |
|  | I = Price, $ |

**Methodology for Scraping Data and Loading CSV File onto Pandas**

We begin to scrape data from Edmunds.com website. Using make, model, and year information from the csv original table, we do a search on Edmunds.com for cars using same criteria listed above to scrape data from results page found using Edmunds.com website. Using Pandas and python for code we specifically look for data on a car’s brand name, model, description, VIN number, mileage, and price. Any data provided in rows that result in a “none” value are dropped from data frame.

After getting results from scraped data a data frame (df) was created. Note, as mentioned above, rows with no found cars were deleted.

Once a cleaned data frame has been created, we connect to the local database (PostgresSQL and the csv file to query both data sets.

* The csv file is converted and loaded into Pandas data frame.
* The new scraped data frame is also loaded into Pandas

Once both data sets are loaded, they can be queried individually or together. See sample provided in Pandas file (Web\_scraping.ipynb).