**ETL (Extract, Transform and Load) Project Report**

Extraction: Extract is the process of reading data from a database. Data sets of interest: Winter tire postings from Kijiji.ca for Toronto and Thunder Bay. Datasets we extracted shows actionable insights into customer interest, Url Link for the detail posting, Title, Price, Location and Date when posted. AND used OpenWeatherMap to get the dataset for weather condition of that city.

Source 1 – Winter Tire ad postings for Toronto area(Web Scraping): Link - <https://www.kijiji.ca/b-city-of-toronto/winter-tires/k0l1700273?dc=true>  Website has more than 10,000 postings and we scraped 30 pages. The dataset included columns are: New\_ID, Link, Title, Price, Location and Date.

Source 2 - Winter Tire ad postings for Thunder Bay area(Web Scraping): Link <https://www.kijiji.ca/b-thunder-bay/winter-tires/k0l1700126?ll=48.380895%2C-89.247682&address=Thunder+Bay%2C+ON&radius=20.0&dc=true>  Website has more than 600 postings and we scraped 10 pages. The dataset included columns are: New\_ID, Link, Title, Price, Location and Date.

Source 3 – OpenWeatherMap API for city of Toronto and Thunder Bay(API):URL

'https://api.openweathermap.org/data/2.5/weather?'

And units are "metric"

Which gives the Weather data for respective cities. And dataset includes location, temperature, humidity, cloudiness, wind\_speed, weathercondition .

Transform: The first step was scraping the data from Urls and stored them into dictionary. Then we converted the dictionary into Pandas data frame. Dataset was not clean so we clean the data by using str.replace() function to replace unwanted (\n, $,)

Symbol’s from Title and Price column. Removed the not date format records and null records. Converted the data type of Price and Date column from string to float and string to date respectively, then cleaned again to ensure data quality.

Load: The Toronto postings dataset were sent to postgresql. For this project, postgresql was chosen for its speed and simplicity when dealing with large datasets, and because there was little need to scale up space on this database.

The final Database, Tables/collections, why this was chosen.

Reproducible ETL steps:

1. Choose data to extract.
2. Read in the data and place it into a dataframe.
3. Group the Data Frame by most relevant data.
4. If there is no obvious choice for a data table merge. Create a normalized category (bins) for joins, merges and comparisons.
5. Merge the normalized category into the data tables.
6. Repeat steps 1-5 for every imported data set.
7. Clean the data: a. Remove duplicate data. b. Drop incomplete data records. (for this project any columns with less than 90% of values filled in. c. Standardize data types. (Language, date/time, encoding, etc.) d. Remove any files with a count beneath your threshold of interest to lessen data static.
8. Create new data frame(s) with the desired columns in the desired order.
9. Compare new data frame(s) to original to compare category lists.
10. Merge tables and set normalized categories as primary keys.
11. Remove any missing values (again), organize data into logical order.
12. Load the file into a relational postgresql database for future business or analytical use.