The Canadian Landscape

For Data Analytics Careers

2019 January 20

|  |
| --- |
|  |

Project 2 ETL

Final Report

The 6 in the Six

Jesse Gray

Sara Hovakeemian

Laurel Lobo

Jose Tomines

***“Choose a job you love, and you will never have to work a day in your life.”***

*- Confucius*

Title of Article from Harvard Business Review: “Data Scientist: The Sexiest Job of the 21st Century:. These are the types slogans and buzz that are bombarding social and business media. This project seeks to uncover the Canadian landscape for those who are seeking careers in this emerging high tech area. By analyzing data jobs being offered in websites like Indeed.com, and correlating them to the population and quality of life data where they are being offered, this project hopes to illuminate where the best opportunities and earning power can be found in Canada.

Table of Contents

[1 Introduction 3](#_Toc535710609)

[2 Methods 4](#_Toc535710610)

[2.1 Extraction & Transformation of Data Analytics JOBS 4](#_Toc535710611)

[2.2 Extraction & Transformation of Location Data 4](#_Toc535710612)

[2.2.1 Extraction & Transformation of Cost of Living Data from Numbeo 4](#_Toc535710613)

[2.2.2 Extraction & Transformation of Popuation Data from Stats Canada 4](#_Toc535710614)

[2.2.3 Merging of Cost of Living Population Data based on Canadian Location 4](#_Toc535710615)

[2.2.4 Aggregation of Cost of Living & Population Data Based on Provinces 4](#_Toc535710616)

[2.3 Loading of data to sql database 5](#_Toc535710617)

[2.3.1 Sub Folder a 5](#_Toc535710618)

[2.3.2 Sub Folder 5](#_Toc535710619)

[3 Findings 6](#_Toc535710620)

[3.1 Job DistriBution Across Canada 6](#_Toc535710621)

[3.2 Average Salaries Across Canada 6](#_Toc535710622)

[3.3 Locations with most opporutnities 6](#_Toc535710623)

[3.4 Locations with most earning power 6](#_Toc535710624)

[4 Conclusions 7](#_Toc535710625)

[5 Discussion 8](#_Toc535710626)

[5.1 Issues with Job data 8](#_Toc535710627)

[5.2 Incompleteness of Numbeo Data 8](#_Toc535710628)

[5.3 Issues regarding correlating data from different sources 8](#_Toc535710629)

[6 aPPENDIX A: Script EXECUTION PLAN 9](#_Toc535710630)

[7 aPPENDIX B: Repository Folder Structure 10](#_Toc535710631)

[8 aPPENDIX C: sql db mODEL 11](#_Toc535710632)

1. Introduction

Inspired by

1. Methods

Using the ETL processes, the following tasks were done:

1. Extraction of the Data Analytics positions and salaries from job search websites
2. Extraction of the Cost of Living information for Canadian cities from Numbeo website
3. Extraction of the population information for Canadian locations from Statistics Canada website
4. Transformation of the Data Analytics positions data focusing on locations and salary into a dataframe
5. Amalgamation and transformation of the Cost of Living and population data into a dataframe based on Canadian Locations and their provinces/territories.
6. Transformation Cost of Living and population dataframe to produce a dataframe of aggregated data for each province/territory.
7. The Load of all dataframes into a SQL database
8. The use of matplotlib for visualization of the relationship between Data Analytics open positions, their locations, and their related cost of living and population information, through database queries

The methods are described in greater detail below:

* 1. Extraction & Transformation of Data Analytics JOBS, AND SALARIES

The jobs listings were scraped from indeed.ca by searching for 'Data Analyst', 'Data Engineer', and 'Data Scientist' jobs in Canada. A multi-nested loop was created in Python that would extract city, company, job link, job title, location, salary, and company from the most recent 10 pages of listings for each or the three job titles. The script would detect whether the job title contained Analyst, Engineer, Scientist, or non of the above and assign the listing with one of four jobIDs. The script also split location into city and province and created new columns for each. Becuase the postings are dependent on users generating the content, the script had to handle for errors, such as filling location in as Canada rather than city and province, and placing job titles in multiple areas of the post. Variables were set and objects creted inside the loop and appended to an array of job postings outside the loop. Once the job list was created, it was able to be turned into a pandas dataframe for further munging and clean up.

The salaries for data analytics jobs were obtained by scraping the website indeed.ca/salaries. A loop was constructed to query the website by modifying the url itself, and if a results table was detected, the loop would then scrape all available data from the table using beautiful soup, and continue to the next page of results as long as a <span> tag with a class (“cmp-pagination-link”) existed. Once results were gathered for all three data science jobs in all provinces, the loop ended and the results were parsed into separate lists for province, job id, job title, salary and a salary classifier (per month, per year etc.). The lists were then used to create a temporary dataframe, which adjusted the salary column to ensure all rows yielded a ‘per/year’ value. The final dataframe then dropped the classifier column, and was exported for loading subsequent analyses.

* 1. Extraction & Transformation of Location Data

The final scenario used for the official test consisted of 10 vusers of script ALCR2 and 10 vusers of script ALCR3, ramping in at a rate of 1 vuser every 30 seconds, with a steady state of 30 minutes, and

* + 1. Extraction & Transformation of Cost of Living Data from Numbeo
    2. Extraction & Transformation of Popuation Data from Stats Canada
    3. Merging of Cost of Living Population Data based on Canadian Location
    4. Aggregation of Cost of Living & Population Data Based on Provinces
  1. FINAL Transformation AND Loading of data to sql database

Before loading data into SQL, the extracted inputs needed to be transformed relative to each other to ensure clean data is loaded. The file used to do the final transformation and load into SQL is ‘Transform\_IntoSQL.jpynb’.

The extracted inputs to this file are as follows:

* Indeed.ca scrape for salaries of Data Analyst, Data Scientist, Data Engineer by province in csv form (Salary\_extracted\_input.csv).
* Indded.ca scrape for job postings for Data Analyst, Data Engineer, Data Scientist positions in Canada is csv form (Job\_Posting\_extracted\_input.csv)
* Canadian city information from Stats Canada API and Numbeo API in csv form (complete\_city\_df.csv)
* Canadian province information from Stats Canada API and Numbeo API in csv form (province\_df.csv)

The purpose of this section of the project is to create 5 clean tables that are interlinked and loaded onto MySQL. The 5 tables are: Job\_Class, Jobs, Cities, Provinces, Salaries.

* + 1. Job\_Class Table

The job\_Class table was a manually made dataframe to assign a Class\_ID (eventually will act as primary key in SQL) to Data Analyst, Data Engineer, and Data Scientist. We included an Other job\_Class because the job posting scrap yielded job titles other than Data Analyst, Data Engineer, and Data Scientist, and we did not want to discard them as they had ‘Data’ in the job title.

* + 1. Provinces Table

The base of the provinces table was made from a pd.read\_html scrape of the Wikipedia page for Canadian provinces. It was merged with the province\_df.csv on the province abbreviation. Each province was given a province\_ID that will act as a primary key in SQL. ‘Other’ was also included in the provinces table because some of the cities yielded from the job postings scrape of indeed.ca could not be matched to city on the city table and therefore could not be matched to a province.

* + 1. Salaries Table

The salaries table needed to be merged with the job\_Class Table and Province Table. Prior to the merges, the provinces column of the raw salaries table (from the Salary\_extracted\_input.csv) needed to be match the provinces column of the provinces table in terms of string. The python library ‘fuzzy wuzzy’ (string matching library) was used to accomplish this.

* + 1. Cities Table

The raw city table (from complete\_city\_df.csv) needed to be merged with the provinces table. Prior to merging, an ‘Other’ city needed to be created with ‘Other’ as the province because the jobs table included cities that are not in the city table that will be categories under ‘Other’. A unique city\_ID is assigned to all the cities in the city table (to act as the primary key in SQL).

* + 1. Jobs Table

The raw data of the jobs table (Job\_Posting\_extracted\_input.csv) included columns city, company, job title, province, and summary. This table needed to be merged with the job\_class table, provinces table, and city table. Prior to the merges, several steps had to be performed to ensure clean merges:

* Python library unidecode was used on the city column to remove any accents from the French cities (as the city table did not include city names with accents).
* A list of cities that were in the jobs table but not in the city table were generated. The python library ‘fuzzy wuzzy’ with a score cutoff (80) was used to determine any semantic differences (e.g. Toronto vs. Greater Toronto Area, St. Catherines vs. St. Catharines, Saanich vs. North Saanich). Cities in the jobs table identified to have minor semantic differences to cities in the cities table were changed. All remaining cities in the jobs table that were not in the cities table were categoried under ‘Other’.
* The provinces in the jobs table were also corrected with fuzzy wuzzy as mentioned before.

* + 1. Load onto SQL

The final dataframes for the job\_Class, provinces, cities, salaries, and jobs were loaded onto SQL through python by creating an engine and using the pd.to\_sql method. Primary keys were set using the engine.execute method where the SQL query 'ALTER TABLE `table name` ADD PRIMARY KEY (`column name`);' was called.

* + 1. FINAL database

Figure 1 below shows the final database loaded into MySQL. The black lines denote the connections between the tables (i.e. similar to foreign keys).

|  |
| --- |
| \    Figure : Database Schema |

1. Findings

The following data relationships were explored:

1. The Distribution of Data Analytics Jobs Across Canada
2. The Comparison of Data Analytics Average Salaries Across Canada
3. The Top 5 Cities with the most Data Analytics Job Opportunites
4. The Top 5 Cities with the most Earning Power for Data Analytics Jobs

Visualization of these relationships will show the locations in Canada where people desiring a Data Analytics career will have the best opportunities to not only find a job, but to have a higher quality of life.

* 1. Job DistriBution Across Canada

Ontario, Quebec, and British Columbia (BC) appear to have the highest variety of data science jobs available in the country. In fact, three of these provinces account for ≈ 90% of the data science jobs in the country. However, when adjusted for population, an interesting pattern emerges. Ontario and BC have nearly the same amount of data science jobs per 100,000 people, while Quebec falls quite a bit shorter. This may suggest that competition in Quebec will be considerably higher than in Ontario and BC.

Interestingly, Alberta, PEI and Sasketchwan all have data science opportunities that are in line with Ontario and BC, however given their slightly higher cost of living index, the salary attained in these provinces should be considerably higher in order to convince an individual to forego pursuing opportunities in Ontario, BC, and Quebec, where the cost of living index is quite a bit lower, presumably because these provinces tend to pay higher salaries than the national average. However, it should be noted that PEI has one of the highest quality of life scores across the country, making it a very viable option

* 1. Average Salaries Across Canada
  2. Locations with most opporutnities
  3. Locations with most earning power

1. Conclusions
2. Discussion

The following are items and issues that were encountered during this project, and should be further discussed to provide insight on the observations and conclusions.

* 1. Issues with Job data
  2. Incompleteness of Numbeo Data
  3. Issues regarding correlating data from different sources

1. aPPENDIX A: Script EXECUTION PLAN

The following are the script designs of the three scripts used in the performance testing. This provides a clear understanding of the steps used when recording the script, the data selected during recording, the variable names used, data preparation instructions, and the LoadRunner transaction names and the functional steps they comprise.

The first script could not be recorded as a LoadRunner script, but the script design document provided clear instructions on how to manually execute the functionality.

1. aPPENDIX B: Repository Folder Structure
2. aPPENDIX C: sql db mODEL

The following is the SQL Data Model used in this project: