Aalto University
School of Science

CS-C3230 DATA SCIENCE PROJECT FINAL REPORT - REAKTOR GROUP

Kodimpi - Relocation service using Statistics Finland Data

Name: Tran Anh Thong

BSc programme: Data Science

Email: thong.tran@aalto.fi

University Instructor: Jorma Laaksonen

Organization Instructor: Jaakko Särelä

Date of submission: 19.12.2019

Table of Content

1. Course background

Courses completed before the project:

MS-A0111	Differential and integral calculus 1
MS-A0402	Foundations of discrete mathematics
MS-A0503	First course in probability and statistics
MS-A0011	Matrix Algebra
MS-C1343	Linear algebra
MS-C2111	Stochastic Processes
MS-C1620	Statistical inference
MS-C2105	Introduction to Optimization
CS-A1110	Programming 1
CS-A1120	Programming 2
CS-C2120	Programming Studio 2: Project
CS-C2150	Theoretical Computer Science
CS-E4620	Introduction to Analytics and Data Science
CS-C3160	Data Science
CS-C1000	Introduction to Artificial Intelligence
CS-EV	Machine Learning with Python

Courses taken concurrently with the project:

CS-E3210	Machine Learning: Basic Principles
CS-A1140	Data Structures and Algorithms
CS-C3190	Principles of Algorithmic Techniques
MS-C2128	Prediction and Time Series Analysis

2. Project motivation

The original guideline for the project was to explore and examine interesting aspect of Staticstics Finland open database by postal code area (Paavo). As the project evolved, it was decided that the data collected would be used to build Kodimpi, a service for relocation suggestion within Finland. Information about population structure, income, housing, education, environment, and public transportation was collected both from Paavo and other sources was utilized to build the service.

3. Data sources

3.1. Paavo

The Paavo data played a central role in the project. The database provided information about Finland demographics for each postal code area from 2012 to 2017. For each postal code area, the data includes population structure, educational structure, individual and household income, size and stage in life of households, buildings and dwellings, workplace structure, main type of activity, and map data.

The data was available on stat.fi and can be freely downloaded manually or through the PX-Web API with various formats including csv. There were more than 3000 postal codes and 100 variables for each year of complete data on Paavo. The amount of data for each year could be estimated to be less than 5 MB, and the map data was around 100 MB.

There were some inconsistencies that needs considering when processing the data. Firstly, the data published in a certain year can be information from 2 or 3 years in the past, depending on the sub-table. Secondly, there were changes in the list of postal codes from year to year as some areas were merged or split.

3.2. Other data sources

Aside from Paavo, some other data sources were explored and integrated into the service. Those data sources provided information about housing, transportation, environment and housing.

Väylä and Traficom provided data on public transportation including trains, buses, trams, and ferries. The original data was the coordinates of the stops and stations, which were converted to postal code area using map data.

Copernicus provided raster data of tree cover density as well as water and wetness.

Vero provided information on tax rate of municipals.

ARA provided informations on house selling prices and rent for the latest 12 months.

Lastly, stat.fi housing data provided quarterly housing data for old dwellings.

4. Methods

4.1 Overview

For data processing and machine learning methods, the main language of choice for the project was Python. The specific libraries that were used in the project can be found in the requirements.txt file on the project's Github repository. The raster data for tree and water coverage was processed with QGIS and GRASS GIS.

4.2 Data fetching

Data from Paavo and more generally, from stat.fi, was collected through the PX-Web API by using HTTP POST method with JSON queries. By querying the database, precise data can be downloaded as csv files or imported directly into Pandas DataFrames. Similarly, Traficom API also allows data fetching by querying. For ARA, the data was fetched by web scrapping using the Beautiful Soup library. Other data was downloaded manually and integrated in the final DataFrame.

4.3 Data cleaning and imputing

Some of the data from Paavo was hidden due to privacy reasons and replaced with either '.' or '..' in the data table. Depending on the type of data that was missing, a value would be replaced with either the median of the attribute or scale with the minimum which was stated in the data documentation. After scaling, the data was checked again, and erroneous values would then be replaced with zeroes.

For the case of missing surface area, the data was calculated using population density values predicted using linear regression on the available data and the population data, which was available for every year from 2012 to 2017 in Paavo.

Another more specific case would be the housing data from Paavo. In this case, data of a postal code was inferred using data from "neighbors" (areas with the postal code with the first same digits) and population density.

4.4. Data processing

Coordinates data for transportation was converted by using the map data. For a specific coordinate, each postal code area areas were sorted in term of distance from the coordinate and each was checked to determine if the coordinate was within its polygon until there was a positive.

The original forest and water data was in the form of raster images. The images were then cropped and projected onto the map data. Each 20-by-20 meter square was projected on the map data to determine its postal code area. From this, the green and water average coverage can be calculated for each area.

After cleaning, the data from Paavo was scaled according to the type of attributes. Some attributes were scaled to population or surface area, and some were normalized by calculating percentages with respect to another attribute. Scaling would make comparing attributes between different area more sensible and help with clustering and measuring distances between areas. Services in each postal code was combined with those of each neighbor area to better reflect the accessibility of the service in each area.

4.5. Data analysis, prediction, and suggestion

Since the time span of the data was low with only 5 data table with every attribute, the data analysis focused mostly on the latest data table (from year 2017). Only the housing data from Paavo had enough data points (52 quarters) for a acceptable prediction.

After cleaning and integrating all the data, there were more than 170 attributes in the data. Therefore, the first step in data analysis was to apply PCA to reduce the number of attributes in the data. The dimension of the data was reduced by half to around 80 principal components while retaining 99.5% of the original variance.

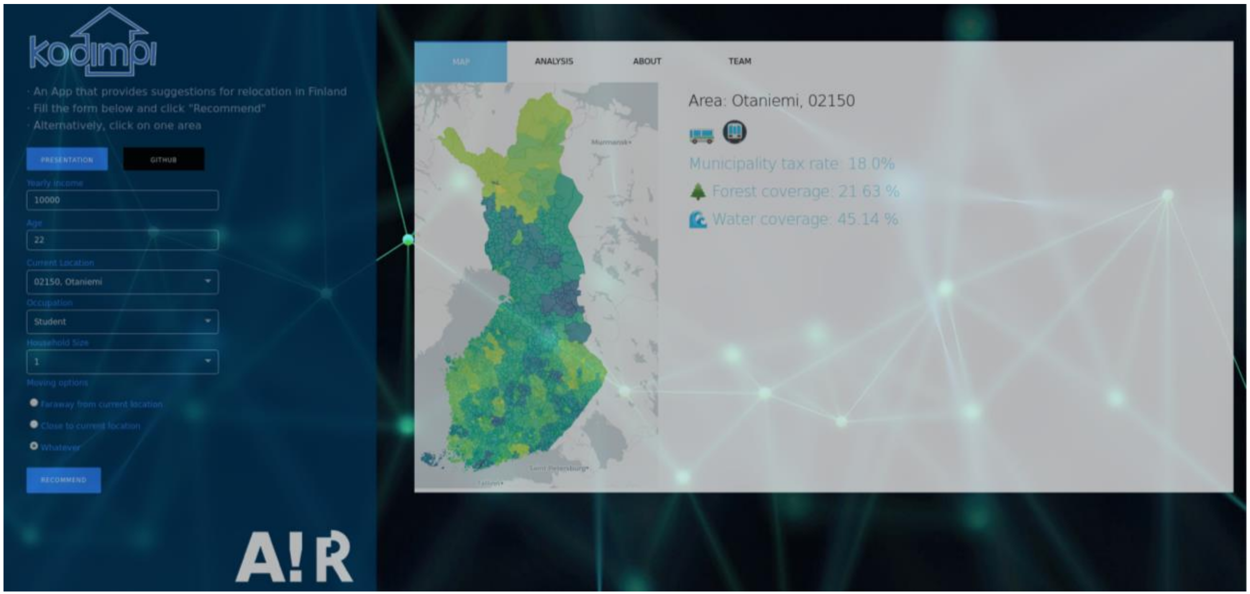
The next step was to cluster similar postal code areas. K-mean clustering was chosen after testing with several different clustering methods. The optimal number of clusters was determined by plotting inertia against the number of cluster as well as calculating Davies-Bouldin and Silhouette score. For this specific data set, the optimal number was found to be 60.

The suggestion system was built upon the result of the PCA and clustering. The system would take age, income, current location, household size and preferred distance from current location as input. The input would then be used to modify the feature space of the current location to make a new data point. The area with the lowest weighted Euclidean distance to the synthetic data point would be the suggested location.

For prediction with housing data, ARIMA model was considered. Preliminary tests were carried out using the average housing price data of all the areas. As the plot of the time series had a linear trend, the difference of it was taken and the augmented Dickey-Fuller test was applied to determine the stationarity of the differenced series. Then, the autocorrelation function (ACF) and partial autocorrelation function (PACF) was plotted to initially determine the range of the order (p, d, q) for the ARIMA model. The final order, (0, 1, 1), was determined by running a grid search with p and q parameters from 0 to 5 and d from 0 to 2 with the determining factor was the predicted root mean square.

5. Results

The Kodimpi web service utilized the results from the data clustering to make relocation suggestions from user input. The service also compared and visualize main factors (services, education, population, income, transportation) between the suggested area and the current area. Basic information such as housing prices and trends, tax rate, and environment was also displayed for user reference.



- An App that provides suggestions for relocation in Finland
- Fill the form below and click "Recommend"
- Alternatively, click on one area

Yearly income: 10000

Age: 22

Current Location: 02150, Otaniemi

Occupation: Student

Household Size: 1

Moving options

○ Faraway from current location

○ Close to current location

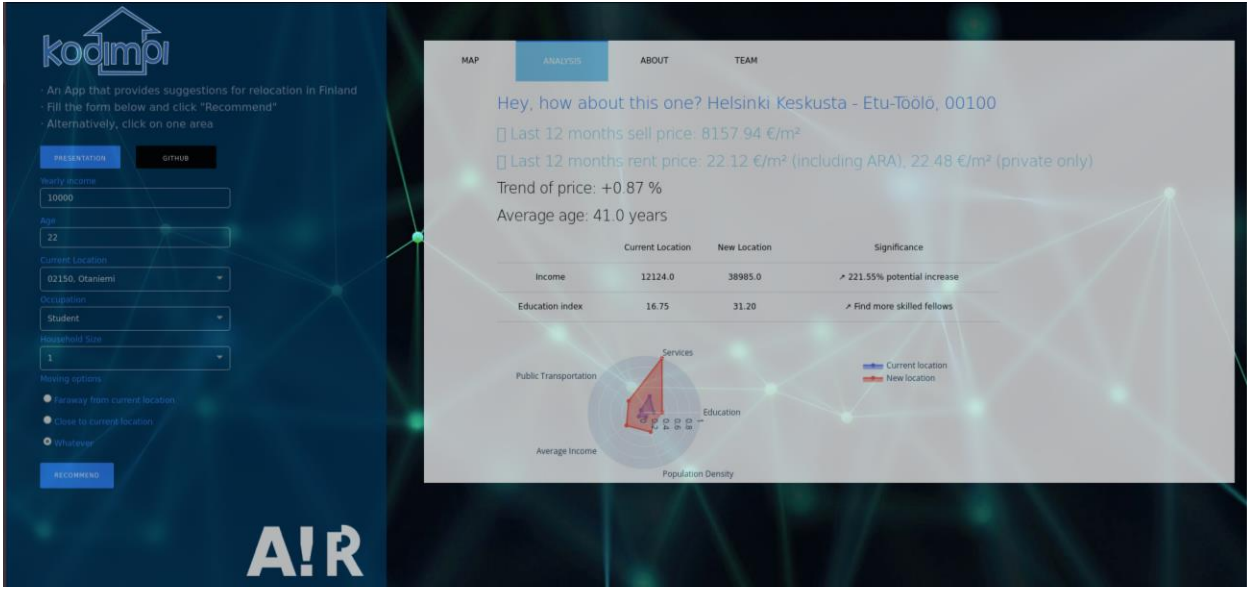
● Whatever

Area: Otaniemi, 02150

Municipality tax rate: 18.0%

Forest coverage: 21.63 %

Water coverage: 45.14 %



- An App that provides suggestions for relocation in Finland
- Fill the form below and click "Recommend"
- Alternatively, click on one area

Yearly income: 10000

Age: 22

Current Location: 02150, Otaniemi

Occupation: Student

Household Size: 1

Moving options

○ Faraway from current location

○ Close to current location

● Whatever

Hey, how about this one? Helsinki Keskusta - Etu-Töölö, 00100

Last 12 months sell price: 8157.94 €/m²

Last 12 months rent price: 22.12 €/m² (including ARA), 22.48 €/m² (private only)

Trend of price: +0.87 %

Average age: 41.0 years

	Current Location	New Location	Significance
Income	12124.0	38985.0	↗ 221.55% potential increase
Education index	16.75	31.20	↗ Find more skilled fellows

The final user interface of Kodimpi

6. Conclusion

In this project, data from the Paavo database was integrated with other sources to form a feature space that provide relevant insights which were utilized for informing and suggesting ideal relocation area within Finland.

The project employed machine learning methods, statistical tools and model such as PCA, K-mean, ARIMA, linear regression to transform and make use of the available data. The Kodimpi web service was deployed with Heroku using Dash and visualization from Plotly as well as a custom-made map using OpenGL.

7. Future prospect

The Paavo database can be applied for several different applications, especially in marketing and business planning, and the dataset can be further expanded with other open data sources. Improvements can be made to make the data imputation and analysis more robust by tuning parameters, validation, testing, and exploring more methods. More features such as the custom-built map and housing data that can be further integrated into the service. New features such as collecting feedback from users or fully automating in updating the data set is also a feature that can be contemplated.

8. Course Feedback

8.1. The most important learnings of the course?

For me the most important outcome of the course is learning the process of a data science project, from exploring and processing the data to applying machine learning method and visualization. The next most important would be knowing the tools and when to use them. Last but not least is teamwork and project management.

8.2. What was easy and difficult for me on the course?

I am not sure if there is anything that can be counted as easy in the course. Almost every step of the project, there were new things that we needed to address, and in most occasion, they can be challenging. At the beginning, it was figuring out the API to fetch the data from Paavo. Then it was the imputation of the missing value. I still strongly feel that the process should be more robust, but with the lack of time, we settled for an acceptable result to move on with other parts of the project. Generally, choosing to keep working and expanding on something like trying out different methods or to move on with other things can be difficult. Another thing that can be difficult is keeping up with others' progress. But at the end of the day, I think we each managed to have just enough information from each other the keep things running.

8.3. What necessary knowledge I was missing when the course started?

The most important missing mechanical knowledge for me would be Python and its libraries. Though I did some basic programming with Python, there were a lot of smaller things that need figuring out along the way. In term of more abstract knowledge, I would say that I did not know what the steps for a data science project would generally be. I suppose it is also a part of the course but having some idea beforehand would be much better.

8.4. What I learned about project and group work and practices?

I think that I did learn a lot about those things. Through the project, I think that I appreciate the role of communication a lot more. Information sharing is a key part in pushing the progress along. Having said that, I think that the communication can be improved.

About project practices, I think that knowing and practicing some of them can be helpful. As a data science project is quite complicated and demanding, having guidelines to follow can increase efficiency a lot.

8.5. What was good and bad in our group's topic?

The good thing about our project topic is that it is quite open, and the data can be easily understood as most of it is numerical data, and the information is nothing too complex that it would require domain expertise. The openness of the topic was also a bit of a problem because we didn't really know that what kind of direction should we choose to use the data. I guess in the end, having the freedom to do what see fit is more gain than lost for us.

8.6. What was good and bad in our group's instruction?

I think the initial instruction for our project is quite vague, and maybe that is the way that it was intended to be. What was good about it was that it mentioned the example project from Reaktor. Their web application is more or less a reference as we try to make our own. Now that I think of it, maybe in the instruction, there should be a line that says that we may choose to use all the information on Paavo or just part of it and some other sources. If there were that line, maybe we would have a very different idea for the project.

8.7. How was our group's final outcome compared to my expectations?

I think what we have done exceeded my initial expectation for the project. I would not believe in the beginning that it would be developed into a web app. But I guess our expectation for the projects grows as we make more progress. There were features that we laid the groundwork for, such as integrating our own map and plots, but was not implemented because we ran out of time at the end. Despite that, I think we did a lot, and the outcome was much better than anticipated.

8.8. How was the other group's topic, instruction and outcome compared to ours?

I personally think that the other topic is more interesting. It was more specific too, as it was clear that the data is finance related. However, as it has to do with natural language processing, it was more intimidating. And as they showed, extracting the information from the report was very challenging because of the way they were written. The instruction was a bit clearer, there were a good suggestion as to what to do with the data and how to expand to other data as well.

8.9. The most and least important ones of the three reading tasks?

I think the least important reading task is the project management reading task. The topic is important, but I guess that reading about it did not help as much as doing a project would. The other two importance, in my opinion, depends on the stage of the project. When I began to start working on the first few pieces of code for the project, knowledge of Python, numpy and Pandas is more important. As the projects is more about finding and testing model, learning about scikit-learn is more relevant. At the end of project, much of what I used was actually statsmodel, so I think the students should know to search and apply whatever tools that they need.

8.10. Should the reading tasks have been graded?

I think that the reading tasks should be more free form, maybe write a few hundred words about what we think was useful from the resources that was given.

8.11. The most and least important ones of the four lectures L2-L5?

I think the one I value the most was lecture 5 on Data science toolbox. It really helped me to know what tools was out there for the whole process of a project. The least important may be the project management or the real-world cases. It is very hard to decide. The real-world cases lecture was very interesting and informative, but I guess not quite vital for the project.

8.12. How I would change the timing of lectures, check-ups and site visits?

The most straightforward thing I would change is swapping the real cases lecture to the last. The others are trickier to decide as there are contents that can be useful at the start of the project. I guess introduction to the libraries is a must at the beginning, but then again, I think we must learn them by practicing anyways. The content of the Data science toolbox covered a pretty broad range of topics. I think that it can be pushed to be the first lecture after the introduction.

I think one more check-up would be appreciated. Site visits are great if there is enough time for more of them.

8.13. How I would change the course when it changes from 10 to 5 credits?

To go from 10 to 5 credits, I think the scope of the projects should be reduced. There may be more detailed instructions and requirements and less freedom to explore. I think there should be a guideline to limit what has to be done. Our group spend a lot of time for the app, and it was kind of acceptable for a 10-credit course in my opinion but would be too much for 5 credits. Therefore, I would say that it should be explicitly said that students should only need to make a local app if they want to, and that they should not spend time on making a web app or learning html for only that purpose. Maybe point out that demonstrating the project with Jupyter notebook with sufficient visualization is good enough. Maybe there should be more support and guidelines for data cleaning, but his is quite hard to do as it is also important to learn data cleaning. Having a teaching assistant to keep track of the progress of the groups and give more frequent feedback can help with the progress and ease the workload, but I am not sure how it would work out.

8.14. My overall opinion about the course in general and its implementation this year?

I think the course was great. Without the workload of the other course, I think I would prefer the 10-credit version of the course as there likely more freedom. With the course, I was able to experience a full process of a data science project and learned a lot along the way. I would prefer the first few weeks to be more informative, especially in term of the expectations of the outcome of the course. Given it is the first implementation, it is quite ok. Having feedback and receiving questions from people from Reaktor and Futurice is quite nice, so I hope that the future implementation will somehow preserve that.

8.15. Other comments

I can't really think of much else right now. Maybe I will have more to say in the meeting next year.