CITY: Characterize the Israeli Town for You

A tool for comparison and recommendation of Israeli cities based on favorable properties

Reut Barack, **Email**: reut.barak2@mail.huji.ac.il , **CS id**: reut.barak2
Liora Itenberg, **Email**: liora.itenberg@mail.huji.ac.il, **CS id**: liorait
Gil Ben Cohen, **Email**: gil.bencohen@mail.huji.ac.il, **CS id**: gilbc

Problem description:

Many people in Israel today face making the very important choice of where to live their lives. Young families in particular regard this as important, as they need to choose a location for buying an apartment, or for raising their children. As of today, no tool exists to help Israeli families and individuals making these decisions. Although the Central Bureau of Statistics (CBS) have published several data attributes on Israeli settlements, the data is insufficient and not well organized to allow a smart and fast decision process. In addition, certain features regarding Israeli settlements that may be very important for certain people are missing. Proximity to nature, integrated education evaluations, pollution data and job demands are some examples of important features that would require creative solutions to assemble. Even after the data is collected, it is not trivial how a city, or a range of cities, should be selected. There is therefore a need not only in recommending a place to live, but also in presenting several cities that show similarity based on selected preferences.

Data:

Education features, district, population size, economic & social features as well as occupations per district were extracted and integrated from multiple **CBS supplementary tables** (https://www.cbs.gov.il/he/Pages/default.aspx). City code or city name were used to integrate tables which led to 204 settlements with a relatively small number of missing values.

Crime rates were extracted from https://www.meida.org.il/ as an absolute number, summarizing all open case records in each settlement. Absolute numbers were modified to crime percentages by dividing in the population size and multiplying by 100.

Nature proximity is not an available trait for Israeli cities online. Therefore, an integration was performed of 10-figures coordinate system and of ITM (Israel Transverse Mercator) coordinates for every city and for every nature reserve in Israel, taken from: https://www.gov.il/he/departments/general/nature_reserves_2. Coordinates of center points were calculated, and the minimal distance from every city to any nature reserve was computed. Since

some cities lay “far” from the reserve’s center point simply because the reserve is very large, a modification was done to account for the size of the reserve. Code for the preprocessing of crime and nature features is available at crimes_nature_proximity.py.

Occupation demands were extracted from https://www.jobmaster.co.il/ using **crawling techniques.** The number of overall demanded workers in each city was extracted by searching the city name in jobmaster’s engine with selenium (selenium package in python. Code is available in jobmaster_extract_num _jobs.py). The second feature was a division of workers to 15 job titles, and the percentage of workers in each job title, for 16 Israeli districts.

Pollution levels were extracted from **https://www.svivaaqm.net/**. Pollution values as measured in real time are given for different streets or areas nearby cities. Pollution levels of NOX, NO2, NO, PM2.5 were taken into account in the final score. Values were than aggregated for each city based on areas in or nearby that city. Finally, the values were also normalized to allow aggregation across different pollutants, and the values for the 4 mentioned gases were summarized into one feature.

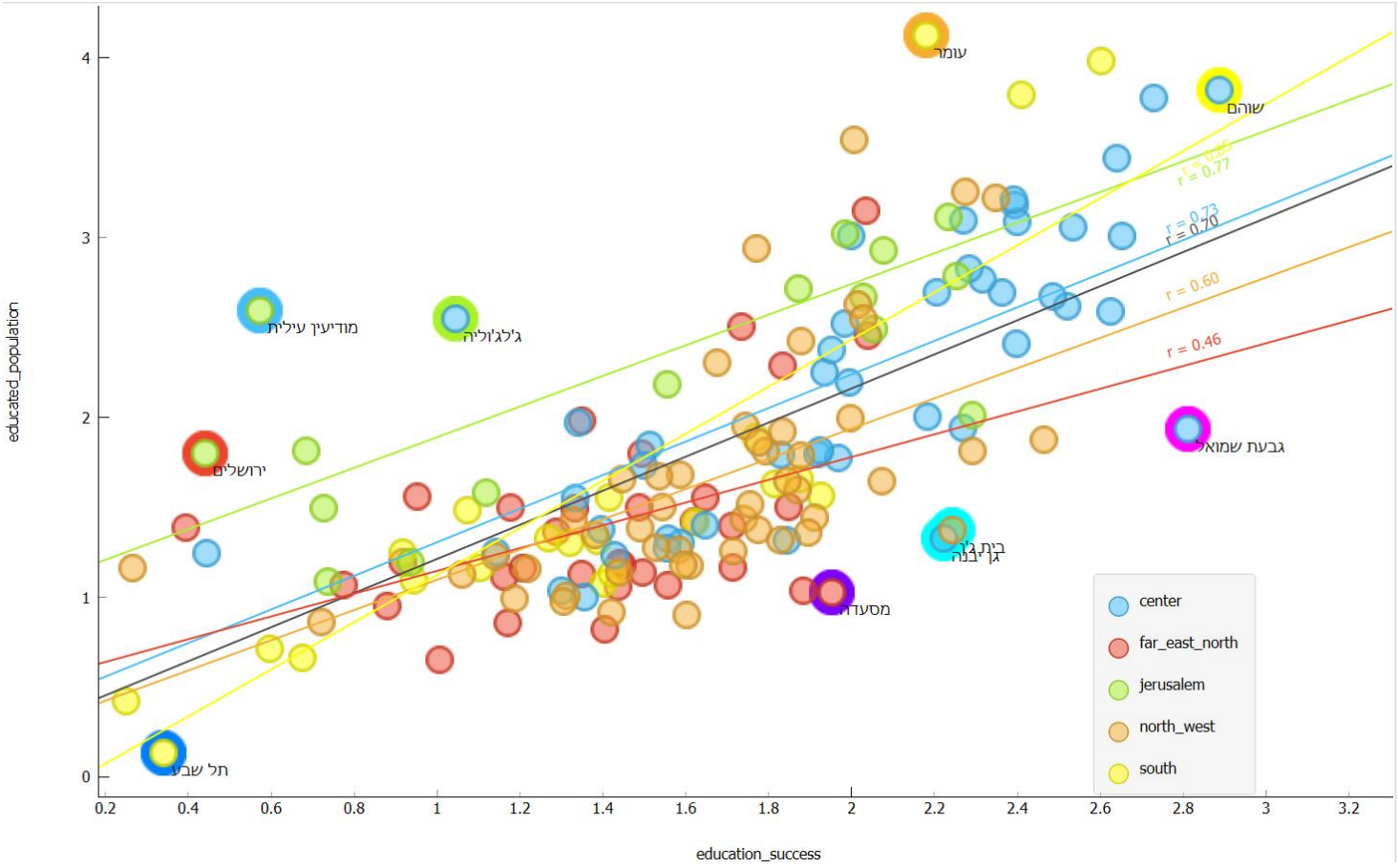
Thirteen **Education features** were collected from CBS and as expected, many of them were highly correlated. To improve user interface and to better calculate recommendations based on education, we combined the 13 features into 3 categories: **“Education quality”,** as measured by class size, number of teachers with master’s degree etc. **“Education success”**, as measured by number of graduates, academy acceptance rate etc. and **“Educated population”**, as measured by number of academy graduates in ages 35-55, number of university students in the population etc. Highest correlations between education features are presented below with spearman correlation coefficient values. Indeed, highly correlated features are accumulated to the same feature by our methodology:

-0.954	avg_students_per_teacher	avg_teacher_hours_per_student
+0.932	academy_acceptance_elig_percent	bagrut_elig_percent
+0.907	academy_graduates_norm	edu_years_avg_norm
+0.900	academy_manager_workers_norm	edu_years_avg_norm

The new features were less correlated although as may be expected, ‘education success’ and ‘educated population’ still had a strong correlation:

+0.745	educated_population	education_success
-0.124	educated_population	education_quality_measures
+0.007	education_quality_measures	education_success

We examined educated_population and education_success looking for outliers. Overall, the strong correlation wasn't disturbed much. "מודיעין עלית" and "ג'לג'וליה" had relatively educated population with relatively low education success, while "גבעת שמואל", "בית ג'ן", "גן יבנה" and "מסעדה" showed higher education success for a relatively less educated population. Separating to districts show that education differences are somewhat bigger in the sout. No settlement reached the extremes of the scatter plot (high left or low right corners). Some interesting settlements are boldened:



To correctly incorporate the data, settlement names needed to be unanimous for comparison, which was not the case for all data sources. To achieve uniformity, we created a dummy variable of the settlement's name, without special characters (' ', '-', ' " ' etc.) and without

the letters “י” and “ו”, which may vary in writing the name (e.g. “קרית טבעון” and "קריית טבעון"). The idea was to do something similar to PorterStemmer and it worked very well in creating maximal uniformity. After ensuring no duplicate city names occur through this method, it led to an improved intersection across all datasets and a significant reduction in missing values. The final dataset consists of 171 settlements with 50 features. There remain 201 missing values, most of which (130) are in the pollution feature, which was not available for a majority of settlements. mean value is used for imputation. Code for preprocessing and final dataset creation is in Unite_data.py

Solution (Demo of the final version is available at: https://youtu.be/Ntk5QRDdNpw)

Our solution is CITY, an Israeli settlements comparison and recommendation tool. CITY is based on properties selected by the user to fit the location best adequate for their needs and preferences. CITY is based on a comprehensive collection of data on the different cities, villages kibbutz etc. in Israel. Lastly, an algorithm-based-interface was developed to recommend places best suit for living according to preferences selected by the user. Cluster analysis is given for the cities along with the recommendation, suggesting settlements with similar properties to the one that fit user preferences.

We focused our work in 4 main efforts: (i) Robust and intelligent data collection and preprocessing. (ii) Friendly user interface with good visuals. (iii) Logical algorithm to integrate user preferences. (iv) Deliver relevant information regarding the selected settlement to the user. In addition, we made several analyses and visualizations in the process.

Efforts regarding data collection were described in the previous section. Regarding the interface, we used HTML and JavaScript to create a web interface and invested a significant amount of time to make it visually friendly. To allow integration of the python algorithm run on the server with the web interface, we used the Flask package. Code for flask integration is in main.py. HTML and JS code is given in Cities.html

The algorithm was designed based on what we believe is most logical when approaching the decision of where to live. Therefore, we first allowed the user to select a religion and a district of will, so that people who wish to live nearby Jerusalem will not get recommended Metula for example. Each parameter of the 9 available for choice is normalized and they are summarized together to form a recommendation. Three parameters are chosen to have stronger effect on the recommendation. To make sure higher ranking makes significant difference in recommendation, we used the rank value (1-5) squared times the normalized value of the selected feature. The algorithm is given in CITY_algorithm.py

Finally, regarding data given to the user, we used PCA for dimensionality reduction of all relevant features across cities. A 2D plot of all settlements is given to the user after recommendation. Settlements are nearby in the plot not based on geographical proximity, but rather based on feature resemblance. The user can therefore find other cities, possibly in different districts, that have similar features to the one recommended for him. Dimensionality reduction and paragraph computations are given in PCA_dimensionality_reduction.py and in open_paragraph.py

Demo of the final version is available at: https://youtu.be/Ntk5QRDdNpw.

Since we have collected a diverse and interesting data, we wanted to also examine the correlations between the different features. We used the spearman correlation coefficient. The 10 strongest correlations (both positive and negative) are presented below:

+0.887	Internet_norm	Salary_norm
-0.611	job_demands	nature_rate
-0.607	Internet_norm	religion_num
-0.536	Salary_norm	religion_num
-0.459	dist_num	nature_rate
+0.451	job_demands	log_population
-0.394	dist_num	religion_num
-0.342	log_population	nature_rate
+0.321	Salary_norm	crime_score_percent

Most correlations were not surprising and could be logically explained. We use those as a validation of our data collection and integration techniques: Internet availability and average income salary. The ricer the population, the more likely they are to have internet connections; job demand and nature proximity, which likely indicate periphery or closer to center settlements; Internet and religion, likely due to less internet connection in non-Jewish villages; population and job demand, since larger population likely means more jobs; population and nature in negative correlation, as large cities tend to have less nature proximity.

Other correlations were slightly more surprising, like average salary and crime percentages, which may be explained by the fact the wealth attracts theft, but could also hint that one of these features is not accurate enough.

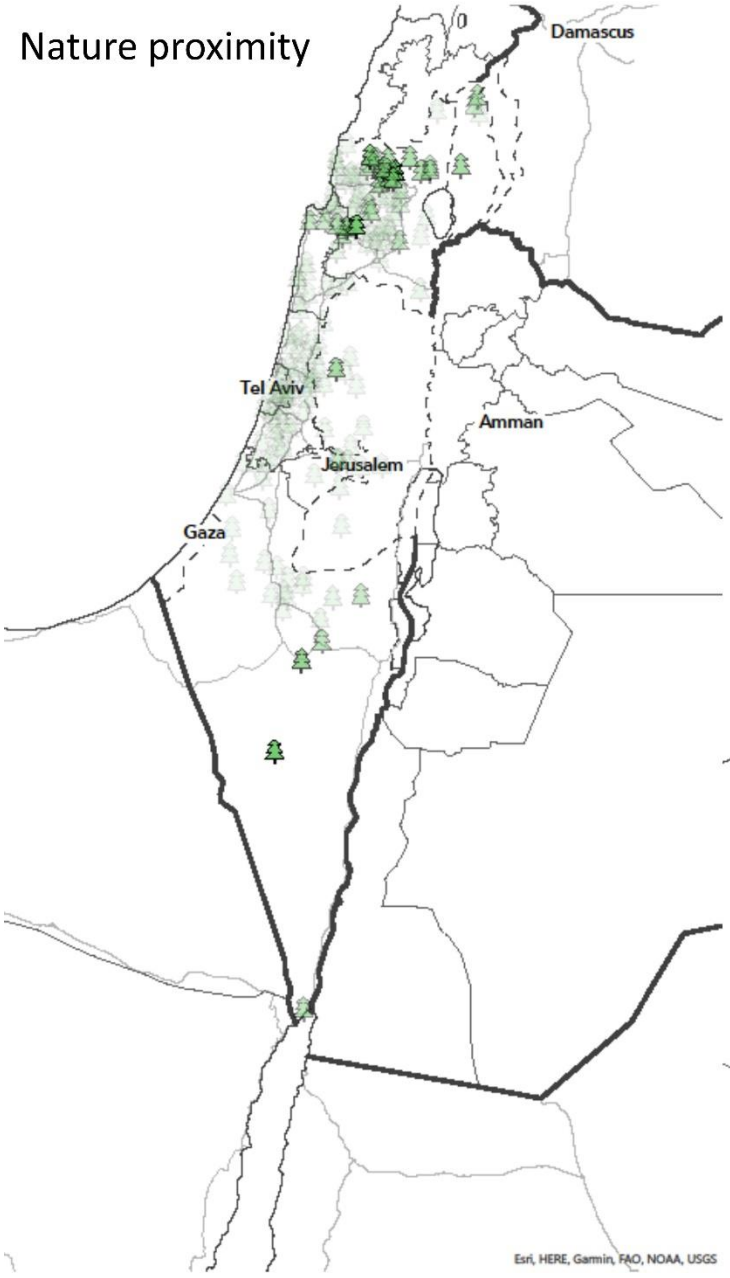
Evaluation

We think our evaluation criteria should be based on the 4 mentioned efforts. Setup and results are described in each section:

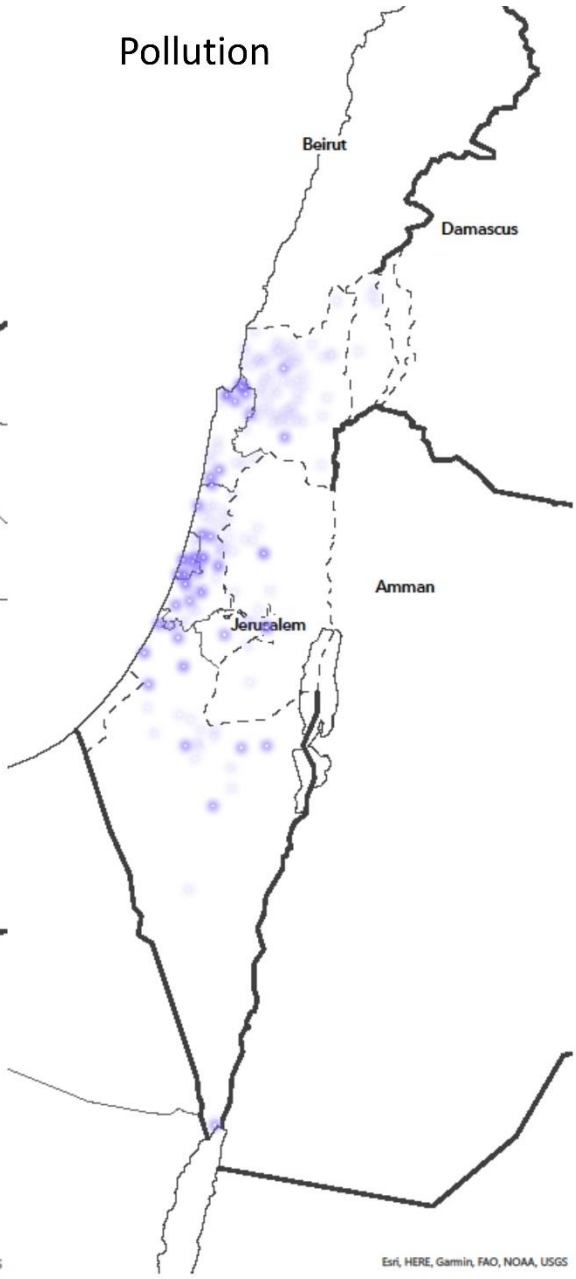
1. **the diversity and sophistication of data accumulation and preprocessing**. Using CBS tables would be the most trivial solution, and every interesting feature we were able to extract or manipulate adds significant power to our tool. We believe this part is in fact very important and we have invested a large portion of the time invested in this project on retrieving the final presented features.
2. **User interface friendly and visually attractive.** Using the command line to run a python script would be the trivial solution in this case. We wanted a web-based interface that would also interact with python. Since none of us had background with web interfaces, this was challenging, and we looked for multiple solutions. Eventually, we have experimented with HTML, JavaScript, Flask and python to form a single environment that produced the desired result.
3. **Algorithm**. The level of freedom given to the user makes creating both the interface and the algorithm more challenging and should be taken as a parameter for evaluation. We really tried to create a tool that fits what we would want to use. The algorithm gives robust recommendations based on even small variations in preferences (as we also tried to capture in the demo).
4. **Information delivered to the user**. The most trivial here would be to give out a name of a settlement and that's it. We wanted to give the users the ability to assess settlements with similar features to the one recommended for them. For that, we created the 2D dimensionality reduction visualization, and included it in the interface. We also wanted to give a brief description of each settlement and used our data to produce that in the interface as well.

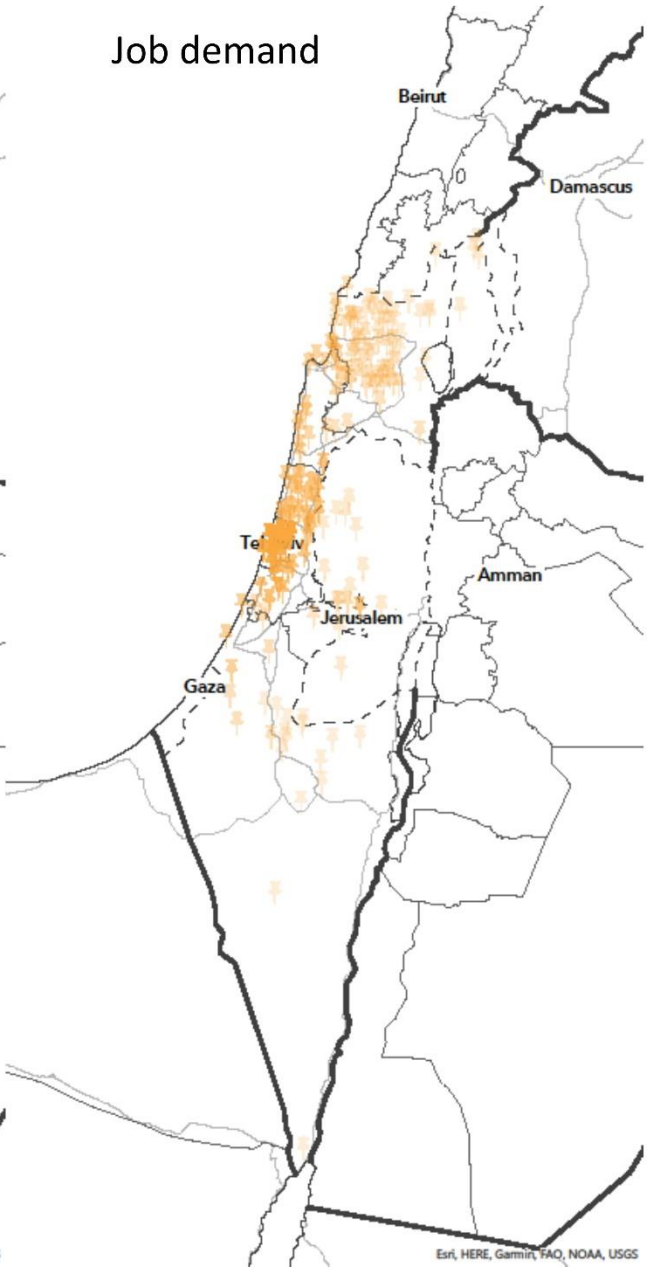
Visualizations. Some of the visualizations were already presented in previous sections. We wanted to see the effect of geographical areas on the different features. Therefore, we created maps of Israel showing the values of the different features, as presented below:

Nature proximity



Pollution





Impediments. Despite significant efforts done to produce the final result, we think a lot can be improved in this project, including evaluations metrics. We think evaluations can be more measurable. For example, we could examine the amount of diversity the algorithm produces for different preferences and use that as a measure. However, since the project took us to experiment with many different tools and we think this is a measurable metric of the effort invested in the final result (visualization techniques in HTML and JS, Flask to integrate python, PCA and dimensionality reduction, crawling techniques and a lot of feature extraction and engineering and more).

Future work.

This project can be extended to many fun and interesting directions. Our suggestions are: Use crawling to google search each settlement. Use the texts in first choice sites to create a word cloud representing each city.

We also thought to possibly create a graph structure to represent the different settlements. Categorical divisions for the features can be used to create edges based on settlement proximity for the different features. Then we could try to identify communities in the graph instead of PCA, to identify similar settlements. User preferences can also be integrated in the PCA analysis, or to strengthen edges relevant to the features he selected.

We could further examine correlations between features and look for outliers in the data. We could raise a recommender system that keeps information of past users and integrates it into the algorithm for better recommendations, or to suggest new users with preferences that may suit them. If we had more time, we would want to collect more features, for example: Temperature measures, taxes, rent levels, health systems, proximity to the sea, pub and restaurant availability, cultural activities, university availability, public transportations, corruption measures and more. We could also try to extend the available data for more settlements in Israel.

Brief conclusion

We have created CITY, an algorithm-based interface recommending on a place to live in Israel based on user preferences. Imagining and then executing the product to best suit the user we would want to be was fun and challenging. We implemented multiple tools and had to invest a lot of thought and creativity in retrieving some of the features we wanted, and then in creating the right format for preferences, in creating the algorithm, and eventually in building up the interface. This project can be useful to people facing this important decision in Israel, and can also be further upgraded in many fun and productive directions.