ML Project

ATM Machines

Introduction

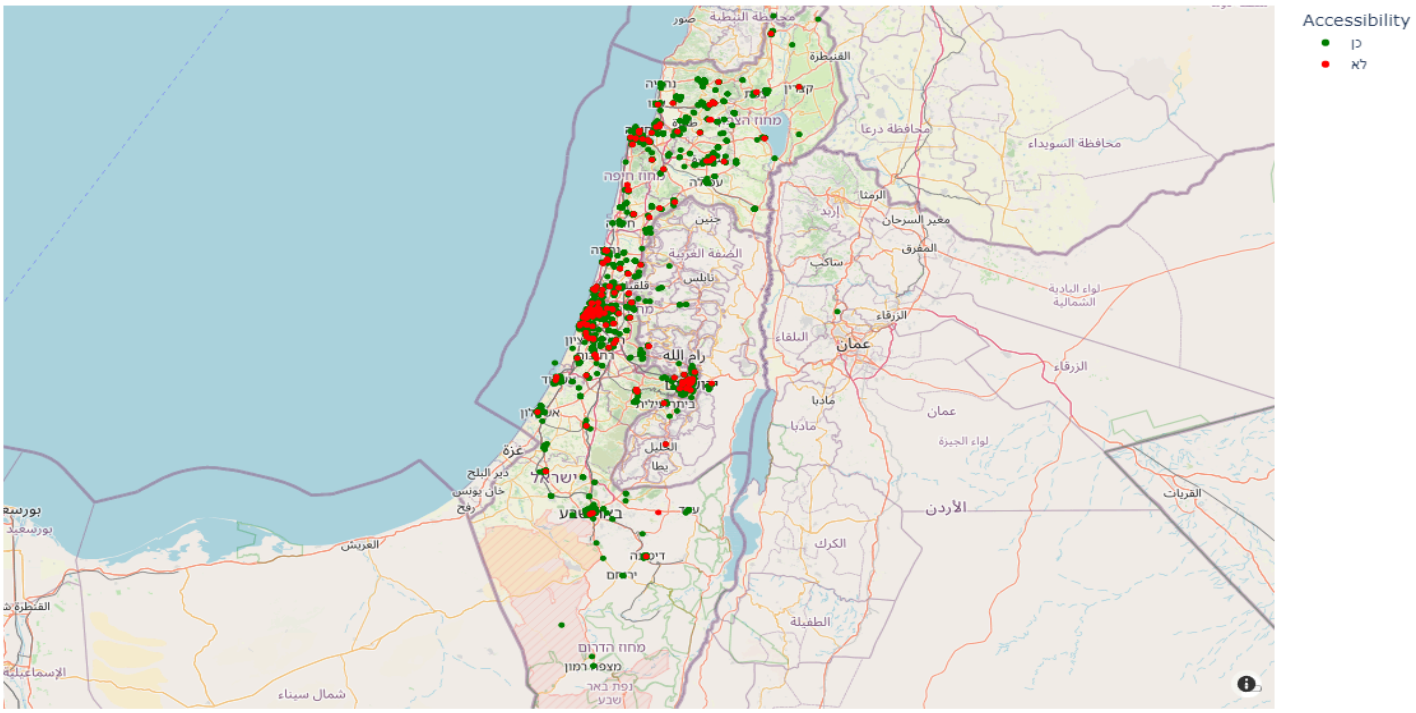
Access to basic financial services is super important, but not everyone has the same level of access to them. Even though there are more ATMs showing up all over the country, a lot of them aren't built to help people with mobility challenges. Our project is all about tackling this problem how many ATMs aren't handicap accessible, especially in big cities where more people rely on them and the need for accessibility is even higher.

By focusing on this problem, we aim to shed light on the areas where accessibility is lacking and provide insights that can improve handicap accessibility

Our project has four main objectives:

1. **Predict underserved areas**: Using Classification algorithms we are trying to predict which areas/cities need to improve their handicap accessibility.
2. **Visualize accessibility distribution**: We will map the distribution of accessible and non accessible ATMs across the country to better understand the current state of accessibility.
3. **Cluster underserved regions**: By employing clustering techniques, we will identify areas that are underserved and need more attention.
4. **Provide actionable recommendations**: Based on our findings, we will suggest locations where banks should consider adding more or improve their handicap accessible ATMs to ensure better access for everyone.

ATM Accessibility Map



Dataset and Features

For this project, we utilized the ATM dataset obtained from the government portal. This dataset includes the following key features:

- **Location**: Geographic coordinates of each ATM ,city and Address
- **Bank Information**: Bank ID, bank name, and branch details.
- **ATM Type**: Classification of the ATM (Cash only, ATM Information Device)
- **Accessibility**: Indicates whether an ATM is equipped with handicap access or not.

Data Collection and Preprocessing
The dataset was acquired via a publicly accessible API, which provided structured JSON data. The API calls were automated using Python scripts, ensuring efficient and consistent data retrieval.

Preprocessing steps included:

1. **Data Cleaning**: Handling missing values by filling or removing incomplete records,changing the type of every feature to its right type ,dealt with outliers and wrong typing by removing the rows .
2. **Encoding :** We took every category feature that is nominal and one hot encode it .
3. **Scaling :** We used Standard scaling on our coordinates so our algorithms can manage and understand better the data .

Feature Selection
We focused on features most relevant to accessibility analysis, such as location and accessibility status, while excluding non-essential attributes like ATM branding,ATM branch code ,ATM address extra , all of these features do not contribute to our analysis and we chose to remove them .

	_id	Bank_Code	Bank_Name	Branch_Code	Atm_Num	ATM_Address	ATM_Address_Extra	City	Commission	ATM_Type	ATM_Location	Handicap_Access	X_Coordinate	Y_Coordinate
0	1	14	בנק אוצר החייל בע"מ	355	3551	שד' מנחם בגין 7	מרכז צימר, ככר הסיטי	אשדוד	לא	משיכת מזומן	על קיר הסניף	כן	31.788451	34.641896
1	2	14	בנק אוצר החייל בע"מ	397	3976	שד' דואני 18	שד' דואני 18	יבנה	לא	מכשיר מידע/או מתן הוראות	במרחק של יותר מ- 500 מטר מהסניף	כן	31.873128	34.738887
2	3	14	בנק אוצר החייל בע"מ	355	3552	שד' מנחם בגין 7	מרכז צימר, ככר הסיטי	אשדוד	לא	משיכת מזומן	על קיר הסניף	כן	31.788451	34.641896
3	4	14	בנק אוצר החייל בע"מ	377	3773	שד' התמרים 11	שדרות התמרים 11	אילת	לא	משיכת מזומן	בתוך הסניף	כן	29.555192	34.952591
4	5	14	בנק אוצר החייל בע"מ	382	3821	יוספטל 92	קניון בת ים	בת ים	לא	משיכת מזומן	על קיר הסניף	כן	32.015182	34.756167
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3364	3365	52	בנק פועלי אגודת ישראל בע"מ	188	1882	יעקב לנדאו 4		בני ברק	לא	משיכת מזומן	על קיר הסניף	כן	32.079522	34.833233
3365	3366	52	בנק פועלי אגודת ישראל בע"מ	160	1601	אהרונוביץ 10		בני ברק	לא	משיכת מזומן	על קיר הסניף	כן	32.091008	34.838876
3366	3367	26	יובנק בע"מ	279	2791	הרצל 182		רחובות	לא	משיכת מזומן	על קיר הסניף	כן	31.896860	34.811048
3367	3368	26	יובנק בע"מ	288	2881	קרן היסוד 32		ירושלים	לא	משיכת מזומן	על קיר הסניף	כן	31.772074	35.221880
3368	3369	26	יובנק בע"מ	280	2801	אחוזה 124	פינת רחוב בר אילן	רעננה	לא	משיכת מזומן	על קיר הסניף	לא	32.180927	34.873921

3369 rows × 14 columns

Methodology

In our project, we focused on two main aspects ,classifying ATM accessibility and performing geographical clustering for regional analysis.

Classification Analysis
We tested five classification algorithms to predict ATM accessibility:

1. Logistic Regression
2. Random Forest
3. Support Vector Machine
4. XGBoost
5. K-Nearest Neighbors

We chose these models to balance traditional algorithms like Logistic Regression, Random Forest with more advanced ones like XGBoost to handle the dataset's mix of numerical and categorical features. After evaluating performance using various metrics, XGBoost emerged as the best due to its strong results and ability to manage complex interactions between features.

Clustering Analysis
For clustering, we explored three methods:

- **DBSCAN**:

  DBSCAN was chosen because it's good for identifying dense areas of data, which is critical when working with spatial data like ATM locations. However, it showed limitations in covering broader regions, so we looked at other options.

- **K-Means** :
  - We used K-Means to see how it would perform with a simple, fixed cluster setup. It provided basic geographical segmentation but was limited because the rigid cluster boundaries didn't necessarily reflect the natural distribution of ATM locations.
- **Gaussian Mixture Models** :
  - GMM was selected as the final clustering approach because it yielded the highest silhouette score, which indicated that the model created well-defined and meaningful clusters. It also allowed for more flexible clustering by modeling the data as a mixture of several Gaussian distributions, which suited the regional analysis better.

The GMM clustering technique was ideal for our needs because it identified six key regions and also outliers: North, South, Eilat, Jerusalem, Central, and Outliers. This flexibility in the model helped us map out the ATM locations in a way that was both meaningful and actionable. We were able to pinpoint specific regions, like Tel Aviv and the area, where accessibility improvements were most needed

Experiments and Results

Classification

As we wrote earlier , we used for the classification task 5 models we thought were the best for the task we were aiming for , For this section we evaluated the performance of each model which to our surprised almost every model did pretty well so we didn't feel like tuning the parameters will change the outcome so we used the default parameters for this one .

Later, we briefly experimented with hyperparameter tuning using grid search. Due to high running times, we focused only on Logistic Regression and K-Nearest Neighbors (KNN). The selected parameters were:

- Logistic Regression: C=1, penalty='l2', solver='lbfgs'
- K-Nearest Neighbors: metric='euclidean', n_neighbors=15, weights='uniform'

The evaluation metrics we used were accuracy, F1-score, precision, recall, and AUC. We picked these metrics because the dataset had an imbalance between classes, so precision and recall were helpful for checking how well the model handled positive and negative cases. Accuracy was calculated using cross-validation to make sure the results were consistent across splits. Precision showed the percentage of correctly predicted positive cases, and recall focused on finding all the actual positive cases. The F1-score combined precision and recall into one number, which was useful for dealing with imbalanced data. AUC measured how well the model could separate positive and negative classes, giving us an idea of the overall performance.

Here's our evaluations :

Algorithm	Accuracy (CV)	F1 SCORE	AUC	Recall	Precision
Random Forest	0.9318	0.9693	0.7397	0.9828	0.9561
Support Vector Machine	0.9352	0.9772	0.5058	1	0.9554
XGBoost	0.9262	0.9699	0.8073	0.9767	0.9632
K-Nearest Neighbors	0.9307	0.9717	0.7101	0.9891	0.9550
Logistic Regression	0.9355	0.9772	0.5683	1	0.9554

Among these methods, XGBoost performed the best in terms of AUC (0.8073), making it the preferred choice for our project. While SVM and LR achieved slightly higher accuracy and F1

scores, XGBoost's superior AUC suggests better performance in distinguishing between accessible and non-accessible ATMs.

Clustering

For Clustering , we evaluate the performance of our models based on our knowledge and based on visualization but also we tried to use Silhouette Score.

DBSCAN Silhouette Score: 0.3872420122264845

K-Means Silhouette Score: 0.4898050498085075

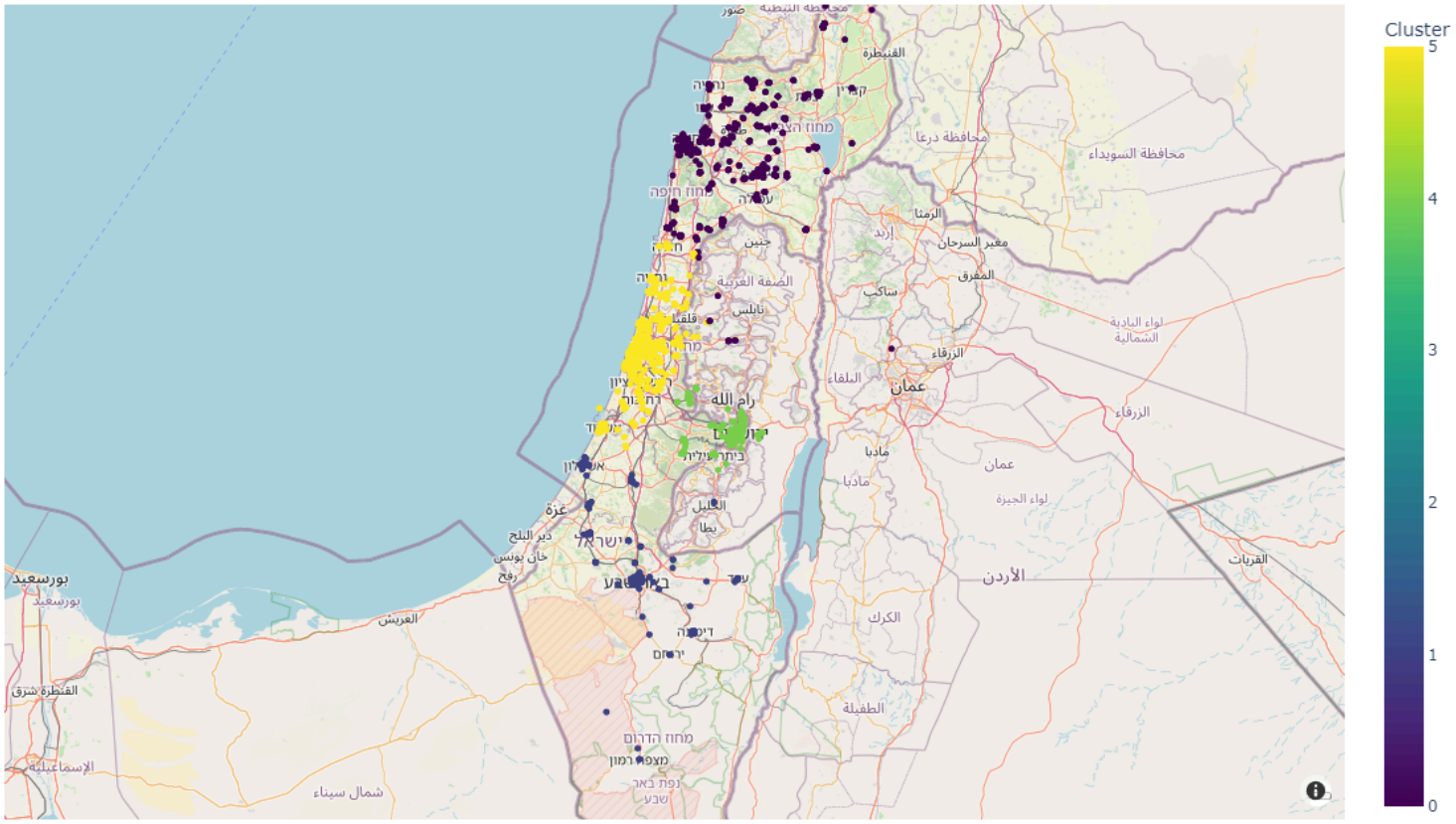
GMM Silhouette Score: 0.5998970523868988

Results and Observations

Classification: XGBoost delivered the best results, showing high precision, recall, and F1-scores. It required more training time compared to other models but performed better. Random Forest also performed well but did not match XGBoost's performance. Logistic Regression was faster but struggled with the data's complexity.

Clustering: GMM was chosen with n_components=6 as the best clustering method based on silhouette scores and visualizations. DBSCAN struggled with noisy data, and KMeans worked well but was sensitive to the number of clusters.

ATM Clusters Map



The results we got from the classification models were pretty surprising most of them actually performed way better than we expected, as you can see in the evaluation section. What also caught us off guard was the running time of XGBoost.

The first issue we ran into was figuring out how to evaluate how good the clusters actually were. The problem is that when you're working with coordinate data, you can't just rely on all the usual evaluation methods. I mean, how are they supposed to know if DBScan, for example, did a good job identifying cities in the country? It’s not that straightforward.

So, we ended up relying on our basic knowledge of regions, along with visualizations, to get a better sense of the clusters. On top of that, we used the Silhouette Score, which measures how well the points fit within their clusters based on distance.

Initially, our plan was to use clustering methods to figure out which cities had the biggest issues with handicap accessibility. But once we applied the models to our data, we quickly realized that they struggled a lot when dealing with really dense data points. For example, DBScan no matter what epsilon values or sample sizes we tried ,we just couldn’t hit the target and identify most of the cities we were focusing on. The same thing happened with k-means (even with different k values) and GMM.

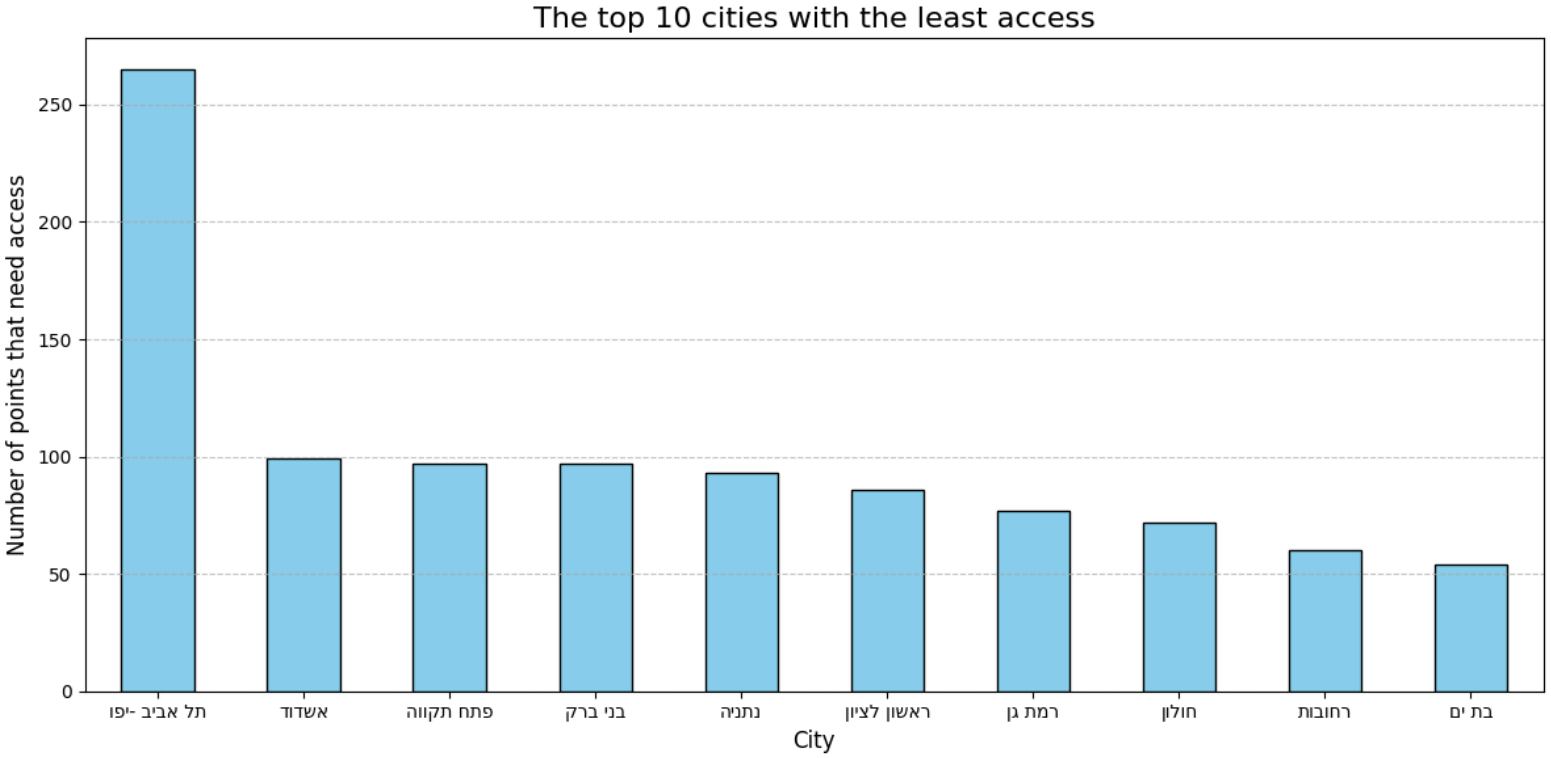
Because of these limitations, we had to take a step back and rethink our approach. Instead of clustering cities directly, we decided to cluster regions or areas first. After that, we focused on identifying which cluster was the most problematic and then 'zoomed in' on it to figure out exactly which cities needed to improve their handicap accessibility.

*A map highlighting areas with handicap accessibility points that need improvement

Conclusion and Discussion

This project analyzed accessibility data to predict whether ATMs were accessible based on their location and features. Classification and clustering approaches were used, and visualizations highlighted accessibility gaps

As u can see below , Tel-Aviv need to improve their accessibility in all terms



Team Contributions

We split the tasks among the team members. Shahar handled the preprocessing and classification tasks, while Ron focused on clustering and visualization. We worked together as a team on the evaluation and conclusion. The final evaluation was truly a combined effort from all areas.

Future Work

Future improvements could include testing ensemble methods like Gradient Boosting or exploring neural networks for classification. More advanced clustering algorithms and fine-tuning DBSCAN parameters might improve results. Deploying the final model as an interactive tool for accessibility planning is another potential direction.