Using Machine Learning to predict population density of Singapore neighbourhoods

K.S. Sujith

Introduction

In this project, I will use Machine Learning to predict the population density in different neighbourhoods of Singapore using the presence of amenities such as Schools, Medical facilities, food and beverage outlets, transportation facilities, sports and recreational facilities, and shopping areas in the area. Foursquare API will be used to get the amenities in an area based on the above categories and the data will be used to train a machine learning model to predict the population density.

Singapore is a highly developed Island city-state with a population of 5 million and land area of approximately 725 km^2 [1,2]. It ranks high in many global indices such as GDP per capita, education, healthcare, quality of life, housing, etc. As a resident of the city, I was interested in knowing the spread on population in the city and the possible reasons behind it. As a highly planned city, it would also be easy to identify the trends observed based on the city planning zones and subzones.

This analysis and insights will be of importance both to the government and citizens. For the residents, this can be used to identify top places to reside-in. This analysis can also be modified easily to include any additional preferences one might have. For the government, this can help in better planning. This can help identify what people value as important, which places needs improvements and in what amenities, etc. This can also be modified to see effects of implementing potential suggestions, which can be a powerful tool in planning.

Data and Tools

Pandas package in python is used for data analysis and manipulation [3]. Population data for different planning subzones in Singapore is obtained from official statistics [4]. This data is available as csv/excel. A Github repository is used as a database. Singapore maps is available in geojson format from official data sources [5]. Folium library will be used to draw maps in python [3]. We will use the Geopy package [3] to assign a latitude and longitude suitable for each subzone and use this to query foursquare for nearby amenities mentioned above using the foursquare API [3]. We will then use this data to perform machine learning using Scikit-learn package [3] and predict the population density in that area. Other packages in python such as Numpy, Scipy, and Matplotlib will also be used for various manipulations and plotting [3].

Data Wrangling and Exploratory Analysis

Detailed population data is obtained in excel format which can be found here. The data is grouped by various categories such as planning zones, age group, gender, etc. This data is imported into a pandas dataframe and cleaned to obtain the total population in each planning subzones as shown below.

	Planning Area	Subzone	Population
0	Ang Mo Kio	ANG MO KIO TOWN CENTRE	4810.0
1	Ang Mo Kio	CHENG SAN	28070.0
2	Ang Mo Kio	CHONG BOON	26500.0
3	Ang Mo Kio	KEBUN BAHRU	22620.0
4	Ang Mo Kio	SEMBAWANG HILLS	6850.0

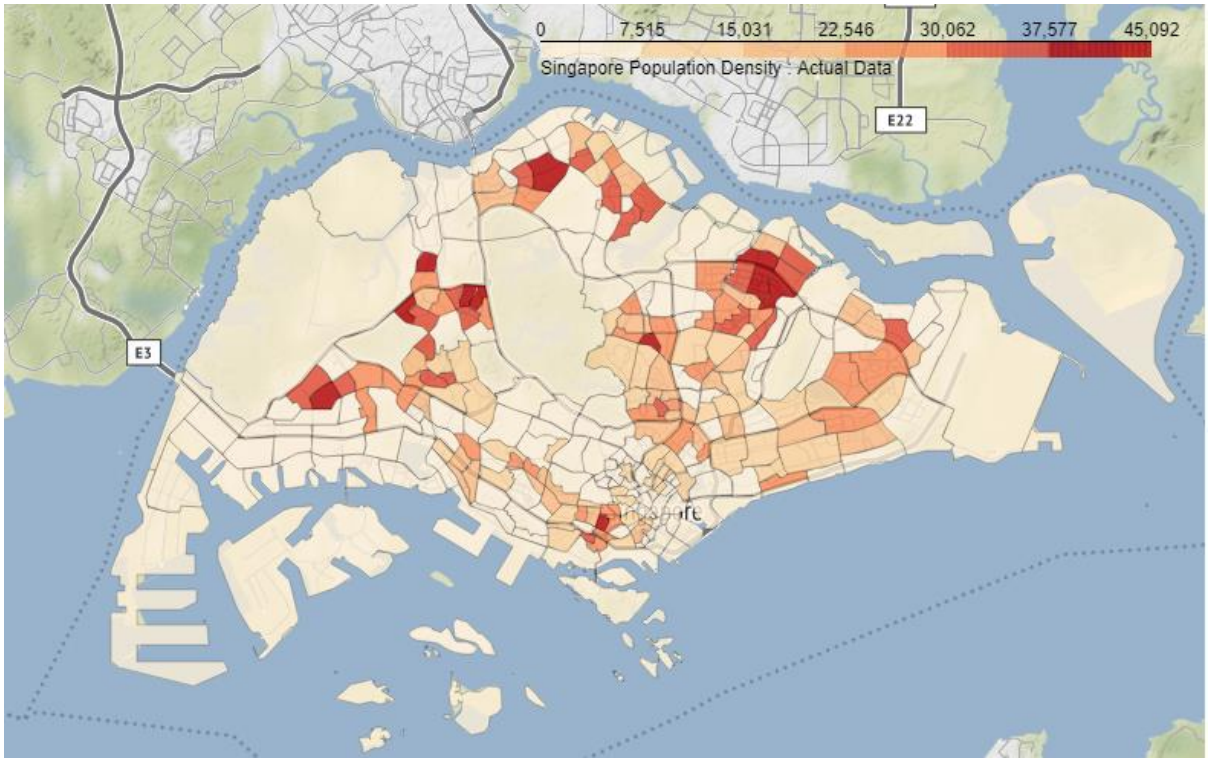
A cleaned version of maps with subzones can be found here. This json file was imported into a dataframe with subzones to ensure that the subzones match the ones in population data. 'geopy.geocoders.Nominatim' was used to obtain coordinates (latitude and longitude) values for each entry. The resulting location table is shown below.

	Subzone	Location	Latitude	Longitude
0	CAIRNHILL	Cairnhill	1.306561	103.839440
1	ANG MO KIO TOWN CENTRE	Ang Mo Kio Town Centre	1.371285	103.846994
2	YUHUA EAST	Yuhua East	1.343852	103.739330
3	SELETAR AEROSPACE PARK	Seletar Aerospace Park	1.417020	103.868082
4	SELETAR	Seletar	1.409849	103.877379

As area data for subzones was not available, the location data from geojson maps file was used to calculate it. The coordinates of polygons were used to calculate area for each subzone in km2. To verify the process, the sum of all areas was obtained to be 785 km^2, which is reasonable, considering the fact that planning subzones contains many neighbouring islands as well. With this, the two tables were merged by matching the subzone values and population density was calculated. The resulting main dataframe is shown below.

	Subzone	Location	Latitude	Longitude	Population	Area_km2	Density
0	CAIRNHILL	Cairnhill	1.306561	103.839440	3880.0	0.455518	8517.773771
1	ANG MO KIO TOWN CENTRE	Ang Mo Kio Town Centre	1.371285	103.846994	4810.0	0.319015	15077.653283
2	YUHUA EAST	Yuhua East	1.343852	103.739330	24770.0	0.930986	26606.212948
3	SELETAR AEROSPACE	Seletar Aerospace	1.417020	103.868082	20.0	3.567072	5.606840

Now, let's visualize the population density in Singapore using Folium choropleth maps.



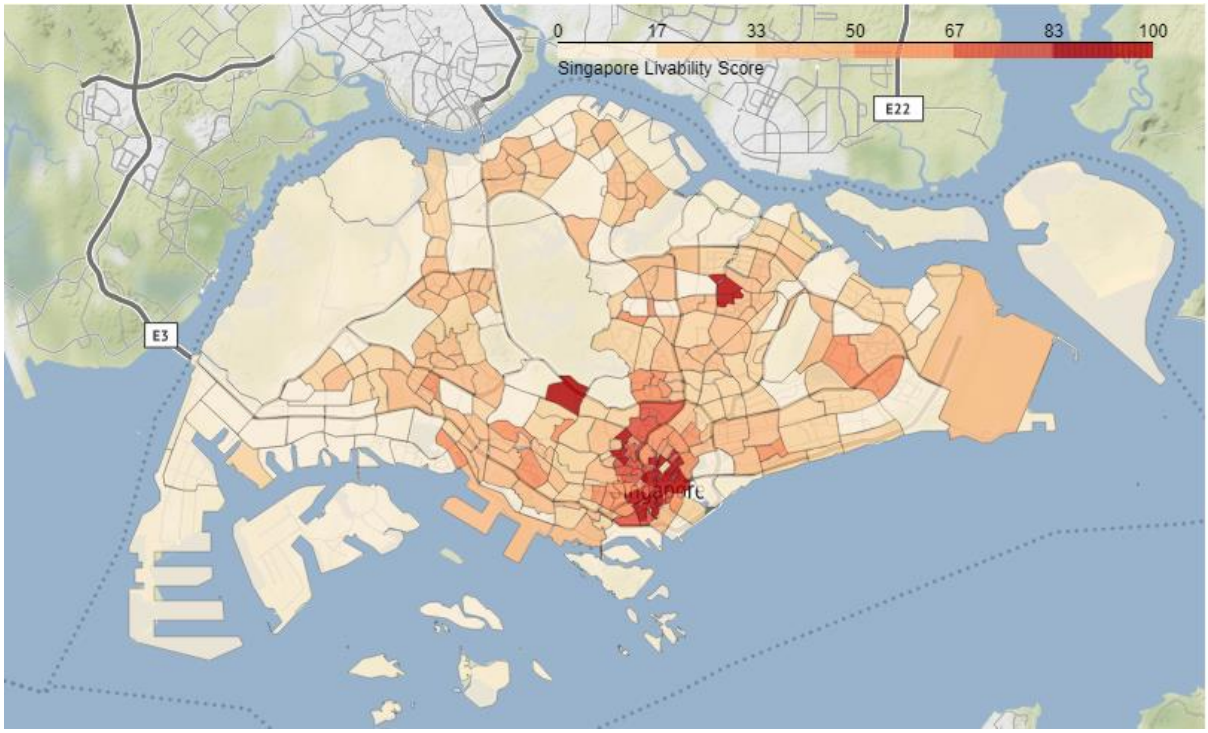
This map is very similar to that of the Nifty map found here, which means our methods are right.

Now, I will use the foursquare API to get nearby venues around each subzone. I selected six categories of venues to look for to judge the quality of life – a parameter I call liveability score. They are: 1) Educational Institutions, 2) Medical Facilities, 3) Food and Beverages Outlets, 4) Transport Facilities, 5) Sports and Recreational facilities, and 6) Shopping.

For each of the above categories, foursquare returns venues nearby the location within a radius of 1 km. 1 km was set because the average radius of subzones was found to be close to 1 km by comparing the total area and total number of subzones. The number of venues returned for each call was stored into a dataframe and the data was normalized as shown below.

	Subzone	School	College	Medical	Recreation	Transport	Shop	Food
0	CAIRNHILL	0.541667	0.439024	0.750000	0.545455	0.477124	0.974490	0.689516
1	ANG MO KIO TOWN CENTRE	0.562500	0.451220	0.481481	0.256198	0.281046	0.418367	0.379032
2	YUHUA EAST	0.375000	0.402439	0.370370	0.330579	0.281046	0.224490	0.225806
3	SELETAR AEROSPACE PARK	0.104167	0.073171	0.009259	0.033058	0.071895	0.015306	0.028226
4	SELETAR	0.020833	0.048780	0.009259	0.024793	0.045752	0.010204	0.032258

As an exploratory model, I define liveability score as the average of all above categories multiplied by 100. It can be viewed as a score given to each subzone based on the amenities nearby. Let's see how well our liveability score predicts the population density.



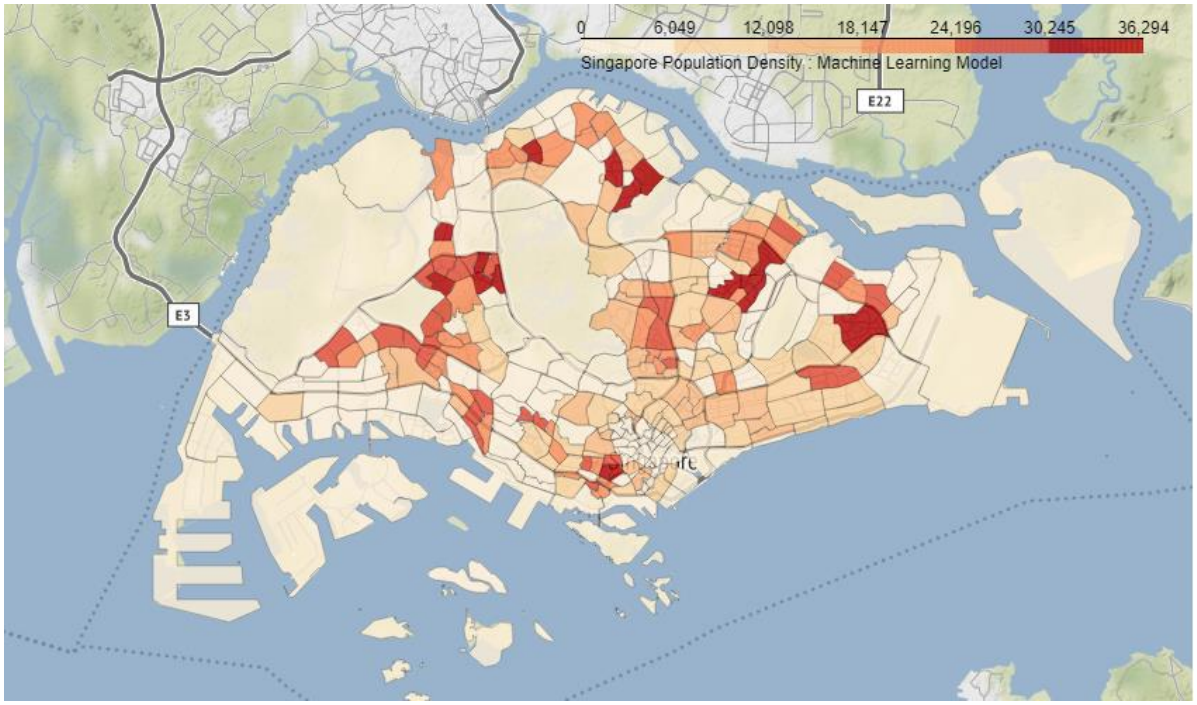
Comparing this to the previous plot with actual data, it can be seen that my liveability score is not a good predictor for the data. However, it should be noted that even this crude model can show the general spread of population. We now proceed to implement machine learning models to predict the data more accurately.

Machine Learning Models and Analysis

The normalized data for our independent variables is copied into X and the corresponding population density is copied into y, which is the target variable. X and y are then split into Xtr, Xtst, ytr, ytst for training and testing the model. Since y is a continuous value, I used regression models. Linear regression and k-nearest neighbour regression was used and the results are compared. Scikit-learn package was used for this.

a) Linear Regression

In linear regression, vector y is a linear combination of vectors in X, ie, MX=y. Liveability score I used earlier as a crude model is a special case of this expression. In linear regression model, this expression is optimized by minimizing the error. 'sklearn.linear_model_LinearRegression' was used for this. The image below shows the results.



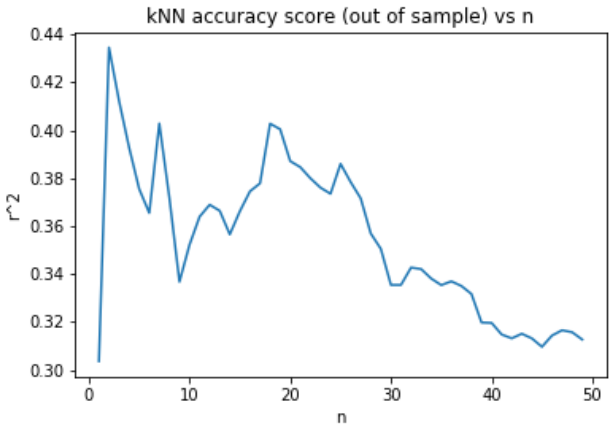
Visually, this looks similar to the actual data. This is a big improvement over the liveability score model. Let's look at the accuracy of this model by evaluating R-squared score. R-squared ranges from 0 to 1 and higher value implies better fitting.

Linear Regression	R-squared value
Out of sample	0.165
In sample	0.135

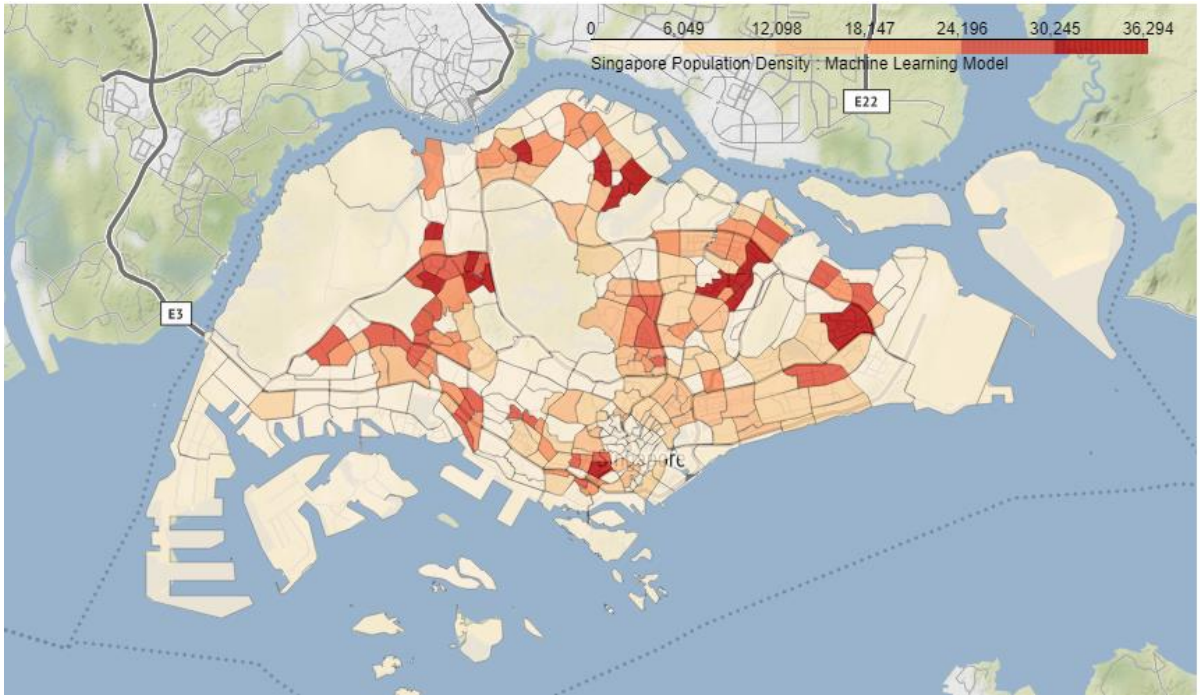
As we can see, even though the model looks reasonable when comparing the plots, from R-squared value, we can see that this is not a good predictor for our data.

b) k-Nearest Neighbour (KNN) Regression

In KNN regression, the distance between all points are evaluated in a multi-dimensional space based on X and the nearest neighbours to each entry are identified. The target variable is calculated as the average of 'k' nearest neighbours. 'sklearn.neighbors.KNeighborsRegressor' is used for this. To identify the suitable k, we run the regression for different values of k and select the one with highest R-squared value.

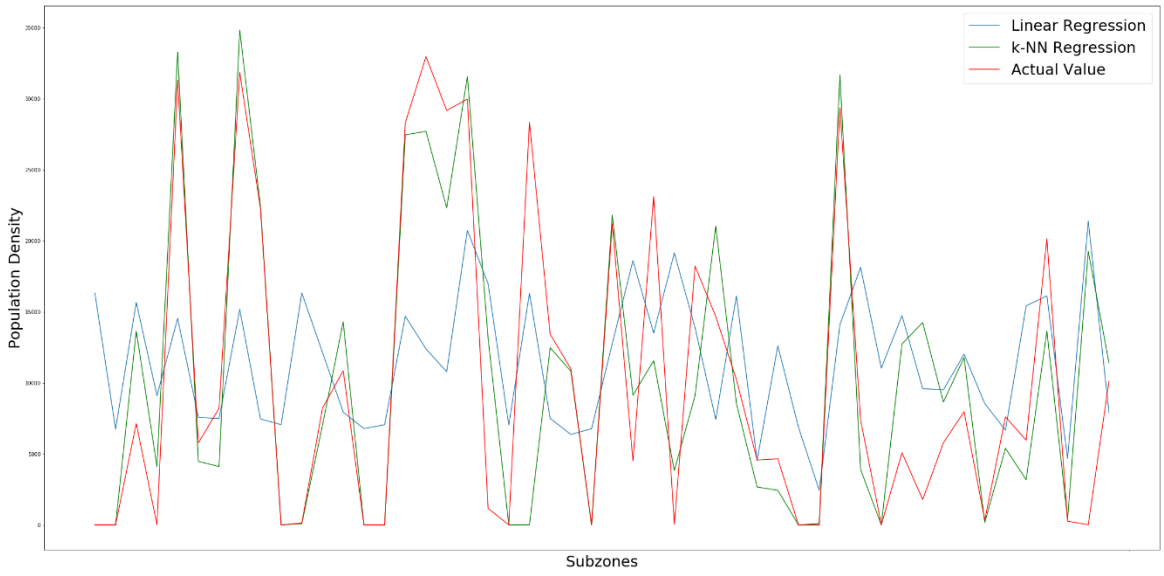


k=2 gives best accuracy. We use this value of k to predict population density. The results are shown below.



KNN Regression	R-squared value
Out of sample	0.435
In sample	0.695

As seen, the results are much better now and the accuracy has improved. We also note that the in-sample accuracy is higher than the out-of-sample accuracy. A comparison of predictions using Linear regression model and KNN regression model with the actual values is shown below (For clarity, only a subset of the data is shown).



As can be inferred from the figure, KNN model does a better job of predicting the population density values compared to linear model.

Limitations

- We only use six independent variables to evaluate our models. In practice, there are many more factors affecting one's choice of neighbourhood such as rent rates, available housing, etc. A more sophisticated model can be created to include more independent variables.
- We only consider one location in each neighbourhood and look at the amenities around it. A more accurate model can look at multiple locations in each neighbourhood.
- We have only looked at a circle of radius 1 km around a location in every subzone. However, subzones are of different shapes and might have differences due to geography.

Conclusions

We have successfully created a map of regions in Singapore showing population densities. Six independent categories were chosen to make a predictive model: educational institutions, medical facilities, food and beverages outlets, transport facilities, sports and recreational facilities, and shopping. Amenities belonging to these categories were retrieved using foursquare API. This was used to build a linear regression and KNN regression model and the results were compared. We find that the KNN model predicts our data with reasonable accuracy.

References

All data, python notebook, and relevant files for this project are available in the Github repository: github.com/kssujith92/Coursera_Capstone/tree/main/Final_Project

[1] Britannica: www.britannica.com/place/Singapore-capital

[2] Wikipedia: en.wikipedia.org/wiki/Singapore

[3] Python Libraries

- Pandas pandas.pydata.org
- Geopy geopy.readthedocs.io/en/stable
- Scikit Learn scikit-learn.org
- Folium python-visualization.github.io/folium
- Matplotlib matplotlib.org
- Scipy www.scipy.org
- Numpy numpy.org

[4] Department of Statistics, Singapore: www.singstat.gov.sg

[5] Singapore Public Data: data.gov.sg