ISSN:0975-9646
Imtiyaz Khan et al / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 14 (2) , 2023, 25-29

Predicting Livability Score

Dr. Imtiyaz Khan[1] , Mohammed Ziyan Ahmed[2] , Salman Awaise[2] , Syeda Ayesha Fatima[2]
[1]*Assistant Professor CS and AI Dept MJCET OU Hyderabad TS India*
[2]*BE CSE AI DS CS and AI Dept MJCET OU Hyderabad TS India*

Abstract-The concept of urban liveability is a crucial consideration in urban planning and policymaking. However, there is limited research on assessing liveability at a finer spatial scale, such as the community level. This article aims to evaluate community livability using an objective indicator system and a comprehensive evaluation framework that combines subjective perceptions and objective indicators. The study reveals diverse preferences of different age groups, significant heterogeneity among districts, and a decreasing spatial pattern of liveability from city centre to surrounding areas. These findings can inform urban planning departments and stakeholders involved in urban development.

Keywords-Socioeconomic Progress, Livability, Spatial Analysis,Urban Well-being, Human Development Index, Socioeconomic Advancement, Edge Detection Techniques

I.INTRODUCTION

Liveability serves as a crucial guiding principle in urban planning and policy making, with its definition and evaluation becoming a significant research topic. As socio-economic development accelerates and evolves into an ever changing sphere there is an increasing need to address the living conditions, especially with diverse groups of people migrating to cities and towns with different cultures, habits and criteria for a habitable environment. With the growing population of such migrant residents, such as students and travellers,seeking safe,comfortable, inclusive, and culturally accepting places to settle, there is a need to establish a uniform code for assessing the liveability of urban areas. Despite the current lack of comprehensive solutions, the evolving trend of migration and the thirst for new experiences demand the provision of the most liveable environments for individuals to thrive in.

II.EXISTING SYSTEM

Existing systems use general attributes like price, walkability score, reviews but it doesn't paint a complete picture. It has failed to catch up with the demand. The existing system is outdated and doesn't consider the modern-day requirements or demands of the people. An information deficit, a lack of multi-dimensional analysis of properties in a fast world where we can't afford to lose and resources are the cries of the previous systems i.e., not that informative. The existing systems are modelled and trained on a century old data. Since then, the demands, features, needs, conditions of the people have changed to accommodate the needs of today and the existing systems have more or less failed us and are not able to keep up with the growing needs of the customers. These systems work on superficial data gathered from peripheral sources which does not give accurate enough information and features about the property and has high chance for human error. The older systems offer a very little variety and diversification to choose from, leaving the customer displeased and leading the customer to adopt and prefer other alternatives. This leaves us to say the existing system was inadequate.

A. Proposed System

The proposed system provides a much more comprehensible and visually diverse image of livability. The proposed system gives the user the resources to finally see and want what they prefer. Any user wants to know the air quality, the crime rates, property type, how much they are getting, facings, accessibility, luxury and amenities of a place they might be frequenting for the foreseeable future.

It enables the city planners, developers and policy makers to bridge a gap between what they provide and what the people need. This promotes equal development throughout an area and prevents concentration of such ideas in select areas. The proposed model successfully bridges the gap the existing model has made by not keeping up with times and saves the users time, resources and mental anguish on a decision that would affect them for a long time to come. The Proposed system is not just adequate, it shows the people what they would want and what's good for them, it's not a Q&A it is the Answer

B. Problem Statement

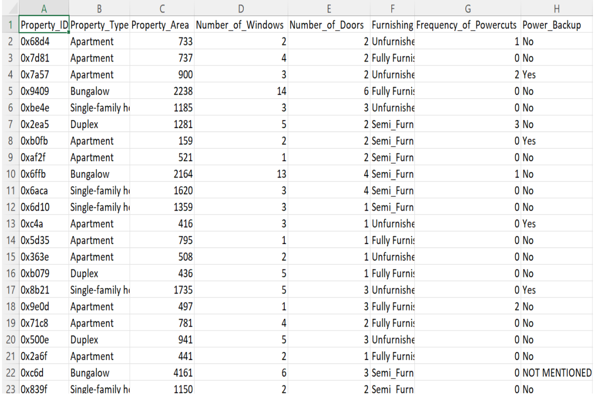
Building a Machine Learning model to identify the livability score of the property based on properties and location-based information.

III. METHODOLOGY

A. Dataset

1)Data Preprocessing:

Datasets from Kaggle used for training and testing with 80% for training and 20% for testing. We used a categorical imputer from the sklearn preprocessing library for cleaning the data, removing the null and discarding the insignificant values and features.



	A	B	C	D	E	F	G	H
1	Property_ID	Property_Type	Property_Area	Number_of_Windows	Number_of_Doors	Furnishing	Frequency_of_Powercuts	Power_Backup
2	0x68d4	Apartment	733	2	2	Unfurnishe	1	No
3	0x7d81	Apartment	737	4	2	Fully Furnis	0	No
4	0x7a57	Apartment	900	3	2	Unfurnishe	2	Yes
5	0x9409	Bungalow	2238	14	6	Fully Furnis	0	No
6	0xbe4e	Single-family h	1185	3	3	Unfurnishe	0	No
7	0x2ea5	Duplex	1281	5	2	Semi_Furn	3	No
8	0xb0fb	Apartment	159	2	2	Semi_Furn	0	Yes
9	0xaf2f	Apartment	521	1	2	Semi_Furn	0	No
10	0x6ffb	Bungalow	2164	13	4	Semi_Furn	1	No
11	0x6aca	Single-family h	1620	3	4	Semi_Furn	0	No
12	0x6d10	Single-family h	1359	3	1	Semi_Furn	0	No
13	0xc4a	Apartment	416	3	1	Unfurnishe	0	Yes
14	0x5d35	Apartment	795	1	1	Fully Furnis	0	No
15	0x363e	Apartment	508	2	1	Unfurnishe	0	No
16	0xb079	Duplex	436	5	1	Fully Furnis	0	No
17	0x8b21	Single-family h	1735	5	3	Unfurnishe	0	Yes
18	0x9e0d	Apartment	497	1	3	Fully Furnis	2	No
19	0x71c8	Apartment	781	4	2	Fully Furnis	0	No
20	0x500e	Duplex	941	5	3	Unfurnishe	0	No
21	0x2a6f	Apartment	441	2	1	Fully Furnis	0	No
22	0xc6d	Bungalow	4161	6	3	Semi_Furn	0	NOT MENTIONED
23	0x839f	Single-family h	1150	2	2	Semi_Furn	0	No

Fig.1: Training Dataset

I	J	K	L	M	N	O	P
Water_Supply	Traffic_Density	Crime_Rate	Dust_and_Noise	Air_Quality_Index	Neighborhood_	Habitability_score	
Once in a day - Ev	4.37	Well below aver	Medium	96	3.55	71.2	
Once in a day - M	7.45	Slightly below av	Medium	121	3.81	71.39	
Once in a day - M	6.16	Well above aver	Medium	100	1.34	31.46	
All time	5.46	Well below aver	Medium	116	4.77	93.7	
Once in a day - M	5.69	Well below aver	Medium	91	4.49	82.94	
All time	7.72	Well above aver	Medium	143	0.96	28.54	
Once in a day - M	6.77	Well below aver	Medium	90	4.48	80.65	
All time	4.14	Well below aver	Medium	89	4.82	73.51	
Once in a day - Ev	6.09	Well below aver	Medium	105	3.53	87.91	
All time	6.08	Well below aver	Medium	143	4.74	76.86	
Once in a day - M	6.96	Well below aver	Medium	84	4.49	71.31	
Once in a day - Ev	6.99	Well below aver	Medium	117	4.14	72.35	
NOT MENTIONED	6.92	Well below aver	Medium	169	4.07	79.82	
All time	6.41	Slightly below av	Medium	131	4.12	72.19	
Once in a day - M	6.7	Well above aver	Medium	102	2.59	83.68	
Once in a day - Ev	6.47	Slightly above av	Medium	164	2.82	77.28	
Once in two days	5.74	Slightly below av	Medium	103	1.96	73.46	
All time	6.51	Well below aver	Medium	101	4.79	84.58	
Once in a day - M	5.19	Well below aver	Medium	100	4.48	81.38	
All time	6.74	Well below aver	Medium	112	4.77	85.51	
All time	7.83	Well below aver	Medium	156	4.71	75.63	
Once in two days	7.01	Slightly below av	High	97	3.16	64.38	

Fig.2: Training Dataset Continuation

The data set initially included numerous irrelevant features that would only add unnecessary complexity to the model without contributing meaningful value to the final score. To ensure optimal results for customers, the features were carefully chosen based on modern needs and their relevance to customers. Thirteen relevant features, including air quality index, crime rates, etc., were identified as the most appropriate for achieving high accuracy and customer satisfaction.

	A	B	C	D	E	F	G	H
1	Property_ID	Property_Type	Property_Area	Number_of_Windows	Number_of_Doors	Furnishing	Frequency_of_Powercuts	Power_Backup
2	0x6e93	Apartment	293	3	1	Unfurnished	0	No
3	0x8787	Apartment	586	4	1	Semi_Furnish	0	No
4	0x6c17	Container Hom	305	1	2	Semi_Furnish	1	No
5	0x9dbd	Apartment	258	2	1	Semi_Furnish	1	No
6	0xbfde	Bungalow	3031	12	4	Fully Furnish	0	No
7	0x6a39	Apartment	350	4	3	Unfurnished	0	No
8	0x47a6	Single-family ho	1163	2	2	Semi_Furnish	0	No
9	0x7687	Container Hom	470	3	2	Semi_Furnish	0	No
10	0x963a	Bungalow	2057	12	4	Fully Furnish	1	No
11	0xbcbb	Single-family ho	1221	2	3	Semi_Furnish	0	No
12	0x4950	Container Hom	203	1	2	Semi_Furnish	1	No
13	0x316a	Duplex	996	6	3	Semi_Furnish	0	No
14	0x6b21	#R%$G&867	155	1	1	NA	1	No
15	0x6252	Bungalow	2174	7	4	Semi_Furnish	1	No
16	0xa67f	Single-family ho	1170	5	2	Fully Furnish	3	Yes
17	0x2cd8	Duplex	1292	6	2	Unfurnished	0	No
18	0x5926	Duplex	942	6	1	Unfurnished	2	No
19	0x4b5e	Apartment	137	3	1	Semi_Furnish	0	No
20	0x76af	Apartment	460	1	3	Semi_Furnish	1	No
21	0xb44b	Apartment	123	2	3	Unfurnished	0	No
22	0x2e49	Duplex	450	5	1	Unfurnished	0	No
23	0x83fc	Apartment	971	2	1	Unfurnished	0	No

Fig.3: Testing Dataset

H	I	J	K	L	M	N
Power_Backup	Water_Supply	Traffic_Density	Crime_Rate	Dust_and_Nois	Air_Quality_Index	Neighborhood_Review
No	Once in a day - M	7.28	Well above ave	Medium	152	2.52
No	Once in a day - Ev	7.63	Well below ave	Medium	92	4.16
No	All time	5.39	Slightly above a	Medium	90	2.92
No	All time	7.53	Slightly below a	Medium	158	3.45
No	All time	8.79	Well above ave	High	186	2.72
No	Once in a day - M	7.44	Well below ave	High	119	4.38
No	All time	6.68	Well below ave	Medium	126	4.75
No	Once in a day - Ev	5.99	Slightly below a	Medium	109	3.52
No	Once in a day - M	6.24	Slightly below a	Medium	118	3.2
No	Once in a day - M	7.46	Well below ave	Medium	94	4.47
No	Once in a day - Ev	5	Slightly below a	Medium	115	2.89
No	All time	6.74	Well below ave	Medium	110	4.77
No	Once in a day - M	3.77	Slightly above a	Low	55	2.73
No	All time	5.02	Well below ave	Medium	109	4.16
Yes	Once in two days	5.09	Slightly below a	Medium	82	1.37
No	All time	7.06	Well below ave	Medium	84	4.8
No	NOT MENTIONED	4.7	Slightly above a	High	110	1.9
No	All time	5.23	Slightly below a	Medium	103	4.16
No	Once in a day - M	5.42	Slightly below a	Medium	86	3.24
No	Once in a day - M	6.88	Well below ave	Low	155	4.46
No	All time	6.65	Well below ave	Medium	111	4.77
No	Once in a day - Ev	7.03	Slightly below a	Medium	148	3.47

Fig.4: Testing Dataset Continuation

Basic exploratory data analysis using pandas, matplotlib, seaborn packages.

Data pre-processing
- Missing value indicator
- Missing value imputation for the columns
- property_type
- number_of_windows
- furnishing
- frequency_of_powercuts
- crime_rate
- Dust_and_noise

The final features for the model are as follows
- 0_property_type
- 1_property_area
- 2_number_of_windows
- 3_number_of_doors
- 4_furnishing
- 5_frequency_of_powercuts
- 6_power_backup
- 7_water_supply
- 8_traffic_density_score
- 9_crime_rate
- 10_dust_and_noise
- 11_air_quality_index
- 12_neighborhood_review

B. Algorithms Used:

1)Linear Regression:

Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable.

This form of analysis estimates the coefficients of the linear equation, involving one or more independent variables that best predict the value of the dependent variable. Linear regression fits a straight line or surface that minimises the discrepancies between predicted and actual output values. There are simple linear regression calculators that use a "least squares" method to discover the best-fit line for a set of paired data. You then estimate the value of X (dependent variable) from Y (independent variable).

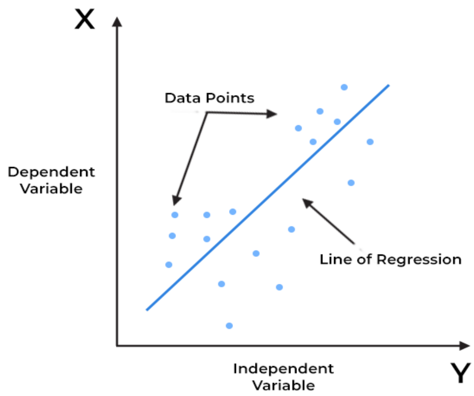


Fig.5: Linear Regression

2)Random Forest Regression:

Random forests or random decision forests is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees.

Fig.6: Random Forest Regression

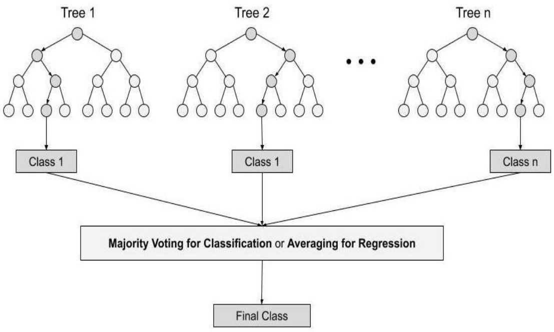


Fig.7: Random Forest Regression Representation

IV. Implementation

A. *Requirements*:

1) Software Requirements:

- Windows 11
- Python 3.9

2) Hardware Requirements:

- NVIDIA 3060
- RAM-32 GB
- 1-TB Hard Disk

3) Libraries Used:

- NumPy
- Pandas
- Matplotlib
- Seaborn
- sklearn
- Streamlit

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Fig.8: Library Installation (1)

```
import numpy as np
import pickle
import pandas as pd
import streamlit as st
```

Fig.9: Library Installation (2)

4) Code Snippets:

```
train_data = pd.read_csv('/content/gdrive/MyDrive/Mini Project/train.csv')
train_data.head()
```

	Property_ID	Property_Type	Property_Area	Number_of_Windows	Number_of_Doors	Furnishing	Frequency_of_Powercuts	Power_Backup
0	0x68d4	Apartment	733	2	2	Unfurnished	1	No
1	0x7d81	Apartment	737	4	2	Fully Furnished	0	No
2	0x7a57	Apartment	900	3	2	Unfurnished	2	Yes
3	0x9409	Bungalow	2238	14	6	Fully Furnished	0	No
4	0xbe4e	Single-family home	1185	3	3	Unfurnished	0	No

Fig.10: Loading the dataset

```
X_train.keys()
X_train.isnull().any()
```

```
Property_Type              False
Property_Area              False
Number_of_Windows          False
Number_of_Doors            False
Furnishing                 False
Frequency_of_Powercuts     False
Power_Backup               False
Water_Supply               False
Traffic_Density_Score      False
Crime_Rate                 False
Dust_and_Noise             False
Air_Quality_Index          False
Neighborhood_Review        False
dtype: bool
```

Fig 11: Checking for null values after cleaning the dataset

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
imputer_windows = SimpleImputer(strategy='most_frequent')
X_train['Number_of_Windows'] = imputer_windows.fit_transform(X_train[['Number_of_Windows']])
```

Fig 12: Importing Categorical Imputer from sklearn for handling the null values

```
model_linear.score(X_train, y_train) * 100
test_data = pd.read_csv('/content/gdrive/MyDrive/Mini Project/test.csv')
test_data.head()
```

	Property_ID	Property_Type	Property_Area	Number_of_Windows	Number_of_Doors	Furnishing	Frequency_of_Powercuts
0	0x6e93	Apartment	293	3	1	Unfurnished	0
1	0x8787	Apartment	586	4	1	Semi_Furnished	0
2	0x6c17	Container Home	305	1	2	Semi_Furnished	1
3	0x9dbd	Apartment	258	2	1	Semi_Furnished	1
4	0xbfde	Bungalow	3031	12	4	Fully Furnished	0

Fig 13: Linear Regression

```
from sklearn.ensemble import RandomForestRegressor
rfr = RandomForestRegressor(n_estimators=250)
model_f=rfr.fit(X_train, y_train)
prediction_rfr = rfr.predict(X_test)
rfr.score(X_train, y_train) * 100
result_rfr = pd.DataFrame([test_data['Property_Type'],test_data['Property_ID'], pre
result_rfr.to_csv('C://Users/Admin/Documents/get_a_room_rfr_250.csv', index=False)
result_rfr.head(100)
```

	Property_Type	Property_ID	Habitability_score
0	Apartment	0x6e93	29.80276
1	Apartment	0x8787	80.61992
2	Container Home	0x6c17	65.34208
3	Apartment	0x9dbd	72.58064
4	Bungalow	0xbfde	74.22992
...	...	...	...
95	Apartment	0x21b6	77.09228
96	Bungalow	0xc799	72.34644
97	Single-family home	0x7011	71.24524
98	Apartment	0x8018	82.65276
99	Bungalow	0xc73f	73.6236

100 rows × 3 columns

Fig 14: Random Forest Regression Predicting the livability Score

5) Execution:

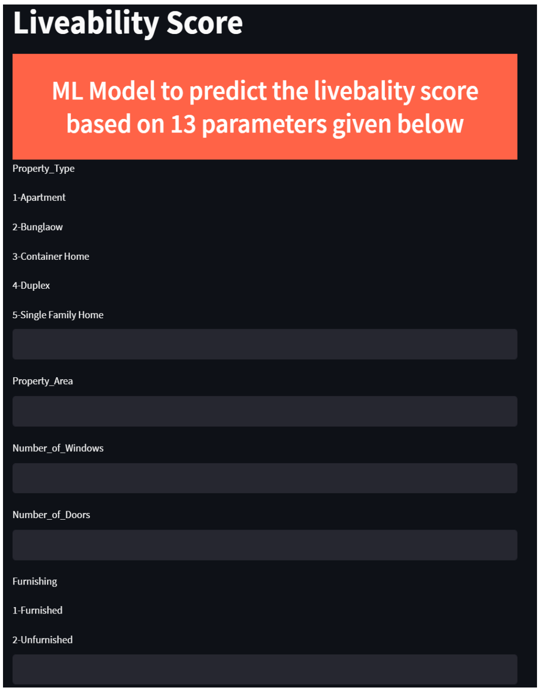


Fig 15: Web app Using Streamlit

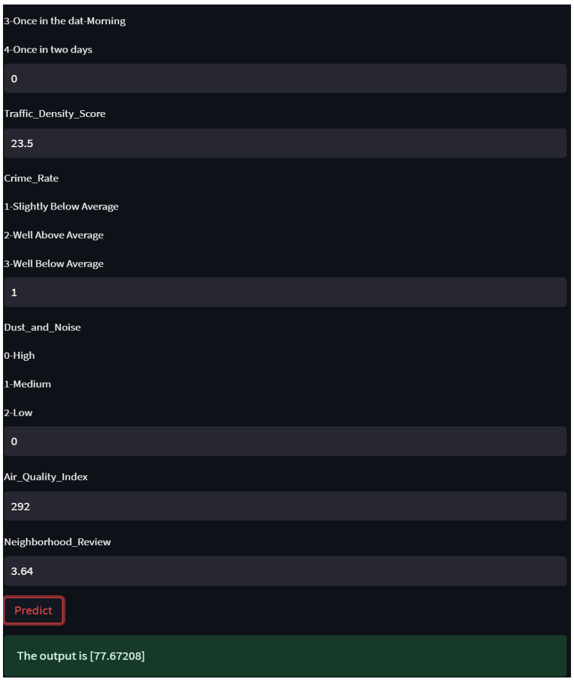


Fig 16: Predicting the livability score

With the help of this model one can get the Livability score of a property based on the certain attributes and basic information. The model is trained and tested to mathematically and visually represent the predicted Livability Score of a property type. The mathematical representation has been predicted visually using data visualisation tools.

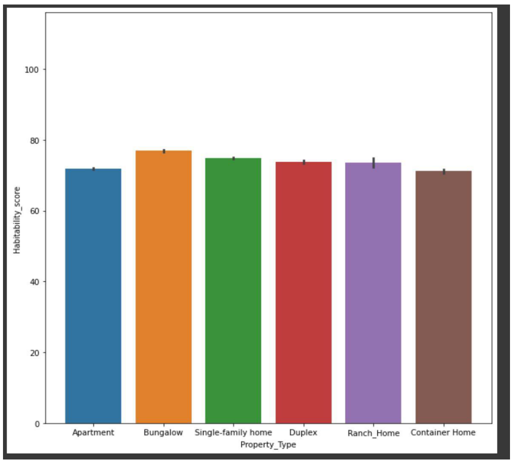


Fig 17: Distribution of Livability of Property Types

The result is clear, the end user can see what's available to them and how they fare, the end user can see what they can do to have a property with the best livability score with their resources. The model provides a 360-degrees in-depth insight into how to choose a property using various visual tools and a mathematical quantifiable number to give the user the best experience and hence protecting the user from wasting their precious human resources and financial resources on bad investments and bridging the gap between the sellers and the consumers.

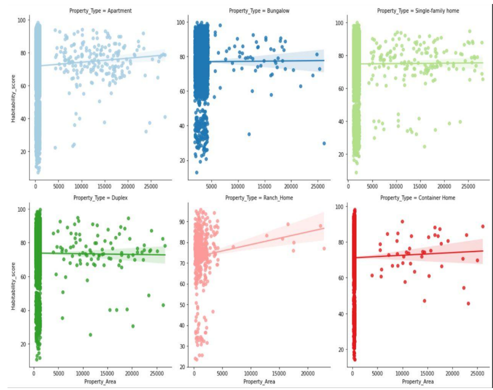


Fig 18: Scatter plot of Livability of Property Types

V. Conclusions and Enhancements

The Livability Score is a comprehensive ranking system that assesses living conditions in various locations. It takes into account 18 different indicators, such as safety, culture, and socio-economic development, to provide a clear understanding of an area's livability. The aim of the Livability Score is to simplify the process of choosing the best neighbourhood for residents and newcomers alike, by providing a uniform code for what constitutes a livable place. By saving time and resources, the Livability Score empowers individuals to make informed decisions about their living environment. In conclusion, the Livability Score is a valuable tool for evaluating the quality of life in different neighbourhoods, helping individuals make well-informed choices about where to live.

REFERENCES

[1] https://numpy.org/doc/
[2] https://scikit-learn.org
[3] https://pandas.pydata.org/docs/
[4] https://matplotlib.org/stable/index.html
[5] https://seaborn.pydata.org/
[6]https://docs.python.org/3/library/pickle.html