A KNN-Linear Regression Fusion Approach for Improved Real Estate Price Estimation

Prit J. Kanadiya[1], Pramila M. Chawan[2]

[1]Final Year B. Tech, Department of Computer Engineering and Information Technology, VJTI Mumbai, Maharashtra, India
[2] Associate Professor, Department of Computer Engineering and Information Technology, VJTI Mumbai, Maharashtra, India

---***---

Abstract – *In this study, we introduce a novel hybrid approach for house price prediction by integrating K-Nearest Neighbors (KNN) with Linear Regression. Our method leverages the strengths of both models to enhance predictive accuracy. Initially, we evaluate the effectiveness of geospatial features in Linear Regression and K-Nearest Neighbors for predicting house prices in Mumbai. We utilize two distinct datasets: one containing traditional features such as the number of bedrooms, square footage, and other property characteristics, and another incorporating geospatial data represented by latitude and longitude. Building on this analysis, we propose a method that first identifies the K nearest houses using KNN, and then applies Linear Regression on this localized subset to predict the price of a test property. Our hybrid model demonstrates significant improvements in predictive performance, highlighting the critical role of spatial information in real estate valuation.*

Key Words: **House price prediction, Hybrid Prediction Model, Geospatial features, Predictive analysis, Feature Engineering, Machine learning**

1. INTRODUCTION

Predicting house prices is a key challenge in machine learning with broad applications in real estate and urban planning. Despite significant advancements in this field, accurately forecasting property values remains difficult due to the numerous factors that influence prices. Traditionally, house price prediction models rely on features such as the number of bedrooms, the size of the property in square feet, and the building's age. However, these factors alone often fail to capture the full picture of property value.

One crucial aspect that traditional models might overlook is the exact location of a property. Location, represented by latitude and longitude, can significantly impact house prices by reflecting factors like connectivity, neighborhood quality, and proximity to amenities. For example, the value of a house in a well-connected area with good schools and nearby public transport can differ greatly from a similar house located in a less desirable area. It might not be possible to capture these features explicitly. The location of a house implicitly models all these features. This spatial information can provide insights into aspects like accessibility and local services that are difficult to quantify but crucial for accurate price predictions.

In this study, we propose a new method that combines K-Nearest Neighbors (KNN) with Linear Regression to enhance house price prediction. We test this hybrid approach by comparing it with Linear Regression and KNN. Using two datasets—one with traditional features like the number of bedrooms and property size, and another with geospatial features such as latitude and longitude—we aim to assess how incorporating geospatial data improves the accuracy of house price predictions.

Our research is structured as follows: Section 2 reviews related literature to contextualize our approach. Section 3 provides details on the methodology which involves dataset, feature engineering, evaluating the impact of geospatial features, and describing our proposed hybrid model. The results of our studies are presented and interpreted in Section 4, and the paper's conclusion and discussion of the ramifications of our findings are covered in Section 5.

2. LITERATURE REVIEW

House price prediction can be modeled as a supervised learning problem, where the goal is to predict the price of a house based on certain characteristics. Mathematically, we have a dataset $D = \{(X_i, Y_i) \mid i = 1, 2, ..., n\}$, where $X_i \in R^m$ represents the feature vector of the i-th house and $Y_i \in R$ denotes the corresponding house price. Our goal is to learn a hypothesis function that accurately models the house prices by mapping the features of the house i.e. Xi to its price Yi. We need to find the optimal hypothesis f that minimizes the cost function represented as J(θ).

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(X_i), Y_i)$$

Here, $\ell(f(X_i), Y_i)$ represents the loss function that measures the error between the actual price Y_i. and the predicted price $f(X_i)$.

The optimization problem involves selecting f such that $J(\theta)$ is minimized, thereby providing the best possible predictions of house prices based on the input features.

Linear Regression and K-Nearest Neighbors (KNN) are popular algorithms to solve this problem. Each of these models has its own assumptions and characteristics, which we will discuss in this section.

2.1 Linear Regression

Linear regression is a common method for predicting continuous values, such as house prices, by fitting a straight line through the data.

Linear regression makes two core assumptions about the data. First, it assumes a linear relationship between the features and the label. For example, house price might have a linear relationship with the area of the house. The second assumption is that the noise, indicated by ε, follows a Gaussian distribution. This means the data points will be scattered around the true linear regression line in a bell-shaped curve.

Mathematical Formulation: Given a dataset $D = \{(X_i, Y_i) \mid i = 1, 2, ..., n\}$, where $X_i \in R^m$ represents the feature vector of the i-th house and $Y_i \in R$ is the targeted price, and the aim is to find the line that best fits the data.

Hypothesis Function: $f(Xi) = \theta_0 + \theta_1 X_{i1} + \theta_2 X_{i2} + \cdots + \theta_m X_{im}$ where $\theta_0, \theta_1, ..., \theta_m$ are parameters to be learned.

Cost Function (Mean Squared Error): The objective is to minimize $J(\theta)$ to find the optimal parameters θ.

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}(f(X_i) - Y_i)^2$$

Linear regression works well for traditional features like the number of bedrooms or the square footage of a house, as these often have a linear relationship with house prices.

2.2 K-Nearest Neighbors

The core assumption that KNN makes is that instances that are near in the feature space have similar target values. Thus, the target value of a new instance is likely to be the same as its nearest neighbors. In this model, the algorithm looks at the prices of the k nearest houses (its "neighbors") and takes the average of their prices when predicting the price of a new property.

Mathematical Formulation: Given a dataset $D = \{(X_i, Y_i) \mid i = 1, 2, ..., n\}$, where $X_i \in R^m$ represents the feature vector of the i-th house and $Y_i \in R$ denotes the corresponding house price, and a new query point X, we need to find the average of the target values of the k nearest neighbors.

$$\hat{Y} = \frac{1}{k}\sum_{j=1}^{k} Y_{(i_j)}$$

where $Y_{i1}, Y_{i2}, ..., Y_{ik}$ are the target values of the k nearest neighbors of query point X.

This model works well in real estate, especially because houses in the same area often have similar prices. For example, houses in the same locality in Mumbai are likely to have prices that are close to each other or that follow a common trend.

3. METHODOLOGY

This section outlines the methodology used for obtaining the proposed solution and explains the algorithm for the new solution. First, we cover the dataset and the feature engineering steps. Next, we evaluate the influence of geospatial features on existing machine learning models. Finally, we propose our hybrid algorithm.

3.1 Dataset and Feature Engineering

We utilized the Mumbai House Prices dataset on Kaggle, which contains house prices for several locations across Mumbai. The dataset includes essential features such as the number of bedrooms (BHK), type of house, locality, region, area in square feet, price, and additional attributes such as whether the house is new or resale and whether it is ready to move in or still under construction. The dataset consists of 76,038 instances.

Although the dataset includes valuable geospatial information, such as locality and region, this information is not directly usable in its raw form. Additionally, several fields, such as the type of house, are categorical and require conversion to numerical values before applying machine learning algorithms. To make the data more applicable to our analysis, we performed several steps of feature engineering.

3.1.1 Feature Engineering

1. Adding Geographical Coordinates

To integrate location information into our analysis, we added latitude and longitude coordinates for each property based on its region. We identified 228 unique locations and converted these into latitude and longitude using geocoding. We experimented with two geocoding libraries, Nominatim and Geocoder. Geocoder proved to be more effective, offering support for more regions and providing more accurate results. Some locations, such as “Khar, Mumbai, India” and

"Belapur, Mumbai, India," were incorrectly mapped. For these cases, we manually corrected the latitude and longitude.

Although we considered using both locality and region to define unique locations, this approach resulted in 9,842 unique locations, which requires more processing without significantly enhancing performance. Therefore, we chose to use region alone for geocoding, which simplified the process while still effectively incorporating geospatial features.

2. Encoding Ordinal Data

Some features in our dataset are categorical and have a natural order or ranking. To employ these categorical features in machine learning models, we need to first convert them into numerical values. Here's how we handled these features:

We assigned numerical values to different house characteristics to simplify their representation in the model. For the type of house, there are five categories, ranging from a "Studio Apartment," which is coded as 0, to a "Penthouse," coded as 1. The intermediate categories such as Apartment, Independent House, and Villa are coded as 0.25, 0.5, and 0.75 respectively.

Similarly, the age of the house was represented numerically, with "Resale" coded as 0, "New" as 1, and "Unknown" as 0.5. The house status was also translated into numerical values, where "Under Construction" was assigned a value of 0 and "Ready to Move" was assigned a value of 1.

By encoding these categorical features in this way, we preserve their inherent order or ranking, which helps our machine learning models understand and use this information effectively.

3. Converting Price Data to a Common Unit

The dataset features prices in two different units: Lakhs and Crores. To ensure consistency, we converted all prices to Crores. For instance, if a price was listed in Lakhs, we divided it by 100 to convert it to Crores.

4. Removing Unnecessary Columns

Columns such as "locality" and "region" are redundant since latitude and longitude effectively describe the geospatial feature of a house. Additionally, the "price-unit" column was no longer necessary after the price conversion. We removed these columns to streamline the dataset and focus on the most relevant features.

3.1.2 Final Dataset

After feature engineering, our dataset includes the following fields: BHK, type, area, price, status, age, latitude, and longitude. This enhanced dataset, which integrates both traditional features and geospatial information is used for training and evaluating our machine learning models.

The entire preprocessing and feature engineering process is executed using Python, and the final dataset was saved for further analysis.

3.2 Evaluating the Impact of Geospatial Features

Tables 1 and 2 show the performance of Linear Regression and K-Nearest Neighbors (KNN) on traditional datasets and geospatial datasets, respectively. Both the models are compared using MAE, MSE, and R-squared score. The traditional dataset includes only traditional features like the number of bedrooms and square footage, while the geospatial dataset also adds geospatial features, such as latitude and longitude.

Table -1: Model Performance on Traditional Dataset

	Linear Regression	KNN
MAE	0.7366	0.5688
MSE	1.5355	1.2095
R^2	0.5889	0.6762

Table -2: Model Performance on Geospatial Dataset

	Linear Regression	KNN
MAE	0.7379	0.3738
MSE	1.5353	0.7150
R^2	0.5890	0.8086

Chart 1 illustrates the MAE for each model across the two datasets. Red bars represent performance with the traditional dataset, while blue bars show results with the geospatial dataset. KNN notably benefits from the inclusion of geospatial features, with a substantial reduction in MAE. This indicates that KNN effectively captures the spatial proximity of houses.

The higher errors observed with KNN on the traditional dataset suggest that traditional features alone are insufficient for accurate predictions in urban areas like Mumbai, where spatial factors play a crucial role.

Linear Regression shows almost the same MAE for both datasets, implying that the model does not significantly benefit from the additional geospatial data. In fact, the MAE for the traditional dataset is slightly lower than for geospatial dataset. This may be because Linear Regression struggles with non-linear relationships in spatial data. Since house prices do not necessarily follow a linear pattern relative to geographic location, incorporating latitude and longitude does not substantially improve its predictive performance.

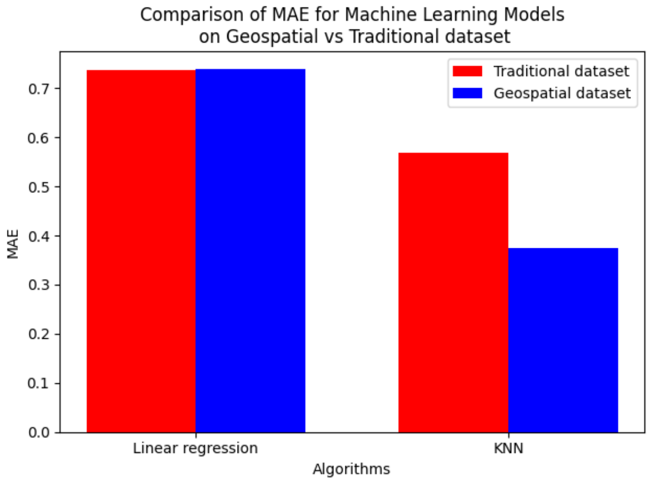


Chart -1: Comparison of MAE

Chart 2, which depicts the MSE, supports the MAE findings. The geospatial dataset consistently results in lower MSE for KNN compared to the traditional dataset. Linear Regression shows minimal improvement in MSE with the inclusion of geospatial features, reinforcing the idea that linear models do not effectively utilize spatial data for non-linear problems.

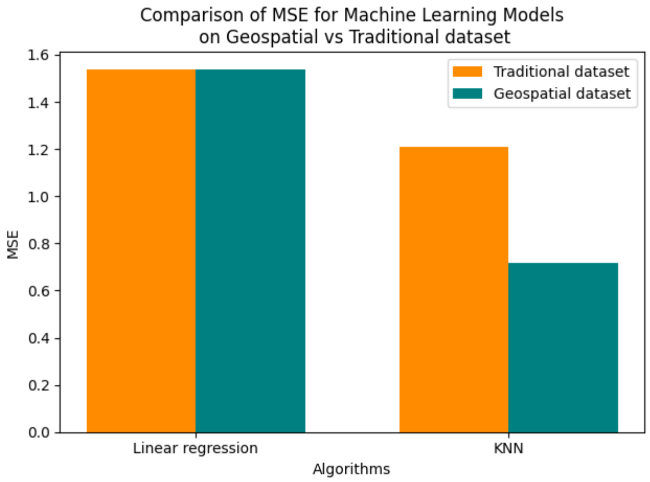


Chart -2: Comparison of MSE

Chart 3 displays the R-squared scores for both models. KNN demonstrates a higher R-squared score with the geospatial dataset, indicating a better fit for the variability in house prices. The score for linear regression only improves slightly after the addition of geospatial features.

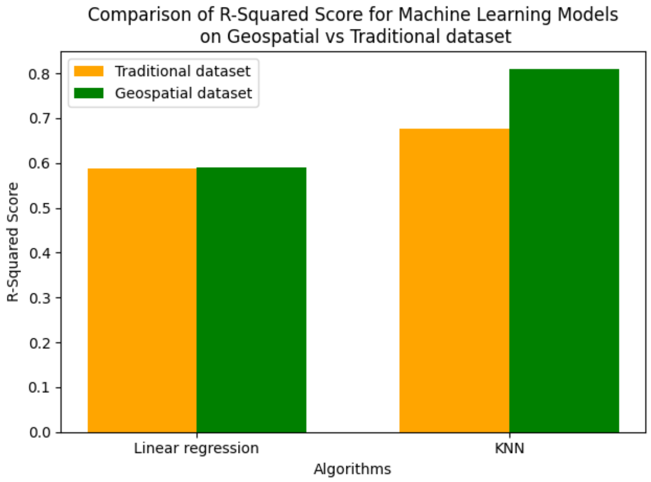


Chart -3: Comparison of R-Squared Score

These results indicate that KNN is most benefited when geospatial features are included. We only observe a marginal improvement in the linear regression model after adding the geospatial features.

3.3 Proposed Algorithm

Both Linear Regression and K-Nearest Neighbors (KNN) have their own advantages and limitations. From the results in Section 3.2, we observe that geospatial features significantly enhance the performance of KNN. While KNN excels at capturing the impact of location, Linear Regression offers benefits, especially in cases where the test and training data differ substantially in certain features, such as the number of bedrooms.

One drawback of Linear Regression is its reliance on the mean, which makes it sensitive to outliers. This characteristic could explain its weaker performance compared to KNN in house price prediction tasks, where outliers can skew predictions. In such cases, the median often provides a better representation of typical values. Overall, while KNN benefits greatly from incorporating geospatial data, Linear Regression shows limited improvement, highlighting its limitations in capturing complex, non-linear spatial relationships.

We provide a hybrid method that incorporates the best features of both models in order to overcome these drawbacks. Our approach integrates KNN's ability to incorporate spatial information with Linear Regression's capability to model relationships between traditional features like property size and number of bedrooms.

Let $D = \{(x_i, y_i) \mid i = 1,...,n\}$represent the dataset, where $x_i = (l_i,f_i) \in R^d$ is a vector containing the features of the i-th house, with $l_i \in R^2$ representing the geospatial features (latitude and longitude), and $f_i \in R^{d-2}$ representing the traditional features. $y_i \in R$ is the house price. The goal is to predict the price $\hat{y}$ for a new house $x = (l, f)$.

For each house i in the dataset, calculate the Euclidean distance between the geospatial features l of the test house and those of the i^{th} house, denoted as l_i.

$$d_i = \|l - l_i\|$$

Sort the distances $\{d_i\}$ in ascending order and select the indices of the k smallest distances, denoted as $N_k = \{i_1, i_2, ..., i_k\}$, where k is the number of nearest neighbors.

Next, we extract the subset of data corresponding to the K-nearest neighbors.

$$\{(f_{ij}, y_{ij}) \mid j \in N_k\}$$

Here, f_{ij} represents the traditional features of the j^{th} nearest neighbor and y_{ij} is the corresponding house price.

We now have a subset of the original data that is located near point x. This subset of data is what we utilize to build the linear regression model. The following is the expression for the linear regression model:

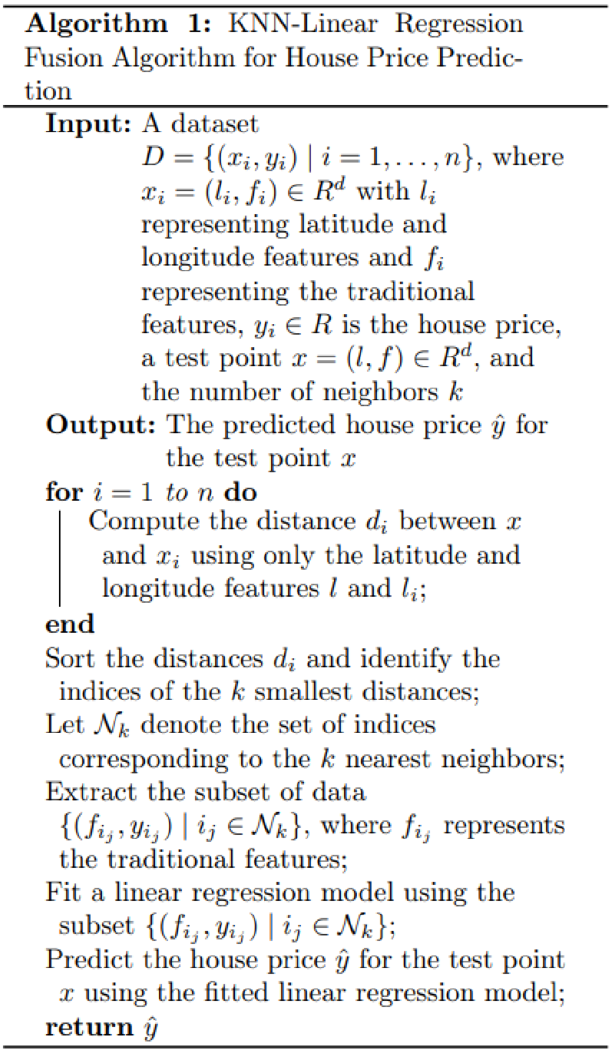
$$\hat{y} = \beta_0 + \sum_{m=1}^{d-2} \beta_m f_m$$

where f_m are the traditional features of the test property, and $\beta_0, \beta_1, ..., \beta_{d-2}$ are the regression coefficients estimated using the nearest neighbors' data.

The predicted house price for the test property x is obtained by applying the trained linear model to the traditional features f of the test property.

The pseudocode for the algorithm is outlined in Figure 1. Figure 2 displays the algorithm's flowchart. The algorithm consists of two main steps: the KNN step and the Linear Regression step.

This hybrid approach captures the spatial influence through KNN while modeling the price based on traditional features using Linear Regression. By first leveraging location-based neighbors and then applying regression on these localized data points, the algorithm balances spatial context with traditional house features to provide more accurate predictions.



```
Algorithm 1: KNN-Linear Regression Fusion Algorithm for House Price Prediction
Input: A dataset
        $D = \{(x_i, y_i) \mid i = 1, \ldots, n\}$, where
        $x_i = (l_i, f_i) \in R^d$ with $l_i$
        representing latitude and
        longitude features and $f_i$
        representing the traditional
        features, $y_i \in R$ is the house price,
        a test point $x = (l, f) \in R^d$, and
        the number of neighbors $k$
Output: The predicted house price $\hat{y}$ for
        the test point $x$
for $i = 1$ to $n$ do
    Compute the distance $d_i$ between $x$
      and $x_i$ using only the latitude and
      longitude features $l$ and $l_i$;
end
Sort the distances $d_i$ and identify the
  indices of the $k$ smallest distances;
Let $\mathcal{N}_k$ denote the set of indices
  corresponding to the $k$ nearest neighbors;
Extract the subset of data
  $\{(f_{i_j}, y_{i_j}) \mid i_j \in \mathcal{N}_k\}$, where $f_{i_j}$ represents
  the traditional features;
Fit a linear regression model using the
  subset $\{(f_{i_j}, y_{i_j}) \mid i_j \in \mathcal{N}_k\}$;
Predict the house price $\hat{y}$ for the test point
  $x$ using the fitted linear regression model;
return $\hat{y}$
```

Fig -1: Pseudocode for Algorithm

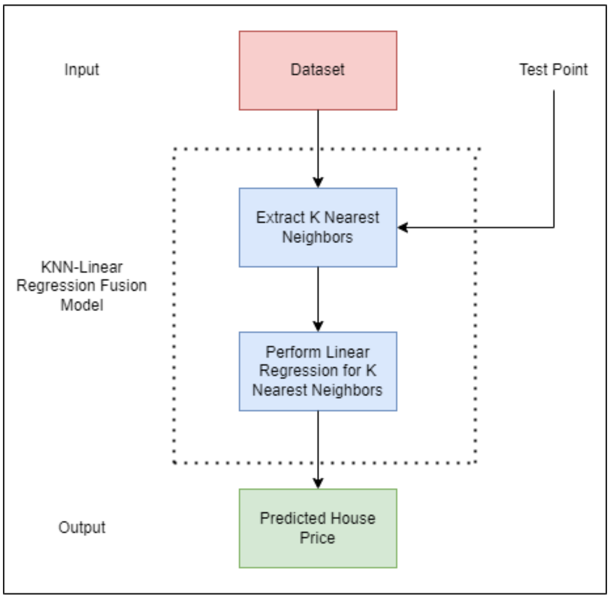

Input
Dataset
Test Point
Extract K Nearest Neighbors
KNN-Linear Regression Fusion Model
Perform Linear Regression for K Nearest Neighbors
Output
Predicted House Price

Fig -2: Flowchart

4. EXPERIMENTAL RESULTS

4.1 Experimental Setup

In this study, we compare the performance of three models: Linear Regression, K-Nearest Neighbors (KNN), and a hybrid KNN-linear regression model. The dataset used comprises 76,038 instances, which are divided into training, validation, and test sets. Specifically, the test and validation set each consist of 5% of the total dataset, that is 3,801 instances each. To ensure the reproducibility of our experiments, we set a fixed random state of 42, which allows for consistent sampling of the test and validation sets. For the KNN model, we select the hyperparameter KKK as 17. In the case of the hybrid KNN-linear regression model, the hyperparameter KKK is set to 201. The Linear Regression model utilizes the Ordinary Least Squares (OLS) method for estimating parameters. All the hyperparameters are tuned over the validation set, and the final results are reported based on the test set.

4.2 Results

Table 3 presents a comparative analysis of our hybrid model against Linear Regression and K-Nearest Neighbors (KNN) using three evaluation metrics: Mean Absolute Error, Mean Squared Error, and R-squared score.

Table -3: Comparison of models

	Linear Regression	KNN	Hybrid Model
MAE	0.7379	0.3738	0.4436
MSE	1.5353	0.7150	0.6057
R^2	0.5890	0.8086	0.8379

Chart 4 illustrates that KNN attains the lowest MAE, indicating the smallest average error. However, our hybrid model exhibits an MAE only marginally higher than KNN, demonstrating its competitive performance.

Chart 5 illustrates that our hybrid model significantly outperforms all other models in terms of MSE. The hybrid model achieves a lower MSE compared to both Linear Regression and KNN, highlighting its superior accuracy in error minimization.

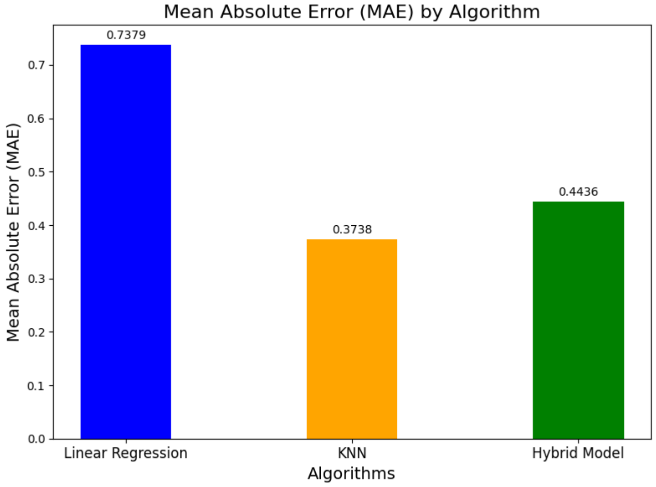


Chart -4: MAE Comparison of Models

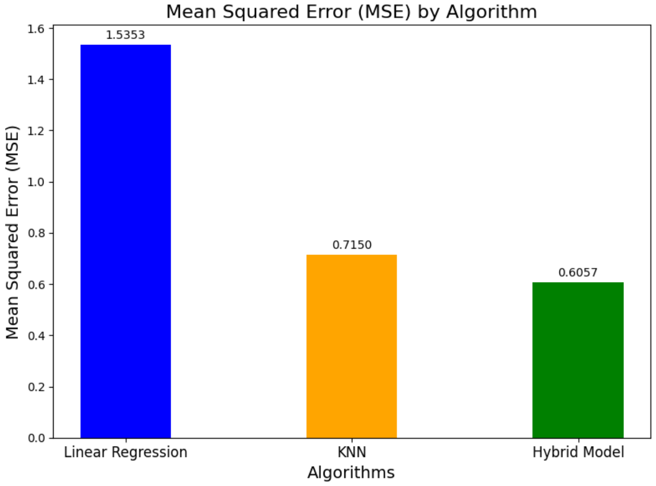


Chart -5: MSE Comparison of Models

Chart 6 reveals that our hybrid model has the highest R-Squared score among the models tested. This indicates that our model explains the greatest proportion of variance in house prices, thus providing the best fit to the data.

Our hybrid model demonstrates superior performance compared to both Linear Regression and KNN. Despite KNN having the lowest MAE, our hybrid model achieved the lowest MSE and highest R-Squared score. This indicates that the hybrid approach not only reduces prediction errors more effectively but also provides a better fit to the data. Thus, incorporating the advantages of both of the models allows us to enhance accuracy in house price prediction.

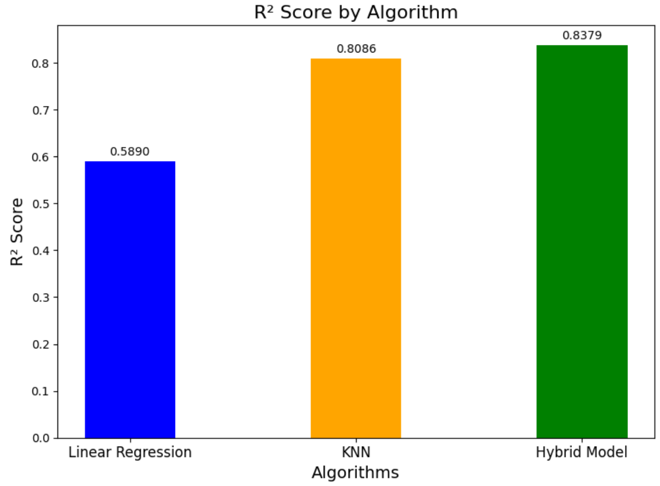


Chart -6: Comparison of R-Squared Score

5. CONCLUSIONS

In this study, we developed a new hybrid approach for predicting house prices by combining K-Nearest Neighbors (KNN) with Linear Regression. Our goal was to see if this combined method could provide better predictions than using either model alone, particularly with the inclusion of geospatial features like latitude and longitude.

Our results show that although KNN achieved the lowest Mean Absolute Error, our hybrid model outperformed it overall. It had the lowest Mean Squared Error and the highest R-squared score, meaning it was the most accurate and best at explaining the variability in house prices. This suggests that our hybrid method is more accurate at predicting house prices than using Linear Regression or KNN alone.

The success of our hybrid model highlights the importance of incorporating geospatial data into the prediction process. By combining KNN's strength in handling local data with Linear Regression's ability to model relationships between features, we created a model that takes advantage of both approaches.

In summary, our hybrid approach offers a new algorithm to predict house prices more accurately. Future research could build on this work by combining the advantages of more models or including additional types of data, such as economic indicators or market trends, to see if we can make even better predictions.

REFERENCES

[1] Mohd, T., Jamil, N.S., Johari, N., Abdullah, L., Masrom, S. (2020). An Overview of Real Estate Modelling Techniques for House Price Prediction. In: Kaur, N., Ahmad, M. (eds) Charting a Sustainable Future of ASEAN in Business and Social Sciences. Springer, Singapore. https://doi.org/10.1007/978-981-15-3859-9_28

[2] Guangliang Gao, Zhifeng Bao, Jie Cao, A. K. Qin, and Timos Sellis. 2022. Location-Centered House Price Prediction: A Multi-Task Learning Approach. ACM Trans. Intell. Syst. Technol. 13, 2, Article 32 (April 2022), 25 pages. https://doi.org/10.1145/3501806

[3] G.Naga Satish, Ch.V.Raghavendran, M.D.Sugnana Rao, Ch.Srinivasulu "House price prediction using machine learning". IJITEE, 2019.

[4] David Emmanuel Aniobi, Chukwuemeka Oluebube Ochuba, Saater Benedicta Nguideen "House price prediction: comparative analysis of regression-based machine learning algorithms", International Journal for Research in Applied Science & Engineering Technology (IJRASET), Volume 11 Issue X Oct 2023
doi: https://doi.org/10.22214/ijraset.2023.56232

[5] Sarkar Snigdha Sarathi Das, Mohammed Eunus Ali, Yuan-Fang Li, Yong-Bin Kang, Timos Sellis "Boosting House Price Predictions using Geo-Spatial Network Embedding", arXiv:2009.00254
doi: https://doi.org/10.48550/arXiv.2009.00254

BIOGRAPHIES

Prit Jignesh Kanadiya,
B. Tech. Computer Engineering,
Veermata Jijabai Technological Institute (VJTI), Mumbai

Prof. Pramila M. Chawan, is working as an Associate Professor in the Computer Engineering Department of VJTI, Mumbai. She has done her B.E.(Computer Engineering) and M.E.(Computer Engineering) from VJTI College of Engineering, Mumbai University. She has 28 years of teaching experience and has guided 85+ M. Tech. projects and 130+ B. Tech. projects. She has published 143 papers in the International Journals, 20 papers in the National/ International Conferences/ Symposiums. She has worked as an Organizing Committee member for 25 International Conferences and 5 AICTE/MHRD sponsored Workshops/ STTPs/FDPs. She has participated in 16 National/International

Conferences. Worked as Consulting Editor on – JEECER, JETR,JETMS, Technology Today, JAM&AER Engg. Today, The Tech. World Editor – Journals of ADR Reviewer -IJEF, Inderscience She has worked as NBA Coordinator of the Computer Engineering Department of VJTI for 5 years. She had written a proposal under TEQIP-I in June 2004 for 'Creating Central Computing Facility at VJTI'. Rs. Eight Crore were sanctioned by the World Bank under TEQIP-I on this proposal. Central Computing Facility was set up at VJTI through this fund which has played a key role in improving the teaching learning process at VJTI. Awarded by SIESRP with Innovative & Dedicated Educationalist Award Specialization : Computer Engineering & I.T. in 2020 AD Scientific Index Ranking (World Scientist and University Ranking 2022) – 2nd Rank- Best Scientist, VJTI Computer Science domain 1138th Rank- Best Scientist, Computer Science, India.