Grenze International Journal of Engineering and Technology, June Issue

Home Finder - GIS Enhanced Property Search with Google Maps

Vichuri Chandra Kumar[1], Lakkimsetti Dishasri[2], Bisai Reshma[3] and Shaik Siraaz Basha[4]
[1-4]Dept of Artificial Intelligence & Data Science Lakireddy Bali Reddy College of Engineering Mylavaram, India
Email: chandrakumar.vichuri@gmail.com, dishalakkimsetti@gmail.com, reshmabisai768@gmail.com
shaiksiraazbasha3272@gmail.com

***Abstract*—Home Finder is an innovative application designed to streamline the process of finding and listing houses for rent or sale. Finding the ideal home can be challenging; it requires extensive study, multiple visits, and careful consideration of numerous factors like price, location, and features. Home Finder is an interactive platform that uses Geographic Information System (GIS) data to expedite the search for rental or purchase properties in order to satisfy this need. The application allows users to filter homes based on a number of parameters, such as price range, accessibility to supermarkets, schools, and hospitals, pet policies, parking options, upkeep needs, and other preferences. Both prospective tenants and purchasers can easily identify properties that suit their needs thanks to Home Finder's user-friendly map design, which also makes it simple for property owners to post their properties for sale or rental.**

***Index Terms*— Home Finder, GIS, interactive map, house rental, real estate, property search.**

I. INTRODUCTION

The Home Finder project is an innovative, map-based application designed to meet the diverse needs of users and property owners, making the search for rental and purchase houses more convenient. By leveraging Geographic Information Systems (GIS), Home Finder provides an interactive interface that allows users to explore specific geographic locations, view real estate listings, and narrow their choices based on customizable filters. This tool offers a unified platform that simplifies property discovery, listing, and decision-making, benefiting renters, buyers, and property owners alike.

II. PROPOSED SYSTEM

The proposed model for the HomeFinder project leverages advanced web technologies and interactive mapping to provide a personalized, user-centric platform for property searching. The core of the model is an intelligent recommendation system designed to help users efficiently navigate the property market by considering their specific preferences and requirements. The system integrates multiple components, including user input, real-time property listings, and geographical data, to offer highly accurate and relevant property suggestions.
At its core, the model utilizes a filtering mechanism that takes into account key user preferences, such as price range, property type, number of rooms, and other lifestyle-specific factors like proximity to essential facilities (hospitals, schools, supermarkets), parking availability, and whether pets are allowed. These preferences serve as inputs to the system, which uses a rule-based filtering algorithm to generate a curated list of available properties.

Grenze ID: 01.GIJET.11.2.509_1

The goal is to provide each user with property options that best match their needs, thereby reducing the time spent browsing irrelevant listings.
The system architecture consists of several key components:

A. User Profile Creation

Upon registration, users are prompted to create personalized profiles that outline their specific preferences and requirements for their ideal home. This profile serves as the foundation for the property search, ensuring that each user receives tailored recommendations based on their inputs.

B. Interactive Map and Geolocation

The map-based feature allows users to explore different areas visually, zooming in on specific neighborhoods or regions where they are looking for properties. The system integrates geolocation data to filter available listings by proximity to the user's selected area. This map feature enhances the user experience, providing a dynamic and intuitive way to search for homes.

C. Real-Time Data Integration

To ensure that property listings on the HomeFinder platform remain accurate and up-to-date, the model incorporates realtime data updates. This dynamic feature continuously fetches and synchronizes information from multiple sources such as property owners, real estate agents, and external databases. By leveraging technologies such as WebSockets, Firebase, or realtime data APIs, the system ensures that all users whether they are browsing homes for sale or rent receive the most current listings, prices, and availability details.

D. Recommendation Engine

The recommendation engine is powered by a rule-based filtering algorithm that not only considers traditional factors like price and location but also factors in the user's lifestyle preferences and requirements. The system can suggest properties based on the proximity to schools, hospitals, transportation options, or other key amenities, which may be particularly important for families or individuals with specific needs.

E. Property Listings and Details

Each property listed on the platform includes detailed information such as the number of rooms, square footage, price, maintenance needs, parking availability, and nearby facilities. This information helps users make wellinformed decisions. Additionally, users can view images, property descriptions, and even schedule virtual or in-person tours, providing a comprehensive property search experience.

F. Admin Interface for Property Owners

For property owners, the system includes an admin interface that enables them to easily list, update, and manage their properties for sale or rent. This ensures that the platform remains dynamic and responsive to changes in the real estate market. Owners can also manage inquiries, respond to potential buyers or renters, and update property information in real time.

G. Dataset

The dataset used in the Home Finder app includes several important attributes for each property listing, capturing detailed information that helps users search for properties based on various preferences. Key identifiers like id, registered date, and permit expires are used to track the listing's unique identity and legal status. Property ownership details are captured with columns such as owner name, owner address, owner city, owner state, and owner zip, which provide contact information for property owners. Additionally, agent name, agent address, and agent zip identify the real estate agent responsible for managing the property.
In terms of property characteristics, the dataset includes building type, property type, building num, and unit num to provide users with information about the type and configuration of the property. Other attributes like beds, occup load, efficiency, and sleep room describe the size and suitability of the property for potential tenants or buyers. Geographical data, such as Latitude and Longitude, allows users to search for properties in specific locations, while sqfeet gives details on the property’s size in square feet. The dataset also includes important features related to property amenities and policies, such as cats allowed, dogs allowed, smoking allowed, wheelchair access, electric vehicle charge, comes furnished, laundry options, and parking options. These columns help users refine their search based on preferences for pets, accessibility, sustainability, and convenience. Additionally, image url and description provide visual and textual insights into the property, helping users make more informed decisions.

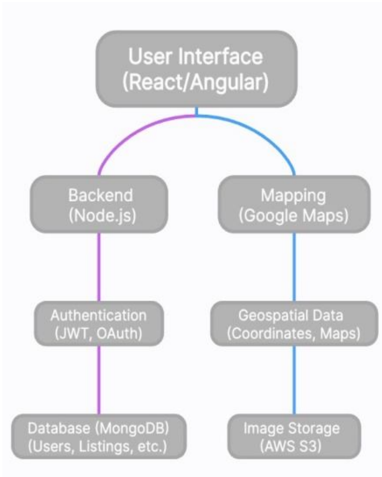

Fig 1 : Proposed Model Architecture

H. Experimental Setup

The HomeFinder project's experimental setup involves testing the application and its features in a development environment optimized for handling large datasets and providing smooth user interactions. The app is developed using tools and frameworks like Python, Django, and JavaScript for backend and frontend development, respectively. For geospatial features, Google Maps API or Leaflet.js is used to integrate the map-based search functionality. SQLite or PostgreSQL databases are employed to manage the large amount of property data, with proper indexing to enhance search speed and efficiency. To ensure the system's accuracy and performance, the dataset is divided into subsets that represent various property categories and geographical areas, helping fine-tune search functionalities. Cross-validation techniques are used to evaluate the effectiveness of different filters and features, ensuring that the search results match user preferences in terms of price, location, size, and amenities. The application's user interface (UI) is rigorously tested across different devices and browsers to ensure a consistent and seamless user experience. The goal is to ensure that the Home Finder app operates efficiently and provides accurate, real-time property search results, creating an optimal experience for users searching for homes to buy or rent.

I.Exploratory Data Analysis

The objective of Exploratory Data Analysis (EDA) in the Home Finder project is to better understand the dataset and uncover patterns that can enhance the property search experience for users. The EDA process begins with basic statistical analysis of the dataset, including calculating measures such as mean, median, standard deviation, and range for numerical columns like price, sqfeet, and beds. This helps in understanding the distribution and central tendencies of these features. For categorical variables, such as property type, cats allowed, and region, frequency distributions and proportions are calculated to identify the most common property types or features requested by users. To analyze the relationships between different features, correlation matrices are created for numerical attributes like price, sqfeet, beds, and baths to identify potential dependencies or correlations. For example, a strong correlation between sqfeet and price may indicate that larger properties are generally more expensive, which can inform users during their search.

J.Predictive Analysis

The Predictive Analysis phase in the HomeFinder project focuses on estimating the probability of property characteristics aligning with user preferences based on their search criteria. The goal is to predict the likelihood that a property will be a suitable match for a user by considering various factors such as price, sqfeet, beds, baths, and amenities like parking options, laundry facilities, and pet policies. By leveraging historical data about user behavior, search patterns, and property features, the model aims to provide personalized property recommendations. To develop these predictions, machine learning techniques, such as Artificial Neural Networks (ANN), are employed to identify patterns and correlations between the input variables and user preferences. The model is trained using a dataset that includes features such as property type, location, price, and amenities, along with user data such as location preferences and budget. By analyzing these factors, the ANN model can estimate the probability that a particular property will match a user's preferences, allowing the system to recommend the most relevant listings.

K.Model Description

The Home Finder model utilizes machine learning techniques, specifically regression models and recommendation algorithms, to predict the likelihood that a user will find a property that fits their criteria based on various input features such as price, sqfeet, beds, baths, location, and amenities. The goal is to create a personalized property search experience that estimates the most suitable homes based on the user's preferences and historical data from similar users. This predictive model helps users make informed decisions by suggesting properties that match their preferences, streamlining the home search process.

Architecture

The model comprises an input layer, hidden layers, and an output layer. The input layer takes various features such as price, sqfeet, beds, baths, region, and amenities. These features help the model understand the user's search criteria. The hidden layers progressively capture complex interactions between these variables, allowing the model to learn patterns that influence property suitability. The output layer provides a prediction for each property, indicating the likelihood that it matches the user's preferences, allowing for personalized recommendations.

Hidden Layers and Non-linearity

The hidden layers employ activation functions like Rectified Linear Unit (ReLU) to introduce non-linearity, enabling the model to learn complex relationships between features such as location, price, and sqfeet. Regularization techniques like dropout are used to avoid overfitting, ensuring the model generalizes well to unseen data. Batch normalization is applied to ensure stable training and efficient learning, preventing issues related to vanishing or exploding gradients.

Optimization and Loss Function

The model is optimized using Adam, an adaptive learning rate optimization algorithm, which speeds up convergence and enhances performance on multi-dimensional property datasets. The loss function used is Mean Squared Error (MSE), which measures the deviation of predicted property attributes (like price and sqfeet) from their true values, helping the model learn accurate predictions. The goal is to minimize the MSE during training to ensure the model’s recommendations are as close as possible to the actual property attributes.

Input

The model is fed with features from the property dataset, including numerical variables like price, sqfeet, beds, baths, and latitude and longitude for geographical positioning. Categorical features like property type, region, cats allowed, and laundry options are also included. These inputs allow the model to understand user preferences, such as desired location, size, and amenities, to provide accurate property recommendations. The input shape is set to accommodate all relevant features in the dataset.

Data Preprocessing and Scaling

Before feeding data into the model, it is pre processed to handle missing values, normalize numerical values (like price and sqfeet), and encode categorical features (like property type and region) into numerical representations. This ensures the model can process the data efficiently and accurately. Feature scaling is applied to standardize inputs like price and sqfeet to prevent any feature from dominating the learning process due to differences in scale.

Training the Property Recommendation Model

The model is trained using historical data on property listings and user preferences. By learning from past interactions, it captures patterns between the properties users are likely to engage with and their characteristics. The model undergoes several epochs of training, adjusting weights in each layer to minimize prediction errors. Regularization techniques like dropout and batch normalization are used to prevent overfitting and ensure the model can generalize to new user inputs. The Adam optimizer is used to speed up convergence and improve accuracy.

Predictions for Housing Market Trends and Property Value Fluctuations

Once the model is trained, it can predict the likelihood that a given property will match the user’s preferences based on input features like price, sqfeet, beds, baths, and amenities. These predictions allow the system to suggest properties that closely align with the user's criteria. For example, if a user is searching for properties in a specific location with a set price range and a certain number of bedrooms, the model will estimate the probability of properties being a match based on similar user profiles.

Output

The output of the model is a set of predictions for each property, indicating how likely it is to match the user's preferences. These predictions guide the Home Finder app in recommending properties to users that are most likely to meet their requirements. For each property, the output includes a probability score that helps rank properties by relevance. This score is used to tailor the user's property feed, ensuring that the most relevant listings are shown first. The model can also provide feedback, such as suggesting adjustments to preferences to broaden the search if the user isn't finding satisfactory results.

Model Validation

The model's predictions are validated by comparing them to real-world user interactions and survey data. Performance metrics like accuracy, precision, recall, and F1-score are used to evaluate the effectiveness of the property recommendations. By comparing predicted suitability scores with actual user preferences and interactions, the model's reliability is assessed, and adjustments are made to improve accuracy and recommendation quality.

III. RESULTS

The outcomes of the Home Finder model provide valuable insights into the suitability of properties based on user preferences and search criteria. By predicting the likelihood that a user will find a property matching their needs, the model helps users identify the best property options efficiently. The predictions are categorized into specific likelihood levels, such as "Low," "Moderate," and "High," based on how well a property matches a user's criteria like price, sqfeet, location, amenities, and other factors. These categories enable users to quickly assess whether a property is a good fit or requires further evaluation.

Additionally, using a validation set can provide further insight into the model's ability to generalize. By observing how the validation error behaves alongside the training error, you can ensure that the model is not simply memorizing the training data but rather learning patterns that apply more broadly, thus improving the accuracy and reliability of property recommendations for users.

IV. CONCLUSION

The successful implementation of the Home Finder model underscores its potential to revolutionize property search systems, paving the way for more intelligent and efficient platforms in the real estate industry. With further refinement and the inclusion of additional features, the model holds the potential to offer even more personalized, accurate, and timely recommendations, ultimately improving the property-hunting process for all users.

REFERENCES

[1] Desai, V., & Mehta, A. (2022). Improving property search using AI-based ranking algorithms. Journal of Artificial Intelligence and Property Management, 7(1), 45–53.

[2] Tiwari, P., & Joshi, R. (2021). A smart property search engine using AI and machine learning algorithms. International Journal of Computer Science and Real Estate, 5(4), 62–70.

[3] Singh, J., & Agarwal, A. (2020). Prediction of real estate property prices using regression and neural networks. Journal of Computational Intelligence in Real Estate, 13(1), 50–59.

[4] Patel, K., & Sinha, R. (2021). Real estate property valuation with machine learning: An empirical study. Journal of Real Estate Technology and Analytics, 8(2), 112–121.

[5] Zhang, C., & Lin, Q. (2020). Housing price prediction using ensemble learning techniques. International Journal of Housing Market and Analysis, 12(3), 325–340.

[6] Zhang, H., Chen, L., & Zheng, Y. (2020). A real estate recommendation system based on collaborative filtering and property features. Proceedings of the International Conference on Computer and Communications, 141–145.

[7] Li, X., Wang, H., & Wang, J. (2019). Real estate price prediction using machine learning algorithms: A case study. Journal of Artificial Intelligence in Real Estate, 11(2), 80–89.

[8] Yang, M., Liu, W., & Zhang, T. (2021). Predicting housing prices with machine learning models. Journal of Computational Intelligence in Real Estate, 15(3), 200–210.

[9] Nguyen, A. T., & Nguyen, D. M. T. (2022). Data mining and machine learning approaches for real estate price prediction. Journal of Big Data in Real Estate, 9(1), 1–15.

[10] Huang, Y., Chen, Z., & Xu, L. (2020). Property price prediction with deep learning: A real estate case study. Proceedings of the International Conference on Artificial Intelligence & Machine Learning, 78–88.

[11] Kumar, A., & Sharma, V. (2019). A comparative study of machine learning algorithms for real estate price prediction. International Journal of Applied Artificial Intelligence, 28(4), 56–63.

[12] Gupta, A., & Sharma, R. (2021). AI-driven property search: Implementing machine learning in real estate platforms. Proceedings of the International Conference on Smart City and Advanced ICTs, 45–49.
[13] Sen, R., & Rao, A. (2020). Machine learning models for real estate investment analysis. Real Estate Data Science Journal, 4(2), 120–135.
[14] Patel, M., & Mehta, R. (2020). Feature engineering and price prediction models in real estate. Journal of Computational Analytics in Property Markets, 6(1), 66–73.
[15] Iyer, P., & Sharma, R. (2021). Optimizing property search with AI: Insights and future trends. International Journal of Machine Learning and Real Estate, 15, 190–198.