IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 26, Issue 5, Ser. 1 (Sept. – Oct. 2024), PP 08-12
www.iosrjournals.org

House Rental Finding System Using Laravel, Vue.Js, And PHP Machine Learning Library

Mohammad Shohel Rana[1, *] Saleh Ibne Omar [2,] Md. Fahim Muntashir Pranta[3]
[1]*Department Of Computer Science And Engineering,*
Southeast University, Dhaka, Bangladesh

Abstract
The use of technology in all parts of our lives has greatly increased in recent years, nowadays, people tend to search for everything online before they make decisions. However, looking for rental houses online is not common in Bangladesh, it is not because people do not want to, but because there are not many online platforms focusing on residential rentals. For the mentioned reason we developed a house rental finding web application using Laravel, Vue.js, and PHP-ML library. Laravel handles the backend whereas Vue.js manages the frontend, we utilized PHP Machine Learning Library's Linear Regression and K-Nearest Neighbours (K-NN) algorithm to build a machine learning model for the search system. The goal is to make it easier to find rental properties. Our system gives personalized suggestions based on user inputs, like location, rental price, property size, and the number of rooms and bathrooms. Our application aims to make searching for rental houses a better experience and more efficient.

Keywords: *K-Nearest Neighbors (K-NN), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Machine Learning (ML), JavaScript (JS), Hypertext Preprocessor (PHP)*

Date of Submission: 26-08-2024 Date of Acceptance: 06-09-2024

I. Introduction

The increasing demand for house rental finding systems in the real estate industry has led to the exploration of advanced technologies to enhance the process of searching for and recommending rental properties. This paper presents a web application developed using the Laravel and Vue.js frameworks, incorporating a machine learning library of the PHP programming language.

The Laravel framework was chosen due to its robustness and extensive features for web development, it provides a solid foundation for building complex web applications, facilitating the implementation of various functionalities required in our house rental finding system. Additionally, Vue.js was selected as the frontend framework because of its speed and lightweight nature.

Linear Regression was employed to predict rental prices based on relevant features such as property size, location, rooms, and bathrooms.

Furthermore, the K-Nearest Neighbors algorithm was applied after obtaining the predicted price from the linear regression model to provide personalized property recommendations. By considering the similarities between properties and user preferences, the system generates suggestions, enhancing the user experience and increasing the likelihood of finding suitable rental options.

The evaluation of the web application involved the use of real-world datasets and simulated scenarios. Performance metrics were applied to assess the accuracy and effectiveness of the machine learning algorithm used.

Overall, this paper contributes to the understanding of utilizing machine learning within the PHP programming language and web development frameworks like Laravel and Vue.js. The developed system offers a practical solution for individuals seeking rental properties, enabling them to make informed decisions based on accurate rental price predictions and personalized recommendations.

II. Methodology

This research was conducted to develop a web application using different tools and technologies (A) that allows users to search for properties based on their specific requirements, such as location, budget, and amenities. We have used a mixed-method approach for this research, comprising both qualitative and quantitative methods. The data required for the development of the system has been collected from various sources, including rental websites, and online user questionnaire submissions (B). The dataset has been

prepared (C) before being used to train the Linear Regression and K-NN algorithm models, we also followed different evaluation metrics (D) to evaluate the accuracy of our models.

Tools and Technologies

To develop our application, we used Laravel as the backend framework which is also a PHP-based web framework, we used Vue.js as the frontend framework, to design the user interface we used Bootstrap, and to do all the machine learning work we used PHP Machine learning library (open source), and as for the database we used MySQL.

Data Collection

'Google Forms' was used to conduct an online survey to collect data. Questions such as the type of apartment, number of rooms, number of washrooms, which floor they are staying on, rent, and location were asked. The collected data was used to train Linear Regression and K-NN algorithm models to predict possible rent ads based on user input.

Dataset Preparation

The dataset was cleaned by discarding missing information and prepared with labels since our algorithms work with labeled data. The dataset was randomly split into a training set (80% of the data) and a test set (20% of the data) for evaluation purposes.

Evaluation Metrics

To assess the accuracy of the Linear Regression and K-NN models, we have used different evaluation metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Classification Accuracy Calculation.

III. System Design

Use Case

The system use case consists of three separate user roles: Admin, User, and Guest. Figure 1 shows the use case of admins, and Figure 2 shows the use case of users and guests.

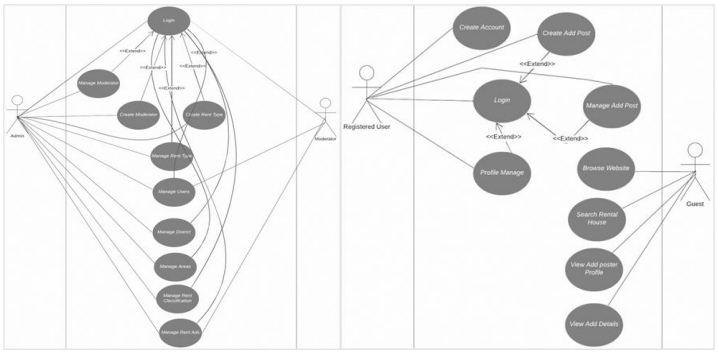

Figure 1: Use Case Diagram of Admin
Figure 2: Use Case Diagram of User and Guest

Entity Relationship

'MySQL' was used as the database of the system, it is a relational database, and creating an Entity Relationship Diagram is a must.

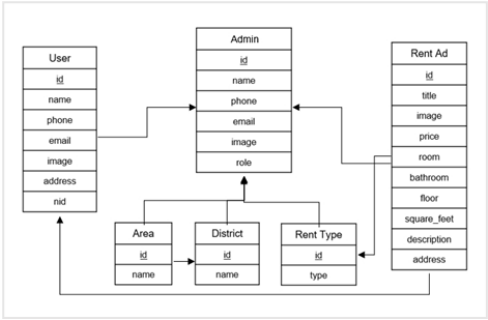

Figure 3: Entity Relationship Diagram

Search Algorithm

The search algorithm has three stages. In the first stage, it predicts the price using the Linear Regression model with inputs from the user, such as the number of rooms, bathrooms, square feet, and location. Then, the algorithm classifies price ranges using the K-NN model, the K-NN model then takes the predicted price and past user inputs as its input. In the second stage, using the predicted nearest categories, the algorithm searches for the closest price range against the predicted price from the first stage. If the predicted price does not fall into any of the predicted price categories, the algorithm then proceeds to the third stage to calculate a relevant price range using the predicted price from the first stage. At the end, the algorithm forms its final price range to create a search query.

First Stage

1. Start
2. Take rooms, bathrooms, square feet, and location as user input.
3. Run Linear Regression using user inputs to get the predicted price
4. Store predicted price as p_price
5. Run K-NN using p_price and other user inputs to get predicted classifiers
6. Store the predicted classifier
7. Declare min_distance = INT_MIN, closest_range = NULL, index = 0

Second Stage

8. Loop: index < classifier.size
9. Declare distance = 0
10. If p_price < classifier[index].start go to step 11, else go to step 12
11. distance = classifier[index].start - p_price
12. distance = p_price - classifier[index].end
13. If distance < min distance go to step 14, else go to step 15
14. min_distance = distance and closest_range = classifier[index]
15. Increment index and go to step 8

Third Stage

16. If p_price is not in closest_range then range difference = closest.to - closest.start and go to step 17 else go to step 24
17. If p_price < closest.start AND closest.start - p_price >= range diff go to step 18 else go to step 19
18. closest.to = closest.end, closest.end = p_price
19. If p_price > closest.end AND p_price - closest.end >= range diff go to step 20 else go to step 21
20. closest.start = closest.to, closest.to = p_price
21. If p_price < closest.start go to step 22, else go to step 23
22. closest.start = closest.start - (closest.start - p_price)
23. closest.end = closest.end + (p_price - closest.end)
24. End

The time complexity of the 'Second Stage' is O(K), where the ideal value of K ranges from 3 to 7, and K is the value of nearest neighbors which is the number of closest classifiers we get from the K-NN model.

User Interface

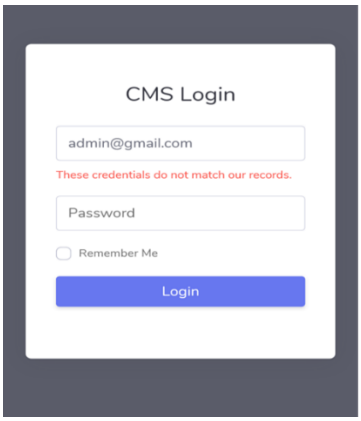

Figure 4: Admin Login Panel

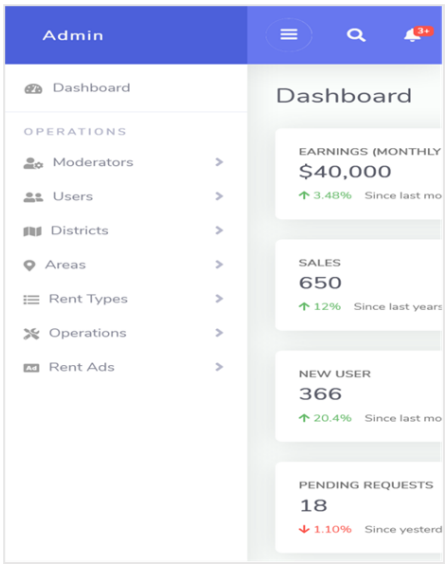

Figure 5: Admin Panel Navigation

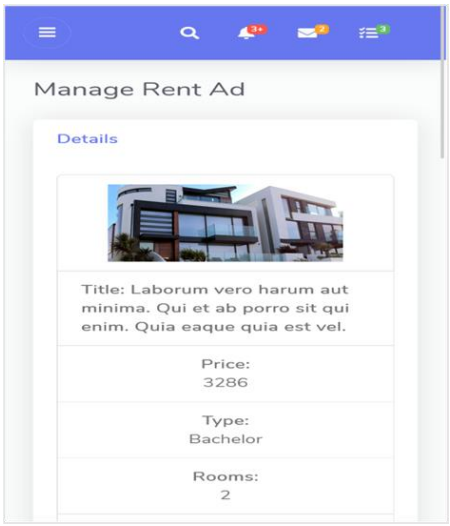

Figure 6: Posted Rent Ad Management for Admin

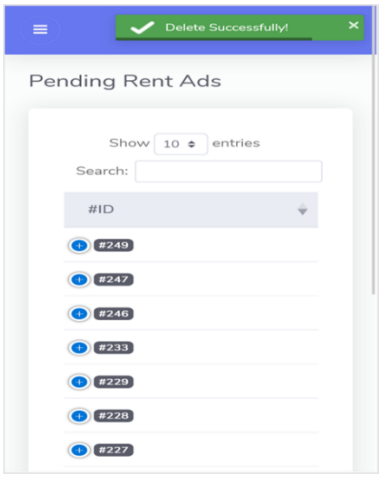

Figure 7: Posted Rent Ad Review Panel for Admin

IV. Findings And Results

Our research was focused on developing a web application that utilizes frameworks like Laravel, Vue.js, and PHP Machine Learning Library (A). While conducting this research, we observed results (B) of

using Linear Regression, and K-NN algorithms and evaluated their accuracy using different metrics such as MAE, RMSE, and Classification Accuracy Calculation.

PHP for Machine Learning

When it comes to machine learning people are drawn to use languages like Python and its libraries, PHP may not even come to their mind since it is well known for its web development purposes, but we have found that there are machine learning libraries based on PHP programming language such as PHP-ML Machine learning library and Rubix-ML, these libraries are easy to use and we have used PHP-ML machine learning library to develop our searching system of our application, so it is a great finding of our study and we believe this will help PHP developers to use machine learning approaches whenever required for better optimization of PHP based applications.

Result and Evaluation

The accuracy comparison between the Linear Regression model and the K-NN model is shown in Figure 14 below. The figure visualizes a pie chart to display the accuracy in percentage (Y-axis) and the number of test cases (X-axis).

Through the analysis of datasets comprising 25, 50, and 100 apartments, we evaluated the performance of our system. Using Linear Regression, we successfully predicted rental prices based on various property features. The evaluation yielded promising results, demonstrating a Mean Absolute Error (MAE) of approximately 8,421.96 for 25 apartments, 9,500 for 50 apartments, and 10,200 for 100 apartments. The Root Mean Squared Error (RMSE) was approximately 10,767.1 for 25 apartments, 11,600 for 50 apartments, and 12,100 for 100 apartments. These metrics indicated reasonably accurate rental price predictions.

To further enhance the accuracy and recommendation capabilities of our system, we incorporated the K-Nearest Neighbors (K-NN) algorithm. By leveraging the characteristics of similar apartments, the K-NN algorithm provided valuable recommendations to users, matching their preferences with suitable rental properties. The accuracy of the K-NN algorithm was approximately 92.00% for 25 apartments, 92.37% for 50 apartments, and 95.72% for 100 apartments. These results showcased the system's ability to effectively match users with rental properties that closely align with their needs and preferences.

V. Conclusion

Through our research, we developed a House Rental Finding System using the Laravel framework and leveraged two machine learning algorithms: Linear Regression and K-nearest neighbors (K-NN), to enhance the accuracy and efficiency of the system. Our goal was to create a reliable and user-friendly platform that assists users in finding suitable rental properties. Through the evaluation of datasets containing 25, 50, and 100 apartments, we demonstrated the system's ability to make accurate rental price predictions and provide efficient property recommendations. Future work may involve expanding the dataset, refining the algorithms, and conducting user studies to further improve the system's accuracy and usability. The findings of this research contribute valuable insights into leveraging PHP machine learning libraries and web development frameworks to create effective and user-friendly web platforms. Since there are no dedicated online platforms in Bangladesh for searching for houses or apartments, our research can be of great help in this particular domain.

References

[1] Afzal, S., Rouf, T., Qadir, S., & Shah, S. (2021). Online Rental Housing. “Journal Of Emerging Technologies And Innovative Research (Jetir)”, 8(11). Retrieved From Https://Www.Jetir.Org

[2] Bproperty. (2018). “Bproperty.” Retrieved From Https://Www.Bproperty.Com/Blog/Successful-Launch-Bproperty-Marketplace-Banani
G. Gommans, H. P., Njiru, G. M., & Owange, A. N. (2014). Rental House Management System. “International Journal Of Scientific And Research Publications”, 4(11). Retrieved From Www.Ijsrp.Org

[3] L. Laravel. (2023). “Laravel - The Php Framework For Web Artisans.” Retrieved From Https://Laravel.Com

[4] N. Neehal, N. (2017). “Machine Learning Algorithm (1st Ed.).” Packt Publishing

[5] P. Php-Ml. (2021). “Php-Ml: Machine Learning Library For Php.” Retrieved From Https://Php-Ml.Org

[6] S. Satapathy, S. M., Jhaveria, R., Khanna, U., & Dwivedi, A. K. (2014). Smart Rent Portal Using Recommendation System Visualized By Augmented Reality. “Procedia Computer Science”, 197-206. Retrieved From Www.Elsevier.Com/Locate/Procedia

[7] S. Shibwabo, B. (2018). System For Recommending Rental Properties. “International Journal Of Science And Research (Ijsr)”, 6(7). Retrieved From Https://Www.Researchgate.Net/Publication/322652906_A_System_For_Recommending_Rental_Properties

[8] Voumick, D., Deb, P., Sutradhar, S., & Khan, M. M. (2021). Development Of Online Based Smart House Renting Web Application. “Journal Of Software Engineering And Applications”, 14, 312-328. Retrieved From Https://Www.Scirp.Org/Journal/Jsea

[9] V. Vue.Js. (2023). “Vue.Js - The Progressive Javascript Framework.” Retrieved From Https://Vuejs.Org

[10] W. Wei, J., Chu, X., Sun, X.-Y., Xu, K., Deng, H.-X., Chen, J., Wei, Z., & Lei, M. (2019). Machine Learning In Materials Science. *Information*, 10(9), Article 275. Https://Doi.Org/10.1002/Inf2.12028