Vol. 21, No. 1, 2024, ISSN 15965538, pp 1-8

Journal of Civil and Environmental Systems Engineering

Department of Civil Engineering, University of Benin, Nigeria

Journal homepage: https://j-cese.com/

AN AI-DRIVEN CONSTRUCTION COST ESTIMATION MOBILE APPLICATION

[a]Olaye, E., [a]Ehizuenlen, E.P, and [a]Oaihimire, O. A.

***[a]Department of Computer Engineering, Faculty of Engineering, University of Benin, Benin City, Edo State, Nigeria*.**

Abstract

This paper presents the software development of a mobile application for accurate estimation of the cost of construction materials and personnel. The system was developed to mitigate the discrepancy that exists between estimated and actual quantities; and improve cost management during building construction. Using an iterative and incremental approach, mobile application development consists of three major components namely; the frontend, the backend, and the machine learning (ML) model. The front end was built with the DART programming language (Flutter framework), providing users with an intuitive interface for capturing essential parameters required for accurate cost estimation. The backend was developed using Python programming language (Flask micro Web framework) and the entire codebase was hosted on the Render platform, thereby eliminating the need for a traditional database. A Machine Learning (ML) model was trained using the Random Forest Regressor algorithm from the sklearn.ensemble module. The model was integrated into the mobile application designed for cost estimation through the mechanism of retrieving and utilizing a pickle file located in the same directory as the main code. User input, such as cost-related data, is collected through the application's interface. Instead of using a static testing dataset, this real-time user input serves as a continuous evaluation mechanism for our model. The developed mobile application was shown to generate accurate building cost/material estimates. An accuracy score of 74.3% using the Random Forest Regressor algorithm was achieved for the ML model deployed on the mobile application.

Keywords: *Mobile Application, Building Cost, Mobile App, Building Construction, Cost Estimation*

1. Introduction

A building is a constructed structure designed to provide shelter, house activities and accommodate human needs (Ali, 2023). The main purpose of constructing a building is primarily to provide shelter from weather, security, and living space, to store belongings, and to comfortably live and work. Before a building is constructed, planning is necessary to ensure that it is not left halfway into the project but completed. Planning in this instance entails structural, financial, and resources, etc. However, financial and resource planning or cost management is deemed to be the most important as they determine if the building project can be completed or not, and also determine the overall timeframe for which the building project can be completed. The process of cost management involves planning, estimating, budgeting, and controlling costs so that the said building can be completed within the approved budget (Haruna *et al.,* 2017).

Building cost planning is the costing of the different parts of a building project and generating a predetermined sum of the overall project which in turn is used to advise the client as regards the cost of the project. The total cost of construction in a normal circumstance includes material, labour, site overheads, equipment/plant, head office cost, and profit (Eshofonie, 2008). The cost estimations for buildings are mostly done by a quantity surveyor who will inspect the building project. Generally, it has been observed that building costs done manually are usually faulted with exaggerated figures mostly for selfish interest or gains. Most times, these estimations by humans are done without taking into consideration the rising cost of goods and services which are usually experienced because of inflation.

The introduction of artificial intelligence (AI) in modern times has brought about the ability of computers to perform some tasks such as planning that were previously done alone by humans. These computers can learn from human inputs, from their immediate environment, and make predictions such as stock prices, prices of goods and services, etc. based on learned inputs. Some of the popular AI tools include advanced web search engines such as Google, recommendation systems used by YouTube, Amazon, and Netflix, understanding human speech such as Siri and Alexa, self-driven cars such as Tesla, automated decision-making making and competing at the highest level in strategic game systems such as Chess and Go (Google, 2006). AI-powered applications have also been used to predict the cost of building projects which tends to be free from human bias exaggeration, and personal interest with a higher probability of being more accurate than the human prediction of building projects.

A few researchers have considered employing mobile applications in the cost of building projects. Adedeji *et al.*, (2017) in their work developed an online Builder's estimating application for construction waste reduction. The study utilized the use of a questionnaire and designed an offline estimating app. The data obtained from fifty (50) selected professional Builders were analyzed using SPPS v. 2 considering statistical tools such as frequencies, percentages, mean scores, and Krustai Wallis H test. The builders' offline estimating app was developed using Javascript language. Results proved that the estimating app was able to perform estimation for selected building materials on site in the required quantities. Abdullahi *et al.*, 2019, present the design and implementation of a web-based tender evaluation system for Nigerian public sector tendering. The system design was based on fuzzy multi-attribute group decision-making techniques using object-oriented methodologies and unified modeling language. The validation result showed that the system has an average rating of 74% accuracy and accurate modeling of the existing manual tendering domain and an average rating of 67.6% for its potential to enhance the proficiency of public sector tendering in Nigeria.

In yet another work, Adedeji *et al.*, (2019) developed an Android-based estimating app for builders on construction sites with the view to increase the accuracy of estimated building materials. The study involved two stages namely, a survey to ascertain existing practices and the development of the Android application. The survey research was conducted on Builders in Lagos state, Nigeria and Mean scores and correlation matrix were used for the descriptive and inferential statistics. The Android estimation mobile application for Builders was developed using Android Studio and Java programming language. Results showed that the Android-based estimation app was developed to estimate concrete works, sandcrete blocks, ceramic tiles, and mortar during the construction process. In 2023, Bhavyashree *et al.(2023)*, developed a mobile application for cost prediction of residential buildings utilizing Android studio, Java, and centreline method which enables the house owners to assess the value of their properties and potential buyers to make informed decisions on the purchase of properties. The developed mobile application considers factors such as property size, dimensions, amenities, and market trends to generate reliable estimates. The result shows that the developed mobile application promotes transparency and fairness in quantity take-off by promoting objective information and reducing reliance on subjective opinions.

From these studies, there is a need to incorporate the use of intelligent mobile applications in the process of building construction to eliminate the issues prevalent with employing manual cost estimation processes. The current study presents a significant advancement in machine learning model evaluation within mobile building cost estimation applications by leveraging real-time user input for continuous assessment.

2.0 Methodology

The methodology involves the design of the mobile application as well as the algorithms required for its proper functioning. The architecture of the system includes:

i. Frontend
ii. Backend
iii. ML Model

An iterative and incremental approach was employed for the design methodology. This approach involves iteratively training multiple models with different configurations to identify the best-performing model thus allowing for experimentation, evaluation, and improvement of the models over time. The incremental approach incrementally builds and refines the components that are required for saving the model and API endpoint setup.

The system architecture was designed such that the user can interact with the frontend application and the backend application without going through any API gateway or load balancer. The backend was built using Python Flask micro Web framework while the frontend was built using Flutter framework with a successful communication via API endpoint. The mobile application was designed such that as the HTTP requests made from the Flutter application, data is being sent to the backend for processing using a trained machine learning model. The backend returns a response in JSON format while allowing the Flutter app to display the predicted values and perform actions that are based on the received data. Table 1 outlines the software and programming languages used in mobile application design.

Table 1: Software/Languages Employed in the Design

S/N	Software/Language	Use
1.	Flutter Framework	For cross-platformmobile development
2.	DART	Used with Flutter Framework
3.	Python Programming Language	For building the backend
4.	Python Flask	Used to develop web application
5.	Python Interpreter	Interpreter software for Python
6	Visual Studio Editor	For developing applications in different languages

2.1 Front End Design

In designing the front end, DART programming language in Visual Studio Code was used alongside the Flutter framework. The front end comprises several screens as shown in Figure 2. The screens include: the splash screen that shows immediately the mobile application is running, the onboarding screen that gives short details of the mobile application, the landing page screen which is the main screen of the mobile application where users put in details of their building structural needs, and lastly, a details page screen that shows the result of the cost computation of the data inputted in the landing page screen.

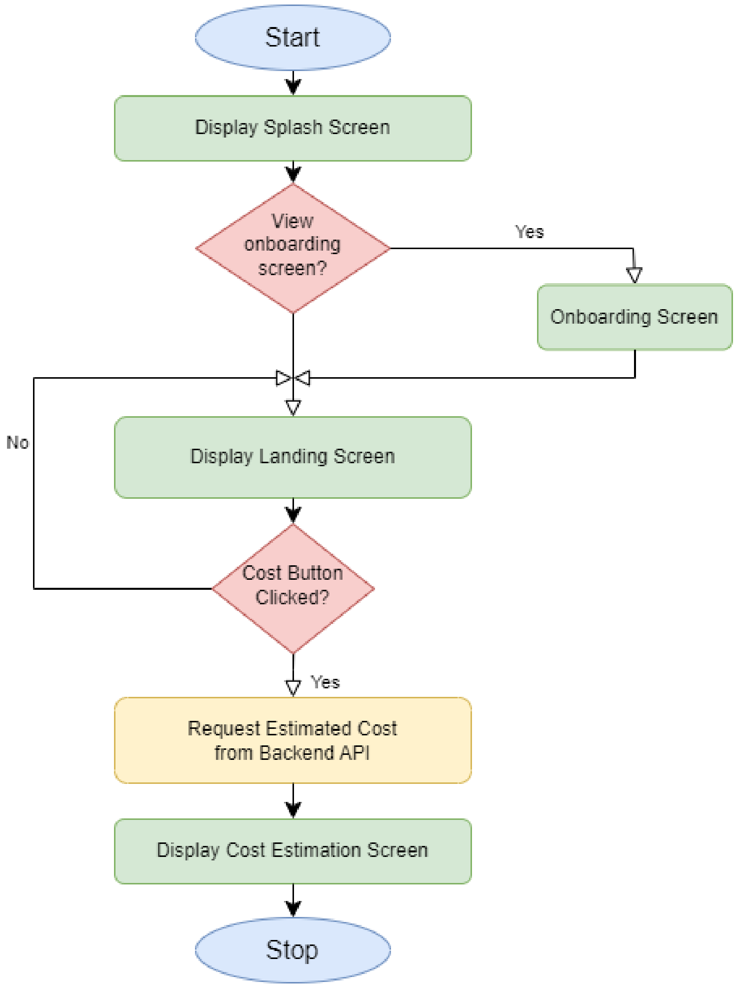

Figure 1: Flowchart for the Front End

2.2 Back End Design

The backend design was achieved using Flask a lightweight web framework for Python. Flask was used to configure the API endpoint, handle user requests, and generate responses in communication with the Flutter front end. The flask framework offers a high level of simplicity and flexibility, thus making it suitable for the backend implementation of the system. The backend operation is divided into four phases as shown in Figure 2. The phases include model training and saving, API endpoint configuration, request handling and data processing, and response generation.

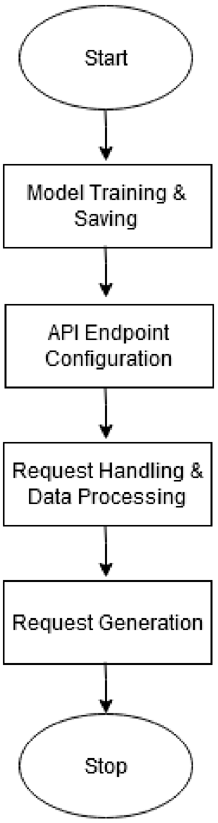

Figure 2: Flowchart for the Backend

2.3 Model Development

This process involved training a machine learning model using the Random Forest Regressor algorithm from the sklearn.ensemble module. The model training process followed an iterative approach, where multiple models were trained with different configurations. The traditional approach of splitting the dataset into training and testing sets is modified. Since the mobile application allows users to input new data for cost estimation, the model is evaluated using this real-time user input obtained from the application's interface. This approach ensures that the model's performance is continuously assessed on new, unseen data, allowing for a realistic evaluation of its generalization capabilities. The performance of each model is evaluated using a scoring metric. The best-performing model, which achieves the highest score, is selected for further use. Once the optimal model is determined, it is saved using the Pickle library. Serializing the model into a Pickle file allows for easy storage and future reuse without the need for retraining. Figure 3 shows a screenshot of the data used to train the model. Figure 4 shows the screenshot for the model accuracy score analysis.

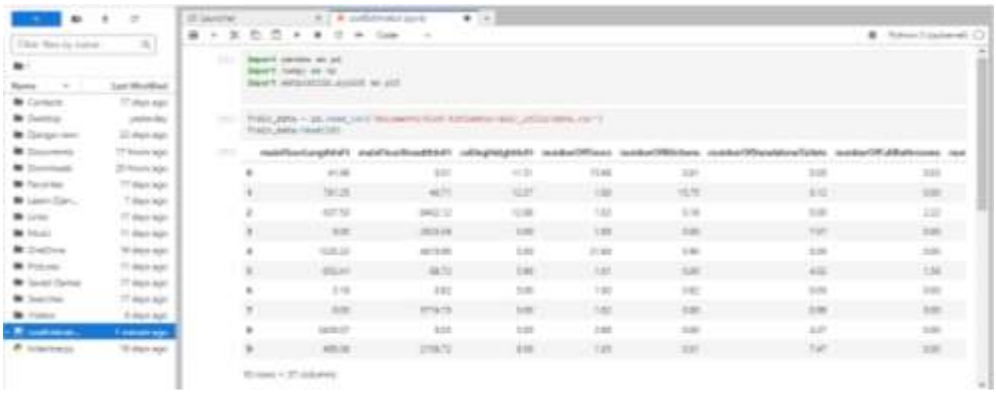

Figure 3: Data for Training the Model

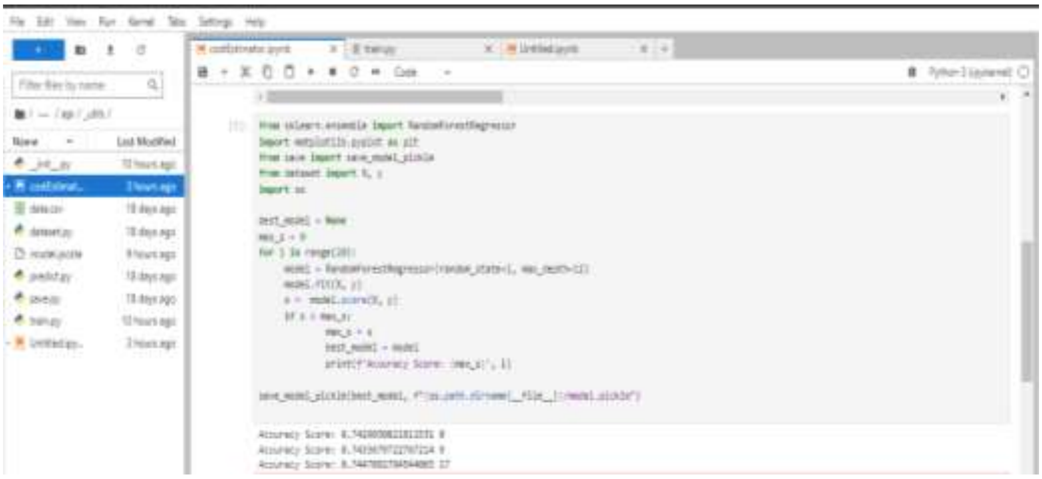

Figure 4: Model Accuracy Score Analysis

2.4 API Endpoint Configuration

To establish communication between the backend and the front end, an API endpoint is configured. Flask, a lightweight web framework for Python, was employed to define the API endpoint route. This route, such as '/predict', serves as the entry point for receiving HTTP requests from the frontend app. The API endpoint configuration includes handling different types of HTTP requests, such as GET and POST, to accommodate various functionalities and data exchange requirements between the front end and back end.

2.5 Request Handling and Data Processing

When an HTTP request is received at the API endpoint, the backends' code in index.py handles the request. The backend extracts the necessary data from the request payload, which typically contains input parameters or data required for processing. The extracted data is validated and preprocessed as needed. For instance, data may undergo data type conversion, normalization, or feature engineering to ensure compatibility with the machine learning model. The preprocessed data is then utilized to perform computations or predictions using the trained model. This step involves passing the data through the model and generating the desired output.

2.6 Response Generation

Once the computations or predictions are performed, the backend generates a response that contains the results. The response was formatted in JSON (JavaScript Object Notation) to ensure a consistent structure can easily be consumed. It includes relevant information such as predicted values, computed metrics, or any other output deemed necessary. The response is sent back to the frontend as the HTTP response, allowing the frontend app to parse and extract the desired information. The response generation also involves handling potential errors or exceptions to provide meaningful error messages or appropriate status codes, ensuring a smooth communication flow between the backend and the front end.

2.7 Performance Evaluation

The performance of the developed model is assessed using various evaluation metrics. The primary metric used is the R-squared (coefficient of determination), which measures the proportion of the variance in the target variable that can be explained by the input features. Additionally, other metrics such as mean absolute error (MAE) and root mean squared error (RMSE) are employed to provide a comprehensive evaluation of the model's performance. The results are analyzed to determine its effectiveness in accurately estimating building materials and costs.

3. Results and Discussions

The developed mobile application was demonstrated to be functional and usable as presented in Figure 5. The simple user interface makes it easy for stakeholders and construction professionals to visualize the mobile application's interface and understand how it assists in estimating building materials and construction costs.

Figure 5: Developed mobile application (a) Cost Parameter Page; (b)Breakdown of Cost Page; (c) Cost Estimate Page

The developed ML model's performance was evaluated using R-squared score, MAE, and RMSE values. Results from the model accuracy score analysis show that the best model has an accuracy of 0.7447 (74.47%). This accuracy score indicates that the model can capture and learn the underlying patterns and relationships within the dataset, resulting in valuable cost estimates for construction projects. Our results demonstrate the effectiveness of this approach in assessing the model's performance under real-world conditions. By continuously evaluating the model with new, unseen data from user inputs, we gain valuable insights into its ability to generalize and adapt to dynamic data. However, it is essential to consider the limitations and potential sources of errors in the model. Factors such as data quality, feature engineering techniques, and the representativeness of the training dataset can influence the accuracy and generalizability of the model's predictions. Continuous refinement and optimization efforts are necessary to address these limitations and improve the model's performance further.

4. Conclusion

The integration of the Flutter frontend with the backend architecture provides a scalable and adaptable solution for construction professionals and stakeholders. The practicality of using the mobile application for cost estimation can lead to improved decision-making, resource allocation, and overall project success in the construction industry. Furthermore, the potential impact of the developed model on the construction industry should be considered. Accurate cost estimation plays a crucial role in cost-effective and efficient construction projects. By providing reliable estimates of building materials and construction costs, the developed model has the potential to significantly enhance cost management and decision-making processes.

References

Afolabi Adedeji, Abimbola Mayowa, and Ojelabi Rapheal (2019): Android-Based Estimating App for Builders on Construction Sites. International Journal of Mechanical Engineering and Technology, Vol. 10, No. 01, January 2019, pp. 1942 – 1951.

Bello Abdullahi, Yahaya M. Ibrahim, Ahmed D. Ibrahim, Kabir Bala (2019): Developing a Web - Based Tender Evaluation System Based on Fuzzy Multi-Attributes Group Decision Making for Nigerian Public Sector Tendering. World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering. Vol. 13, No. 7, pp. 353 – 361.

Afolabi, Adedeji; Fagbenle, Olabosipo; Tunji-Olayeni, Patience and Abimbola, Olumayowa (2017): Development of an on-site Builder's estimating app for construction waste reduction. Development of an On-site Builder's estimating App for Construction Waste Reduction. International Conference on Computing Networking and Informatics. Conference Paper, pp. 1 -10.

Bhavyashree H.D, Reetha raj S, Rekha M3, DhanushKumar H.L, Chandandev M, Ravi Kiran L (2023): Cost Prediction of Residential Buildings, International Advanced Research Journal in Science, Engineering, and Technology. Vol. 10, Issue 9, September 2023.

Eshofonie, F. Patience (2008): Factors Affecting Cost of Construction in Nigeria. Master's Degree Thesis in Construction Management, Department of Building, University of Lagos.

Abdulrahman Haruna, Musa Adamu, Bashar Sami Mohammed, Adamu Bakoji, Musa Mohammed (2017): Problems Faced by Nigerian Contractors in Controlling Cost of Building Project.American Journal of Environmental Policy and Management. Vol. 3, No. 3, 2017, pp. 19 - 23.

Amjdi Ali (2023): Building Definition in Civil Engineering: An Insight into the Construction Industry. AYAL YUSR Article, Al Yusr International, Oman.