AI-Based Real Estate Price Prediction and Location Intelligence

Ayesha Mirza[1] Divya Thakare[2] Ravina Mozar[3]
[1,2,3]Department of Artificial Intelligence and Machine Learning
[1,2,3]Zeal Polytechnic, Pune, Maharashtra, India

Abstract — Buying real estate requires more than checking the advertised price. Buyers also need to understand the location, nearby facilities, property age and condition, seller credibility, legal-document status and ongoing ownership expenses. These details are often scattered across listing websites, local sources and separate service providers. This paper proposes an AI-based real estate price prediction and location intelligence mobile application that combines these factors in one decision-support platform. The system is designed to estimate the total property price and price per square foot using machine-learning regression, describe locality conditions, show available condition insights, and organise seller and document-review statuses. It also provides side-by-side comparison of two or three properties and an estimate of total ownership cost, including applicable charges and an optional equated monthly instalment (EMI). Linear Regression, Random Forest Regressor and Gradient Boosting are proposed as candidate algorithms. Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and R^2 are proposed for evaluation. This paper presents the system design and methodology; it does not claim a completed implementation or measured prediction results.

Keywords: Real Estate, Price Prediction, Machine Learning, Regression, Location Intelligence, Mobile Application, Property Comparison

I. Introduction

Real estate is one of the most important financial decisions for individuals and families. A home buyer may compare several flats or houses before choosing one, while an investor may be interested in the relationship between price, locality development and future demand. In both cases, the advertised price is only one part of the decision. The value of a property can be influenced by its size, property type, age, condition, location, nearby facilities and the reliability of essential services.

A useful prediction system must also define the exact question it is answering. The estimated value may refer to the likely transaction price, the current listing price, or a broad price range for similar properties. These values are related, but they are not identical. The application should state which target variable was used during training and should avoid presenting a listing price as proof of the final sale value. If the dataset contains only advertised prices, the model output should be described as an estimate based on advertised prices rather than as a confirmed market valuation.

The intended users include first-time buyers, families comparing homes, and people who want an initial view of a property before contacting an agent. Their needs are not exactly the same. A buyer may care most about affordability and transport, while a family may prioritise schools, hospitals, safety information and daily services. The interface should therefore make the same underlying information understandable to different users without implying that one location is best for everyone.

Online listing platforms make property discovery easier, but they do not always provide a complete picture. One listing may contain detailed area information while another may omit the building age or maintenance history. A buyer may need to check maps for transport and amenities, ask local residents about water and electricity, and contact separate people to understand documentation. Because these sources are not organised in a common format, comparing properties can become time-consuming and confusing.

Machine learning offers a way to estimate a numerical property value from historical examples. A model can learn relationships between recorded property attributes and their prices, then use those patterns to produce an indicative estimate for a new property. Location intelligence complements this estimate by adding information about the area surrounding the property. Combining the two can help users consider price together with the context that may affect desirability.

The proposed mobile application is intended to make early-stage evaluation more organised. It is a decision-support tool rather than a replacement for a professional valuation, physical inspection, legal advice or confirmation from relevant authorities. The reliability of the output will depend on the coverage, quality and freshness of the data used by the system.

II. Problem Statement

The current property-search process is fragmented. A listing usually gives an asking price, but the user may not know whether that amount is reasonable compared with similar properties in the same locality. Details about area development, transport, schools, hospitals, water supply, electricity reliability and weather may be available from different sources. Information about property age, physical condition and maintenance may be incomplete. Seller identity and property documents also require separate checks, and buyers must often calculate taxes, registration costs, maintenance expenses and loan payments independently.

This fragmentation makes it difficult to evaluate options consistently. A buyer may focus on the lowest price and overlook the total cost of ownership, or may assume that two properties are similar because their sizes are close even though their locations and conditions differ. The project therefore proposes one interface where price estimation, locality information, property condition, verification progress and ownership costs can be viewed together.

A further objective is to make the prediction understandable. Along with a price estimate, the application can display the main property details entered by the user, the date on which the estimate was generated, and a short explanation of the factors available to the model. Where sufficient comparable records exist, the interface may show a range or a comparison with similar properties. Such explanations should be generated from actual model inputs and data; the application should not invent reasons for a

All rights reserved by www.ijsrd.com

prediction or present correlation as proof that a feature causes a price change.

The project also aims to keep the process practical on a mobile device. The user should be able to enter a small set of essential details, review the information before submission, and correct mistakes without restarting the entire process. Optional details can improve the analysis when they are available, but missing optional information should not automatically prevent a user from exploring a property. Clear labels, readable charts, accessible controls and plain-language warnings are important parts of the proposed system, not merely visual additions.

III. Objectives

The primary objective is to design an AI-supported mobile application that helps users make a more informed preliminary comparison of real estate options. The specific objectives are:

1) To estimate the likely total property price and price per square foot from available property features.
2) To present locality and infrastructure indicators, including area development, nearby amenities, water supply, weather and electricity reliability.
3) To show available details about property age, condition and maintenance.
4) To organise seller identity verification using clear process statuses.
5) To organise property-document review using clear statuses.
6) To compare two or three properties side by side.
7) To estimate total ownership expenses, including applicable purchase-related charges and an optional EMI calculation.

IV. Existing System

In the existing approach, property details are usually collected from listing websites, brokers, advertisements or direct contacts. These sources may use different terminology and may not present the same fields. The advertised price can be influenced by negotiation, seller expectations and market conditions, so it is not necessarily equal to the final transaction value. Users often need to search separately for local infrastructure and nearby facilities.

Another limitation is that property verification and financial planning are often handled outside the listing platform. Buyers may need to contact a professional to inspect documents, arrange an on-site visit to assess condition, and use a separate calculator to estimate loan instalments. Comparing several properties then requires manually copying information into notes or spreadsheets. This increases effort and makes it easy to overlook a cost or an important difference.

The proposed application addresses these issues by presenting relevant information in a consistent structure. It does not assume that all data can be verified automatically; instead, it aims to show which information is available, which remains under review and which items need further checking.

V. Proposed System

The proposed system is a mobile application with seven main modules: AI price prediction, location intelligence, condition insights, seller identity verification, document review, smart comparison and total ownership cost. The user enters property details or selects a listing. The system validates the input, prepares the relevant property features, obtains available locality information, generates a price estimate when a trained model is available, and displays the results in a structured interface.

Input validation is necessary before a prediction is requested. For example, area must use a stated unit, the number of bedrooms cannot be negative, and a property age should fall within a reasonable range. If the user selects a property type that the model did not support during training, the system should explain that the estimate may be unavailable or less reliable. It should not silently convert unfamiliar categories into an unrelated category. These checks reduce avoidable errors and help users understand which information is required.

The model response should be displayed with context. A single number can appear more certain than the underlying data supports, especially when the property is unusual or the dataset has few comparable examples. The application may therefore show an estimate together with a suitable uncertainty indicator, such as a prediction interval if the selected method supports one. If a reliable interval cannot be calculated, the interface should simply disclose that uncertainty remains rather than displaying a fabricated confidence percentage.

A. AI Price Prediction

The price-prediction module estimates the likely property price and price per square foot. Potential input features include locality, property type, built-up or carpet area, number of bedrooms and bathrooms, building age and other fields available in the dataset. Units must be standardised before prediction. If the model estimates a total price, the price-per-square-foot value can be calculated using the selected area measure. The application should state whether the value is based on built-up area, carpet area or another defined measure.

The result should be labelled as an estimate rather than a guaranteed market value. An asking price, a completed transaction price and a model prediction are different kinds of information and should be visually distinguished. Where possible, the app can also show the date range and geographic coverage of the training data so users understand the context in which the estimate was produced.

B. Location Intelligence

Location intelligence presents information about the locality surrounding a property. The project proposes considering area development, nearby amenities, water supply, weather conditions and electricity reliability. Other useful contextual details may include distance to schools, colleges, hospitals, public transport and markets where reliable data is available. The purpose is to help users consider the surrounding environment alongside the property's physical features.

Locality labels such as less developed, developing and well developed should be based on explicit criteria. A

All rights reserved by www.ijsrd.com

single label can hide important differences, so the interface should ideally show the underlying indicators separately. The app should identify the source and date of the information, and should not imply that proximity to a facility guarantees its quality or availability.

The condition module should distinguish user-reported information from professionally inspected information. For example, a listing may describe a property as recently renovated, but the application should label this as a listing claim unless a trustworthy inspection record is available. Where users upload photographs or enter notes, these should be treated as supporting material and not as conclusive evidence of structural quality. The application can provide a checklist of topics to discuss during a site visit, including ventilation, water supply, electrical systems, drainage and visible maintenance concerns.

This separation also helps prevent a misleading overall score. A property may be located in a convenient area while requiring significant repairs; another may be in good condition but have fewer nearby services. Showing these dimensions separately allows the user to judge trade-offs instead of relying on a single unexplained rating. The system should make it easy to identify information that is unavailable and should encourage a qualified inspection whenever a decision depends on the physical condition of a building.

C. *Property Condition Insights*

This module presents available information about property age, reported condition and maintenance. Building age may be calculated from a recorded construction year when that information is known. Maintenance history, repair details and inspection notes should be shown only when supported by a source or user-provided evidence. Missing information must be marked as unavailable rather than interpreted as proof that no problem exists.

A digital summary cannot fully identify structural defects, water leakage, electrical problems or other hidden issues. Therefore, the app should encourage an on-site inspection by an appropriate professional before a purchase decision. Any repair-cost estimate should be clearly marked as approximate and should explain the source of the estimate.

VI. VERIFICATION AND COMPARISON MODULES

A. *Seller Identity Verification*

The seller-verification module organises the progress of an identity check using the statuses Pending, Under Review, Verified and Rejected. These labels help the user understand whether the process has started, is being examined, has been completed or has not been accepted. The interface should explain what the verification covers and who is responsible for performing it. A “Verified” label should not be presented as a guarantee that the seller owns the property or that the transaction is free from risk.

Identity information should be collected only when necessary for the stated purpose. Access should be restricted to authorised users, and the application should use secure handling practices for sensitive records. The design should avoid exposing identity documents in public listings or allowing unauthorised users to download them. The process should also make it clear when a check is incomplete or when additional information is required.

B. *Property Document Review*

The document-review module tracks the status of property documents through Pending Review, Under Review, Approved and Rejected. It can help users see whether a document has been submitted and whether a reviewer has completed the required step. A document checklist can reduce the chance that a buyer forgets to ask for an important record, but the exact documents required may vary by property type and location.

An automated application status is not a legal opinion. Approval should identify the scope of the review and should not imply that all possible ownership, title, dispute or regulatory issues have been eliminated. Buyers should consult a qualified legal professional and relevant authorities when appropriate. The application should store review notes and status changes in a controlled way so that users can understand the progress without being given false certainty.

A comparison view should use consistent units and clearly identify the source of each value. Area reported in square feet should not be compared directly with area reported in square metres without conversion. Similarly, a monthly maintenance charge should not be placed beside an annual charge without making the period clear. The user should be able to select the criteria that matter most, while the application avoids automatically ranking a property as universally superior. A summary can identify differences, but the final choice depends on the user's budget, needs and priorities.

The comparison module can also separate known costs from estimates. For instance, the listed price may be available directly, while registration costs or future maintenance may require an approximate calculation. Each value should carry a label such as listing-provided, user-entered, externally sourced or estimated. This simple distinction improves transparency and helps users decide which figures need to be checked with a seller, local authority, bank or other qualified source before making a commitment.

C. *Smart Property Comparison*

The comparison module is designed to place two or three properties side by side. A consistent set of fields may include asking price, predicted price, price per square foot, area, property type, building age, reported condition, locality indicators, nearby facilities and verification status. Displaying common fields in the same order makes it easier to identify meaningful differences.

The interface should distinguish missing data from a negative result. If one property has no recorded maintenance history, it should be labelled “Not available” rather than assigned a zero value. The same principle applies to seller and document statuses. Users should be able to decide which criteria matter most to them, because a single automatically generated ranking may not reflect every buyer’s needs.

For clarity, each comparison should show the date of the listing and the source of key information. A model-estimated price should not be confused with the seller’s

All rights reserved by www.ijsrd.com

asking price or a legally assessed value. If a location indicator is derived from a third-party service, the interface should disclose that source where possible and handle unavailable service responses gracefully.

VII. TOTAL OWNERSHIP COST

A property's purchase price does not represent the full amount a buyer may spend. Other expenses may include stamp duty, registration charges, brokerage, loan processing fees, maintenance deposits, society charges, insurance, repairs and moving costs. The exact categories and rates depend on the property, the transaction and the applicable local rules. The proposed system can present these items as an itemised estimate so that the user can understand how the overall amount is calculated.

The calculator should make a clear distinction between user-entered amounts and values obtained from an external source. Since taxes and charges may change, rates should be updated from reliable sources rather than permanently embedded in the application. If a fee cannot be confirmed, the app should mark it as an assumption or ask the user to enter a value. The total should therefore be presented as an estimate, not as a final legal or financial statement.

An optional EMI calculator can estimate the monthly instalment using the loan principal, annual interest rate and loan tenure. The result should identify the values used and clarify that processing fees, insurance and other charges may not be included. Showing both the monthly instalment and the estimated total interest can help users understand the trade-off between loan tenure and overall borrowing cost.

VIII. METHODOLOGY

A. Data Collection

The first stage of implementation is to assemble a property dataset from sources that can lawfully be used for this purpose. A record may include location, property type, area, number of rooms, property age, condition, asking or transaction price and observation date. The project should document the geographic coverage, time period, units and collection method. If data from different sources is combined, the fields must be reconciled so that similar values have the same meaning.

Location information may be obtained from reliable maps, public infrastructure sources or other permitted data services. Indicators such as water supply and electricity reliability can be difficult to measure consistently, so their definitions and sources must be documented. Data should not be treated as current simply because it exists in a database; the system should retain timestamps and identify stale information where possible.

B. Data Cleaning and Preparation

Before training, the dataset should be checked for duplicate records, missing values, inconsistent units, invalid areas, unusual prices and inconsistent locality names. A documented process should determine whether missing values are imputed, retained with indicators or removed. Categorical values such as property type and locality may need encoding, while numerical fields may need scaling for some algorithms. Preprocessing parameters must be learned from training data only to prevent data leakage.

C. Feature Selection and Model Training

Features should be selected according to their relevance, reliability and availability at the time a prediction is requested. Property area, locality, property type and age may be useful when they are recorded consistently. Information that would not be available for a new property should not be used as a model input. Sensitive personal characteristics should not be included without a justified and lawful reason.

The proposed algorithms are Linear Regression, Random Forest Regressor and Gradient Boosting. Linear Regression provides a baseline and is relatively straightforward to interpret. Random Forest can represent non-linear relationships by combining many decision trees. Gradient Boosting builds an ensemble in stages, with each stage attempting to improve on earlier errors. The algorithms should be trained and evaluated on comparable data splits, and a simple baseline should be retained for reference.

A training set is used to fit the model, while validation data may be used for model selection and parameter tuning. A final test set should remain separate until the final evaluation. Depending on the dataset, a time-based split or a location-based split may provide a more realistic indication of how the model performs on newer records or areas not well represented in training.

Evaluation should also include checks for data leakage. Information that would not be known at the time of a real prediction must not be included as an input feature. Duplicate listings should be handled carefully because the same property appearing in both training and test data can make performance look better than it really is. Where possible, the evaluation split should reflect the intended use of the system, such as testing on later records or on properties not represented in training. The chosen approach and its limitations should be documented so that results can be interpreted correctly.

Error analysis should go beyond one overall score. The team should inspect whether errors differ by property type, locality, price range or the amount of information available. An average error can hide larger mistakes for less common properties or neighbourhoods with limited records. These checks do not guarantee that every prediction will be fair or accurate, but they help identify where the model needs more data, revised features or clearer warnings. Model selection should consider reliability and maintainability as well as numerical performance.

D. Evaluation Metrics

Mean Absolute Error (MAE) measures the average absolute difference between actual and predicted prices. Root Mean Squared Error (RMSE) also measures prediction error but gives greater weight to large errors. R^2 describes the proportion of variation explained by the model relative to a baseline under the chosen evaluation method. These measures should be reported together with the number of records, price units, test-split method, time period and geographic coverage.

No numerical scores are reported in this paper because the supplied project materials do not include a trained

All rights reserved by www.ijsrd.com

model or test results. Reporting invented accuracy values would be misleading. Once implementation is completed, the results should include a comparison table for each candidate model and should discuss errors across different localities and price ranges, not only a single overall score.

E. Application Development

The proposed backend may use Python with Flask or FastAPI. A mobile interface may be developed with Flutter or React Native, while a database stores property details, comparison selections and workflow statuses. Maps and other location services may be integrated when access is permitted. The conceptual data flow is: user input, validation, feature preparation, model inference, location-data retrieval, comparison and cost calculation, followed by presentation of results.

A practical system architecture can be organised into a mobile presentation layer, an application service layer, a prediction service and a data layer. The mobile layer collects information and displays results. The service layer validates requests and manages application rules. The prediction service loads the selected model and returns an estimate, while the data layer stores permitted property records, model versions and limited application metadata. This separation can make the system easier to test and update, although the final architecture should be chosen according to the available development resources and project requirements.

Security and privacy should be considered throughout development. The application should collect only information needed for its stated purpose, protect stored records, restrict access to identity and document details, and avoid exposing sensitive information in error messages or logs. Users should be told why a document is requested and how its review status will be used. External map, geocoding or listing services may have their own terms and limitations, so the project should confirm permission to use their data and communicate when an external service is unavailable.

The implementation should validate inputs, record model versions and handle external-service failures. If a map or weather service is unavailable, the app should indicate that the information could not be retrieved rather than display it as current. Security controls should protect user accounts, documents and identity information. These are proposed implementation choices, not claims that the application has already been built.

IX. Expected Advantages

The proposed system is expected to make property evaluation more organised by presenting a price estimate alongside locality and condition information. A common comparison layout can reduce the effort of moving between several listings and can help users identify missing information. The verification modules can make process status more visible, while the ownership-cost calculator can remind users to consider expenses beyond the advertised price.

Another expected advantage is improved transparency. By showing whether a value is an asking price, a model estimate or a user-entered cost, the application can reduce confusion between different kinds of information. Showing the date and source of locality indicators can also help users judge whether the information is recent enough for their needs. These benefits are expected outcomes and must be tested through implementation and user feedback.

X. Evaluation and Testing Plan

Technical testing should evaluate prediction quality on records that were not used to train or tune the model. MAE, RMSE and R^2 should be reported for each candidate algorithm and compared with a simple baseline. The evaluation should also examine different price bands and localities to determine whether error is concentrated in specific groups. The dataset size, missing-data treatment and limitations of the split should be documented.

Functional testing should check that users can enter property details, view an estimate, open locality information, compare two or three properties, and understand verification and document statuses. Cost calculations should be checked with known test inputs, including missing or invalid values. The application should be tested when external data services fail and when the user enters an unsupported property type or an unusual area value.

A further test scenario should cover incomplete or contradictory information. If a user enters an area in the wrong unit, leaves a required field empty, or provides a locality that cannot be matched, the application should give a specific and understandable message. Tests should also cover slow network responses, unavailable map services, expired data and unsuccessful document uploads. The system should preserve entered details when appropriate and should not show an old estimate as though it were a new result. These scenarios are important because users may rely on the displayed information when planning a site visit or discussing a property with a seller.

Usability testing could ask participants to complete tasks such as interpreting an estimate, comparing two properties and identifying costs that are not included in the purchase price. Task completion, time required, clarity of information and common user misunderstandings can be recorded. Any study involving participants should follow suitable consent and privacy practices. User satisfaction and model accuracy should be reported as separate outcomes.

XI. Limitations and Responsible Use

The system depends on the accuracy, coverage and freshness of its data. Asking prices may differ from final transaction prices, and listing descriptions may be incomplete or incorrect. Locality-level averages can conceal differences between individual streets or buildings. Information about water, electricity, weather and amenities may be updated at different intervals. Predictions may therefore be less reliable in locations with few records or in markets that are changing quickly.

A model cannot detect every factor affecting value, including hidden structural defects, legal disputes, negotiation, personal preferences or sudden market changes. Seller verification and document-review labels are process indicators, not guarantees of safety or ownership. Ownership-cost estimates depend on the accuracy of the entered values and the current rules. Third-party data access may also be restricted by cost, coverage or terms of use.

All rights reserved by www.ijsrd.com

Responsible use requires clear communication about uncertainty. The app should disclose the data date and coverage, distinguish missing information from a negative finding, and avoid presenting one score as an unquestionable decision. Personal and identity data should be collected only, when necessary, protected from unauthorised access and retained only according to a defined policy. Users should be encouraged to obtain independent professional, legal and physical checks before making a final purchase decision.

This paper describes a proposed system. The supplied presentation and project paper do not establish a completed deployment, verified seller checks, measured prediction accuracy or proven user benefits. These outcomes require implementation, appropriate data, testing and documented results.

XII. Conclusion and Future Work

This paper proposes an AI-based real estate price prediction and location intelligence mobile application that combines estimated price and price per square foot with locality indicators, property-condition information, seller identity and document-review workflows, side-by-side comparison and total ownership-cost estimates. Linear Regression, Random Forest Regressor and Gradient Boosting are proposed as candidate models, with MAE, RMSE and R^2 for evaluation.

A phased implementation is recommended. The first phase can establish the data schema, input validation and a baseline regression model. The second phase can compare candidate models using a consistent test strategy and record their limitations. The third phase can connect the selected model to a simple mobile interface and add locality information with clear source labels. Later phases can introduce comparison, ownership-cost calculations and document-status tracking after the core prediction flow has been tested.

Future development should include a documented process for refreshing the property dataset. The team should record the source, collection date, permitted use and important limitations of each dataset. Older records may be useful for training, but market conditions and infrastructure can change over time, so the model should be reviewed when the data distribution changes. Retraining should not be automatic without checks: a new model should be compared with the current version using a consistent evaluation procedure before it is released to users.

Future work should implement the application with verified and representative local property data, compare candidate models using a documented test strategy, and assess errors across locations and price ranges. Further development may include interactive maps, carefully sourced infrastructure indicators, secure document workflows, updated ownership-cost calculations and usability testing. The application should remain a support tool: estimates need interpretation, and legal, financial and physical checks remain important.

References

[1] Project presentation, “AI-Based Real Estate Price Prediction and Location Intelligence,” Department of Artificial Intelligence and Machine Learning, Zeal Polytechnic, Pune, India.
[2] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. New York, NY, USA: Springer, 2009.
[3] L. Breiman, “Random forests,” Machine Learning, vol. 45, pp. 5–32, 2001.
[4] R. T. Fielding, Architectural Styles and the Design of Network-based Software Architectures. Ph.D. dissertation, University of California, Irvine, CA, USA, 2000.

All rights reserved by www.ijsrd.com