DSAI 2025

Final Milestone

Title

House Price Prediction System (USA)

By:

ID: 685380020-3	**Khamtan Khamsamsee**
ID: 685380025-3	**Poonyawat Mandee**
ID: 685380035-0	**Anapat Chansong**

Thought by:
Asst. Prof. Chitsutha Soomlek

Course ID: SC348810
Course name: Software Development and Project Management for Data Science and Artificial Intelligence

Semester 1

Data Science and Artificial Intelligence College of Computing

Khon Kaen University

(September 2025)

CONTENTS

1. Project Overview

This project develops an AI-enabled House Price Prediction System targeted at the United States of America (USA). The application aims to provide accurate property value estimates for end-users, including home buyers, real estate agents, and investors. By inputting some key house features such as LotArea, overall quality, number of bedrooms and bathrooms, users receive tailored predictions. The system enhances accuracy by allowing users to customize feature weights and offers comparative pricing for similar properties in the same neighborhood, providing market context for informed decision-making.

2. Privacy Policy

The privacy policy ensures CCPA/CPRA compliance and transparency:

- **Data Collection**: Collects only necessary data (name, email, password, search history, feedback) for user verification, predictions, and app improvement.
- **Consent**: Users provide explicit consent via in-app notifications before data collection.
- **Security**: Data is encrypted, and access is restricted to authorized purposes. The full policy is accessible on the website.

3. Terms of Use

The terms of use outline user responsibilities and system limitations:

- **Acceptable Use**: Users must provide accurate data and use predictions for personal or professional decision-making, not for unlawful purposes.
- **Liability**: Predictions are estimates; the system is not liable for financial decisions based on outputs.
- **Compliance**: Aligns with CCPA/CPRA and FHA, prohibiting discriminatory use. The full terms are available on the website.

4. AI Policy Framework

The system complies with EO 14110, FHA, and CCPA/CPRA:

- **Fairness**: Excludes discriminatory data (race) and audits for bias (redlining) using Demographic Parity.
- **Accountability**: Logs errors and offers a complaint mechanism; audits ensure FHA compliance.
- **Privacy**: Enforces consent, data minimization, and encryption.
- **Safety/Reliability**: Minimizes errors (MAE target < 5-10%) and secures data.

5. Identify stakeholders, including the end usersOrganizational / Business User (Primary Buyer)

Organizational / Business User (Primary Buyer): Real Estate Company

Real estate companies are ideal primary buyers, using the app to estimate house prices, conduct market analysis, offer competitive pricing, and strategize investments. The neighborhood price comparison feature adds value for market competition. Adding ROI analysis or market trend forecasting could attract larger firms.

Professional User (Secondary): Appraisers, Real Estate Agents and Property Investors.

Appraisers, agents, and investors are suitable secondary users, leveraging predictions for decision-making in buying/selling. Appraisers verify valuations, agents offer competitive prices, and investors identify opportunities. The customizable feature weights meet their needs. Adding dashboards for investors and training for agents would enhance utility.

End-User / Customer (Indirect): Home Buyers and Home Sellers.

Home buyers and sellers benefit indirectly via real estate companies or agent platforms, gaining informed pricing decisions through neighborhood comparisons. A user-friendly UI for direct access would improve engagement if companies allow it, ensuring clarity for non-expert users.

Internal Stakeholders: AI Development Team, Data Scientists and DevOps/ML Engineers.

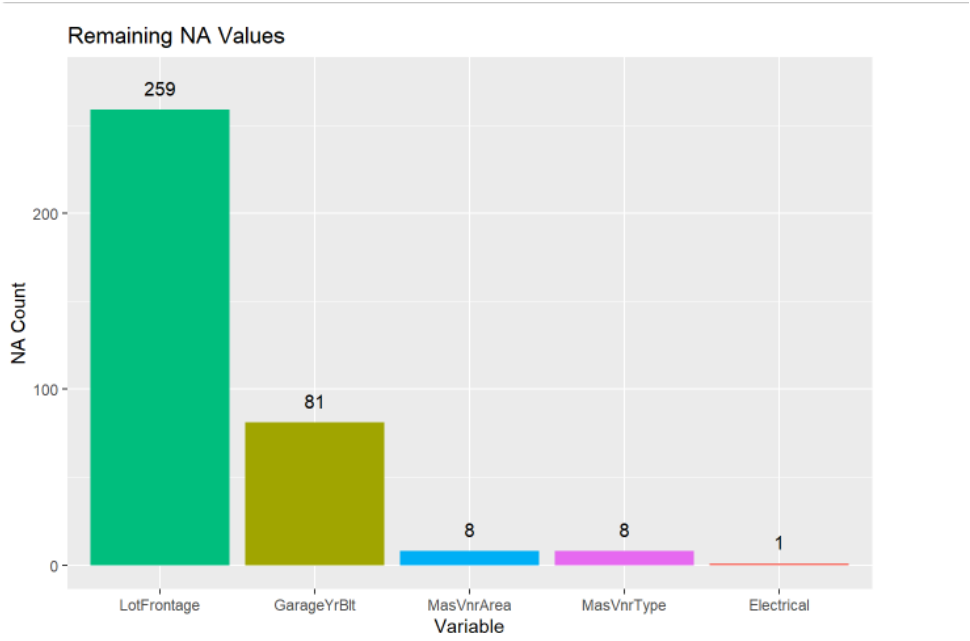
The AI development team, data scientists, and DevOps/ML engineers are critical for building, training, and maintaining the app, ensuring accurate models and robust deployment. Adding cybersecurity engineers would strengthen data protection, vital for compliance with privacy laws like CCPA/CPRA.

External Stakeholders: Government Real Estate Authorities and AI Policy Experts.

Government authorities (HUD) ensure compliance with regulations like FHA to prevent bias, while AI policy experts guide adherence to ethical AI standards (EO 14110). Early engagement with regulators and legal experts ensures the app meets fairness and transparency requirements.

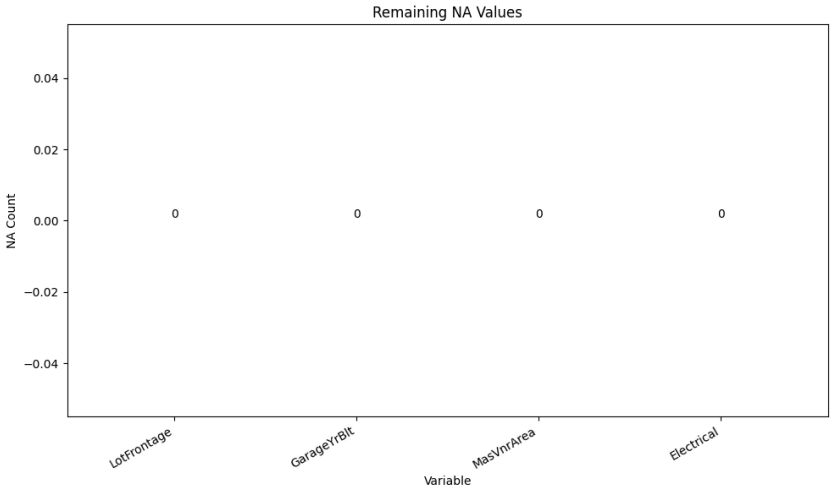
6. Data

The House Price data comes from Kaggle datasets. https://cphaigh.github.io/KaggleProjects/kaggleHousePrices/HousePrices12_19.html?utm_source=chatgpt.com#libraries



Picture 1). Actual DataSet Before clean

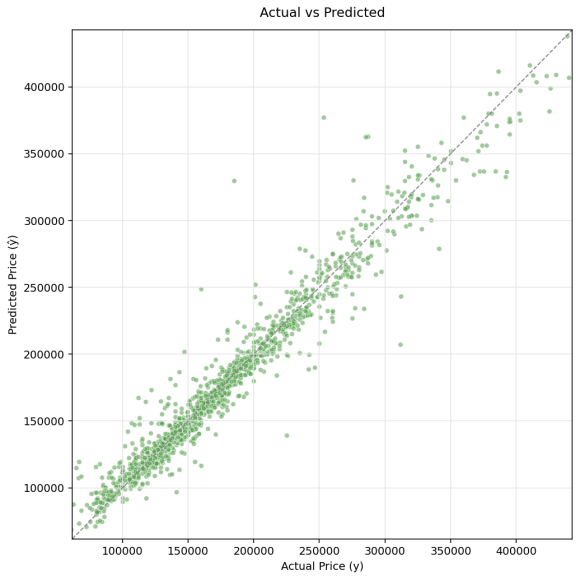
Clean data: The columns which contain a large number of NaN values were dropped from both train and test sets. Missing numeric values were imputed using the mean of each column.



Picture 2). Actual DataSet Cleaned

7. Methodology

- **Feature Selection**: From 81 dataset columns, 10 features were chosen (OverallQual, TotalBsmtSF, LotArea, GarageCars, Fireplaces, BedroomAbvGr, GrLivArea, FullBath, Neighborhood and Sale Price) based on high correlation with house prices, determined via exploratory data analysis (correlation coefficients, feature importance). This reduces model complexity, enhances interpretability, and aligns with user inputs, improving efficiency and user trust.
- **Handling Missing Values**: KNNImputer (5 neighbors) imputes missing inputs by averaging values from the 5 nearest data points based on Euclidean distance. This preserves data relationships (similar houses have comparable LotArea), ensuring robust predictions for incomplete inputs (imputing LotArea: 9379.8 for a sample), unlike simpler methods like mean imputation.
- **Model Choices**:
  - **Random Forest Regressor**: Selected for price prediction due to its ability to capture non-linear feature interactions, achieving high accuracy (R^2 = 0.956, MAE = $9,703.72) with 200 estimators to minimize overfitting.



Picture 3.)
Summary Strong linear relationship between Actual and Predicted
•Random Forest Regressor

  - **KNNImputer**: Chosen for non-parametric imputation, avoiding distribution assumptions, ensuring compatibility with diverse house features and robust inputs for the Random Forest model.

- **Exporting Model**: After training model, the model is exported to the joblib file. Using joblib is more suitable because it is faster, more memory-efficient, and specifically optimized for machine learning models that rely on NumPy and scikit-learn. It ensures smooth loading in the Flask/FastAPI backend and aligns with best practices for model deployment.

8. Website Implementation

The system has been implemented as a bilingual website supporting Thai and English languages. Key user flows and features include:

- **Sign-Up:** New users provide their name, email, and password (minimum 6 characters). They must agree to the Privacy Policy and Terms of Use (aligned with CCPA/CPRA) via a checkbox to ensure informed consent.
- **Privacy Policy and Terms of Use:** Users are informed about the important details of the policies and terms of use.
- **Login:** Returning users log in with their email and password to access the main index page.
- **Index:** Users can input from 1 up to 10 housing features (e.g., lot size, location) via a form.
- **Result:** Predictions are displayed as a set showing all 9 features with price estimates. The system can also perform reverse prediction — when users input a price, it displays the 9 predicted features.
- **User Feedback:** Users can comment on or like/dislike predictions.
- **Security:** Passwords are encrypted, and data collection follows data minimization principles.

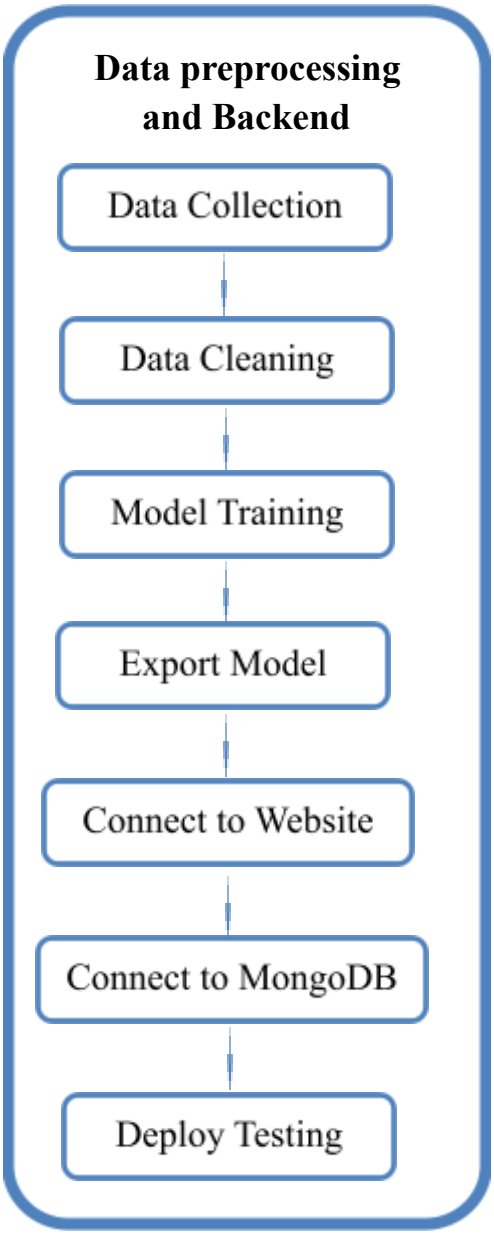
9. User's Data

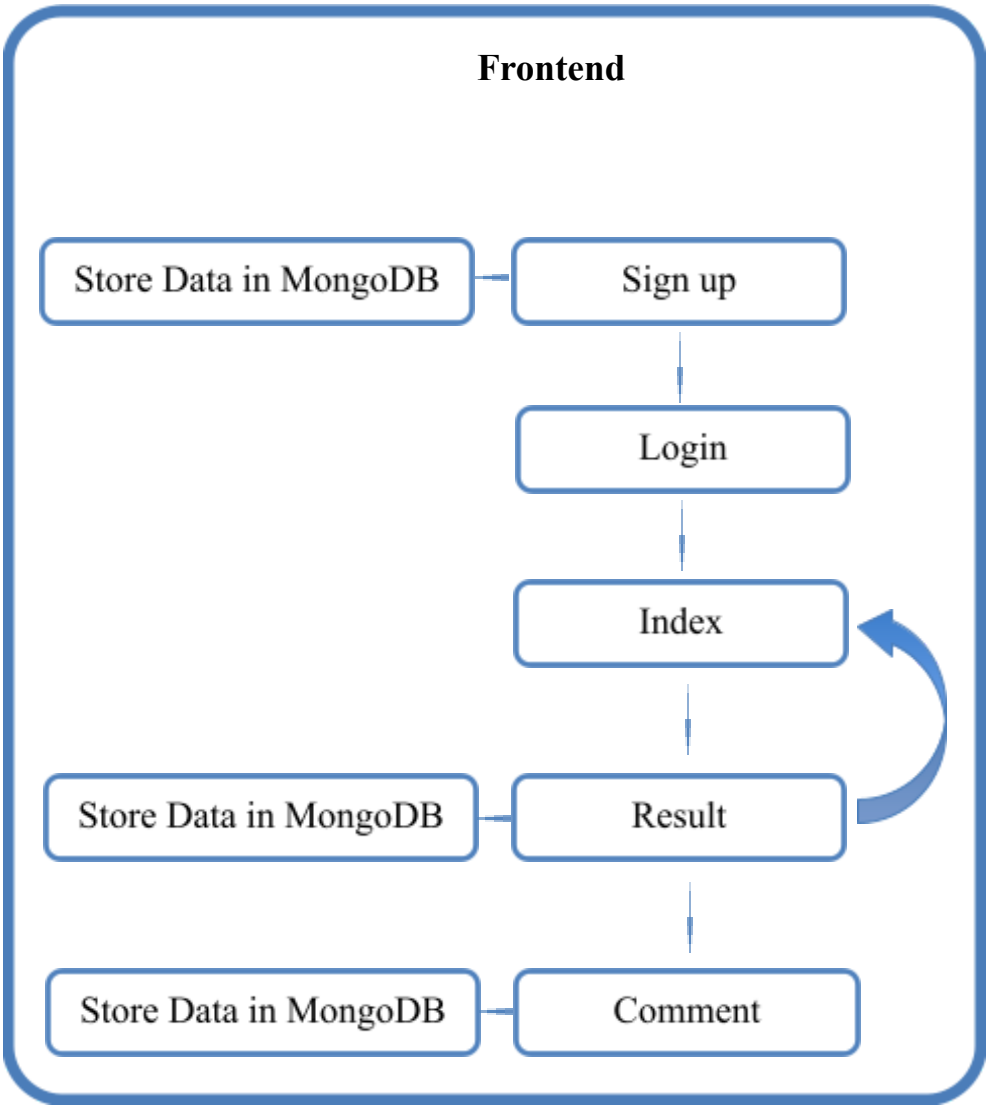
User data (username, hashed password, email, search history, results, feedback and timestamp) is stored in MongoDB Atlas with user consent for model and app improvements.

There are many databases (MongoDB Atlas, Firebase). However, MongoDB Atlas was selected over Firebase as the database solution for this system due to its superior flexibility, integration capability, and data control. The system requires storing diverse and complex data types such as user profiles, prediction inputs, and feedback, which fit naturally into MongoDB's document-based model. Its powerful querying and aggregation functions allow for in-depth analysis of user activities and model performance, which Firebase's simpler query engine cannot easily support. Moreover, MongoDB integrates seamlessly with Python-based machine learning backends, enabling efficient data retrieval for model training and prediction. From a compliance perspective, MongoDB provides advanced privacy and security features, including field-level encryption and detailed access control, aligning well with CCPA/CPRA requirements. Finally, its open-source nature and flexible deployment options minimize vendor lock-in, offering better scalability and long-term maintainability compared to Firebase's proprietary ecosystem.

10. Pipeline

- Collection: Zillow, Kaggle datasets with 81 columns; 10 key features selected for training.
- Steps:
    1. Ingest CSV files (train_cleaned.csv).
    2. Clean data (encode Neighborhood to numeric, remove outliers, impute missing values with KNNImputer).
    3. Engineer features (encode categories, normalize numerics).
    4. Train Random Forest Regressor for price predictions (MAE: $9,703.72, R^2: 0.956) and KNNimputer for Chosen for non-parametric imputation, avoiding distribution assumptions, ensuring compatibility with diverse house features and robust inputs for the Random Forest model.
    5. Deploy via FastAPI/Streamlit.
    6. Monitor weekly for drift/fairness.
- Automation: Scripts and MongoDB for efficient data management; includes flow charts and a timetable.





11. Deployment

The "Staging Deployment" process using Render, a Platform-as-a-Service (PaaS) similar to Vercel or Railway. The process involves deploying a Flask application from GitHub to Render with the following steps:

1. Create a repository (GitHub) and connect it to Render.
2. Set up the repository with a "build Flask" configuration.
3. Define an "environment variable" for staging or production.
4. Obtain the Render URL for access.

https://projectse-9dgx.onrender.com/

12. Conclusion

This system delivers accurate (R^2 = 0.956, MAE = $9,703.72), ethical, and user-friendly house price predictions using 10 key features (OverallQual, GrLivArea, Neighborhood). Random Forest Regressor ensures robust price estimation, Linear Regression enables reverse feature prediction, and KNNImputer handles missing inputs. MongoDB secures user data, and privacy/fairness safeguards enhance trust. Future enhancements could include advanced models or additional features.

13. Tools

- **Microsoft Word**: Writing summary documents.
- **Microsoft Excel**: Inspecting and managing CSV files with house data (lot size, year built) and performing basic analysis, like calculating average prices for neighborhood comparisons.
- **Google Colab**: Online Python coding to train AI models (Linear Regression), clean data, and perform spatial analysis (neighborhood price comparisons using GeoPandas). Supports feature weight customization.
- **Jupyter Notebook**: Offline Python coding and testing, such as model training, data cleaning (lot size, bedrooms), and preliminary data analysis for price comparisons.

- **Visual Studio Code**: Developing frontend (UI for inputting data and displaying price comparisons). Supports creating a web interface for customizable feature weights. Developing backend (push and pull files to Github. Connecting API. Connecting webpages and python code (AI/ML)).
- **AI Large Language Models (ALLM)**: Grammatical error correction and editing code.
- **MongoDB**: Storing data from the backend.
- **Canva**: Making the presentation.
- **Render**: Deploy testing staging.

14. Timetable

Mile Stone 1		Enter P for Present, L for Late, E for Excused absence, and U for Unexcused absence. Use the 'Attendance key' tab to customize.																	
Task		July				August					September				October				
		week1	week2	week3	week4	week1	week2	week3	week4	week5	week1	week2	week3	week4	week1	week2	week3	week4	week5
1	research Document																		
2	Choos Tiltle and collect data																		
3	Preprocessing data																		
4	Train and test model																		
5	Write mile stone report																		
Mile Stone 2																			
Task																			
1	Test model 2																		
2	Test system 1																		
3	Debugging																		
4	Test system 2																		
5	Deploy system																		
6	Monitering																		
7	Final																		