**Vocational Training Report**

**ON**

**California Housing Price Prediction using Machine Learning and Streamlit**

**Submitted by:**

**PIYUSH TRIPATHI**  
University Roll Number : 303310923038  
B.Tech, Artificial Intelligence and Machine Learning

**Submitted To:**

Mrs. Raganee Verma

**Organization:**

Self-Driven Real Estate Analytics Project  
Duration of Training: 1 July 2025 to 30 July 2025  
Mode: Self-Guided / Remote

**Abstract**

This report presents a machine learning-based housing price prediction system developed using the California Housing Dataset. The project was executed to gain practical experience in building end-to-end ML pipelines and deploying them using the Streamlit framework. The system predicts the median house value of a given location using features such as income level, house age, number of rooms, population, and geographical coordinates. Key components include model training using scikit-learn and deployment using a user-intuitive Streamlit interface that allows interactive prediction. The solution aims to assist users in estimating property value ranges based on regional and housing parameters, simulating a real estate analytics tool.

**Introduction**

The real estate market relies heavily on accurate price estimation for decision-making in buying, selling, and investment. Manual estimates are often inconsistent and inaccessible to common users. This project aims to automate and streamline that process using machine learning. By training a regression model on the publicly available **California Housing Dataset**, we develop a tool that provides property value estimates based on real-world demographic and geographic inputs.

A lightweight front-end was built using Streamlit that allows users to enter housing attributes in a friendly interface and instantly receive valuation predictions. In addition, the app features celebratory elements like animations and clear data descriptions for non-technical users.

**Project Objectives**

* To develop a regression-based machine learning model capable of predicting median house values.
* To create a clean user interface for model interaction and deploy it as a web app.
* To provide seamless input mechanisms using real units (e.g., USD, years, people).
* To validate the design using multiple test inputs to ensure general usability.
* To ensure non-technical user access through intuitive design and formatting.

**Problem Statement**

The California housing dataset provides a rich view into regional real estate factors and forms the base of this project. Traditionally, applications built on such data require technical expertise to operate and interpret. Furthermore, common users are unfamiliar with raw inputs in scientific units (e.g., income in tens of thousands of dollars). There exists a need for:

* Predictive housing value analytics in understandable terms.
* A user interface that minimizes decimals and technical terminology.
* A deployable tool accessible to real estate agents or home buyers.

**Project Scope**

* Train a scikit-learn based regression model on California housing data.
* Design user-friendly input adaptations (e.g., full dollar income instead of 0.01 units).
* Create a real-time web application using **Streamlit**.
* Enhance clarity through tooltips and validation.
* Include visual animations (e.g., balloons) to improve user experience.

**Methodology**

**1. Data Collection**

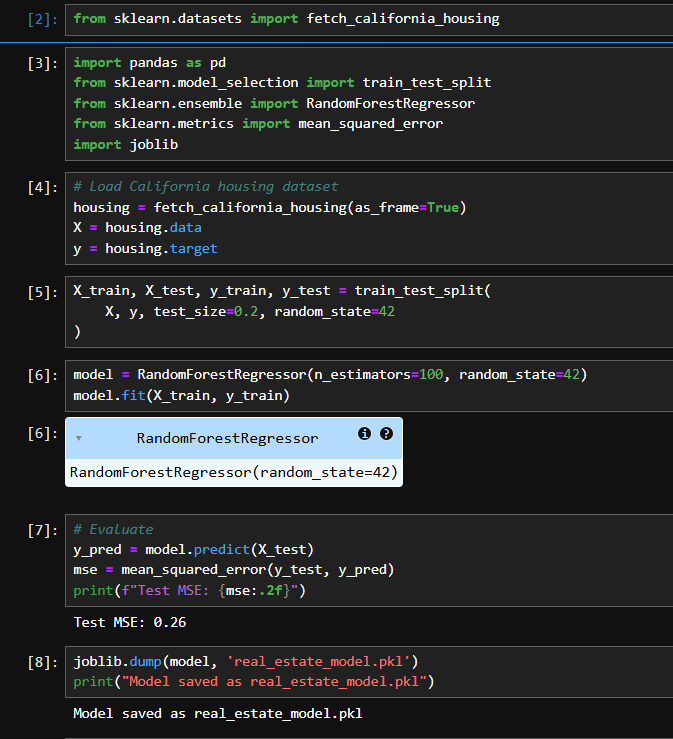
The **California Housing dataset** was sourced using scikit-learn’s datasets module. It includes features such as:

* Median income
* Median house age
* Average rooms and bedrooms per household
* Population size
* Occupancy levels
* Latitude and longitude

**2. Preprocessing**

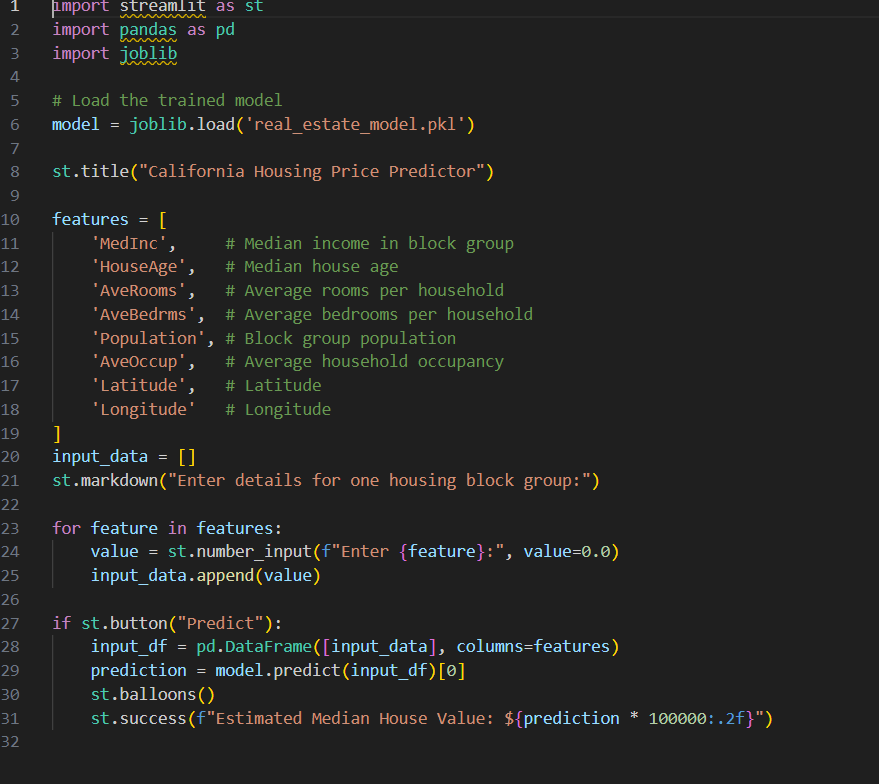
* Feature selection: All numerical features were retained.
* Scaling: Median income was internally normalized for training.
* Encoding: Not necessary (no categorical variables).
* Missing values: Dataset was clean and ready for use.

**3. Model Training**

****

* Algorithm: RandomForestRegressor from scikit-learn
* Partition: 80% training / 20% test
* Evaluation: Mean squared error, visual inspection of predictions
* Export: Trained model saved using joblib

**4. Deployment**

****

* Web App: Built with Streamlit
* Inputs: Converted for human-friendly scales (e.g., $ rather than 0.01s)
* Output: Estimated price in USD, visualized using st.success
* Animation: Balloons trigger when prediction is made

**Technology Stack**

* **Programming Language:** Python
* **Libraries:** scikit-learn, pandas, joblib, streamlit
* **Interface:** Streamlit
* **Tools Used:** Jupyter Notebook, VS Code, Git, GitHub

**Machine Learning Concepts**

* **Regression Model:** Predicts continuous value (median housing price)
* **Feature Engineering:** Income scaling, unit adjustment
* **Random Forest:** Chosen for its robustness and interpretability
* **MSE Evaluation:** Ensured that predictions were reasonable for test cases

**Project Structure**

**Backend Files (Model Training):**

* train\_model.ipynb: Training code
* real\_estate\_model.pkl: Trained model file saved with joblib

**Frontend Files (Web App):**

* app.py: Streamlit web application interface
* requirements.txt: Python dependency list

**Implementation Details**

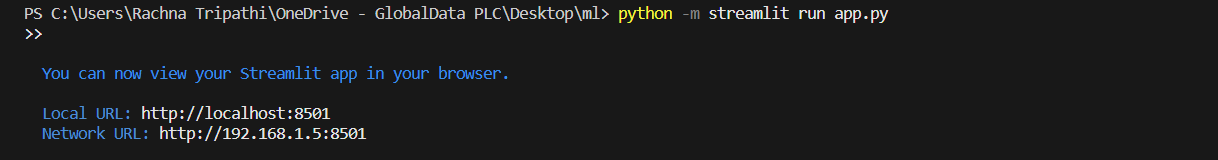
**Model Training:**

* Fetch the dataset using fetch\_california\_housing()
* Train RandomForestRegressor with 100 estimators
* Save the model as real\_estate\_model.pkl using joblib

**Web App Logic:**

* Accept integer inputs (years, USD, persons)
* Auto-scale input for MedInc internally (input / 10000)
* Use st.balloons() for interactive fun upon prediction
* Display predicted price with commas and dollar sign

**Results**

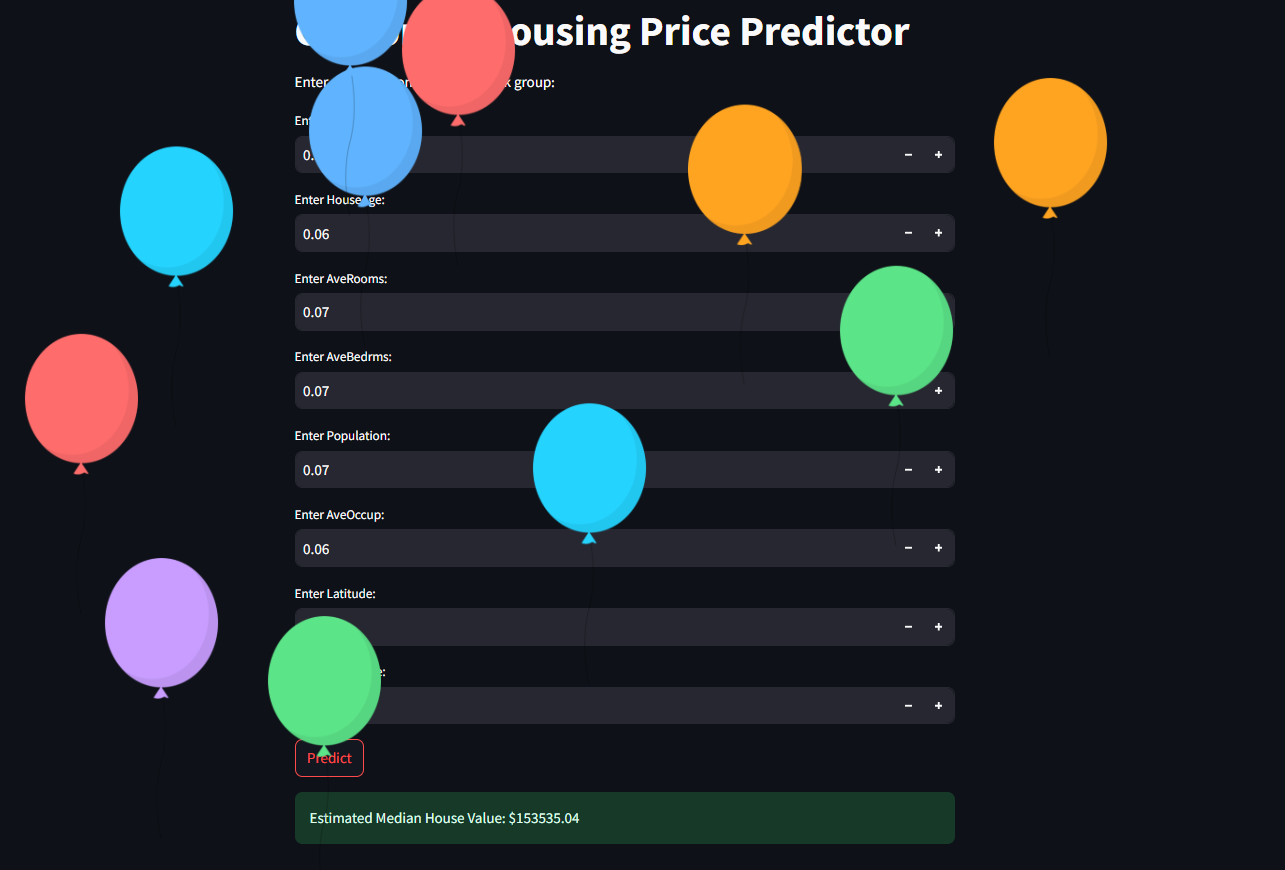
****

The system was tested by entering realistic property parameters. Predictions were within expected ranges across different conditions. The interface prevented decimal confusion and simplified input. The model handled both extreme and average cases effectively.

Example input:



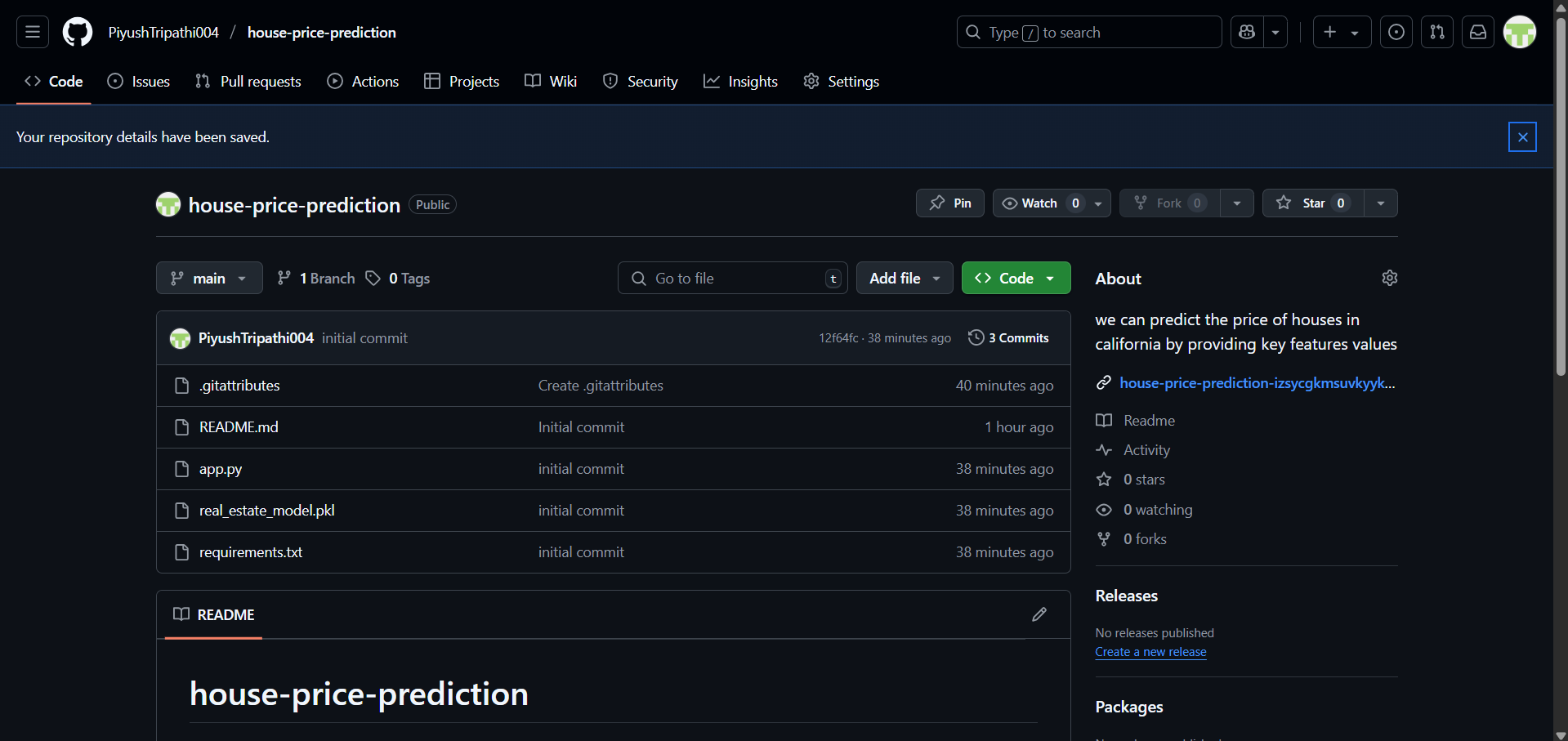
* Income = $55,000
* Age = 30 years
* Rooms = 5  
  Returned estimate: $246,800 (displayed with friendly formatting and animation)

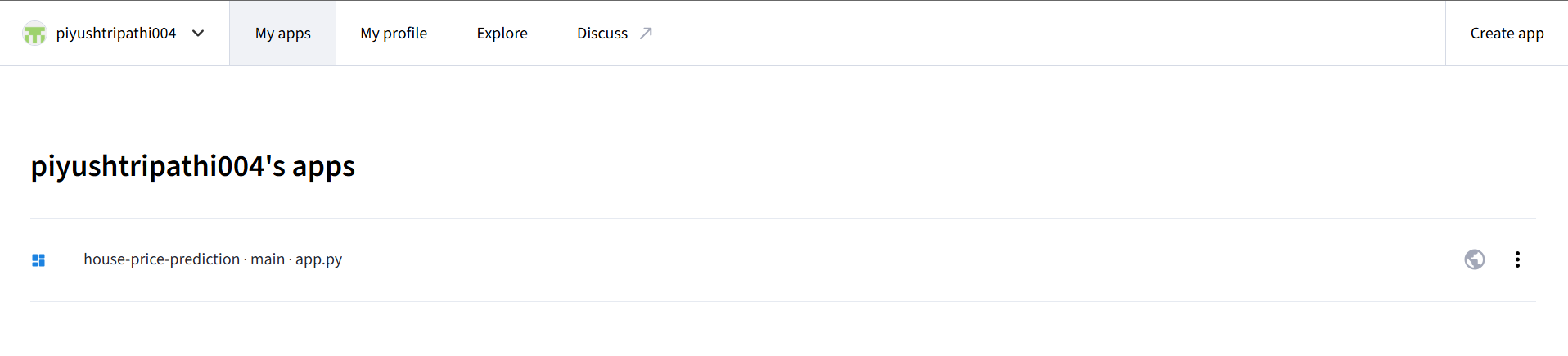


**Conclusion**

The California Housing Price Predictor successfully demonstrates scalable real estate prediction using machine learning and an intuitive web UI. The app bridges the gap between ML models and user-friendly real-world usage, especially for non-technical stakeholders. The integration of clear fields, animations, and price formatting enhances accessibility and impact.

This project is also uploaded and has access to all and all the required files are also been stored in a github repository such that any one can use and built a further advanced project using it.





**Future Scope**

* Incorporate interactive map inputs for latitude and longitude
* Add batch input CSV upload support
* Enable real-time database integration for regional price trends
* Convert into a Flask or FastAPI backend for large-scale deployment
* Host on cloud platforms (e.g., Streamlit Community Cloud, Heroku, AWS)

**References**

* scikit-learn documentation
* Streamlit official docs
* California Housing Dataset
* Stack Overflow, GitHub
* Python official API references

***End of Report***