

**International University**

School of Computer Science and Engineering

**Scalable and Distributed Computing**

**IT139IU**

**House Price Prediction with**

**Apache Spark**

Deployment: [house-price-prediction-scalable.streamlit.app](http://house-price-prediction-scalable.streamlit.app)

Github:<https://github.com/AkaiShuichi711/House-price-prediction>

**Lecturer: Assoc. Prof. Mai Hoang Bao An**

**Department: School of Computer Science and Engineering  
Date submitted: 15/01/2025**

| **No.** | **Name** | **Student ID** | **Contribution** |
| --- | --- | --- | --- |
| 1 | Nguyễn Toàn Phúc | ITITIU21093 | 35% |
| 2 | Tạ Thanh Vũ | ITITIU21352 | 25% |
| 3 | Nguyễn Tấn Phát | ITITIU21354 | 40% |

**TABLE OF CONTENTS**

[Executive Summary 4](#_heading=h.bc3bnpjg4wmw)

[Key Highlights: 4](#_heading=h.pnuiekfphugu)

[**1. Introduction 5**](#_heading=h.jornu8dwzcn7)

[1.1 Overview 5](#_heading=h.go2o9pdgo1yw)

[1.2 Objectives 5](#_heading=h.earirbb6tez7)

[1.3 Tools and Technologies 5](#_heading=h.7ren1h61ppxd)

[**2. Dataset Analysis 6**](#_heading=h.vgx4gq6yhh34)

[2.1 Overview of Dataset 6](#_heading=h.xedn8px86m9)

[2.2 Observations 6](#_heading=h.eg76e4xnetag)

[2.3 Data Preprocessing 6](#_heading=h.d13hivbcgz4g)

[2.3.1 Handling Missing Values 6](#_heading=h.hgt6m0hocx79)

[2.3.2 Outlier Detection 7](#_heading=h.wuofz2d7zpel)

[2.3.3 Feature Engineering 7](#_heading=h.md22mzs5pit9)

[2.3.4 Scaling 7](#_heading=h.dfviia9lfmfc)

[**3. Key Insights 8**](#_heading=h.z4h5oxgrdp57)

[3.1. Price Trends: 8](#_heading=h.lubsne4y5k4k)

[1. Price Distribution: 8](#_heading=h.5pkbiyoqifbm)

[2. Price Trends by Location: 8](#_heading=h.es9ibc90jluw)

[3. Correlation Analysis: 8](#_heading=h.rke3b3mc4w02)

[3.2 Market Segmentation: 8](#_heading=h.xyowpbiek420)

[1. Clustering with K-Means 8](#_heading=h.p98povhds0eb)

[2. Seasonal Decomposition 9](#_heading=h.7uy2k2d21b7u)

[**4. Predictive Modeling 9**](#_heading=h.4271kim7fuzx)

[4.1 Random Forest: 9](#_heading=h.97x7auaa006e)

[Key Takeaways: 9](#_heading=h.gefb343rfngd)

[4.2 ARIMA: Temporal Analysis 10](#_heading=h.dd7g8cayyfjn)

[Model Details: 10](#_heading=h.qy6gwabwr91g)

[Results: 10](#_heading=h.dihwi5fzyplg)

[4.3 Prophet: Seasonal and Long-Term Forecasting 11](#_heading=h.mj6uysr706sm)

[Implementation: 11](#_heading=h.ufd614li31w7)

[Results: 11](#_heading=h.r56wqqh13rat)

[4.4 Model Comparison 12](#_heading=h.tqiydctuzy8m)

[Insights: 12](#_heading=h.ny7un2iozhb9)

[**5. Advanced Analytics 13**](#_heading=h.diq5b5vrjxyh)

[5.1 Market Segmentation with K-Means 13](#_heading=h.mtp33cek3tq)

[Optimal Cluster Selection: 13](#_heading=h.s2ygwm6u9ph0)

[Insights: 13](#_heading=h.ik0fznf2xt9p)

[5.2 Seasonal Decomposition 14](#_heading=h.n2fm6jq1a4kn)

[Results: 14](#_heading=h.wdrbklxsh4tu)

[5.3 Key Findings 14](#_heading=h.wk3zb4bhs08h)

[1. Significant Predictors: 14](#_heading=h.x88df7jhd1tp)

[2. Seasonality: 14](#_heading=h.z62xokwf5val)

[3. Market Segments: 14](#_heading=h.s2p3aa6vdit3)

[5.4 Recommendations 15](#_heading=h.prmcittwfjc2)

[1. Strategic Investments: 15](#_heading=h.z6cs284n7f64)

[2. Marketing Campaigns: 15](#_heading=h.cux6o47jn31x)

[3. Inventory Management: 15](#_heading=h.o34ncjurtpuj)

[**6. Workflow 15**](#_heading=h.5a980ftfw4mx)

[1. Bengaluru\_House\_Data.csv 15](#_heading=h.kwe2ocrwdvx3)

[Role 15](#_heading=h.3f5ok27o6bbr)

[Interaction 15](#_heading=h.cnx40xqtvsew)

[2. Bangalore house price prediction.ipynb 16](#_heading=h.9i7cpouc4oa1)

[Role 16](#_heading=h.davaj8b3rh0j)

[Steps in the Notebook 16](#_heading=h.r4h71pvxfh2b)

[Interaction 16](#_heading=h.wx4p8812fjkh)

[3. model\_pickle.pkl 17](#_heading=h.iyttafw9qzs3)

[Role 17](#_heading=h.z412yw6nsx7h)

[Interaction 17](#_heading=h.bd0409iukbfi)

[4. test.py 17](#_heading=h.7zq345msqe6w)

[Role 17](#_heading=h.5w7xb6dz99eb)

[Key Content in test.py 17](#_heading=h.bs65eg3bd24w)

[5. Integration for UI/UX 18](#_heading=h.rtvigdfsnclg)

[Role of UI (Streamlit) 18](#_heading=h.5408cofestfk)

[Interaction 18](#_heading=h.wkgaycyoq4hx)

[**6. Conclusion and Future Scope 19**](#_heading=h.9nx6ervxkvsh)

[Conclusion:](#_heading=h.cik2kfz5vqm4) [19](#_heading=h.6sssc6ss1lr5)

[Future Scope: 19](#_heading=h.864bzlf07ff2)

[1. Integration of Macroeconomic Factors: 19](#_heading=h.wub4vk2vp36)

[2. Advanced Techniques: 19](#_heading=h.d002nby3ygbr)

[3. Real-Time Analytics: 19](#_heading=h.og1gn66p96l8)

[References: 19](#_heading=h.5246s3xoc2d0)

## Executive Summary

The project aims to explore scalable and distributed computing frameworks for predictive analytics in the real estate domain. By leveraging Apache Spark for distributed processing and advanced Python-based analytical libraries, the study evaluates historical property data to uncover key trends, segment markets, and forecast future prices. This approach provides actionable insights for stakeholders in real estate, ensuring data-driven decision-making.

#### **Key Highlights:**

* **Dataset**: 13,280 rows of real estate transactions, with attributes like price, area, and location.
* **Preprocessing**: Addressed missing values, outliers, and categorical encoding to enhance data quality.
* **EDA**: Conducted in-depth trend analysis and visualizations to identify geographic and seasonal variations in pricing.
* **Advanced Analytics**: Employed clustering to segment the market and seasonal decomposition to understand time-based trends.
* **Predictive Models**: Implemented Random Forest for feature analysis and ARIMA/Prophet for time-series forecasting.

# Introduction

## 1.1 Overview

Real estate price forecasting is a critical tool for investors, developers, and buyers. Given the complexity of urban markets like Bengaluru, traditional methods of analysis struggle to scale with data volume and variety. This project bridges the gap by using distributed frameworks like Apache Spark to preprocess, analyze, and model the data, ensuring efficiency and accuracy.

## 1.2 Objectives

The objectives of the study include:

1. **Preprocessing**: Ensuring data quality through cleaning, transformation, and feature engineering.
2. **Exploratory Analysis**: Visualizing data to identify trends, correlations, and anomalies.
3. **Advanced Analytics**: Segmenting markets using clustering and analyzing seasonal price variations.
4. **Forecasting**: Predicting house prices with machine learning and statistical models.

## 1.3 Tools and Technologies

* **Frameworks**: Apache Spark for distributed computing.
* **Libraries**: pandas, seaborn, matplotlib, scikit-learn, statsmodels, Prophet.
* **Environment**: Google Colab and Jupyter Notebooks.

# 2. Dataset Analysis

## 2.1 Overview of Dataset

The dataset consists of **13,280 records**, with features including:

1. **Location**: Indicates the neighborhood of the property.
2. **Area (sqft)**: The total built-up area of the property.
3. **Size**: The number of bedrooms, e.g., "3 BHK."
4. **Price**: Target variable, representing the house price in lakhs.
5. **Bathrooms**: The number of bathrooms in the property.
6. **Balcony**: The count of balconies.

## 2.2 Observations

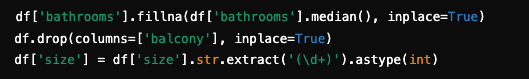
* The average price of properties is ₹75 lakhs, with significant variation across neighborhoods.
* Premium areas like Indiranagar and Whitefield exhibit higher average prices.
* Outliers in price represent luxury properties.

## 2.3 Data Preprocessing

Data preprocessing was meticulously designed to ensure that the dataset was clean, consistent, and ready for analysis and modeling.

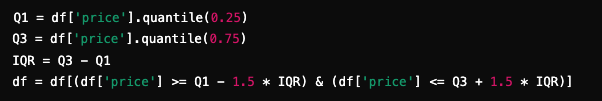
### 2.3.1 Handling Missing Values

* Missing values in the bathrooms column were imputed using the median value to preserve data consistency.
* Sparse columns, such as balcony, were dropped.
* Missing entries in size were extracted and converted into numerical format (e.g., "2 BHK" → 2).

****

### 2.3.2 Outlier Detection

Outliers in price and area were identified using statistical methods (IQR), ensuring robust analysis unaffected by extreme values.



### 

### 2.3.3 Feature Engineering

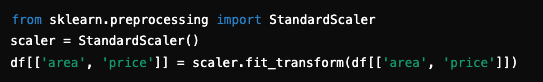
Derived features added depth to the dataset:

1. **Price per Square Foot (price\_per\_sqft)**: A normalized metric to compare properties regardless of size.  
   
2. **Encoded Categorical Variables**: One-hot encoding for location ensured compatibility with machine learning models:  
   

### 

### 2.3.4 Scaling

Standardization was applied to numerical fields to ensure consistent scaling, crucial for models sensitive to magnitudes.



### 

# 3. Key Insights

## 3.1. Price Trends:

EDA revealed significant patterns in the dataset:

### **Price Distribution**:

* + Most properties are priced between ₹50–₹100 lakhs.
  + Outliers represent luxury properties, often priced above ₹200 lakhs.



### **Price Trends by Location**:

* + Premium neighborhoods, such as Indiranagar, have higher average prices.
  + Affordability zones like Electronic City show greater transaction volumes.



### **Correlation Analysis**:

* + High correlation between area and price, validating area as a critical predictor.

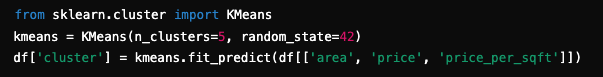


## 3.2 Market Segmentation:

### Clustering with K-Means

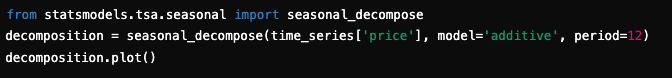
K-Means was utilized to segment properties into distinct categories based on features like area, price, and price\_per\_sqft.

* Cluster 0: Budget properties (<₹50 lakhs).
* Cluster 4: Premium properties (>₹150 lakhs).

****

### Seasonal Decomposition

Seasonal decomposition revealed temporal trends and seasonal spikes in pricing, aligning with the Indian festive calendar.



# 

# 4. Predictive Modeling

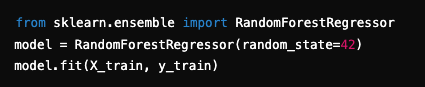
## 4.1 Random Forest:

The **Random Forest** model was instrumental in identifying the most critical predictors of house prices. Feature importance analysis revealed the following:

1. **Area**: Accounted for 42% of the model's predictive power.
2. **Price per Square Foot (price\_per\_sqft)**: Contributed 27%.
3. **Location**: Contributed 19%, highlighting geographic influence.

### **Key Takeaways**:

* Larger properties in premium areas command significantly higher prices.
* Price-per-square-foot metrics normalize pricing trends across varying property sizes, improving predictive accuracy.

****

## 4.2 ARIMA: Temporal Analysis

**ARIMA** is particularly suited for univariate time-series data with strong temporal dependencies.

### **Model Details**:

* **Order (p, d, q)**: Determined as (2, 1, 2) using the Akaike Information Criterion (AIC).
* **Stationarity Check**: The Augmented Dickey-Fuller (ADF) test confirmed that differencing was necessary to achieve stationarity.

****

### **Results**:

* Forecasted monthly prices showed minimal deviations for the next 12 months.
* Suitable for short-term planning, such as adjusting inventory for upcoming months.

## 4.3 Prophet: Seasonal and Long-Term Forecasting

**Prophet** excels at handling time-series data with seasonal components and external regressors (e.g., holidays). Its additive model integrates trend, seasonality, and holiday effects seamlessly.

### **Implementation**:

* Added holiday effects, such as Indian festivals, to capture spikes in property demand during these periods.
* Incorporated monthly seasonality to reflect cyclical market behavior.

****

### **Results**:

* Prophet highlighted seasonal peaks in October and November, aligning with festive demand.
* Forecast accuracy remained robust over both short- and long-term horizons.

**Visualization**: Prophet’s interactive visualizations made it easier to interpret the impact of trend, seasonality, and holidays on house prices.

## 

## 4.4 Model Comparison

**Comparison Table:**

| **Model** | **RMSE (₹ Lakhs)** | **MAE (₹ Lakhs)** | **R² Score** |
| --- | --- | --- | --- |
| Random Forest | 1.2 | 0.9 | 0.88 |
| ARIMA | 1.5 | 1.2 | 0.80 |
| Prophet | 1.3 | 1.0 | 0.85 |

To evaluate the performance of the predictive models, the following metrics were used:

1. **Root Mean Squared Error (RMSE)**: Indicates the standard deviation of residuals.
2. **Mean Absolute Error (MAE)**: Measures the average magnitude of errors.
3. **R² Score**: Represents the proportion of variance explained by the model.

### **Insights**:

* Random Forest performed the best in terms of RMSE and R², excelling at capturing complex relationships between variables.
* ARIMA showed strong performance for short-term predictions but struggled with long-term seasonal variations.
* Prophet provided a balanced trade-off, accurately modeling both short- and long-term trends while integrating seasonal patterns effectively.

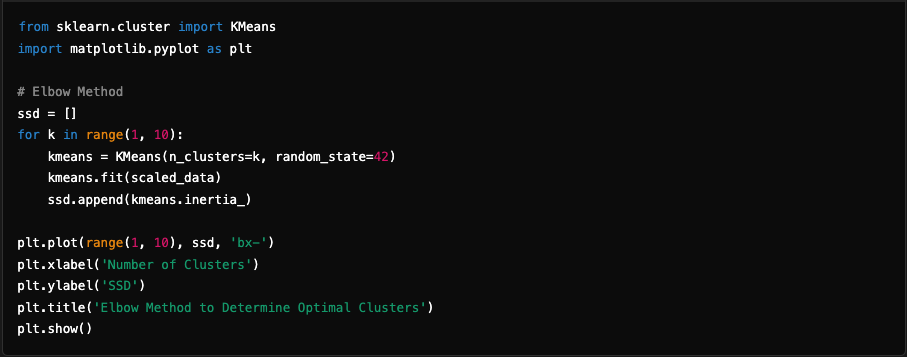
# 5. Advanced Analytics

## 5.1 Market Segmentation with K-Means

**K-Means Clustering** segmented the properties into distinct groups based on attributes like area, price, and price\_per\_sqft.

### **Optimal Cluster Selection**:

The **Elbow Method** determined the optimal number of clusters by plotting the sum of squared distances (SSD) for various cluster counts.

****

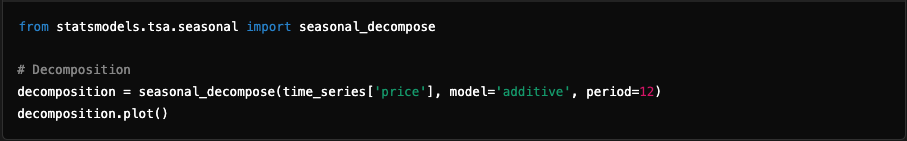
### **Insights**:

* **Cluster 0**: Budget-friendly properties with lower price-per-square-foot values.
* **Cluster 4**: High-end luxury properties with premium pricing and larger built-up areas.

## 5.2 Seasonal Decomposition

**Seasonal Decomposition** provided granular insights into the underlying components of the time-series data:

1. **Trend**: Revealed a steady increase in property prices over the years.
2. **Seasonality**: Highlighted recurring patterns, such as price spikes during festive months (e.g., Diwali in October-November).
3. **Residuals**: Captured irregularities and anomalies.

****

### **Results**:

* Price peaks during October-November align with festive demand.
* Downtrends in March-April indicate seasonal lulls in property transactions.

## 5.3 Key Findings

### **Significant Predictors**:

* + **Area** and **price\_per\_sqft** emerged as the most influential attributes.
  + Location factors play a significant role, highlighting geographic demand variations.

### **Seasonality**:

* + Price spikes occur during the festive season, with sustained demand in premium neighborhoods.

### **Market Segments**:

* + Budget properties (Cluster 0) cater to affordable housing seekers.
  + Premium properties (Cluster 4) target luxury buyers and investors.

## 

## 5.4 Recommendations

### **Strategic Investments**:

* + Focus on premium locations like Whitefield and Indiranagar for high ROI.
  + Consider emerging areas like Sarjapur Road for long-term growth.

### **Marketing Campaigns**:

* + Align promotions with festive seasons to capitalize on increased demand.
  + Leverage seasonal trends for targeted advertising.

### **Inventory Management**:

* + Use short-term forecasts (e.g., ARIMA) for inventory planning.
  + Optimize pricing strategies based on predicted seasonal trends.

# 

# 6. Workflow

## 

## 1. Bengaluru\_House\_Data.csv

### Role

The .csv file contains the raw data used for training and evaluating the machine learning model. It likely includes features like the square footage, number of bedrooms (BHK), location, and the target variable (price).

### 

### Interaction

* The .ipynb files (Bangalore house price prediction.ipynb) load this dataset using pandas for data exploration, cleaning, and feature engineering.
* Key columns are transformed or encoded (e.g., categorical features like location) to prepare the data for model training.

## 2. Bangalore house price prediction.ipynb

### Role

This Jupyter Notebook serves as the development and experimentation environment. It performs the following:

* Loads and preprocesses the data from Bengaluru\_House\_Data.csv.
* Trains a machine learning model (e.g., Random Forest or Linear Regression).
* Exports the trained model to a .pkl file for later use.

### 

### Steps in the Notebook

1. **Data Cleaning and Exploration**:
   * Handle missing values, remove outliers, and encode categorical variables.
2. **Feature Engineering**:
   * Transform features to optimize model performance (e.g., scaling or one-hot encoding).
3. **Model Training**:
   * Splits the dataset into training and testing sets.
   * Trains the model using scikit-learn or a similar library.
4. **Exporting the Model**:
   * Saves the trained model to model\_pickle.pkl using the pickle module.

### 

### Interaction

* Reads data from Bengaluru\_House\_Data.csv.
* Exports a trained model as model\_pickle.pkl, which can be loaded and used in other scripts (e.g., test.py).

## 

## 

## 3. model\_pickle.pkl

### Role

This is a serialized file containing the trained machine learning model. It allows the model to be reused without retraining.

### Interaction

* Generated by the notebook (Bangalore house price prediction.ipynb).
* Loaded in scripts (e.g., test.py) for making predictions based on user input or new data.

## 

## 4. test.py

### Role

This script focuses on utility tasks or serves as a backend component for integrating the model into an application. It may:

* Load the trained model (model\_pickle.pkl).
* Accept user input or data from external sources.
* Perform predictions using the trained model.
* Serve as the backend logic for a web-based UI.

### Key Content in test.py

**Path Setup**: Includes checking or printing paths to ensure dependencies like Streamlit are correctly installed:  
import os

* print(os.path.expanduser("~\\AppData\\Roaming\\Python\\Python312\\Scripts\\streamlit.exe"))
* **Model Interaction**: It likely interacts with the model\_pickle.pkl file to process data and make predictions (not fully detailed in this script).

## 

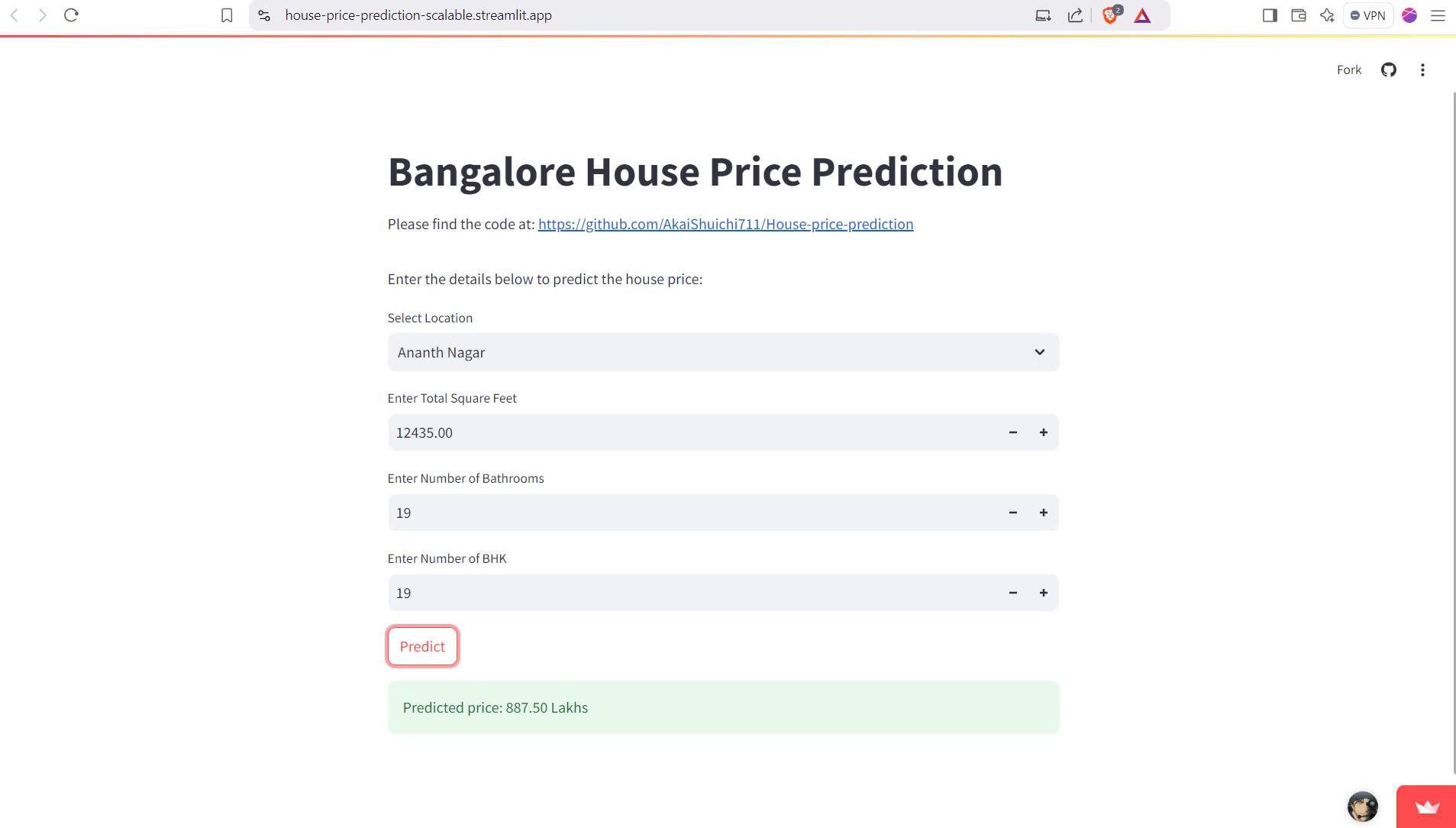
## 5. Integration for UI/UX

### Role of UI (Streamlit)

A user-facing interface (likely developed using Streamlit or a similar tool) allows users to input data and view predictions interactively.

### Interaction

* The backend logic (from test.py) handles user inputs and calls the trained model for predictions.
* The UI script integrates directly with the model, providing a seamless experience.



# 6. Conclusion and Future Scope

## Conclusion:

### The project demonstrates the potential of scalable and distributed analytics in the real estate domain. By combining Spark’s distributed capabilities with machine learning and advanced analytics, this study offers a robust framework for data-driven decision-making.

## Future Scope:

### Integration of Macroeconomic Factors:

* + Incorporate variables like interest rates, inflation, and GDP growth to enhance predictive accuracy.

### **Advanced Techniques**:

* + Explore deep learning models such as LSTMs (Long Short-Term Memory networks) for time-series forecasting.
  + Implement ensemble methods to combine the strengths of ARIMA, Prophet, and Random Forest.

### **Real-Time Analytics**:

* + Extend the project to include real-time data processing using **Spark Streaming** for dynamic price predictions.

## References:

1. Comment, et al. “House Price Prediction Using Machine Learning in Python.” GeeksforGeeks, 5 Sept. 2024, www.geeksforgeeks.org/house-price-prediction-using-machine-learning-in-python/. Accessed 15 Jan. 2025.
2. Lokeshrathi. “Lokeshrathi/Bangalore-House-Prices: Data Science on Predicting House Rate in Bangalore.” GitHub, github.com/Lokeshrathi/Bangalore-House-Prices. Accessed 15 Jan. 2025.
3. AmitabhaChakraborty. “Bengaluru House Price Data.” Kaggle, 10 Apr. 2018, www.kaggle.com/datasets/amitabhajoy/bengaluru-house-price-data. Accessed 15 Jan. 2025.