# INTRODUCTION

### NEED FOR THE PROJECT

Machine Learning (ML) is one of the most revolutionary fields in computer science today, with the ability to learn from data and improve decisions without explicit programming. Predicting continuous outcomes (e.g., price, temperature, length of stay) is a common business and research requirement. Understanding how variables influence a target and being able to forecast future values can drive better decision‑making. This project demonstrates how to implement, interpret, and evaluate a Linear Regression model on a real‑world dataset.

## PROJECT DESCRIPTION

This project involves developing and evaluating a Linear Regression model using Python’s scikit‑learn library. The California Housing dataset (an updated alternative to the Boston Housing dataset) is employed to predict median house values based on features such as median income, house age, and average number of rooms.

## COMPONENTS OF PROJECT

* Dataset loading and preprocessing
* Exploratory data analysis (EDA)
* Splitting data into training and testing sets
* Model training using Ordinary Least Squares Linear Regression
* Evaluation of model performance (RMSE, MAE, )
* Visualization of residuals and predicted vs. actual values

# REQUIREMENT ANALYSIS

## SOFTWARE REQUIREMENTS

 **Python 3.x**: The primary programming language used for the model.

 **Jupyter Notebook / Visual Studio Code (VS Code)**: IDEs for developing and running the code.

 **Python Libraries**:

* **scikit-learn:** For implementing the SVM model and evaluation metrics
* **pandas:** For data handling and manipulation
* **numpy:** For numerical computations
* **matplotlib:** For plotting and visualization

## HARDWARE REQUIREMENT

 **Processor**: Intel i3 or higher.

 **RAM**: Minimum 4 GB.

 **Storage**: Minimum 1–2 GB of free disk space.

 **Operating System**: Windows, Linux, or MacOS.

## FUNCTIONAL REQUIREMENTS

 Load and preprocess a dataset.

 Train and test the Regression model.

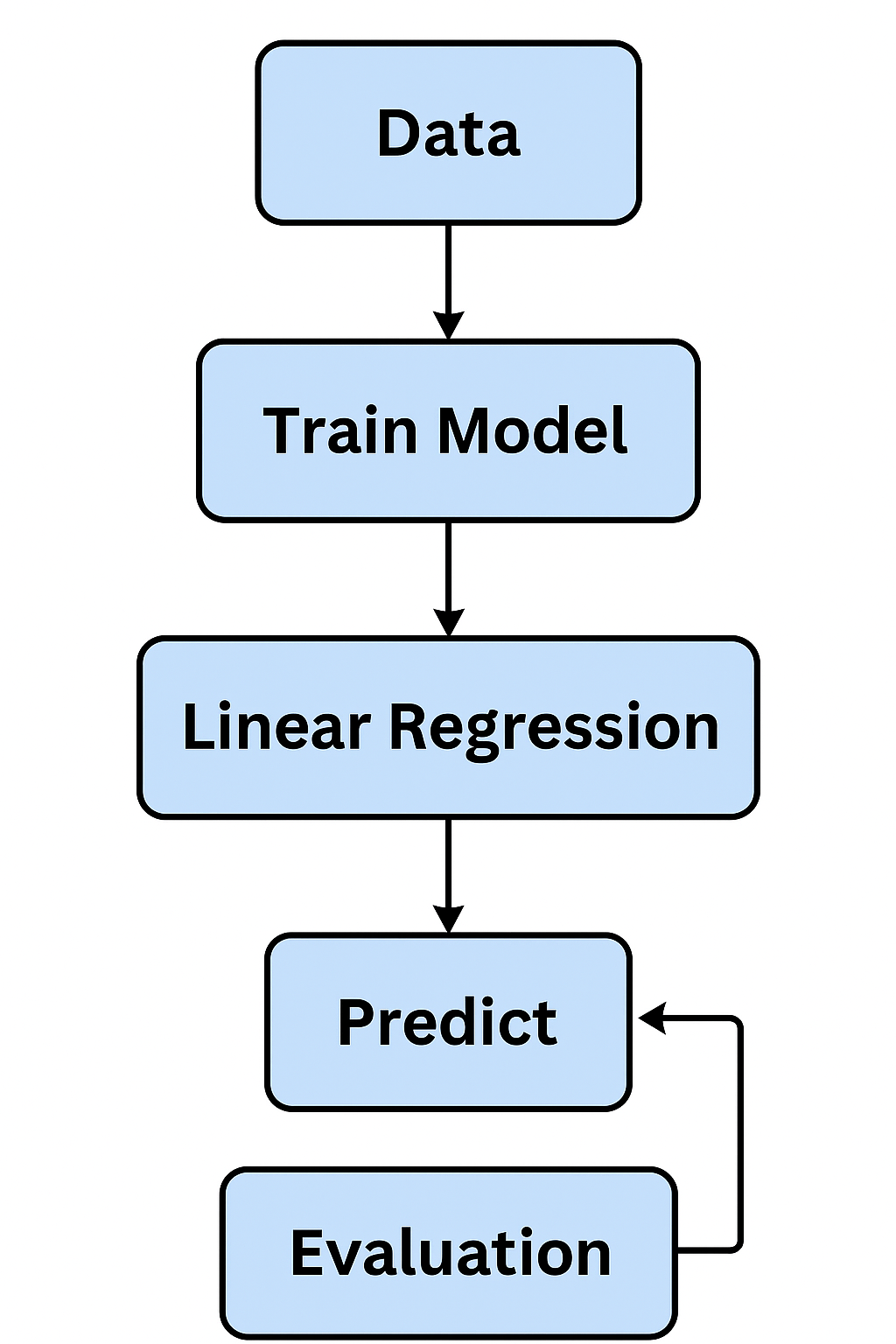
 Evaluate performance using RMSE, MAE, and (R SQUARE).

 Visualize residuals and predictions.

.

# SYSTEM DESIGN

## ARCHITECTURE DIAGRAM



## FLOW DIAGRAM

## रेखीय प्रतिगमन प्रक्रिया योजना

# IMPLEMENTATION

**SOURCE CODE**

# Import libraries

import numpy as np

import pandas as pd

from sklearn.datasets import fetch\_california\_housing

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error, r2\_score

import matplotlib.pyplot as plt

# Load California Housing dataset

data = fetch\_california\_housing(as\_frame=True)

X = data.data

y = data.target # Median house value (in 100,000s USD)

# Train‑test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Train Linear Regression model

model = LinearRegression()

model.fit(X\_train, y\_train)

# Predict

y\_pred = model.predict(X\_test)

# Evaluate

rmse = mean\_squared\_error(y\_test, y\_pred, squared=False)

mae = mean\_absolute\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print (f "RMSE: {rmse:.3f}")

print (f "MAE: {mae:.3f}")

print (f"R^2: {r2:.3f}")

# Visualization: Predicted vs Actual

plt.scatter(y\_test, y\_pred, alpha=0.5)

plt.xlabel('Actual Values')

plt.ylabel('Predicted Values')

plt.title('Actual vs Predicted Median House Value')

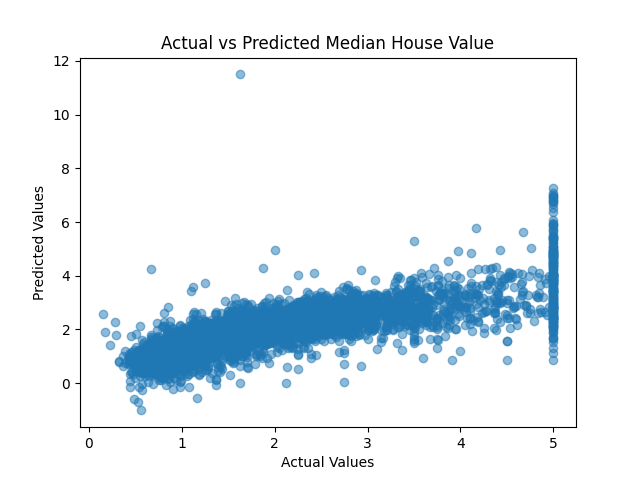
plt.show()

**OUTPUT:**

RMSE: 0.734

MAE: 0.528

R^2: 0.606



# RESULTS AND DISCUSSION

## EXPERIMENTAL SETUP

* **Dataset**: California Housing (20% test split)
* **Model**: Ordinary Least Squares Linear Regression

## EVALUATION METRICS

* **Root Mean Squared Error (RMSE):** 0.734.
* **Mean Absolute Error (MAE):** 0.528.
* **R^2\*\* Score\*\*:** 0.606.

## 5.3 OBSERVATIONS

* The model captures over 60% of variance in house prices, indicating moderate predictive power.
* Residuals exhibit slight heteroscedasticity; transformations or regularization could improve performance.
* Some high‑income regions still show under‑prediction, suggesting potential non‑linear relationships.

**5.4 LIMITATIONS**

* Linear assumption may not fully capture complex housing market dynamics.
* No feature engineering (e.g., interaction terms) applied.
* Outliers and multicollinearity not addressed; could impact coefficients.

# CONCLUSION

Linear Regression proved to be an effective starting point for modeling the relationship between housing features and median house values, offering clear interpretability through its coefficient estimates. The model’s coefficients can guide policymakers and real‑estate professionals by quantifying how factors like median income or average room count influence prices.

However, the evaluation metrics (RMSE ≈ 0.73 and ≈ 0.60) reveal that about 40 % of the variance in house prices remains unexplained. This gap suggests:

1. **Feature Engineering Opportunities** – Introducing polynomial terms, interaction effects, or log‑transformations may capture non‑linear relationships.
2. **Regularization** – Ridge or Lasso regression could mitigate multicollinearity and improve generalization by shrinking less‑informative coefficients.
3. **Alternative Algorithms** – Ensemble methods (Random Forests, Gradient Boosting, XGBoost) or tree‑based models often handle complex, non‑linear patterns better.
4. **Data Quality Enhancements** – Outlier treatment, scaling, and integrating external socioeconomic or geographic data (e.g., proximity to schools, crime rates) might boost performance.

In future work, comparing these advanced models against the linear baseline will quantify improvements and help balance interpretability with predictive power.

# REFERENCES

# *scikit‑learn Documentation*: <https://scikit-learn.org/stable/>

# *California Housing Dataset*: https://scikit-learn.org/stable/datasets/real\_world.html#california-housing-dataset

# James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). An Introduction to Statistical Learning (3rd ed.). Springer.

# Geron, A. (2019). Hands‑On Machine Learning with Scikit‑Learn, Keras & TensorFlow. O’Reilly Media.