JSPM's

RAJARSHI SHAHU COLLEGE OF ENGINEERING

TATHAWADE, PUNE-33

DEPARTMENT OF COMPUTER ENGINEERING

Case Study

on

House Price Prediction using Linear, Ridge and Polynomial Regression

– Submitted by –

Name of Student: Adarsh Gangshettiwar

Roll No: CS3274

PRN No: RBTL24CS145

Class and Division: 3rd year B Division

Course: Machine Learning

1. Title

House Price Prediction using Linear, Ridge and Polynomial Regression

2. Background/ Introduction

Accurate prediction of house prices is essential for buyers, sellers, and investors in the real estate market. Traditional estimation methods rely heavily on manual calculations or expert judgment, which may not be precise or scalable.

Machine Learning provides a powerful approach to predict house prices based on historical data. By analyzing features such as area, number of bedrooms, bathrooms, and stories, regression algorithms can learn patterns in the data and make accurate predictions for new houses.

In this project, we implement Linear Regression, Ridge Regression, and Polynomial Regression to predict house prices and compare their performance using metrics such as R^2 Score and RMSE.

3. Problem Statement

The main challenge is to design an accurate and efficient system that predicts house prices using historical property data. The system should identify the relationships between different property features and price, and provide predictions with minimal error.

4. Objectives

- To collect and preprocess house price data for predictive modeling.
- To explore regression algorithms: Linear Regression, Ridge Regression, and Polynomial Regression.
- To evaluate and compare model performance using R^2 Score and RMSE.
- To predict the price of new houses based on learned models.

5. Libraries required

- NumPy – For numerical operations
- Pandas – For data manipulation and analysis
- Scikit-learn – For machine learning algorithms and evaluation metrics
- Matplotlib / Seaborn – For visualization (optional)

6. Approach/ Methodology

The project follows a standard machine learning workflow:

1. **Data Collection:**
    - Use a dataset containing house details and prices (House_Price.csv).
2. **Data Preprocessing:**
    - Check for missing values and clean the dataset if necessary.
    - Select relevant features such as Area, Bedrooms, Bathrooms, and Stories.
3. **Feature & Target Selection:**
    - Define features X and target y (Price).
4. **Train-Test Split:**
    - Split the dataset into training and testing sets (80%-20%).
5. **Model Training:**
    - Train Linear Regression, Ridge Regression, and Polynomial Regression models.
6. **Model Evaluation:**
    - Evaluate models using R^2 Score and RMSE to compare accuracy and error.
7. **Prediction:**
    - Use the trained models to predict house prices for new input data.

7. Implementation

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, Ridge
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import make_pipeline
from sklearn.metrics import r2_score, mean_squared_error

# Load Dataset
df = pd.read_csv("House_Price_Predictions.csv")
print("Dataset Loaded Successfully")
print(df.head())

# Define Features & Target
X = df[['Area', 'Bedrooms', 'Bathrooms', 'Stories']]
y = df['Price']
```

```
# Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# Linear Regression
lin_reg = LinearRegression()
lin_reg.fit(X_train, y_train)
y_pred_lin = lin_reg.predict(X_test)
print("Linear Regression R² Score:", r2_score(y_test, y_pred_lin))
print("Linear Regression RMSE:", np.sqrt(mean_squared_error(y_test,
y_pred_lin)))

# Ridge Regression
ridge_reg = Ridge(alpha=1.0)
ridge_reg.fit(X_train, y_train)
y_pred_ridge = ridge_reg.predict(X_test)
print("Ridge Regression R² Score:", r2_score(y_test, y_pred_ridge))
print("Ridge Regression RMSE:", np.sqrt(mean_squared_error(y_test,
y_pred_ridge)))

# Polynomial Regression (Degree=2)
poly_model = make_pipeline(PolynomialFeatures(degree=2),
LinearRegression())
poly_model.fit(X_train, y_train)
y_pred_poly = poly_model.predict(X_test)
print("Polynomial Regression R² Score:", r2_score(y_test, y_pred_poly))
print("Polynomial Regression RMSE:", np.sqrt(mean_squared_error(y_test,
y_pred_poly)))

# Model Comparison
results = pd.DataFrame({
    'Model': ['Linear', 'Ridge', 'Polynomial (deg=2)'],
    'R2 Score': [
        r2_score(y_test, y_pred_lin),
        r2_score(y_test, y_pred_ridge),
        r2_score(y_test, y_pred_poly)
    ],
    'RMSE': [
```

```
        np.sqrt(mean_squared_error(y_test, y_pred_lin)),
        np.sqrt(mean_squared_error(y_test, y_pred_ridge)),
        np.sqrt(mean_squared_error(y_test, y_pred_poly))
    ]
})

print("\nModel Performance Comparison:\n")
print(results)

# Sample Prediction
sample = pd.DataFrame({
    'Area': [2000],
    'Bedrooms': [3],
    'Bathrooms': [2],
    'Stories': [1]
})
predicted_price = lin_reg.predict(sample)[0]
print(f"\nPredicted Price (Linear Regression): ${predicted_price:,.2f}")
```

8. GitHub Link:

https://github.com/AdarshG07/House_Price_Prediction.git

9.Results

Dataset Loaded Successfully

	Area	Bedrooms	Bathrooms	Stories	Price
0	1500	3	2	1	320000
1	1800	4	2	2	400000
2	2400	4	3	2	540000
3	3000	5	4	2	750000
4	1200	2	1	1	210000

Linear Regression R^2 Score: 0.9639498581248424

Linear Regression RMSE: 22525.66862254802

Ridge Regression R^2 Score: 0.9793060409244201

Ridge Regression RMSE: 17066.56011001595

Polynomial Regression R^2 Score: 0.9864889879504827

Polynomial Regression RMSE: 13790.123081283811

Model Performance Comparison:

	Model	R2 Score	RMSE
0	Linear	0.963950	22525.668623
1	Ridge	0.979306	17066.560110
2	Polynomial (deg=2)	0.986489	13790.123081

Predicted Price (Linear Regression): $402,170.05

10.Conclusion

The House Price Prediction project demonstrates how different regression techniques can be applied to predict property prices.

- **Linear Regression** provides a simple baseline model.
- **Ridge Regression** improves generalization by reducing overfitting.
- **Polynomial Regression** captures non-linear relationships and performs best on this dataset.

This project highlights the importance of selecting appropriate regression techniques and showcases how machine learning can assist in accurate price prediction for real estate applications.