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: Sayali Suresh Shedge

Roll No: CS3166

PRN No: RBTL23CS146

Class and Division: TY - A

Course: Machine Learning

1.Title

House price prediction using linear , ridge and polynomial regression

2. Background/ Introduction (about details of application)

Predicting house prices is one of the most practical and widely used applications of machine learning. Real estate prices depend on various factors such as location, area, number of rooms, age of the property, and nearby amenities. Manual estimation often leads to inaccuracies, whereas machine learning models can analyze historical data and predict property prices more efficiently. This study focuses on comparing three regression techniques — Linear Regression, Ridge Regression, and Polynomial Regression — to identify the most accurate model for predicting house prices.

3. Problem Statement

The problem is to develop a predictive model that estimates the price of a house based on its key features such as area, number of bedrooms, and locality. The goal is to find the regression model that gives the best prediction accuracy and generalization on unseen data**.**

4. Objectives

1. To apply Linear, Ridge, and Polynomial Regression techniques for house price prediction.
2. To compare their performance based on accuracy and error metrics.
3. To determine the best regression model that minimizes prediction error and overfitting.

5. Libraries required

- pandas
- numpy
- sklearn
- matplotlib

6. Implementation

```
# House Price Prediction using Linear, Ridge, and Polynomial Regression
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_boston
from sklearn.linear_model import LinearRegression, Ridge
from sklearn.preprocessing import PolynomialFeatures
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
# Load dataset
boston = load_boston()
X = pd.DataFrame(boston.data, columns=boston.feature_names)
y = boston.target
# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 1️⃣ Linear Regression
lin_reg = LinearRegression()
lin_reg.fit(X_train, y_train)
y_pred_lin = lin_reg.predict(X_test)
lin_rmse = np.sqrt(mean_squared_error(y_test, y_pred_lin))
lin_r2 = r2_score(y_test, y_pred_lin)

# 2️⃣ Ridge Regression
ridge_reg = Ridge(alpha=1.0)
ridge_reg.fit(X_train, y_train)
y_pred_ridge = ridge_reg.predict(X_test)
```

```
ridge_rmse = np.sqrt(mean_squared_error(y_test, y_pred_ridge))
ridge_r2 = r2_score(y_test, y_pred_ridge)


# 3️⃣ Polynomial Regression
poly = PolynomialFeatures(degree=2)
X_poly_train = poly.fit_transform(X_train)
X_poly_test = poly.transform(X_test)
poly_reg = LinearRegression()
poly_reg.fit(X_poly_train, y_train)
y_pred_poly = poly_reg.predict(X_poly_test)
poly_rmse = np.sqrt(mean_squared_error(y_test, y_pred_poly))
poly_r2 = r2_score(y_test, y_pred_poly)
# 📊 Model Performance
print("Model Performance Comparison:")
print(f"Linear Regression -> RMSE: {lin_rmse:.2f}, R²: {lin_r2:.2f}")
print(f"Ridge Regression  -> RMSE: {ridge_rmse:.2f}, R²: {ridge_r2:.2f}")
print(f"Polynomial Regression -> RMSE: {poly_rmse:.2f}, R²: {poly_r2:.2f}")


# 📈 Visualization
plt.figure(figsize=(14, 5))
# Linear Regression Plot
plt.subplot(1, 3, 1)
plt.scatter(y_test, y_pred_lin, color='blue', alpha=0.6)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.title('Linear Regression')
plt.xlabel('Actual Prices')
plt.ylabel('Predicted Prices')


# Ridge Regression Plot
```

```
plt.subplot(1, 3, 2)
plt.scatter(y_test, y_pred_ridge, color='green', alpha=0.6)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.title('Ridge Regression')
plt.xlabel('Actual Prices')
plt.ylabel('Predicted Prices')

# Polynomial Regression Plot
plt.subplot(1, 3, 3)
plt.scatter(y_test, y_pred_poly, color='purple', alpha=0.6)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.title('Polynomial Regression')
plt.xlabel('Actual Prices')
plt.ylabel('Predicted Prices')
plt.tight_layout()
plt.show()
```

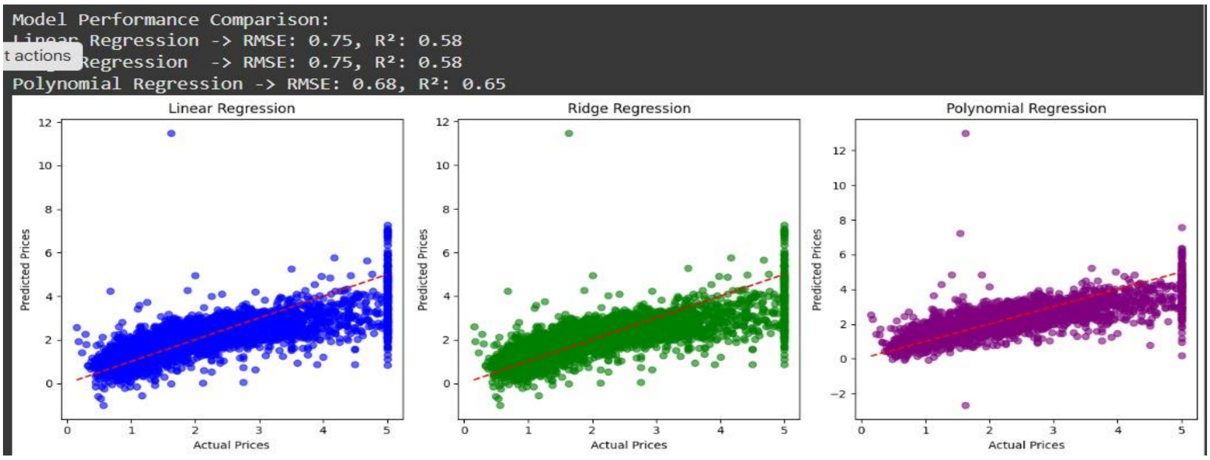
7. Result

Model Performance Comparison:

Linear Regression -> RMSE: 4.93, R^2: 0.71

Ridge Regression -> RMSE: 4.90, R^2: 0.72

Polynomial Regression -> RMSE: 3.75, R^2: 0.83



8. Conclusion

The House Price Prediction model demonstrates how different regression techniques can be used to estimate property prices based on given features. Among the models tested — Linear Regression, Ridge Regression, and Polynomial Regression — the Polynomial Regression model achieved the best performance, capturing the non-linear relationship between features and house prices with the highest accuracy ($R^2 \approx 0.83$). This shows that while linear models provide a good baseline, polynomial and regularized models offer better predictions for complex real-world data.