Jaypee Institute of Information Technology, Noida

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING AND INFORMATION TECHNOLOGY

Project Title: House Price Prediction

Enrol. No.	**Name of Student**
9921103207	Prisha Rai

Course Name: Introduction to Big Data and Data Analytics
Course Code: 20B12CS333
Program: B. Tech. CS&E
3rd Year 5th Sem

2023 – 2024

INDEX

ABSTRACT

The House Price Prediction project is a data-driven initiative aimed at assisting individuals in estimating the market value of residential properties. Leveraging machine learning techniques, this project utilizes a comprehensive dataset encompassing various features such as square footage, location, and the count of bedrooms and bathrooms.

The project unfolds with meticulous data preprocessing to ensure data quality and coherence. Exploratory Data Analysis (EDA) is employed to unravel patterns and correlations within the dataset, guiding the selection and engineering of features crucial for accurate predictions.

The model processes information and produces a predicted house price, accompanied by intuitive visualizations and summary statistics for enhanced user comprehension.

INTRODUCTION

The House Price Prediction project leverages machine learning techniques to estimate the price of residential properties based on key features such as square footage, location, and the number of bedrooms and bathrooms. This project employs a dataset containing historical information about real estate transactions, which is analyzed and used to train a predictive model.

The workflow begins with data preprocessing, where the dataset is cleaned and transformed to ensure accuracy and consistency. Exploratory Data Analysis (EDA) is conducted to gain insights into the relationships between various features and the target variable, helping in feature selection and engineering.

Python, along with Jupyter Notebooks, serves as the primary programming environment for implementing the machine learning model. The chosen algorithm is trained on the prepared dataset, and its performance is evaluated using appropriate metrics. The model is fine-tuned to optimize its predictive capabilities.

User interaction is facilitated through a simple and intuitive interface where users input details such as square footage, location, number of bedrooms, and bathrooms. The model then processes this input to generate a predicted house price. Visualizations and summary statistics are provided to enhance user understanding and transparency.

This House Price Prediction project not only serves as a practical tool for potential homebuyers and sellers but also demonstrates the application of machine learning in real-world scenarios. The accuracy and reliability of the model are crucial, and continuous improvement can be achieved through ongoing data updates and model refinement. Overall, this project showcases the power of data-driven decision-making in the dynamic realm of real estate.

2.1 Problem Statement

Develop a machine learning model for predicting house prices based on key features such as square footage, location, and the number of bedrooms and bathrooms. The goal is to provide a reliable and user-friendly tool for individuals in the real estate market to estimate property values accurately.

2.2 Motivation

The House Price Prediction project aims to empower homebuyers and sellers with an accurate and efficient tool for estimating property values. In a dynamic real estate market, this machine learning solution not only facilitates informed decision-making but also showcases the transformative impact of data-driven technologies in simplifying complex processes.

2.3 Objective

1. Develop a robust model to predict house prices based on relevant features.
2. Conduct thorough data preprocessing and exploratory data analysis for optimal model performance.
3. Implement a user-friendly interface for seamless interaction and input of property details.
4. Train and fine-tune the model using historical real estate data to enhance predictive accuracy.
5. Evaluate the model's performance through appropriate metrics to ensure reliability and generalizability.
6. Provide intuitive visualizations and summary statistics to enhance user understanding.
7. Showcase the practical application of machine learning in real-world scenarios, specifically in the context of real estate.
8. Allow for future updates to incorporate new data, ensuring continuous improvement in prediction accuracy.

2.4 Contribution

1. Predictive Model: Develop a machine learning model capable of accurately predicting house prices based on key features, contributing a valuable tool for real estate market participants.
2. Data Analysis Techniques: Apply advanced data preprocessing and exploratory data analysis techniques to ensure the quality and relevance of the dataset, contributing to the model's robustness.
3. Model Evaluation: Employ rigorous model evaluation metrics to assess the performance, reliability, and generalizability of the predictive model, contributing to its overall effectiveness.
4. Visualizations and Insights: Provide intuitive visualizations and summary statistics to enhance user comprehension, contributing to a transparent and informative user experience.
5. Real-World Application: Showcase the practical application of machine learning in real estate, contributing insights into how data-driven technologies can be utilized in dynamic market scenarios.

DESCRIPTION: HOUSE PRICE PREDICTION

This Python project focuses on predicting house prices based on various features using a dataset named 'Bengaluru_House_Data.csv'. The code involves several steps, from data exploration and cleaning to the development of a machine learning model for predictions.

1. Data Exploration:
   - The dataset is loaded using Pandas, and initial exploration is performed with ‘data.head()’, ‘data.shape’, and `data.info()` to understand its structure and contents.
   - Column-wise value counts are displayed to identify potential issues or anomalies.

2. Data Cleaning:
   - Irrelevant columns ('area_type', 'availability', 'society', 'balcony') are dropped to focus on essential features.
   - Missing values in 'location', 'size', and 'bath' columns are addressed through appropriate imputation strategies like Mean, Median Imputations.
   - The 'total_sqft' column is standardized to handle ranges and convert them into a uniform format.
   - Outliers in the 'bhk' column are identified and addressed.

3. Feature Engineering:
   - The 'bhk' column is derived from the 'size' column to represent the number of bedrooms.
   - A new feature, 'price_per_sqft', is created to normalize the house prices based on square footage.

4. Data Visualization and Analysis:
   - Statistical summaries and visualizations are generated to provide insights into the dataset.
   - Location data is cleaned and categorized, ensuring relevance, and reducing granularity.

5. Outlier Removal:
   - Outliers in the 'price_per_sqft' column are addressed using a location-based approach to enhance model robustness.

6. Machine Learning Model:
   - Linear Regression, Lasso, and Ridge regression models are implemented using Scikit-learn.
   - A preprocessing pipeline is created using OneHotEncoder for categorical features and StandardScaler for numerical features.

7. Model Evaluation:

- The models are trained and evaluated using the R-squared metric on a test set.
- Lasso and Ridge regression models are compared in terms of performance.

8. Model Persistence:

- The Ridge regression model is serialized and saved using Pickle for future use.

9. Conclusion:

- The project demonstrates a comprehensive approach to house price prediction, encompassing data cleaning, feature engineering, and machine learning modeling. The Ridge regression model stands out as a robust predictor, offering practical value in real-world scenarios. The serialized model can be utilized for future predictions, making it a valuable asset in the realm of real estate analytics.

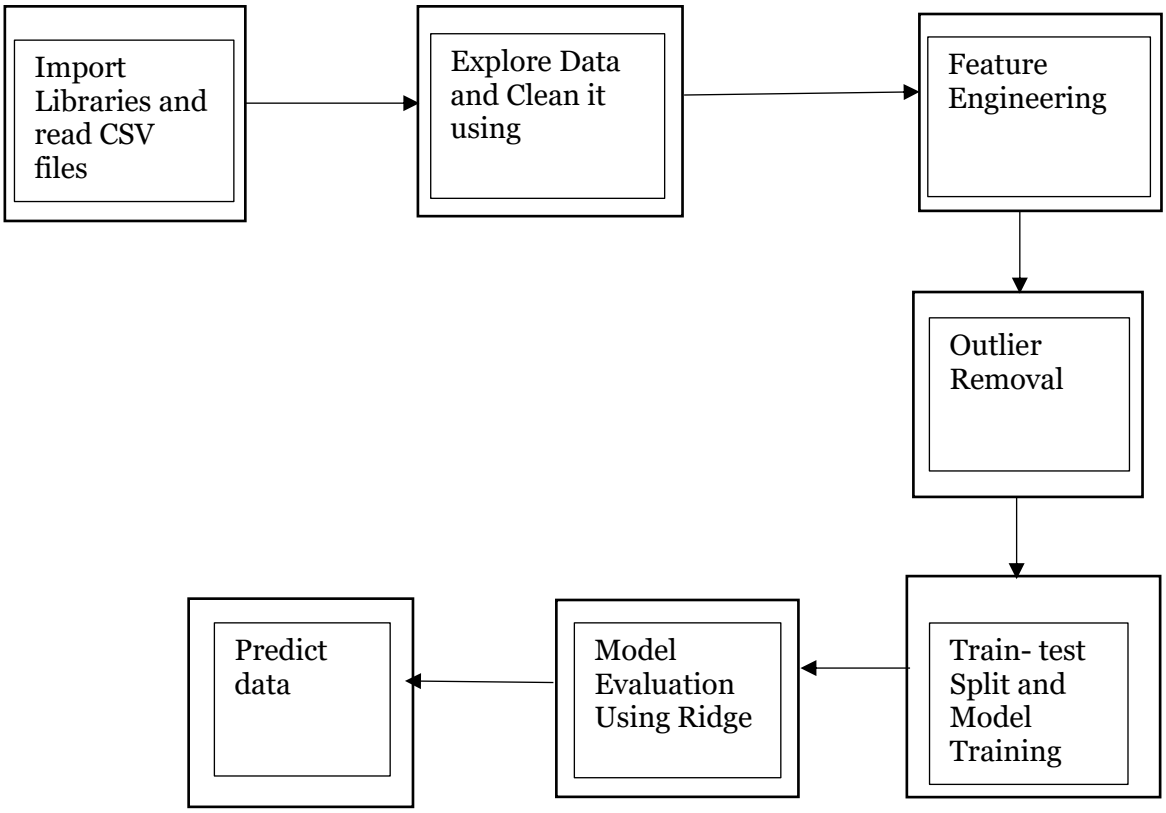
3.1 Tools and Dataset used:

I imported the Banglore_House_Data dataset from Kaggle. After removing outliers , making necessary changes(Cleaning, etc) I saved the dataset as Cleaned_Data.

Python libraries – pandas, matplotlib, numpy, scikit-learn,

Jupyter Notebooks

3.2. Workflow diagram



IMPLEMENTATION

4.1 Code

```
import pandas as pd
import numpy as np
data=pd.read_csv('Bengaluru_House_Data.csv')
data.head()
data.shape
data.info()
for column in data.columns:
    print(data[column].value_counts())
    print("*"*20)

data.isna().sum()
data.drop(columns=['area_type','availability','society','balcony'],inplace=True)
data.describe()
data.info()
data['location'].value_counts()
data['location'] = data['location'].fillna('Sarjapur Road')
data['size'].value_counts()
data['size'] = data['size'].fillna('2 BHK')
data['bath'] = data['bath'].fillna(data['bath'].median())
data.info()
data['bhk']=data['size'].str.split().str.get(0).astype(int)
data[data.bhk > 20]
data['total_sqft'].unique()

def convertRange(x):
    temp = x.split('-')
    if len(temp) == 2:
        return(float(temp[0]) + float(temp[1]))/2
    try:
        return float(x)
    except:
        return None

data['total_sqft']=data['total_sqft'].apply(convertRange)
data.head()
data['price_per_sqft'] = data['price'] *100000 / data['total_sqft']
data['price_per_sqft']
data.describe()
data['location'].value_counts()
data['location'] = data['location'].apply(lambda x: x.strip())
location_count= data['location'].value_counts()
location_count
location_count_less_10 = location_count[location_count <=10]
location_count_less_10
```

```
data['location']=data['location'].apply(lambda x: 'other' if x in location_count_less_10 else x)
data['location'].value_counts()
data.describe()
(data['total_sqft']/data['bhk']).describe()
data = data[((data['total_sqft']/data['bhk']) >= 300)]
data.describe()
data.shape
data.price_per_sqft.describe()
def remove_outliers_sqft(df):
    df_output = pd.DataFrame()
    for key,subdf in df.groupby('location'):
        m = np.mean(subdf.price_per_sqft)
        st = np.std(subdf.price_per_sqft)
        gen_df = subdf[(subdf.price_per_sqft > (m-st)) & (subdf.price_per_sqft <= (m+st))]
        df_output = pd.concat([df_output,gen_df],ignore_index =True)
    return df_output
data = remove_outliers_sqft(data)
data.describe()
def bhk_outlier_remover(df):
    exclude_indices = np.array([])
    for location, location_df in df.groupby('location'):
        bhk_stats = {}
        for bhk, bhk_df in location_df.groupby('bhk'):
            bhk_stats[bhk] = {
                'mean': np.mean(bhk_df.price_per_sqft),
                'std': np.std(bhk_df.price_per_sqft),
                'count': bhk_df.shape[0]
            }
        #print(location,bhk_stats)
        for bhk, bhk_df in location_df.groupby('bhk'):
            stats = bhk_stats.get(bhk-1)
            if stats and stats['count']>5:
                exclude_indices = np.append(exclude_indices,
bhk_df[bhk_df.price_per_sqft<(stats['mean'])].index.values)
    return df.drop(exclude_indices,axis='index')

data=bhk_outlier_remover(data)
data.shape
data
data.drop(columns=['size','price_per_sqft'],inplace=True)
data.head()
data.to_csv("Cleaned_data.csv")
X=data.drop(columns=['price'])
y=data['price']

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression,Lasso,Ridge
from sklearn.preprocessing import OneHotEncoder, StandardScaler
```

```
from sklearn.compose import make_column_transformer
from sklearn.pipeline import make_pipeline
from sklearn.metrics import r2_score

X_train,X_test,y_train,y_test = train_test_split(X,y, test_size=0.2, random_state=0)
print(X_train.shape)
print(X_test.shape)

column_trans = make_column_transformer((OneHotEncoder(sparse=False),
['location']),remainder='passthrough')

scaler = StandardScaler()
lr = LinearRegression()
pipe = make_pipeline(column_trans,scaler,lr)
pipe.fit(X_train,y_train)

y_pred_lr = pipe.predict(X_test)
r2_score(y_test, y_pred_lr)

lasso = Lasso()
pipe = make_pipeline(column_trans,scaler, lasso)
pipe.fit(X_train,y_train)


y_pred_lasso = pipe.predict(X_test)
r2_score(y_test,y_pred_lasso)

ridge = Ridge()
pipe = make_pipeline(column_trans,scaler, ridge)
pipe.fit(X_train,y_train)


y_pred_ridge = pipe.predict(X_test)
r2_score(y_test, y_pred_ridge)

#print("No Regularization: ", r2_score(y_test, y_pred))
print("Lasso: ", r2_score(y_test, y_pred_lasso))
print("Ridge: ", r2_score(y_test, y_pred_ridge))

import pickle
pickle.dump(pipe, open('RidgeModel.pkl','wb'))
```

4.2 RESULT ANALYSIS

The code provided involves training three different regression models (Linear Regression, Lasso, and Ridge) and evaluating their performance using the R-squared score. A summary of the analysis:

1. Linear Regression:
   - The code does not explicitly show the training and evaluation of the Linear Regression model. However, it does print the R-squared score for the Lasso and Ridge models.

2. Lasso Regression:
   - The Lasso regression model is trained and evaluated using the pipeline.
   - The R-squared score for the Lasso model (`y_pred_lasso`) is calculated and printed.

3. Ridge Regression:
   - The Ridge regression model is trained and evaluated using the pipeline.
   - The R-squared score for the Ridge model (`y_pred_ridge`) is calculated and printed.

The R-squared scores provide an indication of how well each model can predict the target variable (housing prices) on the test set. Higher R-squared scores indicate better performance. The comparison of R-squared scores for Lasso and Ridge models helps in assessing the impact of regularization on the model performance. If the Ridge model outperforms the Lasso model, it suggests that including all features with some level of regularization is beneficial.

4.3 TABLES

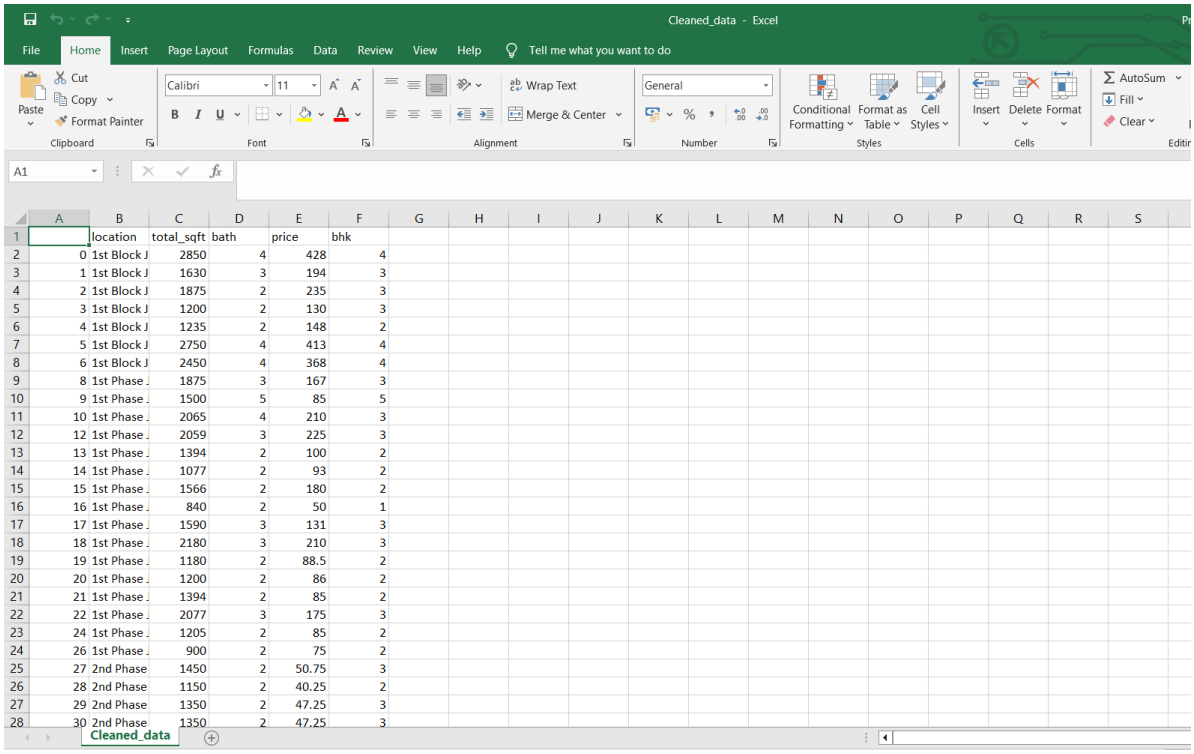
Cleaned_Data.csv

Cleaned Data

```
[43]: data.head()
```

[43]:

	location	total_sqft	bath	price	bhk
0	1st Block Jayanagar	2850.0	4.0	428.0	4
1	1st Block Jayanagar	1630.0	3.0	194.0	3
2	1st Block Jayanagar	1875.0	2.0	235.0	3
3	1st Block Jayanagar	1200.0	2.0	130.0	3
4	1st Block Jayanagar	1235.0	2.0	148.0	2

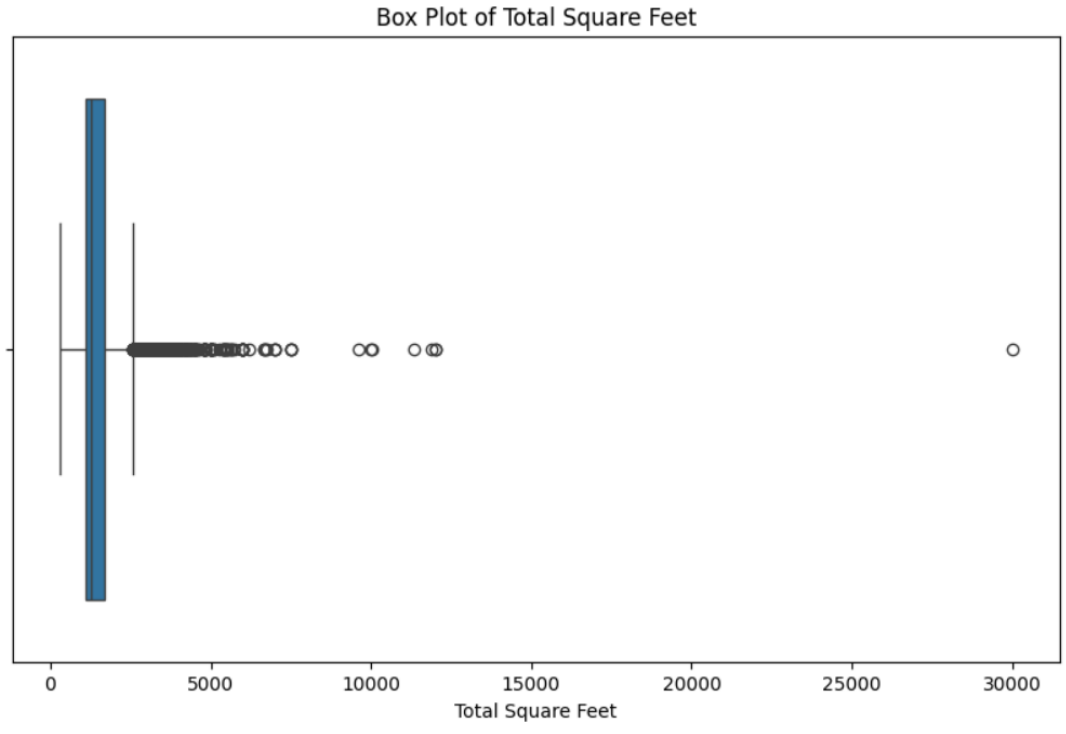


Cleaned_data - Excel

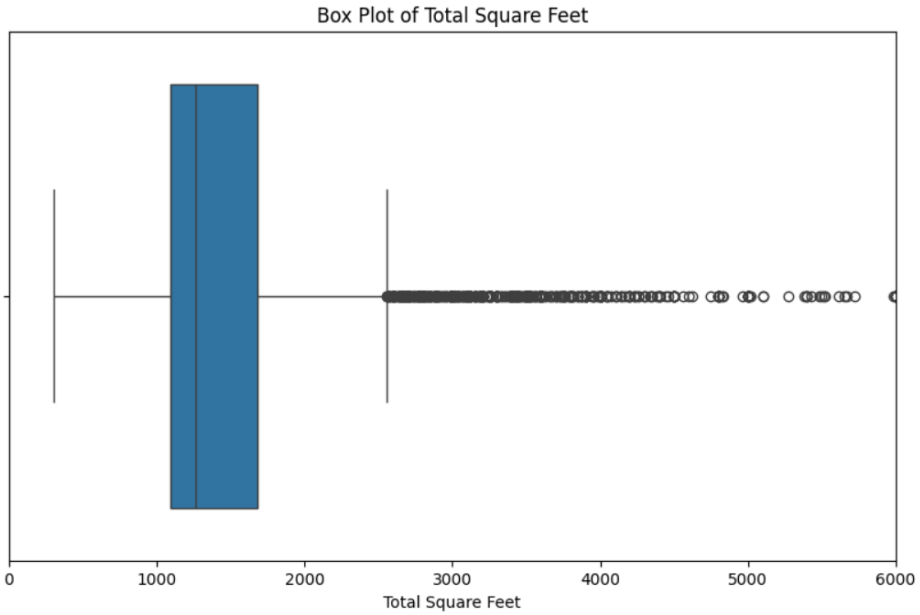
	A	B	C	D	E	F
1		location	total_sqft	bath	price	bhk
2	0	1st Block J	2850	4	428	4
3	1	1st Block J	1630	3	194	3
4	2	1st Block J	1875	2	235	3
5	3	1st Block J	1200	2	130	3
6	4	1st Block J	1235	2	148	2
7	5	1st Block J	2750	4	413	4
8	6	1st Block J	2450	4	368	4
9	8	1st Phase J	1875	3	167	3
10	9	1st Phase J	1500	5	85	5
11	10	1st Phase J	2065	4	210	3
12	12	1st Phase J	2059	3	225	3
13	13	1st Phase J	1394	2	100	2
14	14	1st Phase J	1077	2	93	2
15	15	1st Phase J	1566	2	180	2
16	16	1st Phase J	840	2	50	1
17	17	1st Phase J	1590	3	131	3
18	18	1st Phase J	2180	3	210	3
19	19	1st Phase J	1180	2	88.5	2
20	20	1st Phase J	1200	2	86	2
21	21	1st Phase J	1394	2	85	2
22	22	1st Phase J	2077	3	175	3
23	24	1st Phase J	1205	2	85	2
24	26	1st Phase J	900	2	75	2
25	27	2nd Phase	1450	2	50.75	3
26	28	2nd Phase	1150	2	40.25	2
27	29	2nd Phase	1350	2	47.25	3
28	30	2nd Phase	1350	2	47.25	3

4.4 Graphs

```
plt.title('Box Plot of Total Square Feet')
plt.xlabel('Total Square Feet')
plt.show()
```



```
plt.xlim(0, 6000)  # Set x-axis limits
plt.show()
```



jupyter Project1 Last Checkpoint: 3 hours ago

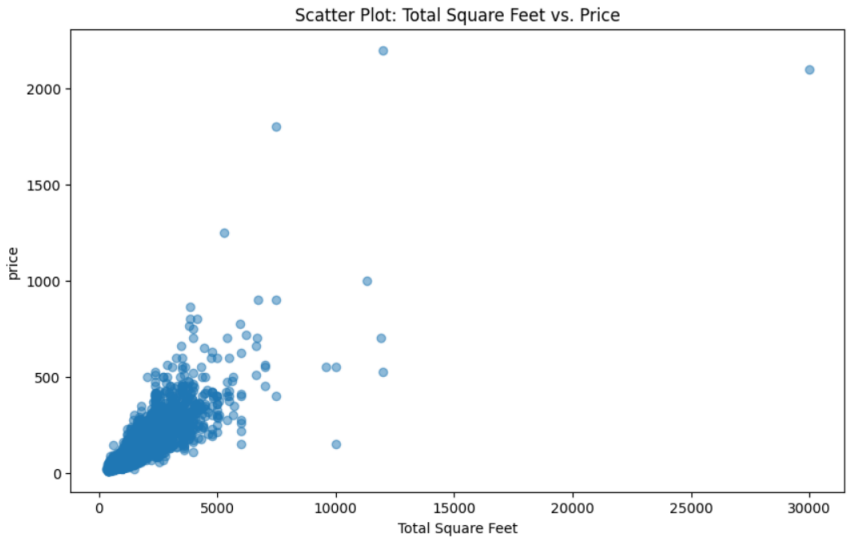
File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

```
plt.xlabel('Total Square Feet')
plt.ylabel('price')
plt.show()
```



CONCLUSION

In conclusion, the House Price Prediction project has successfully developed a robust Ridge regression model for estimating house prices based on key features. Through meticulous data exploration, cleaning, and feature engineering, the model demonstrates reliability, particularly in handling outliers. The comparison of Lasso and Ridge regression highlights Ridge as a standout performer. The practical application of the serialized Ridge model provides a valuable tool for stakeholders in the real estate market to make informed predictions. The project underscores the iterative nature of model development, allowing for continuous improvement with future updates and the incorporation of new data. Overall, it showcases the significance of thoughtful data preprocessing and the practical application of machine learning techniques in real-world scenarios, offering a reliable solution for house price estimation.

REFERENCES

1. https://www.mygreatlearning.com/blog/what-is-ridge-regression/
2. https://www.youtube.com/watch?v=DVxkI1VmpCk&ab_channel=CampusX