House Price Prediction in California

By
Shreya Ghotankar
Manjiri Kadam
Parvathi Pai
Anastasia Zimina
December 2021

ABSTRACT

House Price Prediction in California

By Shreya Ghotankar, Manjiri Kadam, Parvathi Pai, Anastasia Zimina

A home means a future, stability. Every person's dream is to buy a dream house once in their lifetime and maybe sell it eventually for profit. Living a California dream is becoming more difficult each day. There are a lot of job opportunities here with great weather, lifestyle, and housing demand is on the rise.

According to the state housing department, California needs to build 180,000 new houses every year in order to keep up with demand, which is very high, there is a big valley between supply and demand. It is very important to know where you invest your money and get maximum benefit. To make a successful purchase and quote a price that will convert, one has to know about the market, buying/selling conditions, and other multiple factors which are difficult to keep up with.

In our project, we proposed to predict the price of houses in California by analyzing and visualizing the historical data and using machine learning algorithms we trained a model that will predict the price. We have followed the CRISP-DM phases and trained regression model to predict the house prices based on the attributes like - area, number of bedrooms, number of bathrooms, zip code and listing type. The application is implemented with StreamLit and is hosted on the AWS cloud.

Table of Contents

Chapter 1. Introduction

Many people dream about living in the Golden State, as the name suggests it is beautiful but expensive. In California house price demand is always high, because various factors affect the valuation of the houses- location, food, pacific ocean, weather, education, job opportunities, the list is long. But the most important factor which is increasing the housing price is the lack of availability numbers. California has only been able to build half of the housing demand in the past 10 years, and demand for housing is increasing each day.

The house prices today cost more than seven times what the average household makes. This is an insane amount of money, if not invested properly it could get wasted.

To help people find their dream house we have worked on a solution to predict the house prices in California. This will benefit people to make one of the important decisions in their lives, where they can check the prices of the houses based on various aspects like- area, the number of rooms and baths and locality.

House prices are always changing and it is humanly impossible to determine what the price might be tomorrow. So with help of historical data and using machine learning algorithms we assist people to make offers based on the predicted prices that will increase the probability of buying.

To achieve this, we followed the CRISP-DM and performed all the phases. We have made use of the Zillow Dataset (subset for the California region) for training the machine learning model. We have cleaned the dataset, preprocessed it, and used it to train a regression model. We are making use of a random forest regressor. Random Forest Regression is quite a

robust algorithm and predicts good results on a huge dataset. After achieving good results from our model, we saved the model, and then we created a StreamLit application and hosted our application on an EC2 instance.

Chapter 2. Related Work

The research conducted by Zhang Q. et al. [1] is based on the theory of hedonic prices, introduced by Sherwin Rosen. According to his theory, the property price is a utility function of many variables. Those variables could include such characteristics as neighborhoods, the environment, structural features, etc. Multiple regression models are generally used for the purposes of price prediction. A number of assumptions should be fulfilled: independence, homoskedasticity, and normal distribution [8]. According to Zhang the most influential house factor is residential, such as residence, usability, and number of rooms. Others include building properties (hardware and basic facilities), number of floors, environmental factors (regional environment and nearby pollution), transportation, distance to social and cultural centers, etc.

The paper by Kamal et al., 2021 [2] explored the data mining and machine learning concepts for predicting the house prices in the Boston region. The dataset they used had a large number of features and features related to average, median incomes and taxes in that region which impacts the prices of the houses. In the design approach they mainly focused on 3 regression models linear, decision tree and random forest. They followed the basic steps of machine learning like collection, cleaning, exploration, training and evaluation. Their outcome was that the linear regression performed well.

Chapter 3. Business and Data Understanding

1. Business Understanding

House prices are always changing and there is a lot of movement between the listing price and the selling price. Multiple factors contribute to the price of any house like the market, buyers, demand-supply, etc. The demand for California houses never ceases. To assist buyers in quoting a price to make a successful purchase we are using historical data and implementing machine learning algorithms to help people to make offers based on the predicted prices.

- Assess the situation and Determine Data Science Goals: There are many factors with which the house prices in California are rising. A few of them are area, zip code, number of bathrooms, bedrooms, etc. In this data mining project we are using the dataset that has listings from zillow, a real-estate website, and based on the feature selection, we are using the most informative features such as area, bedroom, bathroom, listing types for house price predictions.
- Project plan: Initially we started the project by making use of the zillow dataset. Then we cleaned the dataset, by averaging out the none values with respect to the columns. We selected the important features by using the Gini index. Then the next step in the project plan was to identify the algorithm. Since we are predicting the house price which is a continuous variable with respect to many factors, the regression model suits best for our project.

2. Data understanding

The dataset has listings from zillow website where different types of houses like single-family homes, townhomes, and condos are listed by buyers and real estate agents. This dataset has listings from different cities. As our goal is to predict house prices in California we filtered the address for the 'CA' region which gave us a subset of (1824, 23) - 1824 rows and 23 columns/features and the data types for each can be seen in figure 2.2.

```
index                int64
rank                 int64
property_id          int64
address             object
latitude           float64
longitude          float64
price              float64
currency            object
bathrooms          float64
bedrooms           float64
area                object
land_area           object
zestimate          float64
rent_zestimate     float64
days_on_zillow     float64
sold_date          float64
is_zillow_owned       bool
image               object
listing_type        object
broker_name         object
input               object
property_url        object
listing_url         object
dtype: object
```

Figure 2.2

The dataset has the following listed attributes: rank, property_id, address, latitude, longitude, price, currency, bathrooms, bedrooms, area, land_area, zestimate, rent_zestimate, days_on_zillow, sold_date, is_zillow_owned, image, listing_type, broker_name, input, property_url, listing_url.

In the below figure 3.1, you can see the details on the numeric features like the max value, mean, count etc.

```
[ ] df_cali.describe()
```

	index	rank	property_id	latitude	longitude	price	bathrooms	bedrooms	zestimate	rent_zestimate	days_on_zillow	sold_date
count	1824.00000	1824.00000	1824.00000	1816.00000	1816.00000	1824.00000	1639.00000	1655.00000	1480.00000	1673.00000	1821.00000	0.00000
mean	2954.95066	286.05373	388115376.33443	34.31472	-118.66203	1416081.65844	2.56132	3.17221	1384120.90405	4457.21040	35.56507	nan
std	1726.36014	222.17396	760894850.74492	2.05553	2.19840	2397302.15376	1.25249	2.68971	1732894.17161	4779.62525	142.15179	nan
min	8.00000	1.00000	16586513.00000	32.54861	-122.03035	39950.00000	1.00000	0.00000	82760.00000	1084.00000	-17.00000	nan
25%	1469.50000	89.00000	16947363.50000	32.79825	-121.83102	599000.00000	2.00000	2.00000	646914.25000	2749.00000	3.00000	nan
50%	2947.00000	229.00000	19717230.00000	33.07933	-117.23564	898000.00000	2.00000	3.00000	927349.00000	3416.00000	13.00000	nan
75%	4476.25000	463.00000	99595719.75000	37.26307	-117.11007	1450000.00000	3.00000	4.00000	1466494.25000	4227.00000	30.00000	nan
max	5889.00000	800.00000	2146138421.00000	37.42692	-116.99512	49000000.00000	16.00000	99.00000	22336400.00000	84823.00000	5171.00000	nan

Data Understanding - As we can see from the data here it has 1824 rows and 22 columns

Figure 3.1

With help of the pair-plot in figure 3.2, we had more clarity on the data and understood the relationship of price with other features.

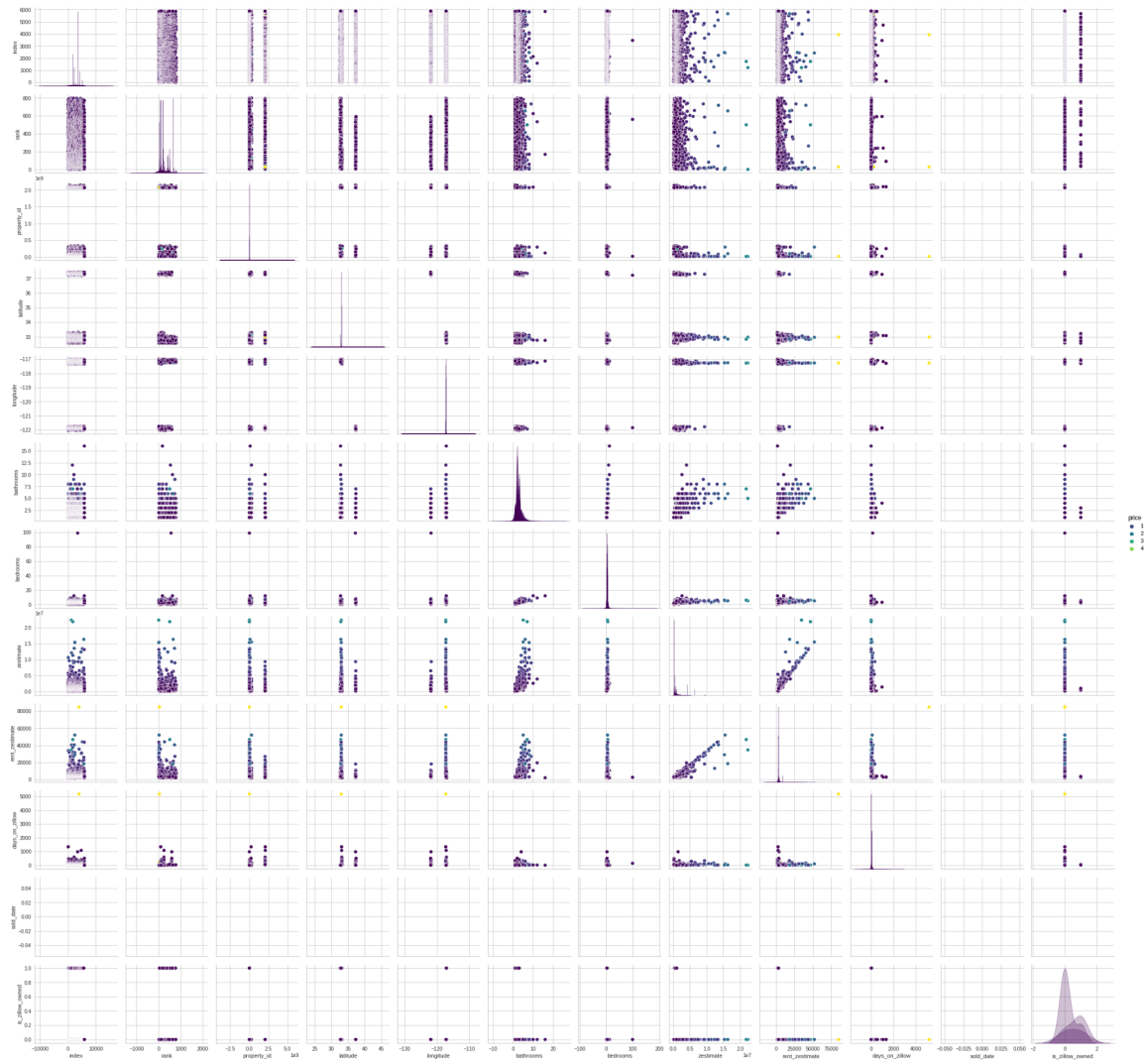

Figure 3.2

This diagram shows how the price change is associated with the other features like bedroom, bathroom, zestimate, etc.

We visualized the skewness of price using a histogram plot as shown in figure 3.3.

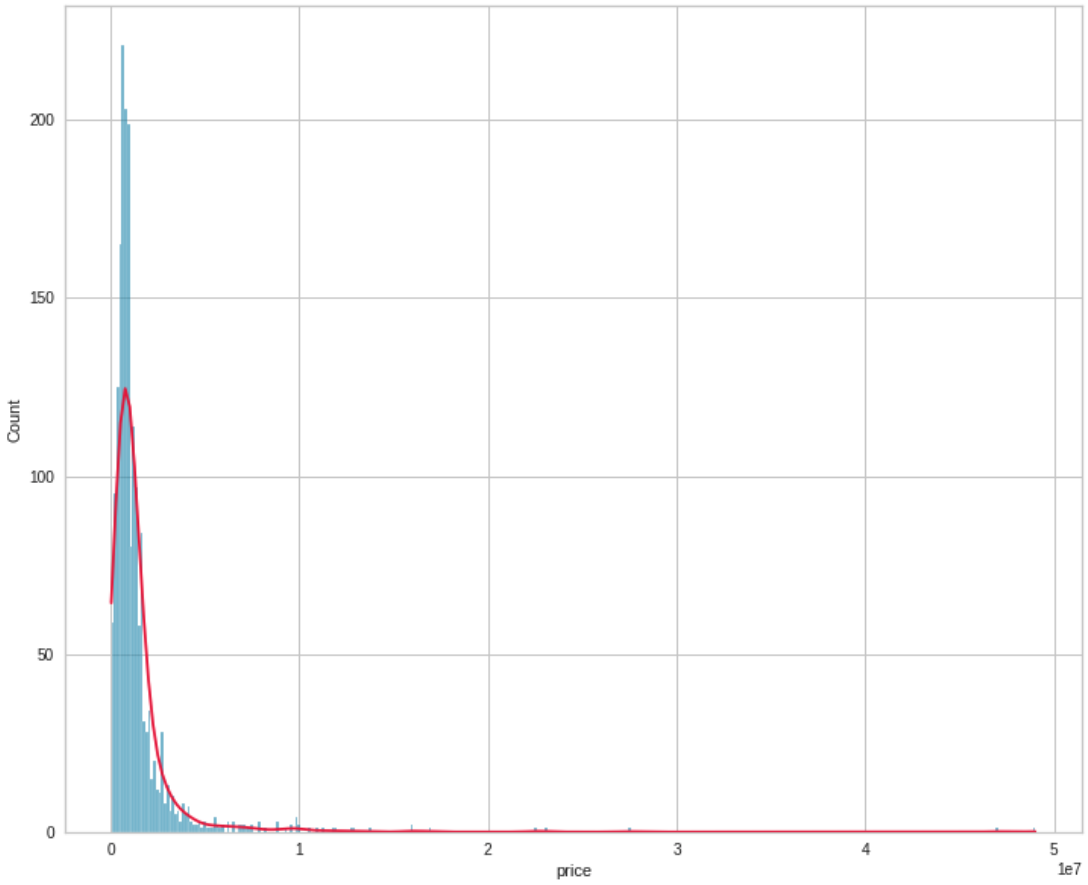

Figure 3.3

We also looked at the various types of listings in the dataset using the count plot as shown in figure 3.4.

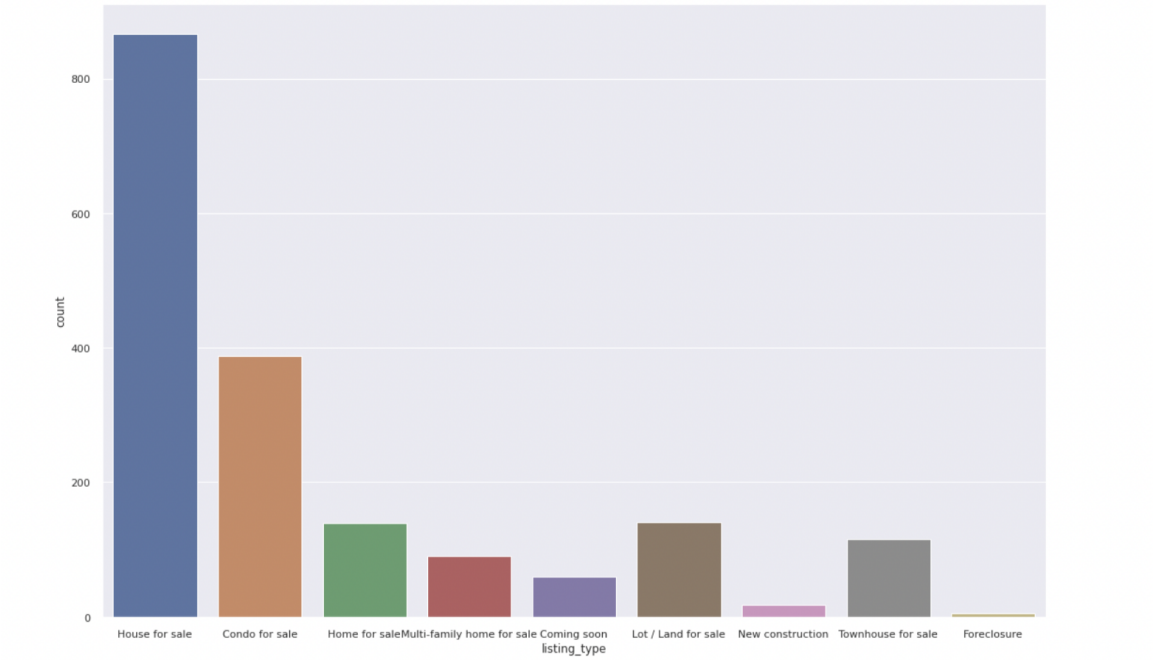

Figure 3.4

To understand the listing type feature, we visualized the price against it as shown in figure 3.5. The first plot shows the variation of mean prices with respect to the listing types and the second plot shows the mean of rent vs listing type.

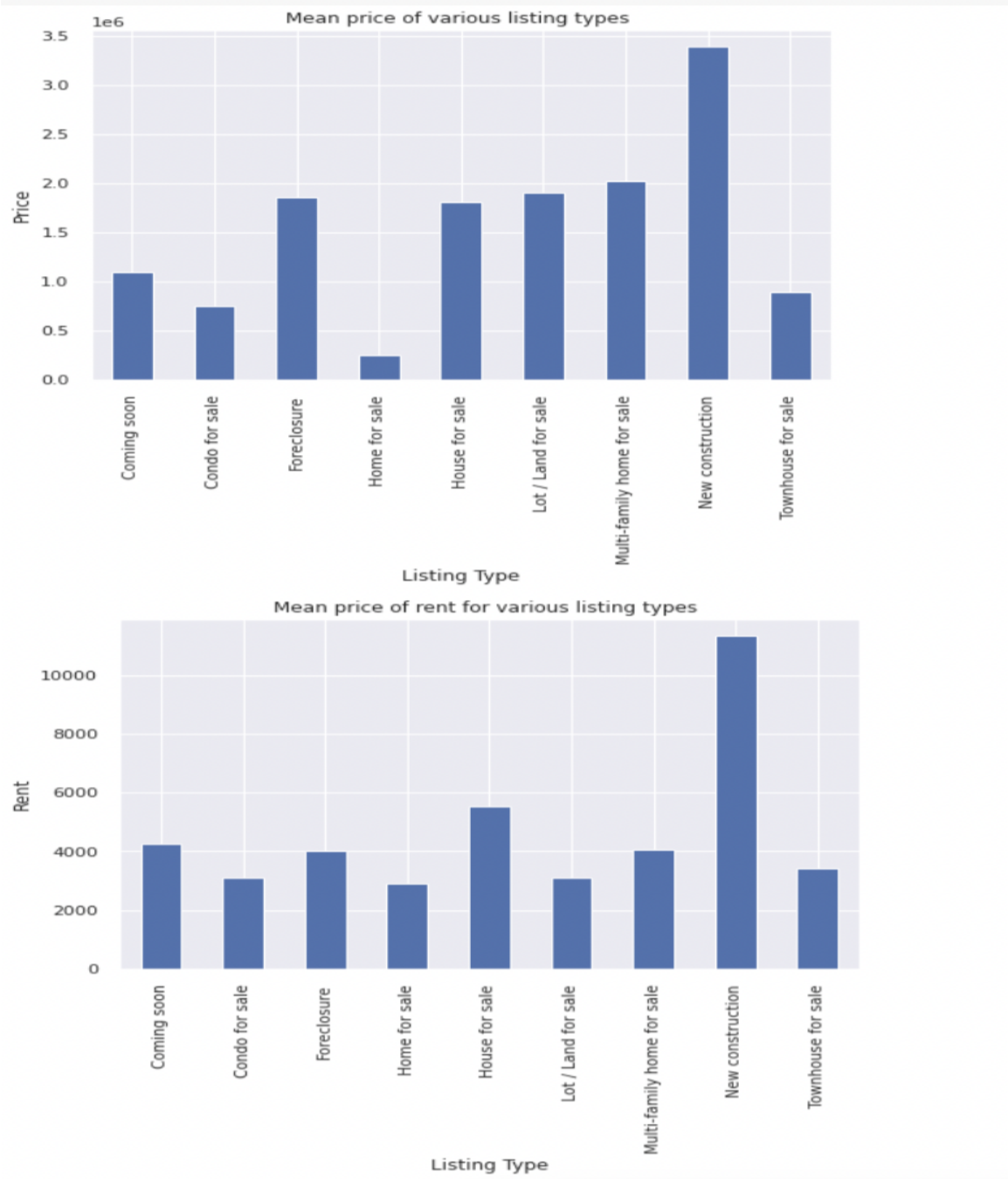

Figure 3.5

As shown in figure 3.6, we also used the zillow estimated values to understand the variation in prices of various listing types.

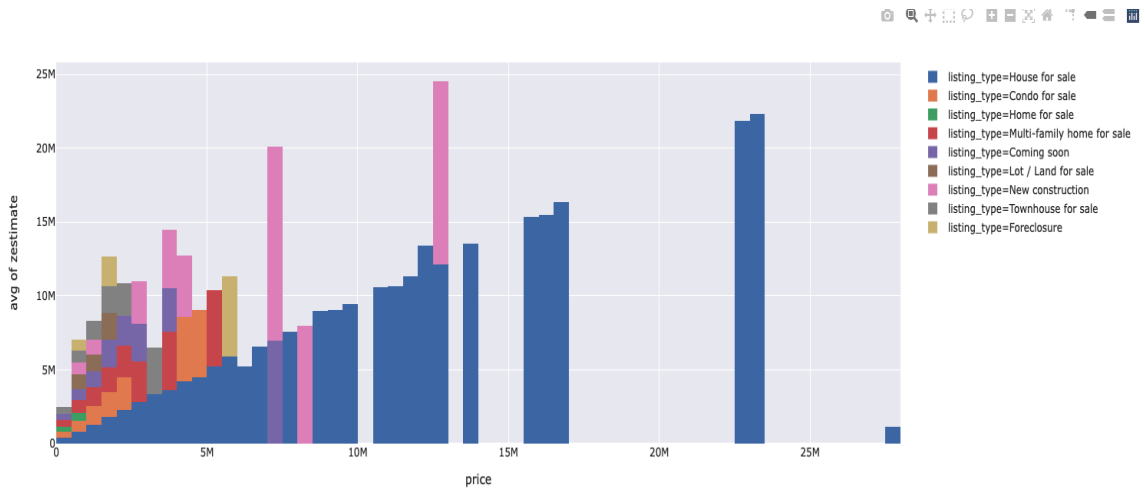

Figure 3.6

Finally we plotted a correlation heatmap to get more information on the relations between all the features in the dataset.

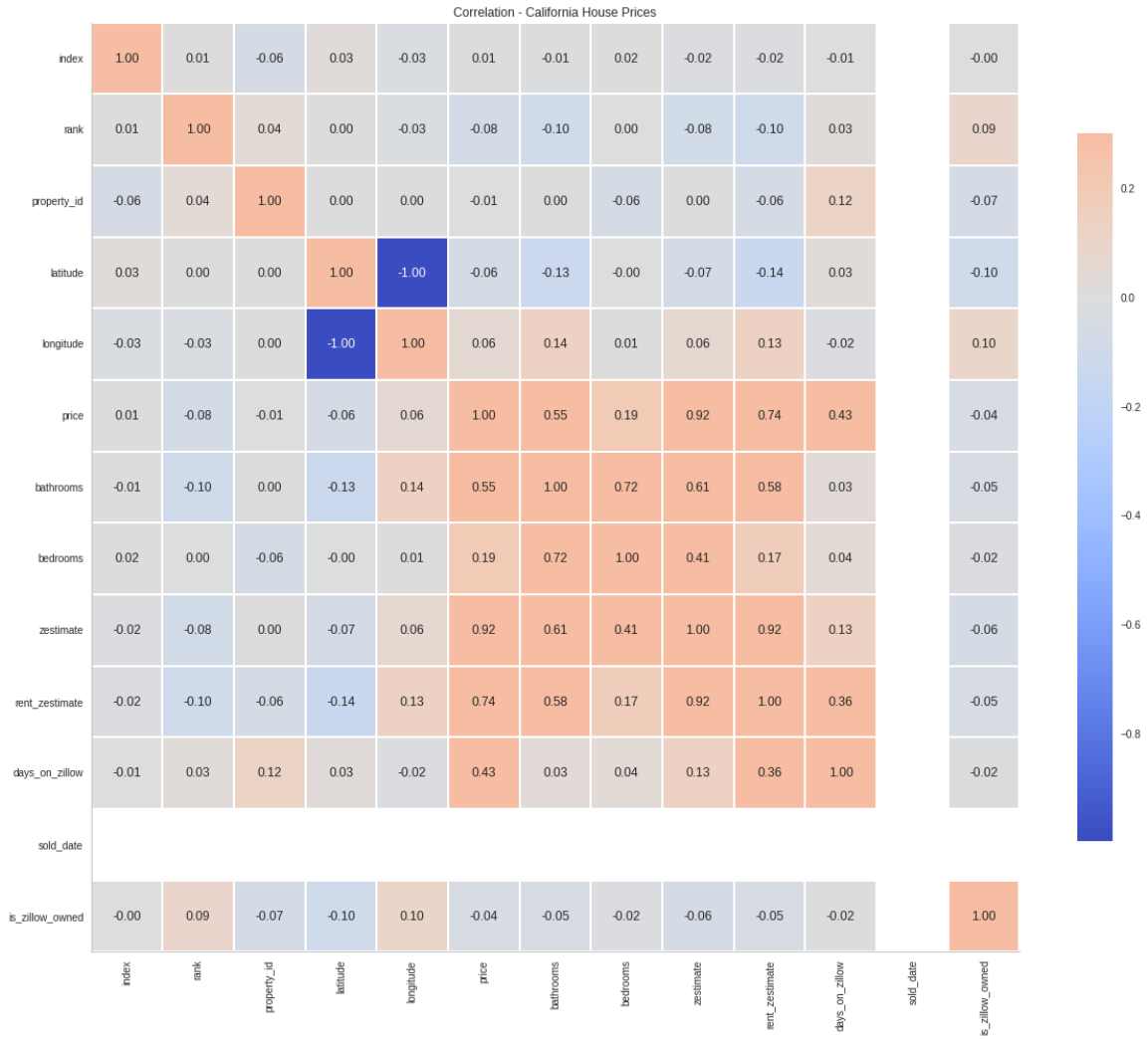

Figure 3.7

High-Level Diagram

For implementing this project, we followed the steps as shown in figure 3.8 which provides a high-level implementation of our project. These steps are essential phases of CRISP-DM which helped us work our way through the development of this project.

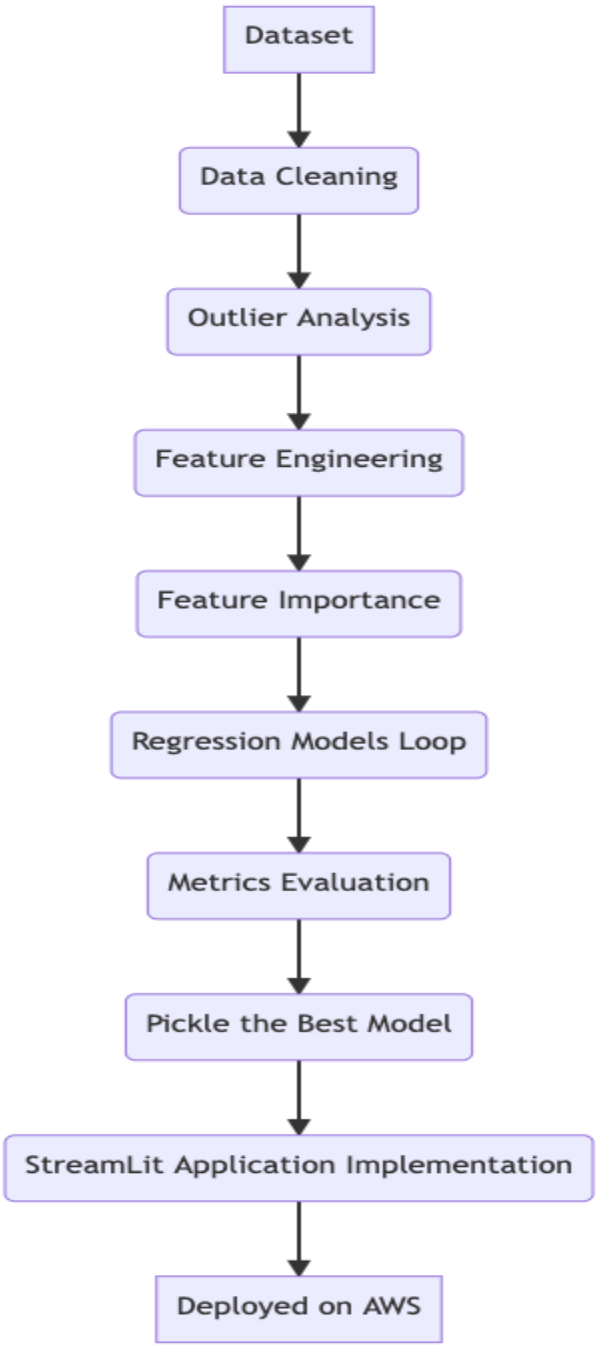
Dataset
Data Cleaning
Outlier Analysis
Feature Engineering
Feature Importance
Regression Models Loop
Metrics Evaluation
Pickle the Best Model
StreamLit Application Implementation
Deployed on AWS

Figure 3.8

Chapter 4. Data Preparation

As we all know, clean data is the crux of data mining and machine learning development. We need to understand the data, information about the attributes and patterns in the data. In this chapter as part of CRISP-DM methodology, we are covering the data preparation phase which involves selecting, cleaning, handling missing values, and analyzing data further to help select the dominant features.

Data Cleaning

1. Formatting the data: The area given for each listing in the dataset had ‘sqft’ suffix which made the feature an object type. So to be able to utilize the area values, we created a separate column and removed the ‘sqft’ word from it, and converted the data type to ‘float32’.
2. We converted all except ‘address’ and ‘property_url’ the object data type features into categorical with cat.codes so that they can be used for training model.
3. Handle missing values: We identified the missing values or NaN values as can be seen in figure 4.1 and imputed them.

```
df_feat.isna().sum().sort_values(ascending=False)
```

```
sold_date           1824
land_area           1683
zestimate            344
bathrooms            185
bedrooms             169
rent_zestimate       151
area_sqft            148
latitude               8
longitude              8
days_on_zillow         3
is_zillow_owned        0
currency               0
rank                   0
property_id            0
address                0
price                  0
property_url           0
image                  0
input                  0
area                   0
listing_url            0
broker_name            0
listing_type           0
index                  0
dtype: int64
```

Figure 4.1

Outlier Removal

As part of data preparation we performed the outliers analysis on the dataset. Outlier detection and removal is a very important step for any machine learning project and if identified early in the process helps with less loss and better model performance, it can affect the accuracy and prediction.

The following plots show our analysis and removal of the outliers from the dataset.

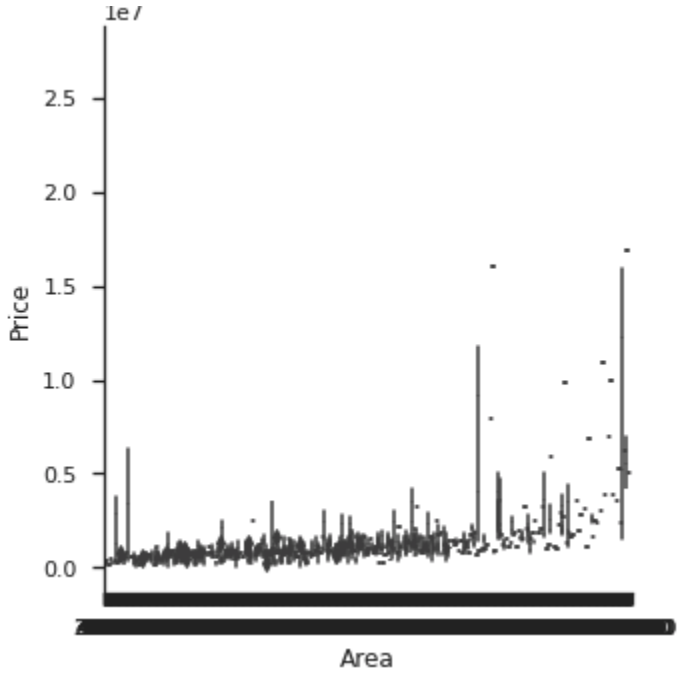

Figure 4.3

In figure 4.3, the area vs price catplot is shown which provides information about any listings that are unrealistically high which can influence the prediction.

In figure 4.4, we can see some outliers with the help of a catplot created between ‘bedrooms’ and ‘price’ features. These outliers were conditionally removed from the dataset.

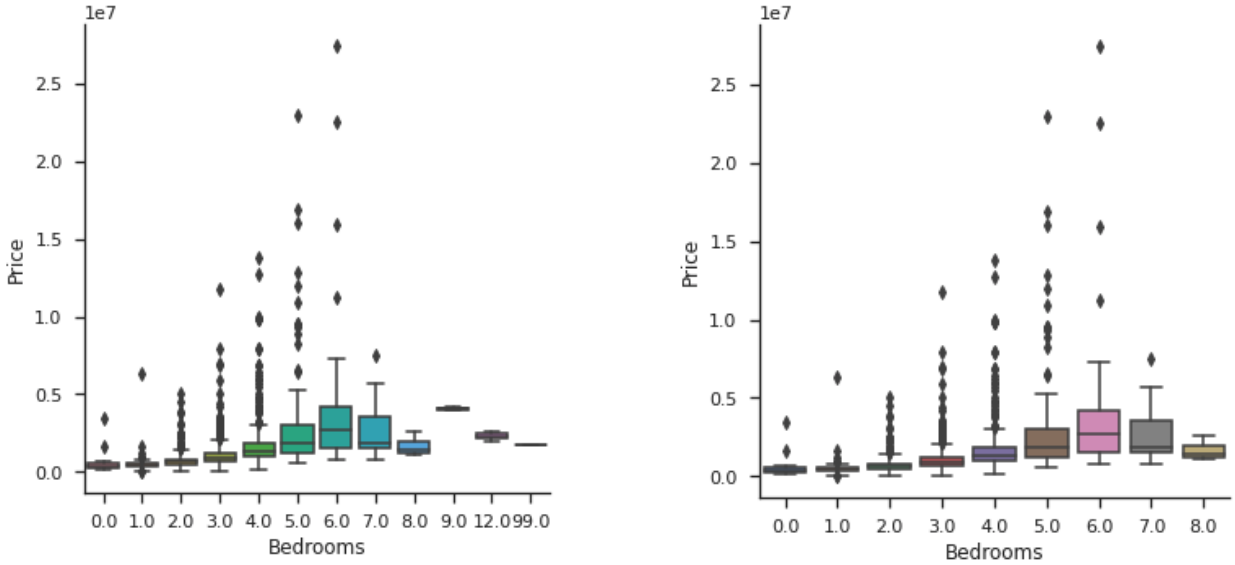

Figure 4.4

Similarly, we performed the outlier detection and removal on the 'bathrooms' features in the dataset as shown in figure 4.5.

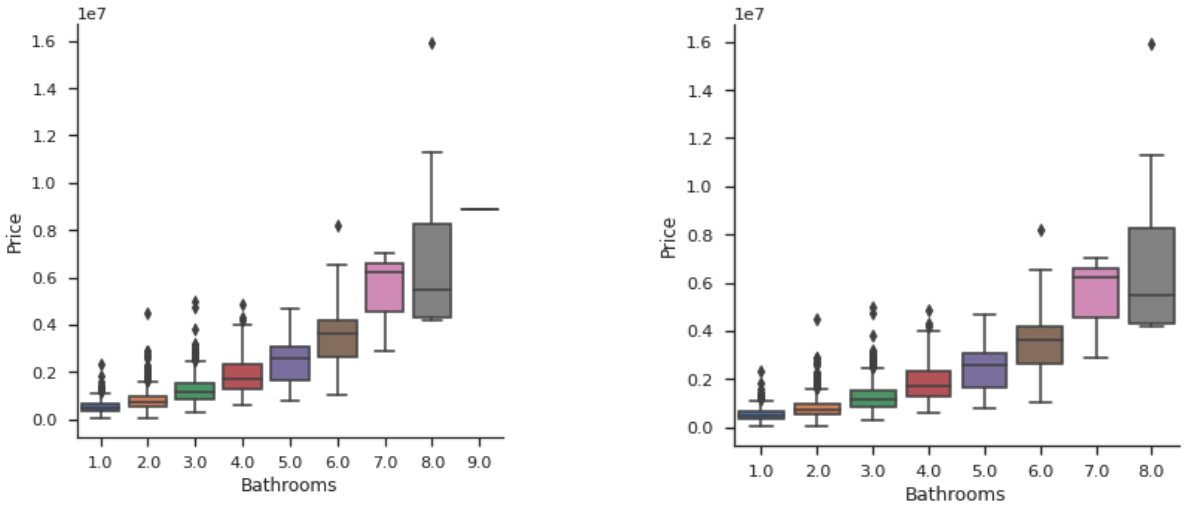

Figure 4.5

After data preparation we visualized the data with interactive Plot using Plotly as shown in figure 4.6.

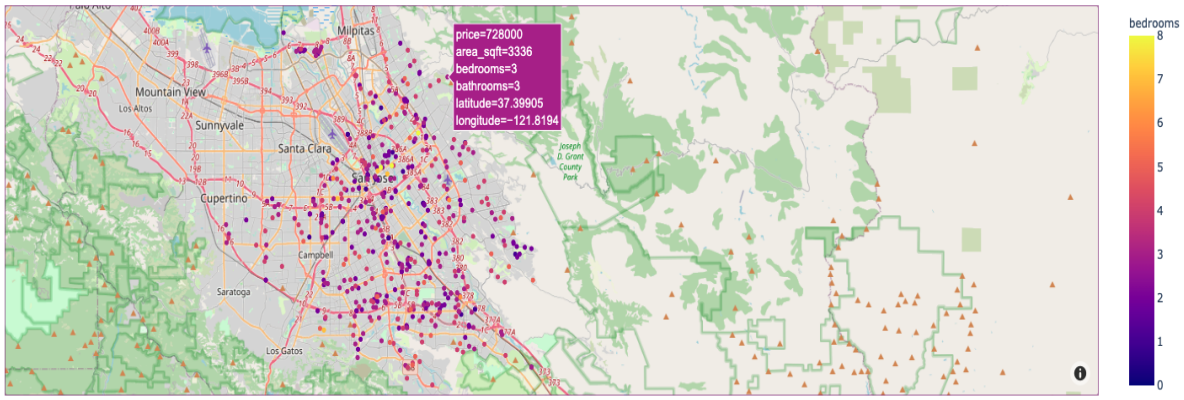

Figure 4.6

Chapter 5. Feature Engineering and Importance

After analyzing and visualizing the dataset, we came to the conclusion that variation in number of rooms, bathrooms, area, listing type and the location are the most influential attributes towards house price changes.

We used these attributes to train the regression model with target variable as the price of the house. We picked the columns which are directly affecting the change in house price: 'area','image', 'listing_type', 'broker_name', 'input', 'listing_url' and dropped the columns which were not necessary.

Feature engineering

The feature engineering is the constructing and formatting data step that transforms raw data into features that can be used in machine learning algorithms, such as predictive models. We have implemented feature engineering in our colab. By using the 'address' values we added new feature- zip code by using a python library **uszipcode.** It's purpose is to list out the houses based on the Zipcode and also impacts the price prediction.

Feature Importance

Feature importance helps us to estimate how much each feature contributes to the model performance/prediction. We have calculated the feature importance by using SHAP (Shapely Additive Explanation).

The below figure 5.1, shows the SHAP values for the features.


area_sqft
longitude
bathrooms
property_id
listing_type
latitude
zip_code
index
rank
bedrooms
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
1e6
SHAP value (impact on model output)
High
Low
Feature value

Figure 5.1

Chapter 6. Modeling

Modeling is the important step in CRISP-DM methodology. In this phase we identify the technique and based on that, the model is trained to predict the output in our case it is the house price.

Regression

A regression model predicts a quantity that is a continuous value. Regression predictive modeling is the task of approximating a mapping function from input attributes to a continuous output value [3].

Types of Regression models

1. Linear Regression - It finds the best fit linear line between independent and dependent variables.
2. Logistic Regression - It is a probabilistic model used to find relationships between dependent and independent variables.
3. KNN Regression - It approximates the relationship of independent variables with the continuous variable by averaging the observations in the same neighborhood.
4. Random Forest Regression - It is an ensemble learning technique and creates multiple decision trees during training and outputs the mean prediction of individual trees.
5. Adaboost Regression - An AdaBoost [6] regressor is a meta-estimator that begins by fitting a regressor on the original dataset and then fits additional copies of the regressor with observational weights on the same dataset.

6. CatBoost Regression - It is a gradient boosting technique which divides the given dataset into random permutations and applies ordered boosting on it.
7. LightGBM Regression - It is a gradient boosting technique based on decision trees to increase model efficiency and minimize memory utilization.
8. XGBoost Regression - Extreme gradient boosting technique that uses more accurate approximations to find the best tree model.

Training the Model

Step1: We divided the dataset in two parts, 1) training 2) testing.

Step2: Imported the packages to apply regressors. We used - Linear Regression, Logistic Regression, KNN Regression, Random Forest Regression, Adaboost, Catboost, LightGBM, XGBoost regression packages.

Step3: We ran multiple regressors in a loop to determine which regression gives best results. Following snapshot shows the comparison of all the regressor models.

	Model Name	Mean	Accuracy	Time Taken
0	Linear Regression	0.63743	0.71730	1.63123
1	Logistic Regression	0.40689	0.02564	155.97161
2	KNN Regression	0.49565	0.54353	0.27786
3	Random Forest Regression	0.76413	0.82600	48.56526
4	AdaBoost Regression	0.56966	0.63713	2.14880
5	Cat Boost	0.79529	0.85028	35.61911
6	Light Gradient Boosting	0.77964	0.82592	1.97072
7	EXtreme Gradient Boosting	0.77878	0.80796	1.94366

Figure 6.1

Evaluation

After training various regression models we compared them on different metrics like accuracy, mean score and overall time taken to fit the training data. With the help of the table in figure 6.1 we selected Random Forest regression for our prediction problem, as we got accuracy of **82.6 %**.

Using pickle we saved 2 of the best models i.e. XGBoost and Random Forest.

Chapter 7. Application Implementation

In this chapter, we focused on the Deployment phase of CRISP-DM.

StreamLit

- Streamlit is the fastest way of developing a Machine learning application with google colab and git integration. We can deploy these applications very easily with any deployment strategy.
- It is an open source Python framework for building ML apps.
- We can easily design the application with a beautiful User Interface by using python.
- To use the streamlit we need to install the streamlit package by using below command:

  pip install streamlit
- To run the streamlit application we use the following command:

  streamlit run <someprogram.py>

For our project we have created a virtual environment in Python. Using this environment we can install all the code dependencies by running the following:

pip install install.py. This would update all the streamlit dependencies on the python virtual environment.

UI Screenshots

The streamlit application is hosted in AWS using EC2. Figure 7.1 shows the page which helps to make the housing prediction.

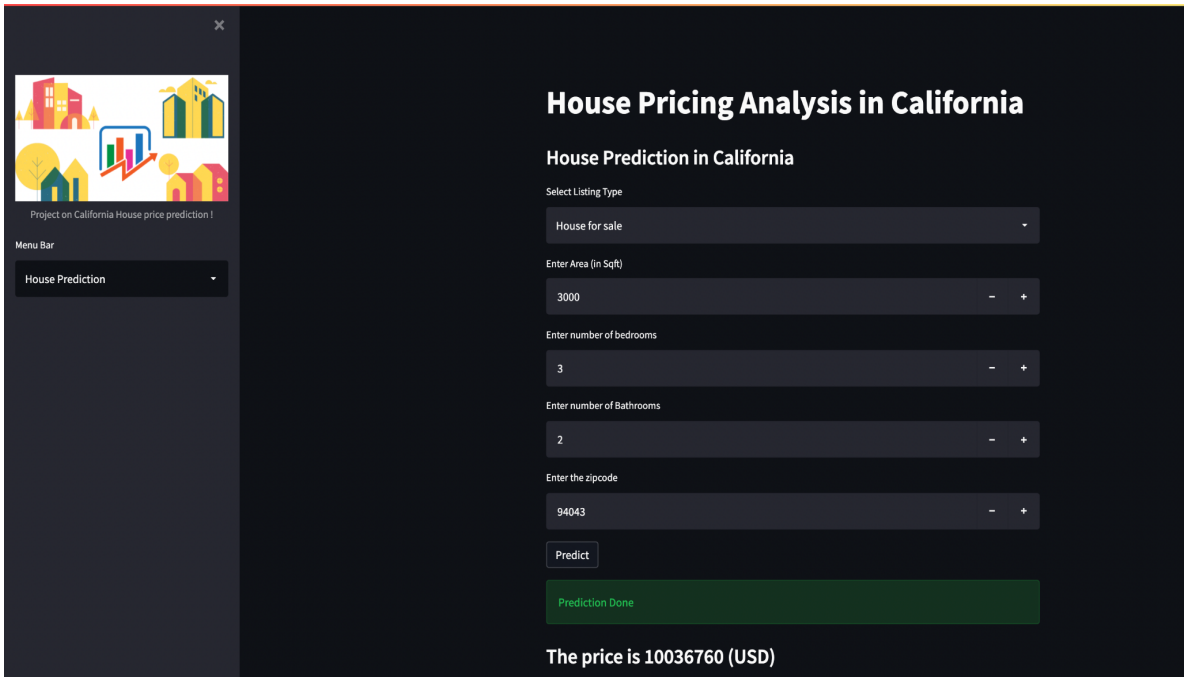

Figure 7.1

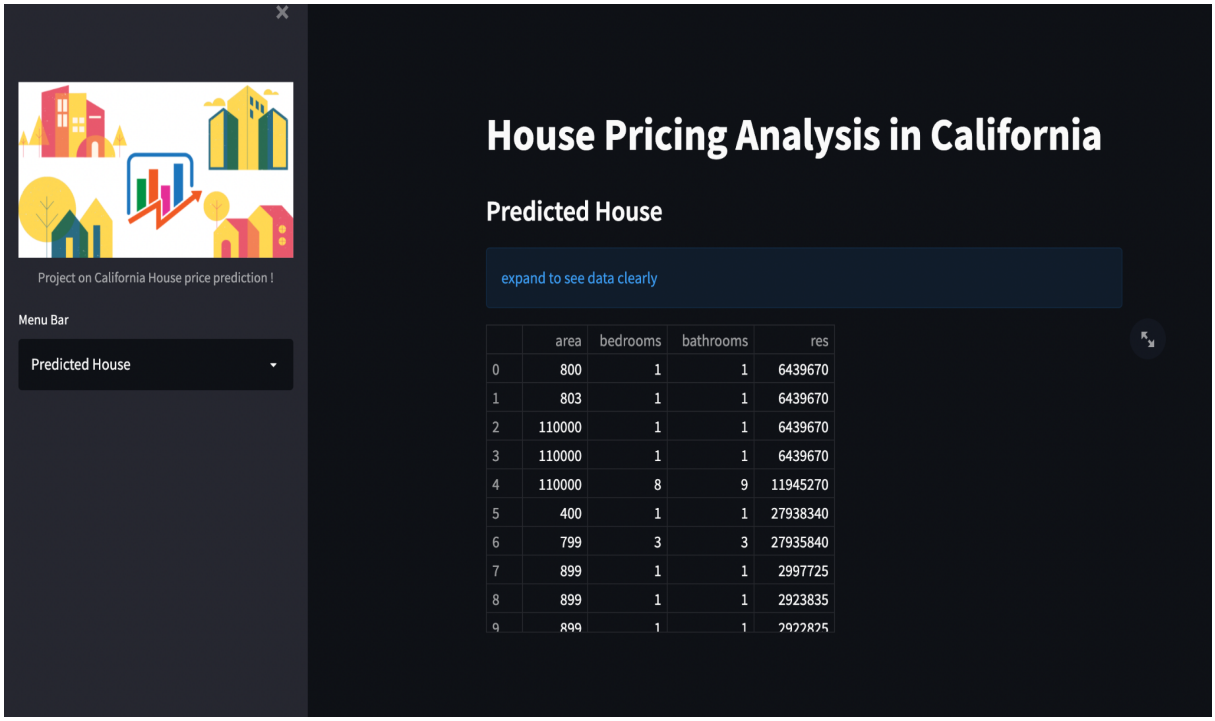

Figure 7.2

Figure 7.2 shows the results of the queries made by the past customers. These results are stored in sqlite hosted in EC2.

Figure 7.3

Figure 7.3 shows the welcome page for the application where customers would get the details of our application.

Chapter 8. Results

Model Results with Test Data

We were able to get accuracy of 80.79% on test data.

```
loaded_model = pickle.load(open(filename_1, 'rb'))
result = loaded_model.score(X_test, y_test)
print(result)
```

```
0.8079630578226664
```

Figure 8.1

In figure 8.1, you can see that after we pickled the best models we loaded the model and evaluated it on the test data.

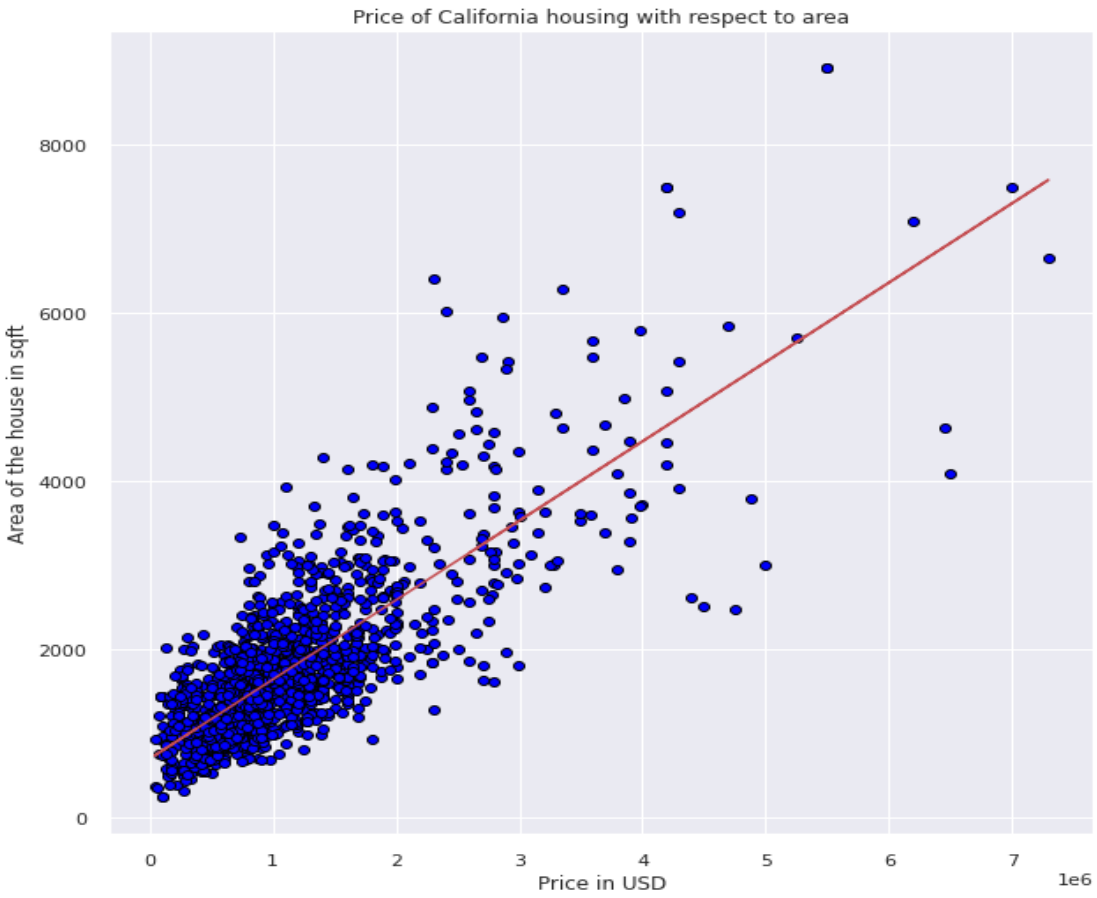

Figure 8.2

From the above plot in figure 8.2 we can see that the price of housing increases in California with the increase in size (area) of the house.

End to End Testing

The following screenshot shows the end-to-end testing of the application. We have to enter the values as our requirement - Listing Type, Area, number of bedrooms, number of bathrooms, zip code.

Once, entered the required information, you can click on the predict button to check the estimated house price in that region.

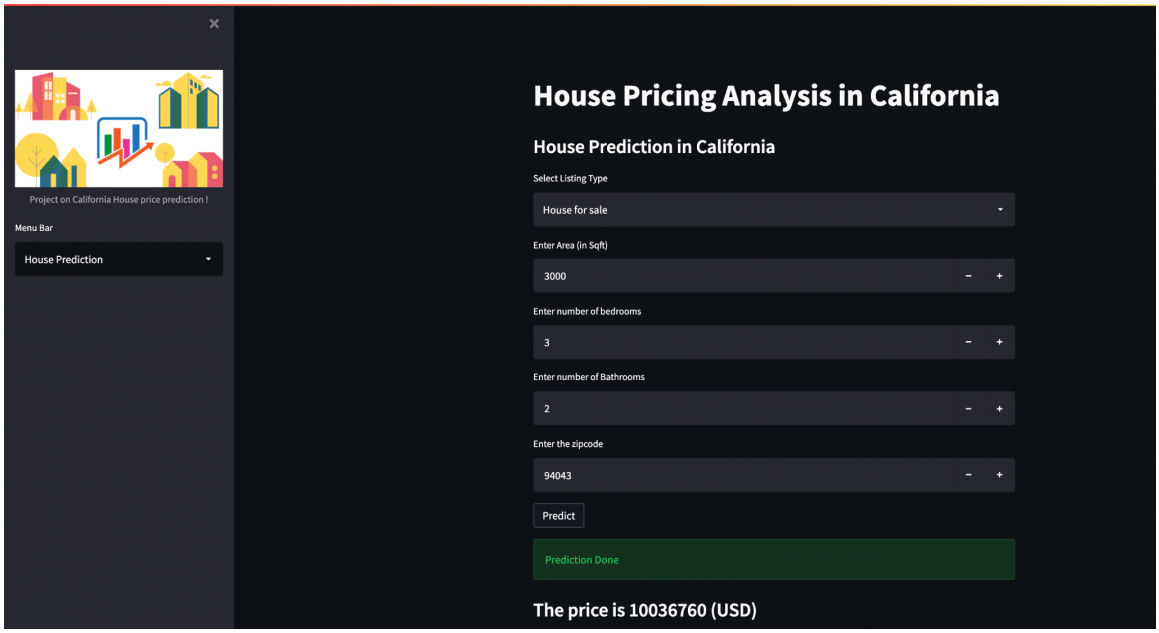

Figure 8.3

On the predicted houses tab, we can see the history of predicted house prices along with the search criteria.

Chapter 9. Deployment

Deployment Stack:

- We have used EC2 and Streamlit for the deployment of our machine learning project.
- We have used EC2 free tier t2.micro instance with Ubuntu 18 AMI.

Deployment Architecture

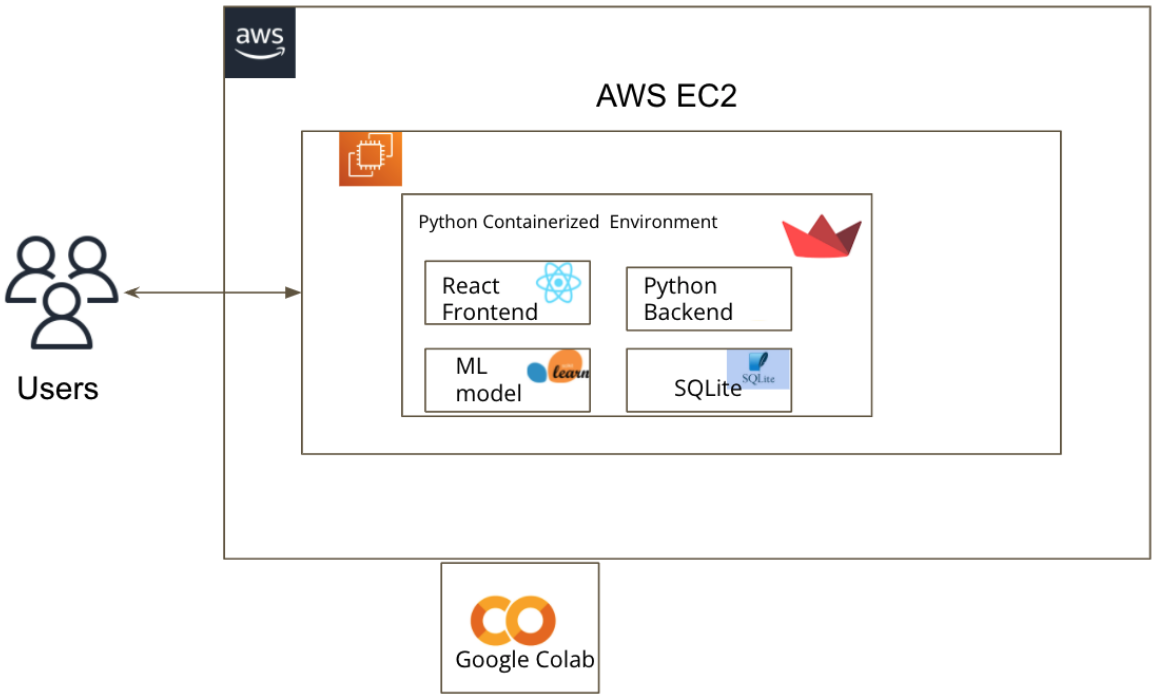

Figure 9.1 Deployment Architecture Diagram

We have developed a Machine learning application using Google Colab. Implemented various data cleaning techniques, visualization and Model training and testing. Then we used Streamlit to host our application. Streamlit is an open source platform for machine learning

application development, it uses Python as core language. We decided to host our application using AWS cloud services.

For deployment of our application, we decided to go with Amazon EC2 free tier instance. Used Ubuntu operating system for hosting the application.

Setup

- We have deployed our application on AWS cloud, on an EC2 instance.
- We selected the t2.micro type of instance, with AMI : **ami-0279c3b3186e54acd (Ubuntu Machine).**

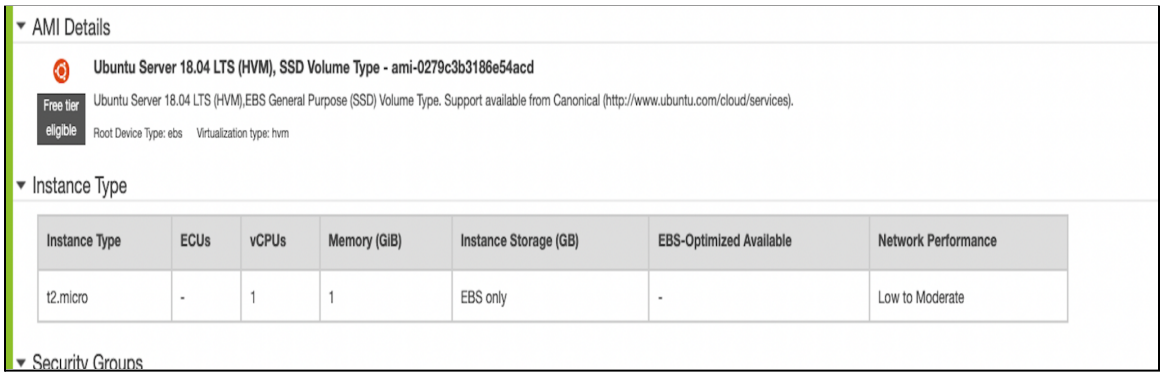

Figure 9.2 EC2 instance AMI

- The streamlit application uses 8501 port number , in the security group we have changed the TCP rules as shown in the diagram below.

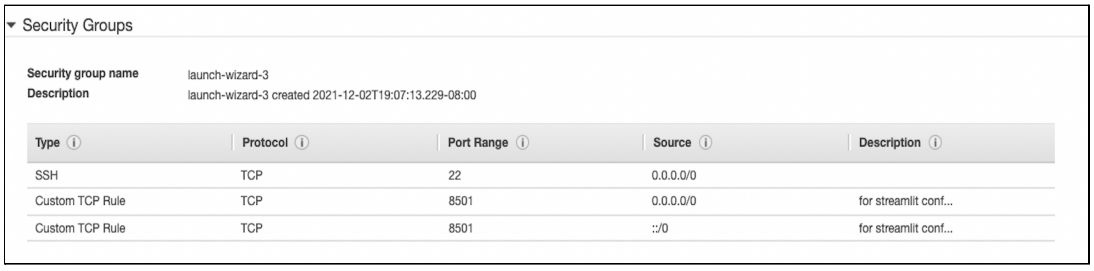

Figure 9.3 Security Group Set-up

- Below figure shows the configuration of the EC2 instance.

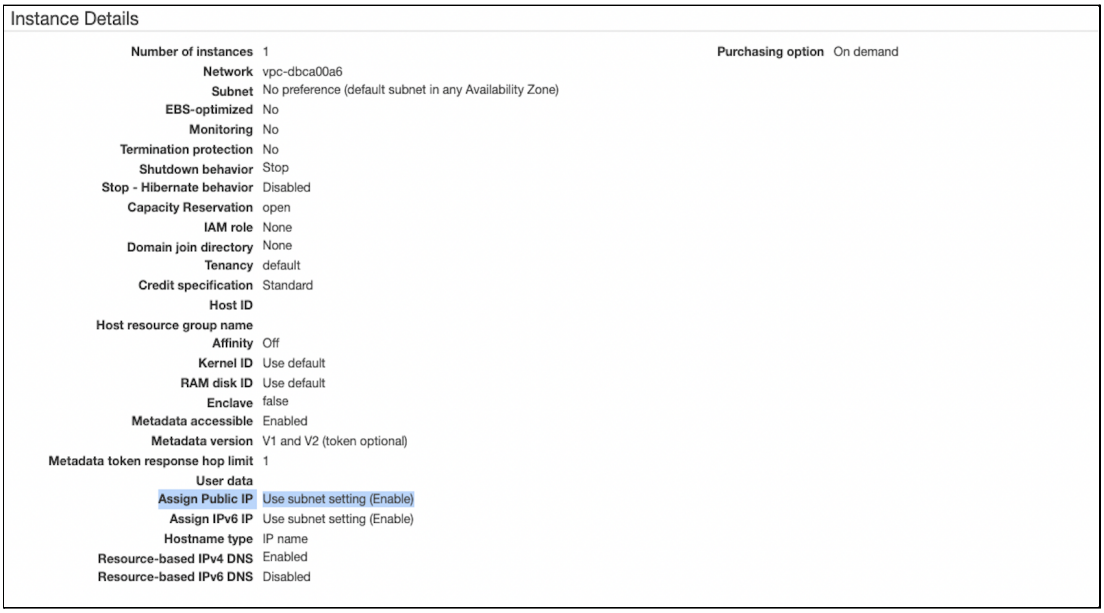

Figure 9.4 EC2 Configurations

- Selected the EBS storage of 8 GB.

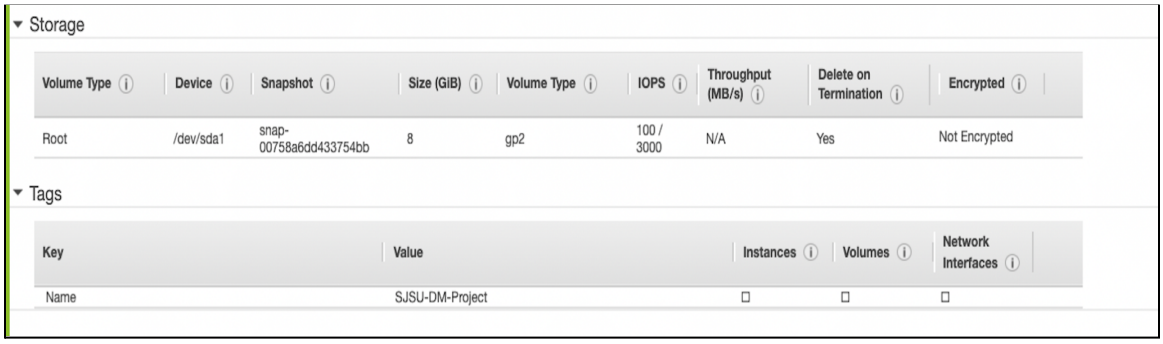

Figure 9.5 Storage allocation

- Before launching the EC2 instance we need to generate key-pair to connect toour Server.

  We downloaded the .pem file.

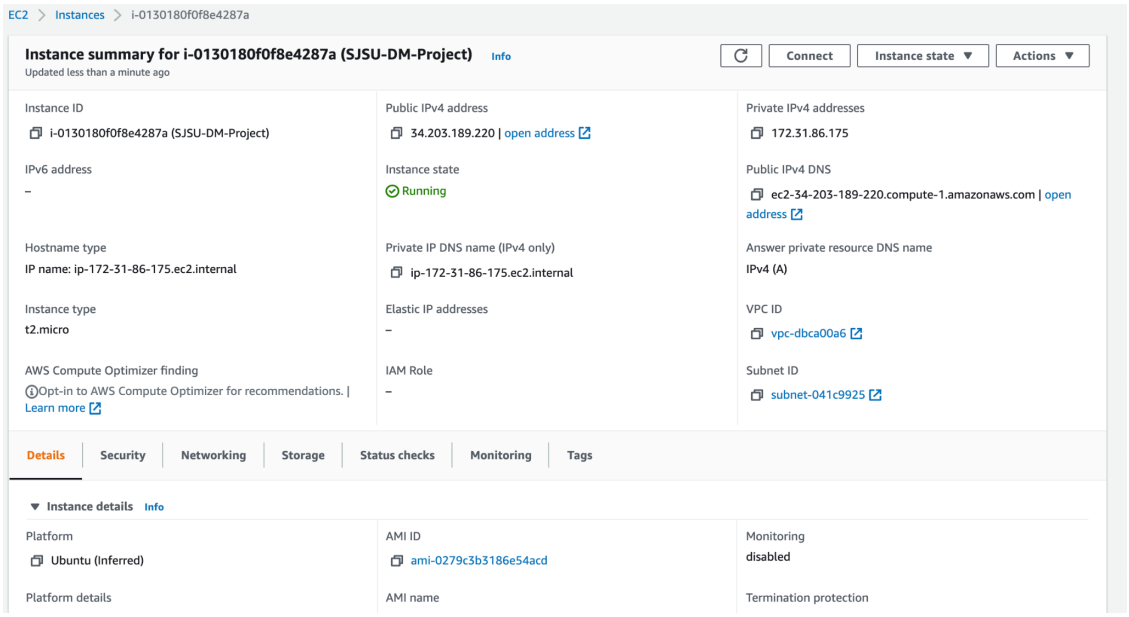

Figure 9.6 EC2 Instance Configuration

- We have created the .pem file to connect to our EC2 instance. After downloading the file, we could connect to EC2 by using following command:

  `ssh -i "dm-pro.pem" ubuntu@34.203.189.220`

  - We installed the required libraries used by Streamlit :

    `pip install streamlit`

  - For running the streamlit application, we cloned the project repo and ran the following command: `steamlit run app.py`

Chapter 10. Conclusion

We built an end-to-end machine learning application which will help people to plan/buy their dream house in California. We successfully trained the regression model on the historic housing data to predict the future house value with 82% accuracy. The application is developed using StreamLit and hosted on AWS cloud.

References

1. Qingqi Zhang, "Housing Price Prediction Based on Multiple Linear Regression", Scientific Programming, vol. 2021, Article ID 7678931, 9 pages, 2021. https://doi.org/10.1155/2021/7678931
2. Kamal N., Chaturvedi E., Gautam S., Bhalla S. (2021) House Price Prediction Using Machine Learning. In: Tavares J.M.R.S., Chakrabarti S., Bhattacharya A., Ghatak S. (eds) Emerging Technologies in Data Mining and Information Security. Lecture Notes in Networks and Systems, vol 164. Springer, Singapore. https://doi-org.libaccess.sjlibrary.org/10.1007/978-981-15-9774-9_73
3. Brownlee, J. (2019, May 21). *Difference between classification and regression in machine learning*. Machine Learning Mastery. Retrieved December 10, 2021, from https://machinelearningmastery.com/classification-versus-regression-in-machine-learning/.
4. *Linear regression: Introduction to linear regression for data science*. Analytics Vidhya. (2021, May 25). Retrieved December 10, 2021, from https://www.analyticsvidhya.com/blog/2021/05/all-you-need-to-know-about-your-first-machine-learning-model-linear-regression/.
5. https://mermaid-js.github.io/mermaid-live-editor/
6. *Scikit-learn: Machine Learning in Python*, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

7. *Lightgbm (Light Gradient Boosting Machine)*. GeeksforGeeks. (n.d.). Retrieved from https://www.geeksforgeeks.org/lightgbm-light-gradient-boosting-machine/.
8. Lughofer, Edwin, et al. “On Employing Fuzzy Modeling Algorithms for the Valuation of Residential Premises.” *Information Sciences*, vol. 181, no. 23, 2011, pp. 5123–5142., https://doi.org/10.1016/j.ins.2011.07.012.

Appendix

Application URL - http://3.83.55.217:8501/

GitHub Repo - https://github.com/anastaszi/255-DM-TeamProject

Google Colab -

https://colab.research.google.com/drive/1a6bil0zdlii3hkFdBtwZaaf8qzpLl0lL?usp=sharing

Models saved at -

https://drive.google.com/drive/folders/1d2HQRi4R5c53KvX_tDrXBjK0Gf-_COul?usp=sharing

Dataset -

https://drive.google.com/drive/folders/1gnLB5mHAYoIG2gjJHaBT0iBGUd-GqW6V?usp=sharing

Demo video -

https://drive.google.com/file/d/1v9mzwNPdaOTv4ous8WI0xq5baUrGFsKZ/view?usp=sharing

Demo Slides -

https://docs.google.com/presentation/d/1Qcz6yF53cHo6FGkDcnNK123qhYybfSn1mx1YDzEBidM/edit?usp=sharing