**TEAM MEMBERS**

|  |  |
| --- | --- |
| **NAME** | **EMAIL ID** |
| Matchimai.K.E | [matchierumporai@gmail.com](mailto:matchierumporai@gmail.com) |
| Mirudula.V | [**preethipriya2128@gmail.com**](mailto:preethipriya2128@gmail.com) |
| Parkavi.KS | [**mirudula2503@gmail.com**](mailto:mirudula2503@gmail.com) |
| Preethi.R | [**parkavisathya982@gmail.com**](mailto:parkavisathya982@gmail.com) |

**Phase 5:** Project Documentation & Submission

**Topic:** In this section we will document the complete project and prepare it for submission.



**Introduction:**

* The real estate market is a dynamic and complex arena, where property values can fluctuate significantly due to a multitude ofϖ factors. For both homebuyers and sellers, accurately determining the fair market value of a property is of paramount importance. In this era of technological advancement, machine learning base merged as a game-changing tool in the realm of real estate. One of its most compelling applications is predicting house prices with remarkable accuracy.
* Traditional methods of property valuation, relying on factors such as location, square footage, and recent sales data, are undoubtedly useful. However, they often fall short in capturing the intricacies and nuances that drive real estate market dynamics.
* Machine learning, on the other hand, has the capability to process vast volumes of data and identify patterns that human app raisers might overlook. This technology has the potential to revolutionize the way we value real estate, offering more precise and data-driven predictions.
* In this exploration, we delve into the exciting world of predicting house prices using machine learning. We will uncover how this cutting-edge technology harnesses the power of algorithms and data to create predictive models that consider an array of variables, such as neighborhood characteristics, property features, economic indicators, and even social trends.
* By doing so, machine learning enables us to make informed, data backed predictions about the future value of a property.
* This transformation of the real estate industry is not only beneficialϖ for buyers and sellers but also for investors, developers, and policymakers. Accurate house price predictions can inform investment decisions, urban planning, and housing policy development, leading to a more efficient and equitable real estate market. As we embark on this journey into the realm of machine learning for house price prediction, we will explore the various techniques, data sources, and challenges involved.

Dataset Link: ( <https://www.kaggle.com/datasets/vedavyasv/usa-housing>)

Here's a list of tools and software commonly used in the process:

1. Programming Language: - Python is the most popular language for machine learning due to its extensive libraries and frameworks. You can use libraries like NumPy,pandas, scikit-learn, and more.
2. Integrated Development Environment (IDE): - Choose an IDE for coding and running machine learning experiments. Some popular options include Jupyter Notebook, GoogleColab, or traditional IDEs like PyCharm.
3. Machine Learning Libraries: - You'll need various machine learning libraries, including: - scikit-learn for building and evaluating machine learning models. - TensorFlow or PyTorch for deep learning, if needed. - XGBoost, LightGBM, or CatBoost for gradient boosting models.
4. Data Visualization Tools: - Tools like Matplotlib, Seaborn, or Plotly are essential for dataexploration and visualization.
5. Data Preprocessing Tools: - Libraries like pandas help with data cleaning, manipulation, andpreprocessing.
6. Data Collection and Storage: - Depending on your data source, you might need web scrapingtools (e.g., BeautifulSoup or Scrapy) or databases (e.g., SQLite, PostgreSQL) for data storage.
7. Version Control: - Version control systems like Git are valuable for tracking changes in your code and collaborating with others.
8. Notebooks and Documentation: - Tools for documenting your work, such as Jupyter Notebooksor Markdown for creating README files and documentation.
9. Hyper parameter Tuning: - Tools like Grid Search CV or Randomized Search CV from scikit-learn can help with hyper parameter tuning.
10. Web Development Tools (for Deployment): - If you plan to create a web application for model deployment, knowledge of web development tools like Flask or Django for back end development, and HTML, CSS, and JavaScript for the front-end can be useful.
11. Cloud Services (for Scalability): - For large-scale applications, cloud platforms like AWS, Google Cloud, or Azure can provide scalable computing and storage resources.
12. External Data Sources (if applicable): - Depending on your project's scope, you might require tools toaccess external data sources, such as APIs or data scraping tools.
13. Data Annotation and Labeling Tools (if applicable): - For specialized projects, tools for data annotation andlabeling may be necessary, such as Labelbox or Supervisely.
14. Geospatial Tools (for location-based features): - If your dataset includes geospatial data, geospatial libraries like Geo Pandas can be helpful.

**Understanding**:

To tackle this problem effectively, we need to understand several key aspects:

1. **Data Collection**: Gather a comprehensive dataset that includes historical house sale prices and relevant features such as size, location, number of bedrooms, bathrooms, amenities, and neighborhood data.
2. **Data Preprocessing**: Clean the data by handling missing values, outliers, and encoding categorical variables. Feature engineering may also be necessary to create new informative features.
3. **Exploratory Data Analysis (EDA)**: Conduct EDA to gain insights into the data. Visualize relationships between features and the target variable (house prices) to identify patterns and correlations.
4. **Model Selection**: Choose appropriate machine learning algorithms for regression tasks. Common choices include linear regression, decision trees, random forests, support vector machines, and gradient boosting.
5. **Feature Selection**: Select the most relevant features to improve model performance and reduce overfitting. Techniques like feature importance analysis and recursive feature elimination can help.
6. **Model Training**: Split the data into training and testing sets to train and evaluate the chosen machine learning models. Perform cross-validation to ensure robustness.
7. **Hyperparameter Tuning**: Fine-tune model hyperparameters to optimize performance. Grid search or random search can be used for this purpose.
8. **Evaluation Metrics**: Select appropriate evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE) to measure the model's accuracy.
9. **Deployment**: Once a satisfactory model is developed, deploy it as an application or service that can provide real-time house price predictions.

**Design Thinking:**

Incorporate design thinking principles to create a user-centric solution:

1. **Empathize**: Understand the needs and pain points of users in the real estate market. Consider the perspectives of both buyers and sellers.
2. **Define**: Clearly define the problem and objectives. For example, if targeting homebuyers, the goal might be to provide them with accurate price estimates to aid their decision-making process.
3. **Ideate**: Brainstorm creative solutions for the problem. Think about how the machine learning model can be integrated into existing real estate platforms or apps.
4. **Prototype**: Create a prototype or mock-up of the user interface for the price prediction tool. Gather feedback from potential users to refine the design.
5. **Test**: Test the prototype with users to ensure it meets their needs and is easy to use. Make necessary adjustments based on user feedback.
6. **Implement**: Develop the full-fledged machine learning model and user interface, incorporating the insights gained during the design thinking process.
7. **Iterate**: Continuously gather feedback and iterate on the model and user interface to improve accuracy and user satisfaction.

**2. DESIGN INTO INNOVATION**

1. Data Collection: Gather a comprehensive dataset that includes features such as location, size, age, amenities, nearby schools, crime rates, and other relevant variables.
2. Data Preprocessing: Clean the data by handling missing values, outliers, and encoding categorical variables. Standardize or normalize numerical features as necessary.

**Content for phase 2 Project: Random Forest**

**Dataset link :(**<https://www.kaggle.com/datasets/vedavyasv/usa-housing>**)**

* **Program:**

**In [1]:**

model\_rf=RandomForestRegression(n\_estimators=50)

**In[2]:**

model\_rf.fit(X\_train\_scal,Y\_train)

**In[3]:**

print(r2\_score(Y\_test,Prediction2))

print(mean\_absolute\_error(Y\_test, Prediction2))

print (mean\_squared\_error(Y\_test, Prediction2))

* **Output:**

**Out[3]:**

-0.0006222175925689744

286137.81086908665

128209033251.4034

**3. BUILD LOADING AND PREPROCESSING THE DATASET**

1. Data Collection: Obtain a dataset that contains information about houses and their corresponding prices. This dataset can be obtained from sources like real estate websites, government records, or other reliable data providers.
2. Load the Dataset: Import relevant libraries, such as pandas for data manipulation and numpy for numerical operations.¬ Load the dataset into a pandas Data Frame for easy data handling.¬ You can use pd.read\_csv() for CSV files or other appropriate functions for different file formats.

preprocessing code :

from google.colab import drive

drive.mount("/content/drive")

import pandas as pd

data = pd.read\_csv('/content/sample\_data/california\_housing\_test.csv')

missing\_values = data.isnull().sum()

numeric\_columns = data.select\_dtypes(include=['number']).columns

data[numeric\_columns] = data[numeric\_columns].fillna(data[numeric\_columns].mean()) missing\_values\_after = data.isnull().sum()

data.to\_csv('preprocessed\_dataset.csv', index=False)

import numpy as np

import pandas as pd

import os

for dirname, \_, filenames in os.walk('/content/preprocessed\_dataset.csv'):

for filename in filenames:

print(os.path.join(dirname, filename))

import matplotlib.pyplot as plt

import seaborn as sb

data = pd.read\_csv('/content/preprocessed\_dataset.csv')

data.head()

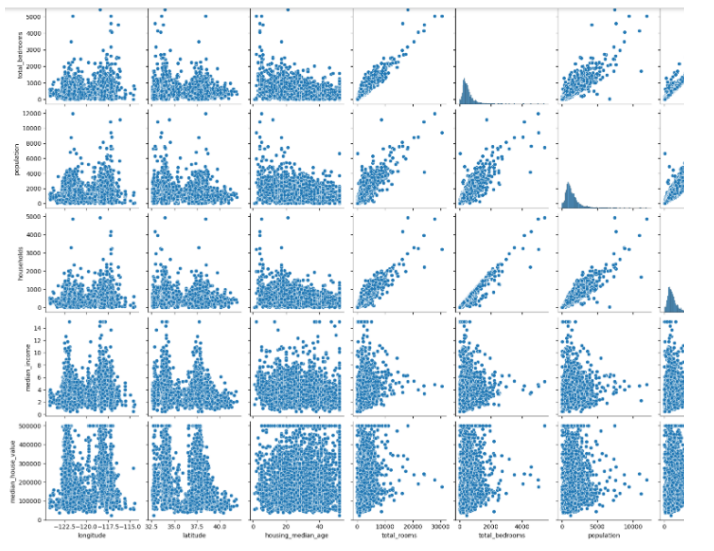
data.shape

data.info()

data.isna().sum()

data.duplicated().sum()

sb.pairplot(data = data)



**4. PERFORMING DIFFERENT ACTIVITIES LIKEFEATURE ENGINEERING, MODEL TRAINING, EVALUATION etc.,**

**Overview of the process:**

The following is an overview of the process of building a house price prediction model by feature selection, model training, and evaluation:

1. Prepare the data: This includes cleaning the data, removing outliers, and handling missing values.
2. Perform feature selection: This can be done using a variety of methods, such as correlation analysis, information gain, and recursive feature elimination.
3. Train the model: There are many different machine learning algorithms that can be used for house price prediction. Some popular choices include linear regression, random forests, and gradient boosting machines.
4. Evaluate the model: This can be done by calculating the mean squared error (MSE) or the root mean squared error (RMSE) of the model’s predictions on the held-out test set.
5. Deploy the model: Once the model has been evaluated and found to be performing well, it can be deployed to production so that it can be used to predict the house prices of new houses.

**Dataset:** ( <https://www.kaggle.com/datasets/vedavyasv/usa-housing>)

**Procedure:**

Feature selection:

1. Identify the target variable. This is the variable that you want to predict, such as house price.
2. Explore the data. This will help you to understand the relationships between the different features and the target variable. You can use data visualization and correlation analysis to identify features that are highly correlated with the target variable.
3. Remove redundant features. If two features are highly correlated with each other, then you can remove one of the features, as they are likely to contain redundant information.
4. Remove irrelevant features. If a feature is not correlated with the target variable, then you can remove it, as it is unlikely to be useful for prediction.

**Feature Selection:**

We are selecting numerical features which have more than 0.50 or less than -0.50 correlation rate based on Pearson Correlation Method—which is the default value of parameter "method" in corr() function. As for selecting categorical features, Is elected the categorical values which I believe have significant effect on the target variable such as Heating and MS Zoning

Model training: 1. Choose a machine learning algorithm. There are a number of different machine learning algorithms that can be used for house price prediction, such as linear regression, ridge regression, lasso regression, decision trees, and random forests are Covered above.

**Model training:**

Once the model is trained, it can be used to predict house prices for new data. For example, you could use the model to predict the price of a house that you are interested in buying

1. Prepare the data. This involves cleaning the data, removing any errors or inconsistencies, and transforming the data into a format that is compatible with the machine learning algorithm that you will be using.
2. Split the data into training and test sets. The training set will be used to train the model, and the test set will be used to evaluate the performance of the model on unseen data.
3. Choose a machine learning algorithm. There are a number of different machine learning algorithms that can be used for house price prediction, such as linear regression, ridge regression, lasso regression, decision trees, and random forests
4. Tune the hyper parameters of the algorithm. The hyper parameters of a machine learning algorithm are parameters that control the learning process. It is important to tune the hyper parameters of the algorithm to optimize its performance.
5. Train the model on the training set. This involves feeding the training data to the model and allowing it to learn the relationships between the features and house prices.
6. Evaluate the model on the test set. This involves feeding the test data to the model and measuring how well it predicts the house prices. If the model performs well on the test set, then you can be confident that it will generalize well to new data.

**Model evaluation:**

1. Calculate the evaluation metrics. There are a number of different evaluation metrics that can be used to assess the performance of a machine learning model, such as R-squared, mean squared error(MSE), and root mean squared error (RMSE).
2. Interpret the evaluation metrics. The evaluation metrics will give you an idea of how well the model is performing on unseen data. If the model is performing well, then you can be confident that it will generalize well to new data. However, if the model is performing poorly, then you may need to try a different model or retune the hyper parameters of the current model

**Model evaluation:** Model evaluation is the process of assessing the performance of a machine learning model on unseen data. This is important to ensure that the model will generalize well to new data.ϖ R-squared: This metric measures how well the model explains the variation in the actual house prices.

* Mean absolute error (MAE): This metric measures the average absolute difference between the predicted and actual house prices
* Root mean squared error (RMSE): This metric is the square root of the MSE.
* Mean squared error (MSE): This metric measures the average squared difference between the predicted and actual house prices.
* There are a number of different metrics that can be used to evaluate the performance of a house price prediction model. Some of the most common metrics include:
* Bias: Bias is the tendency of a model to consistently over- or In addition to these metrics, it is also important to consider the following factors when evaluating a house price prediction model: Interpretability: Interpretability is the ability to understand how the model makes its predictions. This is important for house price prediction models, as it allows users to understand the factors
* Variance: Variance is the measure of how much the predictions of a model vary around the true house prices. underestimate house prices. influence the predicted house prices.

**Feature Engineering:**

Feature engineering is a crucial aspect of building a house price prediction model using machine learning. It involves creating new features, transforming existing ones, and selecting the most relevant variables to improve the model's predictive power. Here are some feature engineering ideas for house price prediction:

1. Total Area Features: Combine individual room areas to create features like "Total Living Area," "Total Bedroom Area," or "Total Bathroom Area." These can be significant predictors of house price
2. Ratio Features: Create features that represent ratios, such as the "Bedroom to Bathroom Ratio" or "Living Area to Lot Area Ratio." These ratios may capture the property's layout and functionality
3. Age of the Property: Calculate the age of the property by subtracting the construction year from the current year. Newer properties might have higher values.
4. Neighborhood Statistics: Aggregate neighborhood-level statistics, such as the average income, crime rate, school ratings, or proximity to amenities, and use these as features.
5. .Distance to Key Locations: Calculate distances from the property to essential places like schools, parks, shopping centers, or public transportation hubs. Closer proximity to such amenities can affect the price.
6. Categorical Encodings: Use techniques like one-hot encoding, label encoding, or target encoding for categorical variables, such as property type, heating system, or garage type.
7. Seasonal Features: Create features indicating the season during which the house was sold. Seasonality can influence property demand and prices.
8. Historical Data: Incorporate historical data on house prices and local real estate market trends. This can help the model account for cyclical patterns.
9. Exterior Features: Develop features related to the property's exterior, such as the presence of a swimming pool, patio, or garden. These features can be valuable for determining a property's appeal.
10. Quality Scores: Create a combined quality score by aggregating the quality ratings of various components of the property, such as kitchen quality, bathroom quality, and overall house quality.
11. Logarithmic Transformations: Apply logarithmic transformations to features like "Lot Area" or" Number of Bedrooms" to make their distributions more normal.
12. Interaction Features: Create interaction terms by multiplying or dividing relevant features. For example, "Number of Bathrooms" multiplied by "Total Living Area" can represent the total bathroom area.
13. Missing Value Indicators: Create binary indicators for missing values in the dataset. The presence of missing data can be an informative feature.
14. Density Features: Compute population density in the neighborhood or the density of certain property types. High density might impact property prices.
15. Sentiment Analysis: Analyze online reviews or social media sentiment related to the property or neighborhood to capture public perception.
16. Time-Related Features: Incorporate time-related features like day of the week, month, or year when the property was listed or sold.
17. Zoning Information: Include zoning information that can affect property use, such as residential, commercial, or mixed-use zoning.
18. .Accessibility Features: Create features to represent accessibility, like the number of nearby public transport stations or major highways.
19. .Energy Efficiency: Include features related to energy-efficient components, such as insulation, energy-efficient appliances, or solar panels.
20. Demographic Data: Use demographic data for the area to understand the potential buyer's income levels, family sizes, and preferences

**Program:**

**Input:**

data.head()

Avg. Area Income Avg. Area House Age Avg. Area Number of Rooms Avg. Area Number of Bedrooms Area Population Price Address

0 79545.458574 5.682861 7.009188 4.09 23086.800503 1.059034e+06 208 Michael Ferry Apt. 674\nLaurabury, NE 3701...

1 79248.642455 6.002900 6.730821 3.09 40173.072174 1.505891e+06 188 Johnson Views Suite 079\nLake Kathleen, CA...

2 61287.067179 5.865890 8.512727 5.13 36882.159400 1.058988e+06 9127 Elizabeth Stravenue\nDanieltown, WI 06482...

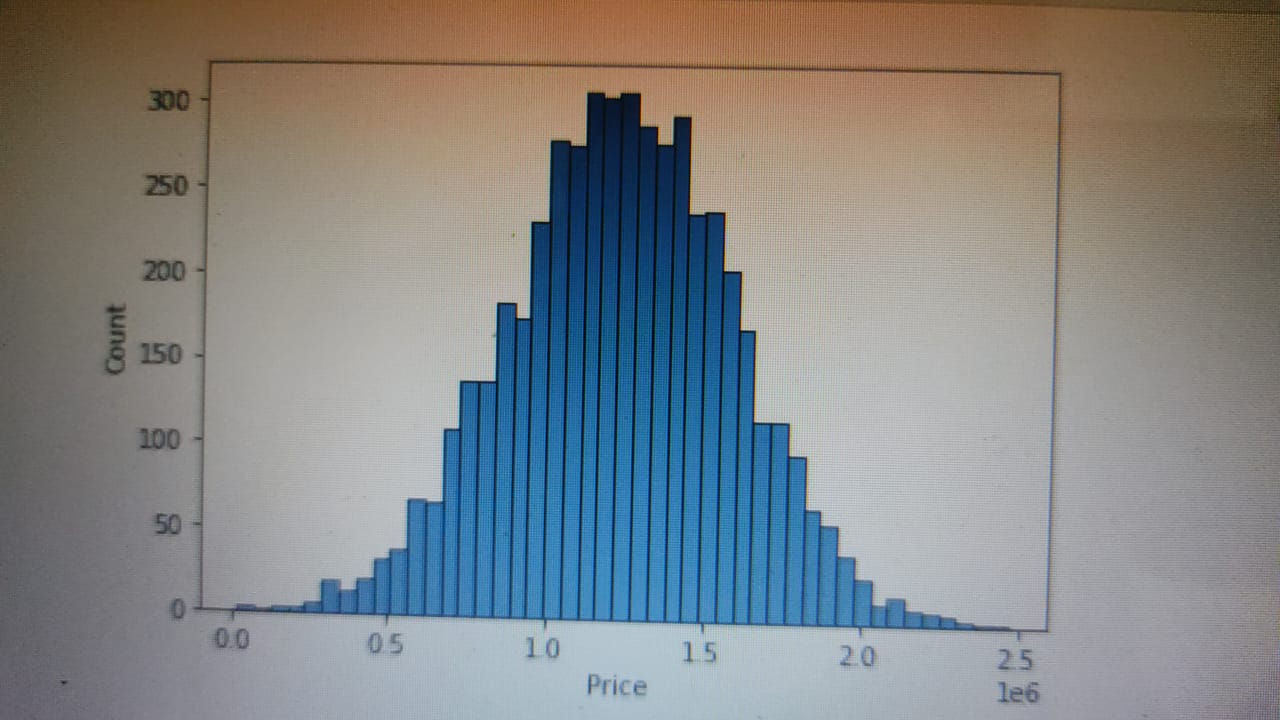
3 63345.240046 7.188236 5.586729 3.26 34310.242831 1.260617e+06 USS Barnett\nFPO AP 44820

4 59982.197226 5.040555 7.839388 4.23 26354.109472 6.309435e+05 USNS Raymond\nFPO AE 09386

**Step:1**

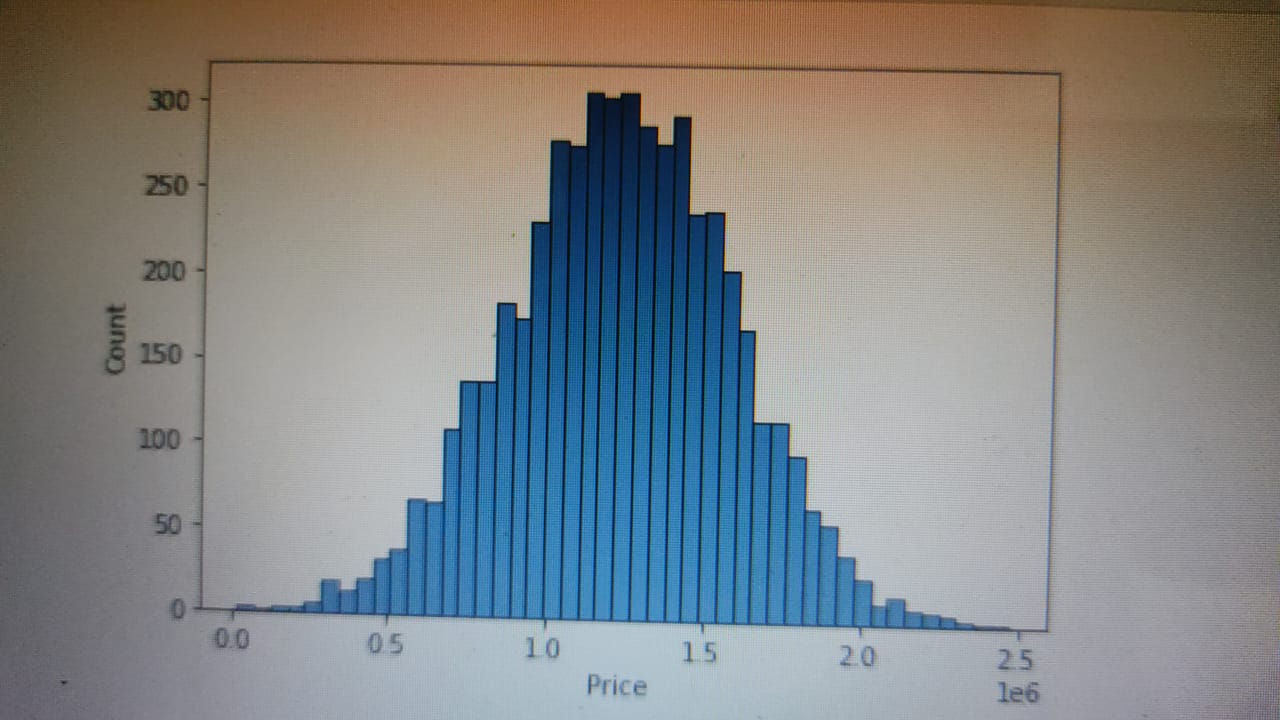
b.histplot(x = data['Price']);

**Output:**

****

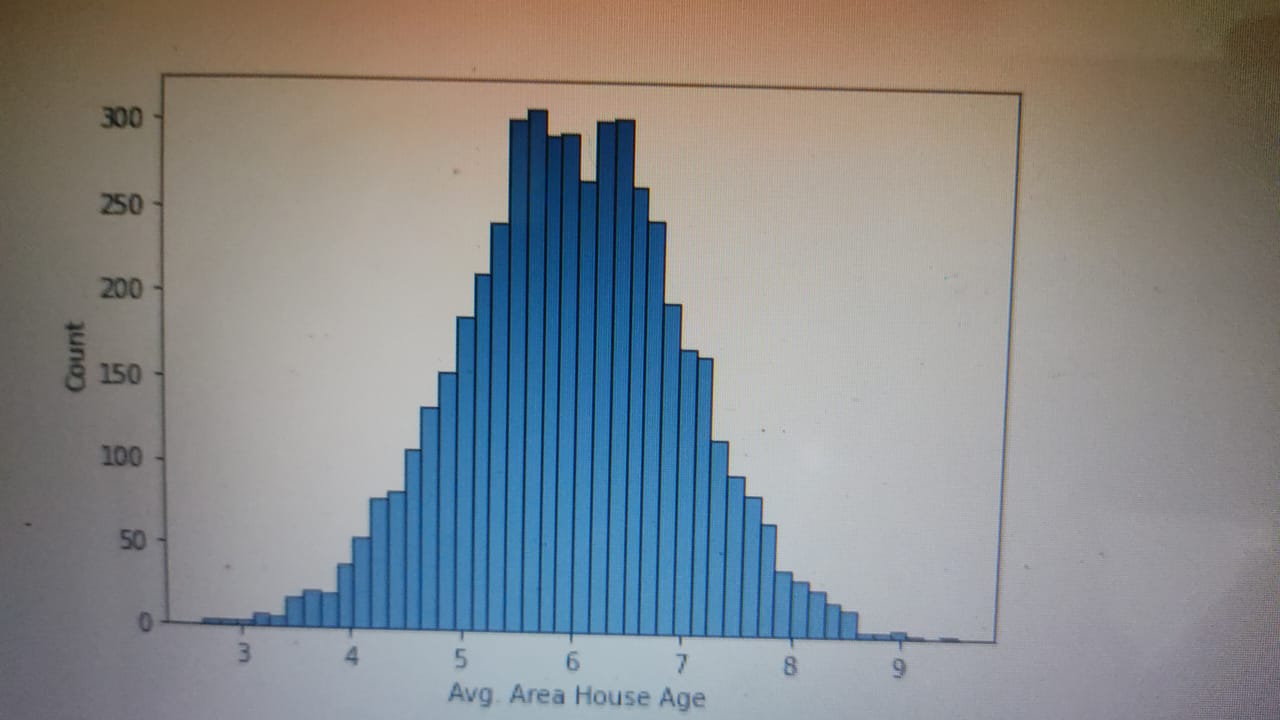
**Step:2**

**sb.histplot(x = data['Avg. Area Income']);**

****

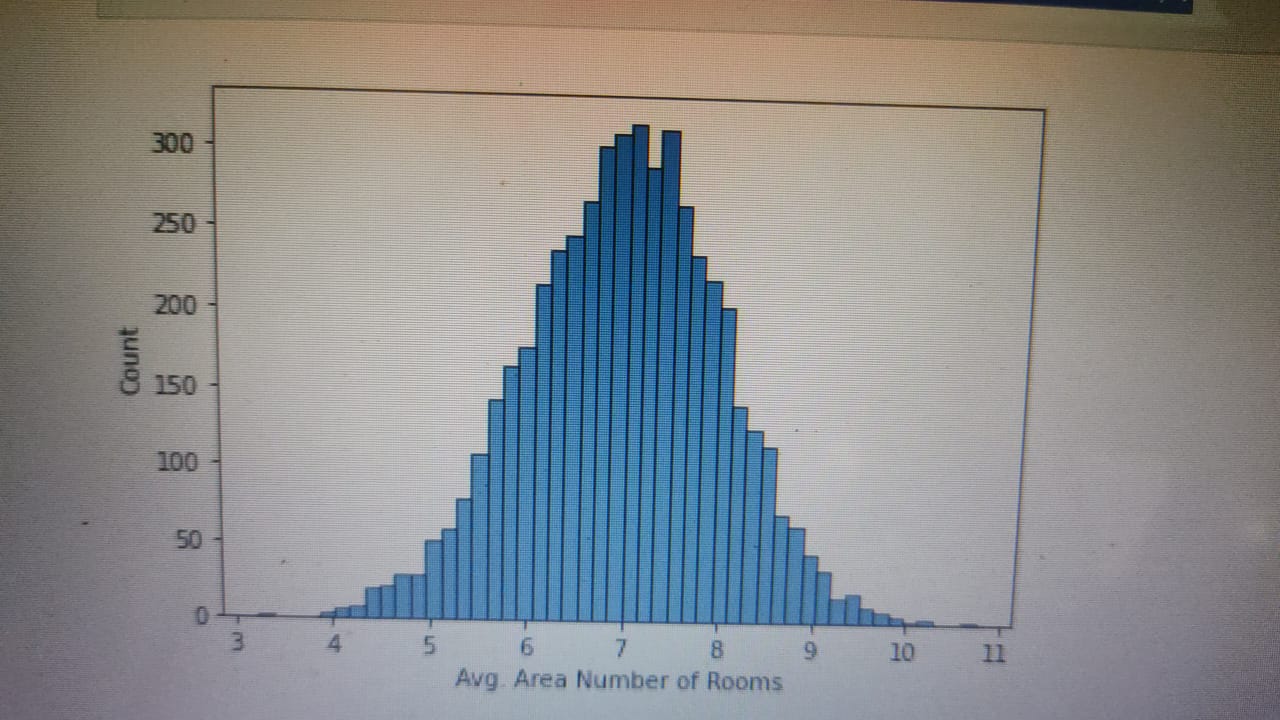
**Step 3:**

**sb.histplot(x = data['Avg. Area House Age'])**

****

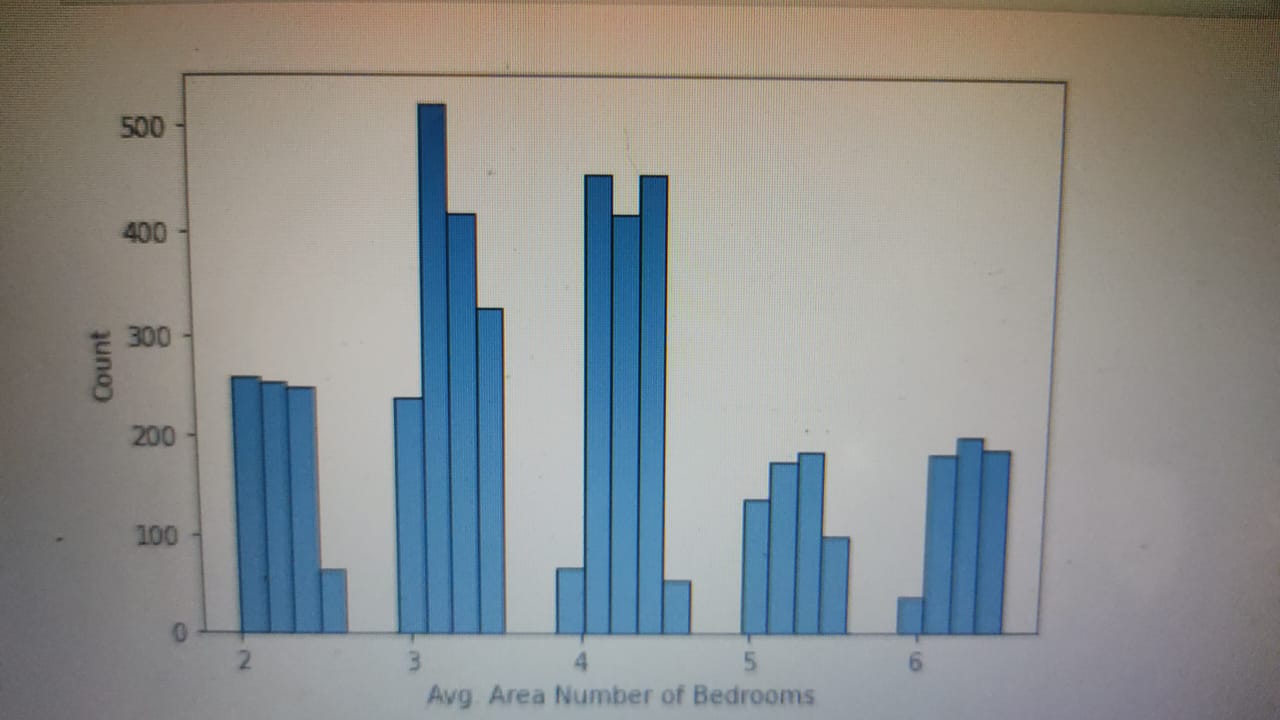
**Step4:**

**sb.histplot(x = data['Avg. Area Number of Rooms']);**

****

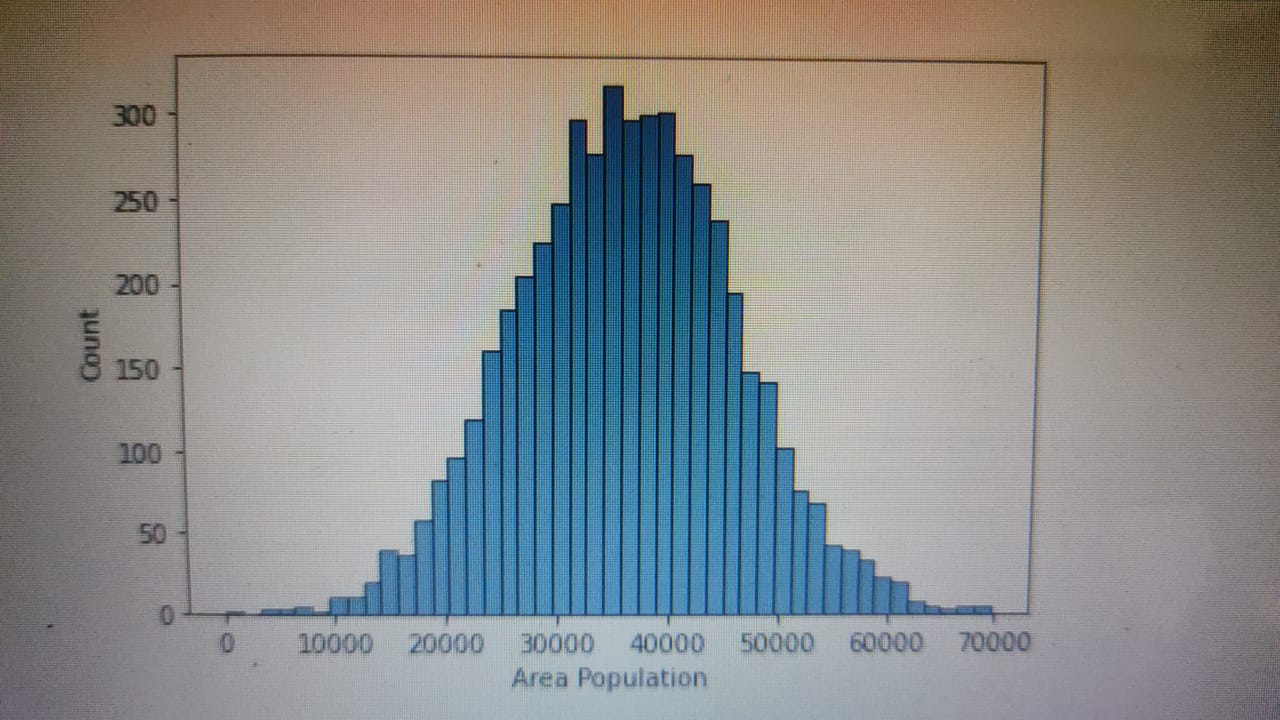
**Step:5**

**sb.histplot(x = data['Avg. Area Number of Bedrooms']);**

****

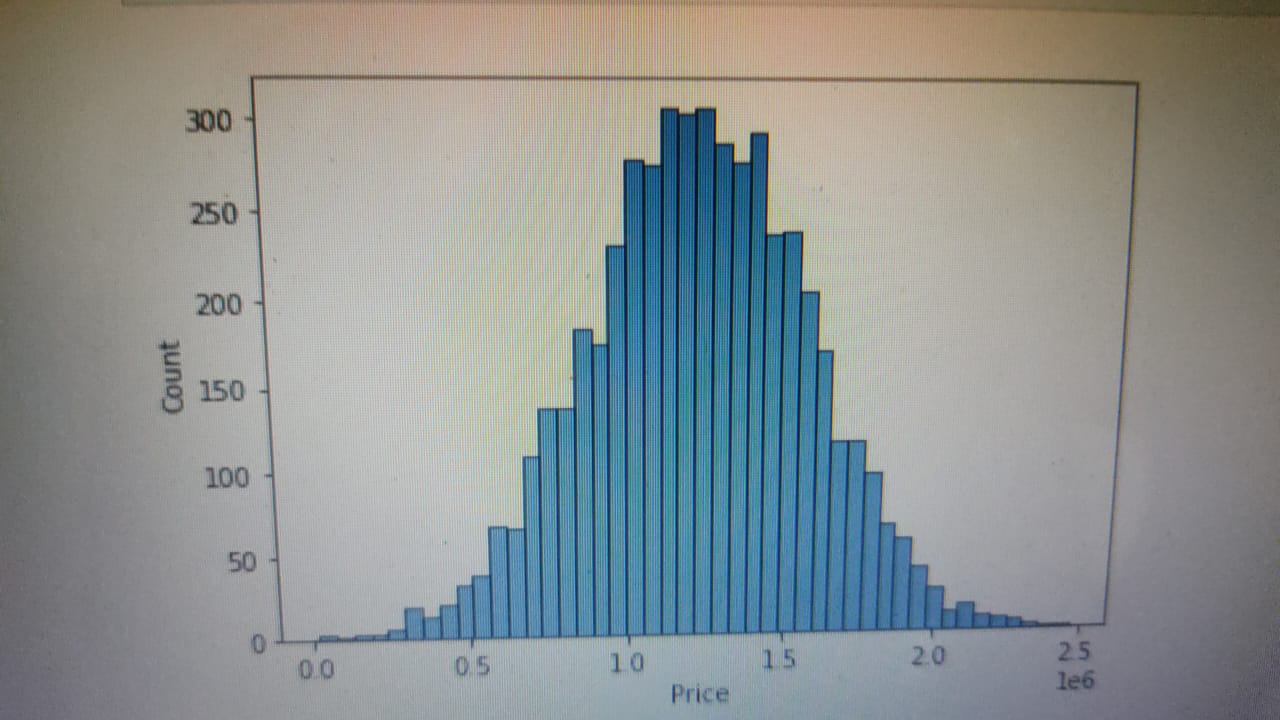
**Step6:**

**sb.histplot(x = data['Area Population']);**

****

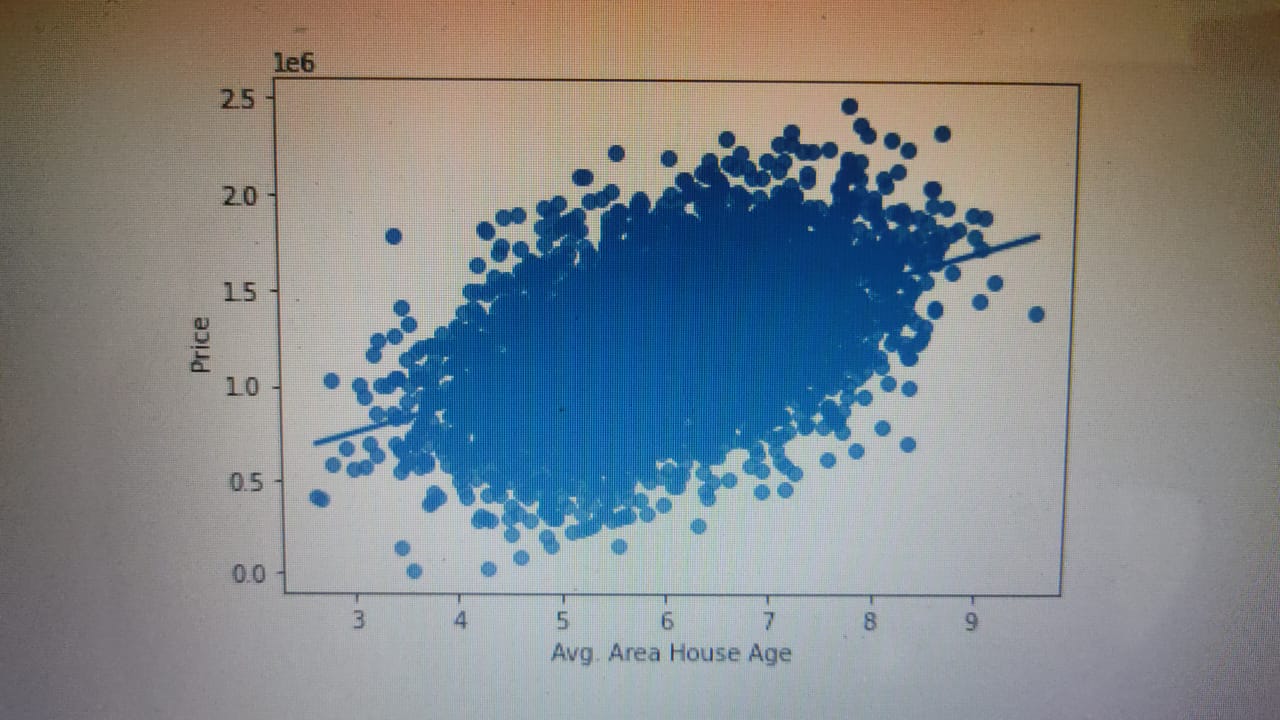
**Step7:**

**sb.histplot(x = data['Price']);**

****

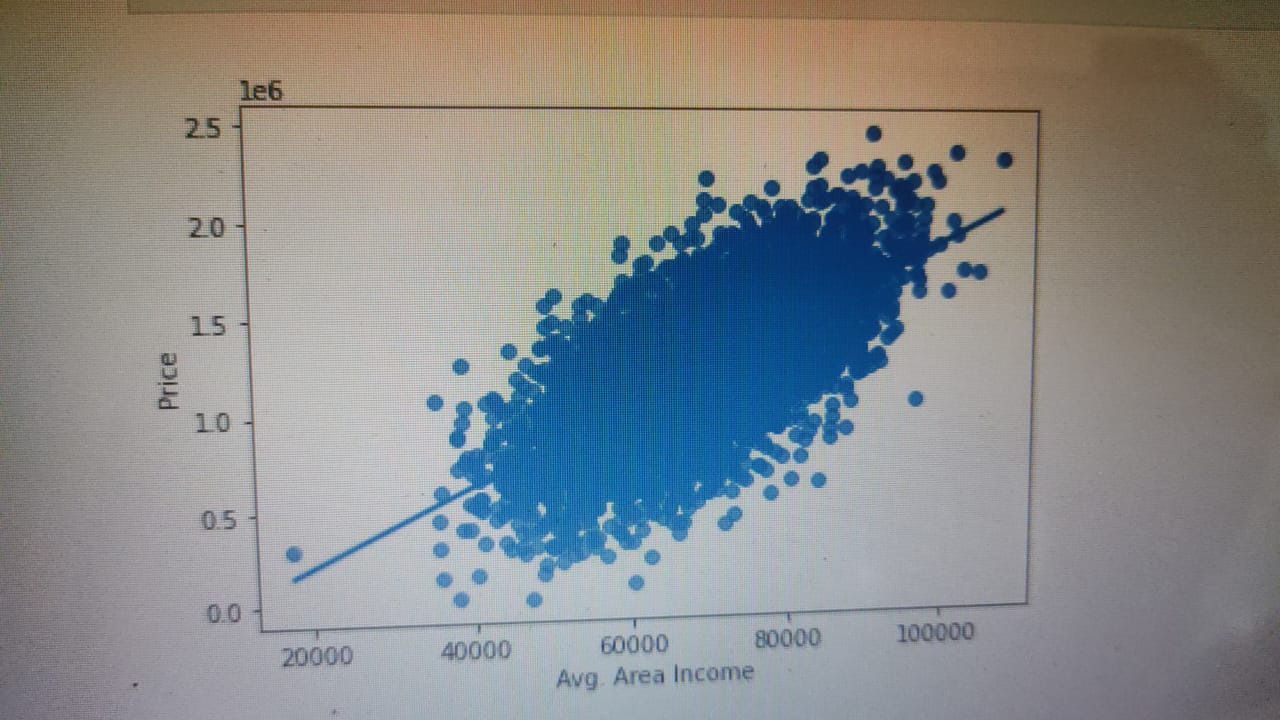
**Step8:**

**sb.regplot(x = data['Avg. Area House Age'], y = data['Price'])**



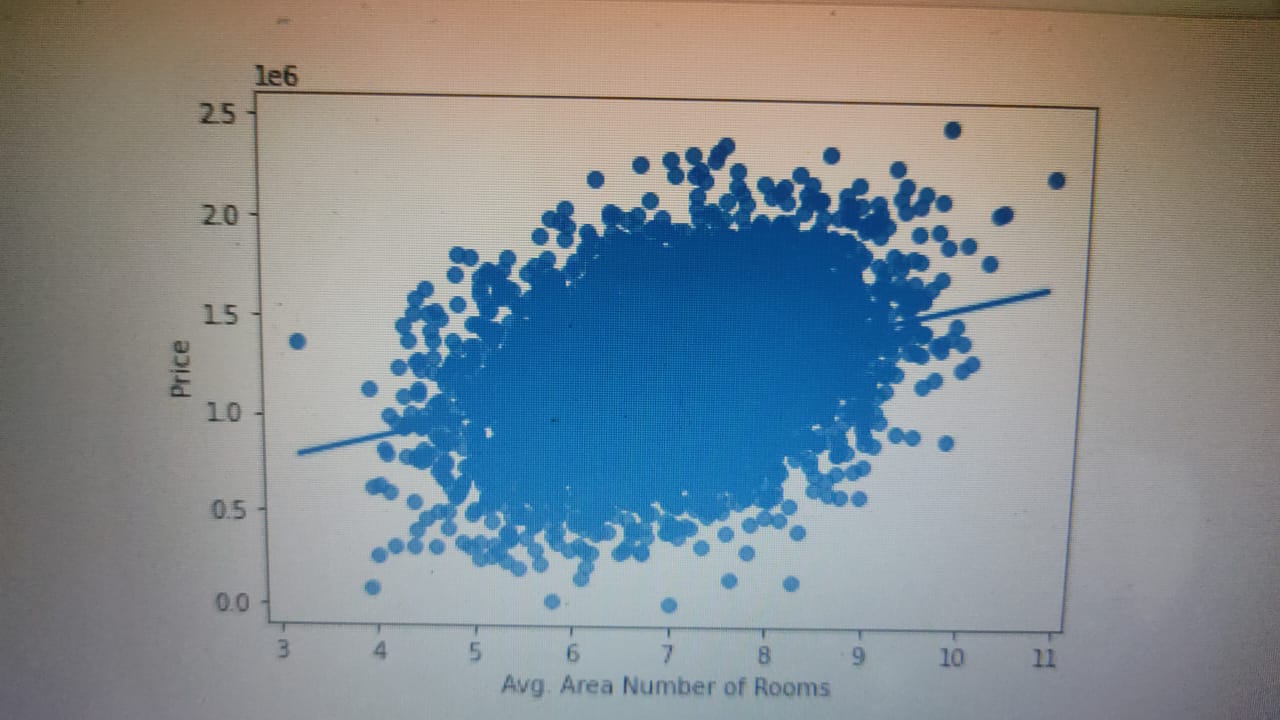
**Step9:**

sb.regplot(x = data['Avg. Area Income'], y = data['Price']);



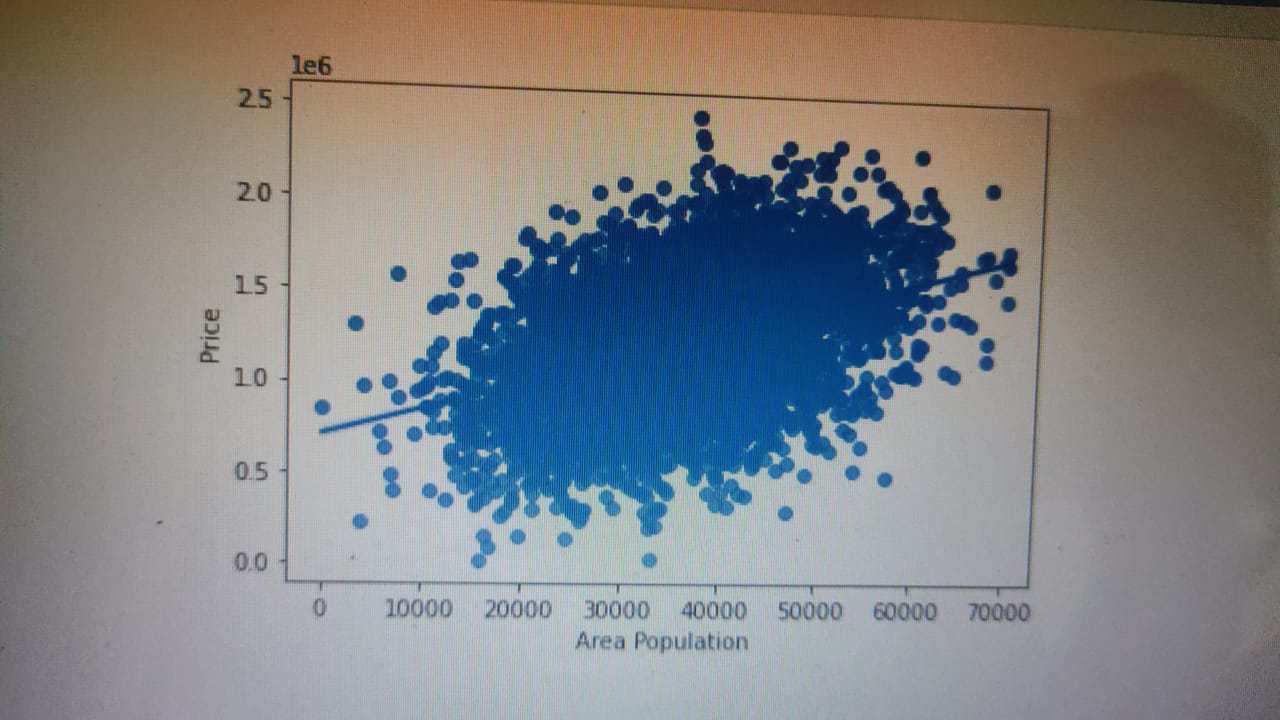
**Step** **10:**

sb.regplot(x = data['Avg. Area Number of Rooms'], y = data['Price']);



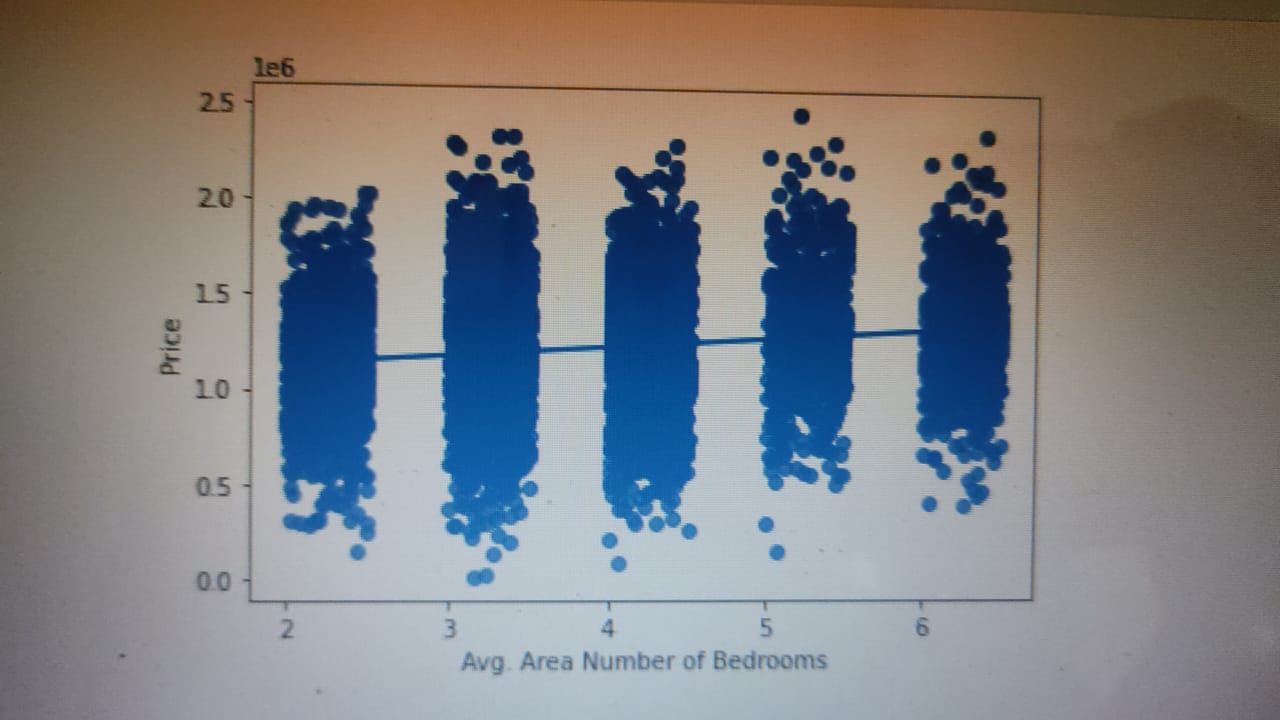
**Step11:**

sb.regplot(x = data['Area Population'], y = data['Price']);



**Step 12:**

sb.regplot(x = data['Avg. Area Number of Bedrooms'], y = data['Price']);



**Analysis**

Price increases with all the variables

Price increases sharply with increase in Average Area Income

plt.figure(figsize = (15, 10))

sb.heatmap(data.corr(), annot = True, cmap = 'mako')



y = data['Price']

X = data.drop(['Price', 'Address'], axis = 1)

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

from sklearn.metrics import mean\_absolute\_error

from sklearn.metrics import mean\_squared\_error

from sklearn.ensemble import RandomForestRegressor

train\_X, val\_X, train\_y, val\_y = train\_test\_split(X, y, random\_state = 42)

model = RandomForestRegressor(random\_state = 1)

model.fit(train\_X, train\_y)

preds = model.predict(val\_X)

print("MAE: ", mean\_absolute\_error(preds, val\_y))

print("RMSE: ", np.sqrt(mean\_squared\_error(preds, val\_y)))

MAE: 93812.37073246129

RMSE: 118380.48325186648

**Conclusion:**

* Predicting house prices using machine learning is a transformative and promising approach that has the potential to revolutionize the real estate industry. Throughout this exploration, we have uncovered the remarkable capabilities of machine learning in providing more accurate, data-driven, and nuanced predictions for property values. As we conclude, several key takeaways and implications emerge: Improved Accuracy: Machine learning models consider a myriad of variables, many of which may be overlooked by traditional methods.
* This result in more accurate predictions, benefiting both buyers and sellers who can make informed decisions based on a property's true value. Data-Driven Insights: These models provide valuable insights into the real estate market by identifying trends, neighborhood characteristics, and other factors that influence property prices.
* This information can be invaluable for investors, developers, and policymakers seeking to make strategic decisions. Market Efficiency: The increased accuracy in pricing predictions can lead to a more efficient real estate market, reducing overvaluation and under valuation of properties. This contributes to a fairer and more transparent marketplace. Challenges and Considerations: Machine learning for house price prediction is not without its challenges. Data quality, model interpretability, and ethical concerns are important considerations. Addressing these issues is crucial for the responsible and ethical deployment of this technology. P a g e|67Continual Advancement: The field of machine learning is continually evolving, and as it does, so will the accuracy and capabilities of predictive models. As more data becomes available and algorithms improve, we can expect even more sophisticated predictions in the future .In conclusion, the application of machine learning in predicting house prices is a groundbreaking development with far reaching implications. It empowers individuals, businesses, and governments to navigate the real estate market with more confidence and precision.
* However, it is essential to approach this technology with a clear understanding of its potential and limitations, ensuring that its benefits are harnessed responsibly for the betterment of the real estate industry and society as a whole. As machine learning continues to advance, we can look forward to a future where property valuation becomes increasingly precise and data-informed.