**A PROJECT REPORT**

**ON**

**BANGALORE HOUSE PRICE PREDICTION**

Submitted in partial fulfilment for the requirement of the award of

TRAINING

IN

Data Analytics, Machine Learning and AI using Python



*Submitted By*

**AMEY THAKUR**

(Student at University of Mumbai affiliated Institute Terna Engineering College)

*Under the guidance of*

**Mr Bandenawaz Bagwan**

**ACKNOWLEDGEMENT**

My sincere gratitude and thanks towards my project paper guide Mr. Bandenawaz Bagwan. It was only with his backing and support that I could complete the report. He provided me with all sorts of help and corrected me if ever seemed to make mistakes. I have no such words to express my gratitude. Without his support, I couldn’t even start the work. So I am grateful to him. And last but not the least, I acknowledge my dearest parents for being such a nice source of encouragement and moral support that helped me tremendously in this aspect. I also declare to the best of my knowledge and belief that the Project Work has not been submitted anywhere else.

**ABSTRACT**

I propose to implement a house price prediction model of Bangalore, India. It’s a Machine Learning project which integrates Data Science and Web Development. I intend to deploy the app on the Heroku Cloud Application Platform. Housing prices fluctuate on a daily basis and are sometimes exaggerated rather than based on worth. The major focus of this project is on predicting home prices using genuine factors. Here, I intend to base an evaluation on every basic criterion that is taken into account when establishing the pricing. This project is a part of my training and internship in Data Analytics, Machine Learning and AI using Python at Diginique TechLabs.

**TABLE OF CONTENTS**

1. INTRODUCTION
   1. Context
   2. Motivation
   3. Objective
   4. Implementation
2. LITERATURE SURVEY
3. METHODOLOGY
   1. Data Collection
   2. Linear Regression
   3. Decision Tree Regression
   4. Classification Trees
   5. Regression Trees
   6. Support Vector Regression
   7. Random Forest Regression
4. PROJECT
   1. Problem Statement
   2. Data
   3. Dataset
   4. Model
   5. Web App
   6. Project Architecture
   7. Data Science
   8. Machine Learning
   9. Frontend
5. MODEL
   1. Steps to create the model
      1. Import Libraries
      2. Load Dataset
      3. Exploratory Data Analysis
      4. Data Cleaning
      5. Feature Engineering
      6. Dimensionality Reductions
      7. Outlier Removal using Business Logic
      8. Outlier Removal using Standard Deviation & Mean
      9. Data Visualization
      10. Building a Model
      11. Test the Model for few properties
      12. Export the tested model to a pickle file
6. DEPLOYMENT
   1. Steps to deploy the model on Heroku
7. RESULTS
8. CONCLUSION

**INTRODUCTION**

**1.1 Context**

This project was created as part of my training and internship at Diginique Techlabs for Data Science Machine Learning and AI using Python. My supervisor was Mr. Bandenawaz Bagwan, and I had one month to finish the job. We had two weeks of training sessions where we studied everything from the ground up.

**1.2 Motivation**

I am highly interested in anything related to Machine Learning, the independent project provided me with the opportunity to study and reaffirm my passion for this subject. The capacity to generate guesses, forecasts, and offer machines the ability to learn on their own is both powerful and infinite in terms of application possibilities. Machine Learning may be applied in finance, medicine, and virtually any other field. That is why I opted to base my idea on Machine Learning.

**1.3 Objective**

As a first project, I intended to make it as instructional as possible by tackling each stage of the machine learning process and attempting to comprehend it well. I picked Bangalore Real Estate Prediction as a method, which is known as a "toy issue," identifying problems that are not of immediate scientific relevance but are helpful to demonstrate and practice. The objective was to forecast the price of a specific apartment based on market pricing while accounting for various "features" that would be established in the following sections.

**1.4 Implementation**

I genuinely believe that sharing resources and knowledge allows us to not only benefit others but also ourselves. The project's sources can be found at the following URL:

Web Application: <https://bangalorehousepriceprediction.herokuapp.com>

Repository: <https://github.com/Amey-Thakur/BANGALORE-HOUSE-PRICE-PREDICTION>

Model: <https://www.kaggle.com/ameythakur20/bangalore-house-price-prediction-model>

**LITERATURE SURVEY**

Real Estate Property is not only a person's primary desire, but it also reflects a person's wealth and prestige in today's society. Real estate investment typically appears to be lucrative since property values do not drop in a choppy fashion. Changes in the value of the real estate will have an impact on many home investors, bankers, policymakers, and others. Real estate investing appears to be a tempting option for investors. As a result, anticipating the important estate price is an essential economic indicator. According to the 2011 census, the Asian country ranks second in the world in terms of the number of households, with a total of 24.67 crores. However, previous recessions have demonstrated that real estate costs cannot see the expenses of significant estate property are linked to the state's economic situation. Regardless, we don't have accurate standardized approaches to live the significant estate property values.

First, I looked at different articles and discussions about machine learning for housing price prediction. The title of the article is house price prediction, and it is based on machine learning and neural networks. The publication's description is minimal error and the highest accuracy. The aforementioned title of the paper is Hedonic models based on price data from Belfast infer that submarkets and residential valuation this model is used to identify over a larger spatial scale and implications for the evaluation process related to the selection of comparable evidence and the quality of variables that the values may require. Understanding current developments in house prices and homeownership are the subject of the study. In this article, they utilized a feedback mechanism or social pandemic that fosters a perception of property as an essential market investment.

**METHODOLOGY**

**3.1 Data Collection**

The statistics were gathered from Bangalore home prices. The information includes many variables such as area type, availability, location, BHK, society, total square feet, bathrooms, and balconies.

**3.2 Linear Regression**

Linear regression is a supervised learning technique. It is responsible for predicting the value of a dependent variable (Y) based on a given independent variable (X). It is the connection between the input (X) and the output (Y). It is one of the most well-known and well-understood machine learning algorithms. Simple linear regression, ordinary least squares, Gradient Descent, and Regularization are the linear regression models.

**3.3 Decision Tree Regression**

It is an object that trains a tree-structured model to predict data in the future in order to provide meaningful continuous output. The core principles of decision trees, Maximizing Information Gain, Classification trees, and Regression trees are the processes involved in decision tree regression. The essential notion of decision trees is that they are built via recursive partitioning. Each node can be divided into child nodes, beginning with the root node, which is known as the parent node. These nodes have the potential to become the parent nodes of their resulting offspring nodes. The nodes at the informative features are specified as the maximizing information gain, to establish an objective function that is to optimize the tree learning method.

**3.4 Classification Trees**

Classification trees are used to forecast the object into classes of a categorical dependent variable based on one or more predictor variables.

**3.5 Regression Trees**

It supports both continuous and categorical input variables. Regression trees are regarded as research with various machine algorithms for the regression issue, with the Decision Tree approach providing the lowest loss. The R-Squared value for the Decision Tree is 0.998, indicating that it is an excellent model. The Decision Tree was used to complete the web development.

**3.6 Support Vector Regression**

Supervised learning is linked with learning algorithms that examine data for classification and regression analysis.

**3.7 Random Forest Regression**

It is an essential learning approach for classification and regression to create a large number of decision trees. Preliminaries of decision trees are common approaches for a variety of machine learning problems. Tree learning is required for serving n off the self-produce for data mining since it is invariant despite scaling and several other changes. The trees are grown very deep in order to learn a high regular pattern. Random forest is a method of averaging several deep decision trees trained on various portions of the same training set. This comes at the price of a slight increase in bias and some interoperability.

**PROJECT**

**4.1 Problem Statement**

Create a model to estimate the price of houses in Bengaluru and host it on Heroku.

**4.2 Data**

The data is the most important aspect of a machine learning assignment, to which special attention should be paid. Indeed, the data will heavily affect the findings depending on where we found them, how they are presented, if they are consistent, if there is an outlier, and so on. Many questions must be addressed at this stage to ensure that the learning algorithm is efficient and correct.

To obtain, clean, and convert the data, many sub steps are required. I'll go through each one to illustrate how they've been used in my project and why they're helpful for the machine learning section.

**4.3 Dataset**

Dataset: <https://www.kaggle.com/amitabhajoy/bengaluru-house-price-data>

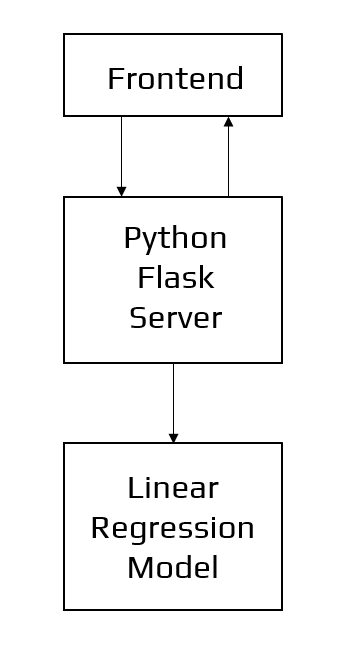
**4.4 Model**

Model: <https://www.kaggle.com/ameythakur20/bangalore-house-price-prediction-model>

**4.5 Web App**

Web Application: <https://bangalorehousepriceprediction.herokuapp.com>

**4.6 Project Architecture**



Architecture of the Application

**4.7 Data Science**

The first stage is standard data science work, in which we take a data set named ‘Bengaluru House pricing data' from Kaggle. We will do significant data cleaning on it to guarantee that it provides reliable predictions throughout prediction.

This jupyter notebook, ‘Bangalore-House-Price-Prediction-Model.ipynb,' is where we do all of our data science work. Because the jupyter notebook is self-explanatory, I'll only touch on the principles that I've implemented briefly. In terms of data cleansing, our dataset needs a significant amount of effort. In fact, 70% of the notebook is dedicated to data cleaning, in which we eliminate empty rows and remove superfluous columns that will not aid in prediction.

The process of obtaining valuable and significant information from a dataset that will contribute the most to a successful prediction is the next stage.

The final stage is to deal with outliers. Outliers are abnormalities that do massive damage to data and prediction. There is a lot to comprehend conceptually from the dataset in order to discover and eliminate these outliers.

Finally, the original dataset of over 13000 rows and 9 columns is reduced to about 7000 rows and 5 columns.

**4.8 Machine Learning**

The resulting data is fed into a machine learning model. To find the optimal procedure and parameters for the model, we will mostly employ K-fold Cross-Validation and the GridSearchCV approach.

It turns out that the linear regression model produces the best results for our data, with a score of more than 80%, which is not terrible.

Now, we need to export our model as a pickle file (Bengaluru House Data.pickle), which transforms Python objects into a character stream. Also, in order to interact with the locations(columns) from the frontend, we must export them into a JSON (columns.json) file.

**4.9 Frontend**

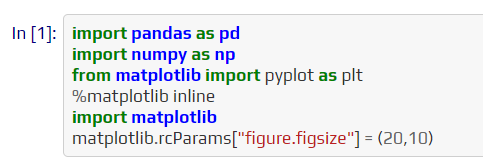
The front end is built up of straightforward HTML. To receive an estimated pricing, the user may fill-up the form with the number of square feet, BHK, bathrooms, and location and click the ‘ESTIMATE PRICE' button. The JavaScript file is in charge of connecting with the backend flask server routes as well as the frontend HTML. It takes the form data entered by the user and executes the function, which employs the prediction model to calculate the projected price in lakhs of rupees (1 lakh = 100000).

**MODEL**

**5.1 Steps to create the model**

1. Import Libraries
2. Load Dataset
3. Exploratory Data Analysis
4. Data Cleaning
5. Feature Engineering
6. Dimensionality Reductions
7. Outlier Removal using Business Logic
8. Outlier Removal using Standard Deviation & Mean
9. Data Visualization
10. Building a Model
11. Test the Model for few properties
12. Export the tested model to a pickle file

**5.1.1 Import Libraries**



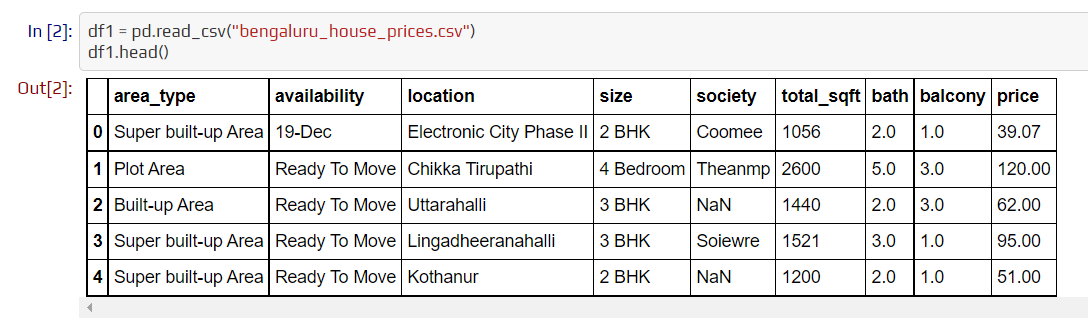
Pandas - Pandas is a data manipulation and analysis software package built-in Python. It specifically provides data structures and functions for manipulating numerical tables and time series.

NumPy - NumPy is an abbreviation for Numerical Python. NumPy is a Python package that is used to work with arrays. It also includes functions for working with linear algebra, Fourier transforms, and matrices.

Matplotlib - Matplotlib is a charting toolkit for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding charts into applications utilizing general-purpose GUI toolkits. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack.

**5.1.2 Load Dataset**

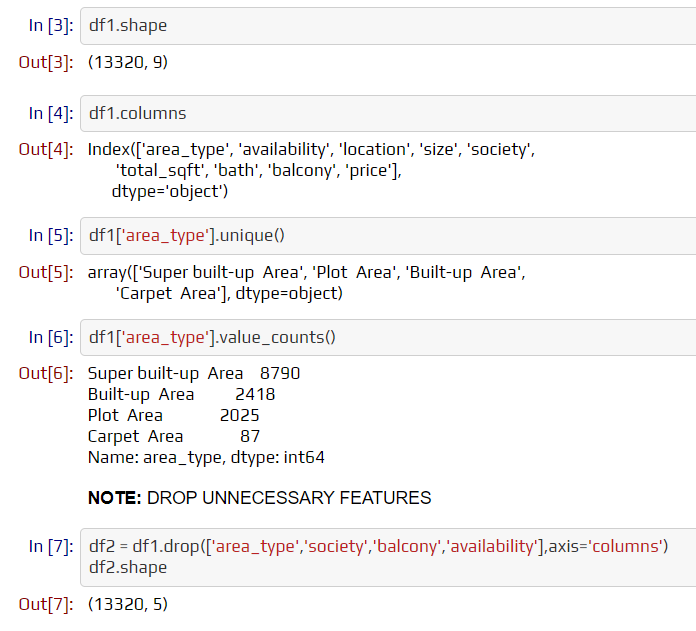
<https://www.kaggle.com/amitabhajoy/bengaluru-house-price-data>

****

read\_csv - read\_csv is an important pandas function to read CSV files and do operations on them.

head() - To obtain the first n rows, use the head() function. This function retrieves the object's first n rows based on location. It is handy for rapidly determining whether your object contains the correct type of data.

**5.1.3 Exploratory Data Analysis**

****

shape - The function "shape" returns the shape of an array. The shape is a tuple of numbers. These values represent the lengths of the appropriate array dimension. In other words, the "shape" of an array is a tuple with the number of items per axis (dimension).

columns - columns return the column labels of the provided Dataframe.

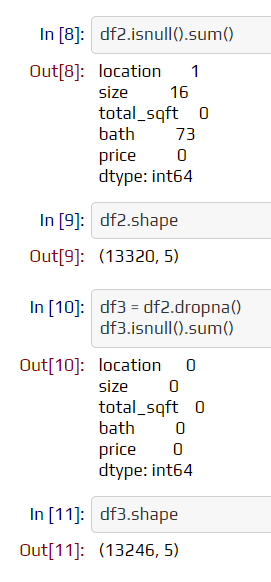
unique() - The unique() function is used to locate the array's unique items. Returns the array's sorted unique items.

NOTE: Drop unnecessary features.

counts() - The value counts() function returns an object with counts of unique values. The resultant object will be arranged in decreasing order, with the first element being the most often occurring. By default, NA values are excluded.

drop() - The drop() function is used to remove a set of labels from a row or column. By supplying label names and associated axes, or by explicitly specifying index or column names, you can remove rows or columns.

**5.1.4 Data Cleaning**

****

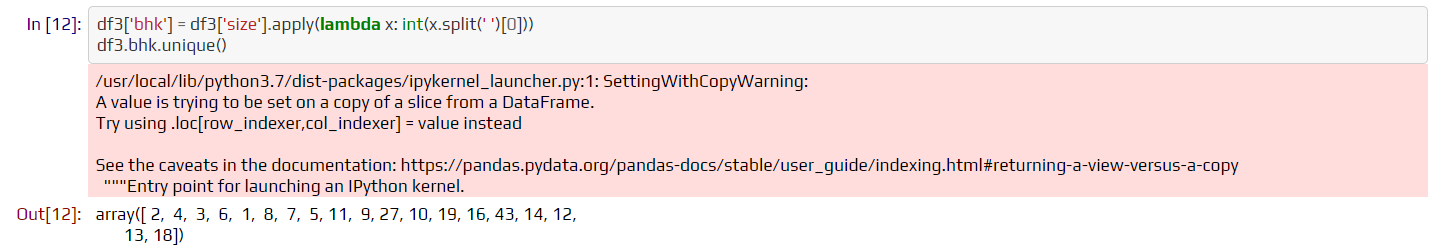
isnull() - The isnull() function finds missing values in a series object. It returns a boolean same-sized object that indicates if the values are NA or not. Missing values are translated to True, whereas non-missing values are assigned to False.

sum() - The sum() function in Python computes the sum of all numerical values in an iterable. sum() may be used with both integers and floating-point values. The sum() function in Python may be used to compute the sum of all values in an inerrable object.

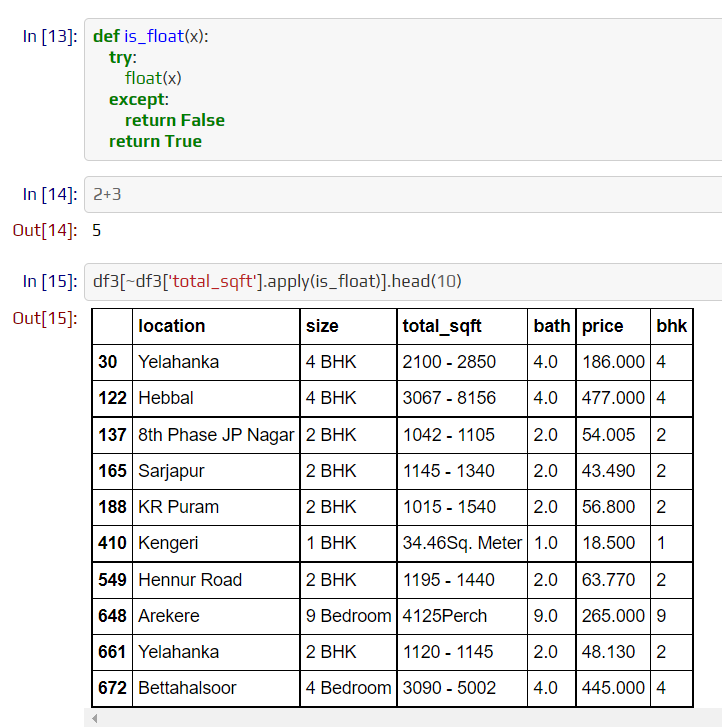
dropna() - dropna() returns Index with no NA/NaN values. All missing values are deleted, and a new object is returned that does not include any NaN values.

**5.1.5 Feature Engineering**

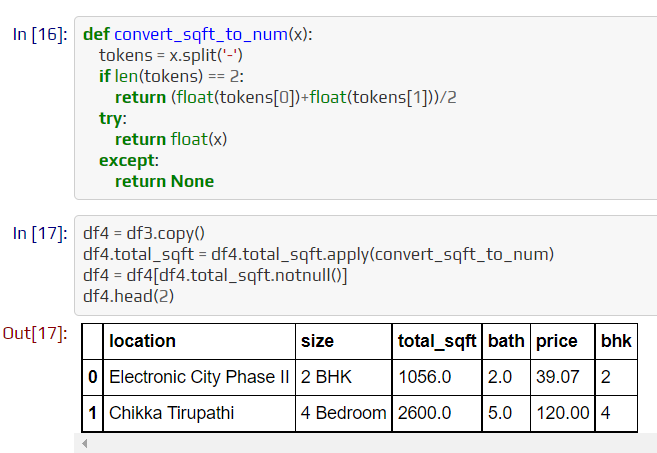
Feature engineering is the process of creating new features from existing data, which is frequently dispersed over many linked tables. Feature engineering is collecting important information from data and consolidating it into a single table that can subsequently be utilized to train a machine learning model.



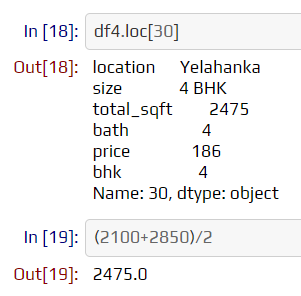
Explore total sqft features.



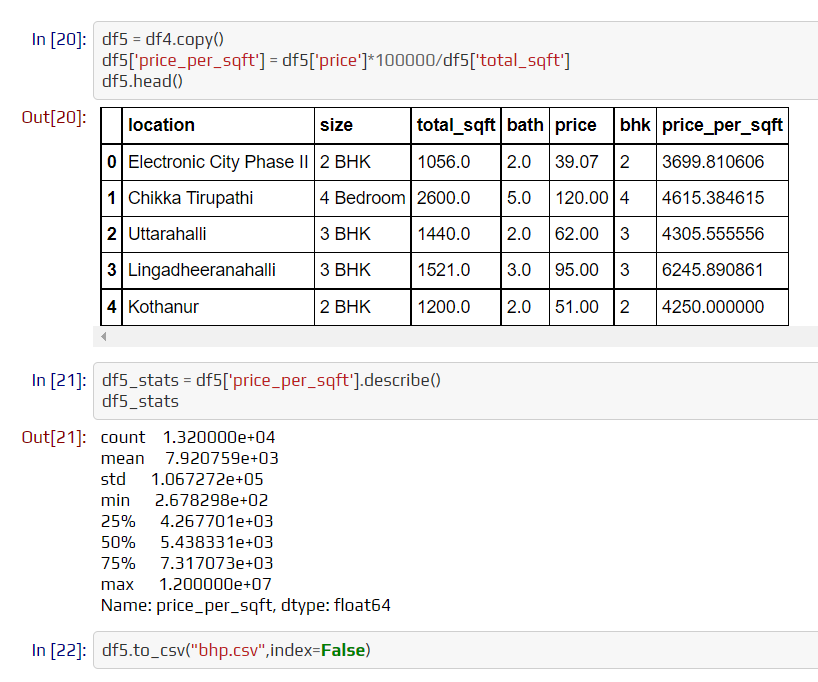
The above shows that total\_sqft can be a range (e.g. 2100-2850). For such a case, we can just take an average of min and max value in the range. There are other cases such as 34.46Sq. The meter which one can convert to square ft using unit conversion. I am going to just drop such corner cases to keep things simple.



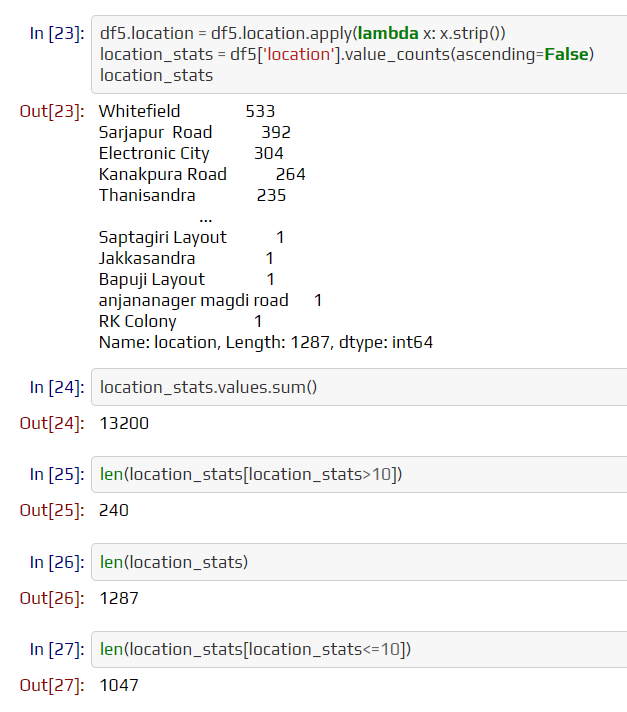
For the below row, it shows total\_sqft as 2475 which is an average of the range 2100-2850.



Add a new feature called price per square feet.



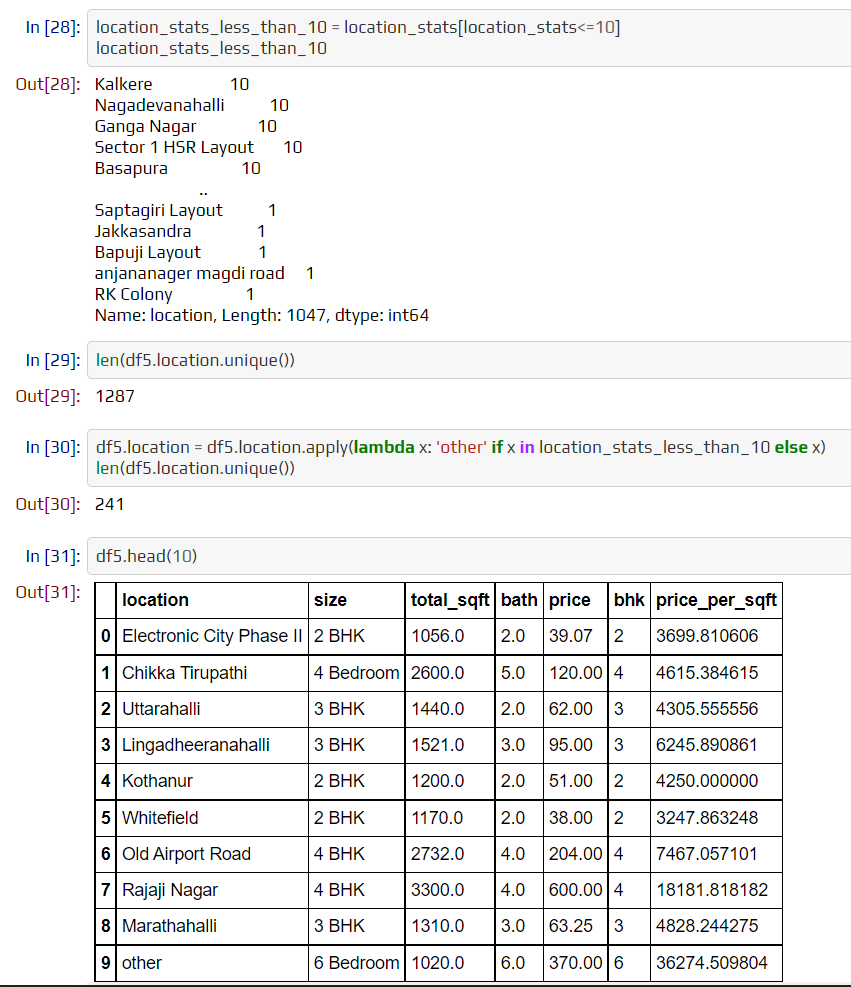
Examine locations that are categorical variables. We need to apply the dimensionality reduction technique here to reduce the number of locations.



**5.1.6 Dimensionality Reductions**

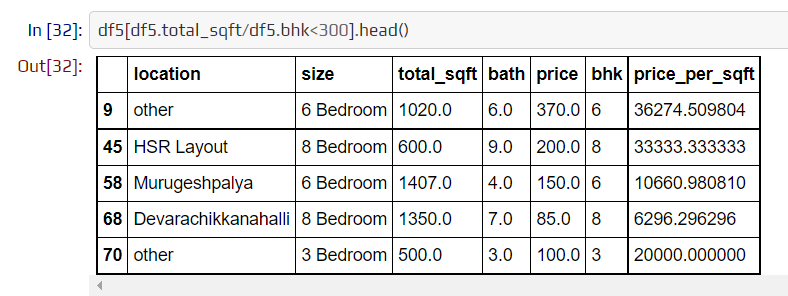
Dimensionality reduction methods are used to reduce the number of input variables in training data. When working with high-dimensional data, it is frequently beneficial to reduce the dimensionality by projecting the data to a lower-dimensional subspace that captures the data's "essence."

Any location that has less than 10 data points should be tagged as another location. This way the number of categories can be reduced by a huge amount. Later on, when we do one-hot encoding, it will help us with having fewer dummy columns.

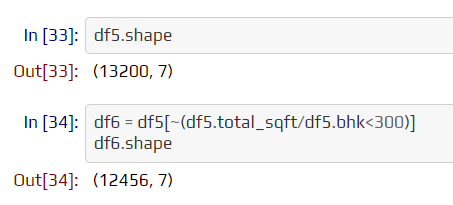


**5.1.7 Outlier Removal using Business Logic**

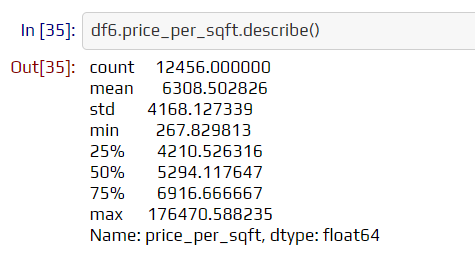
As a data scientist when you have a conversation with your business manager (who has expertise in real estate), he will tell you that normally square ft per bedroom is 300 (i.e. 2 bhk apartment is minimum 600 sqft. If you have for example 400 sqft apartment with 2 bhk then that seems suspicious and can be removed as an outlier. We will remove such outliers by keeping our minimum threshold per bhk to be 300 sqft.

****

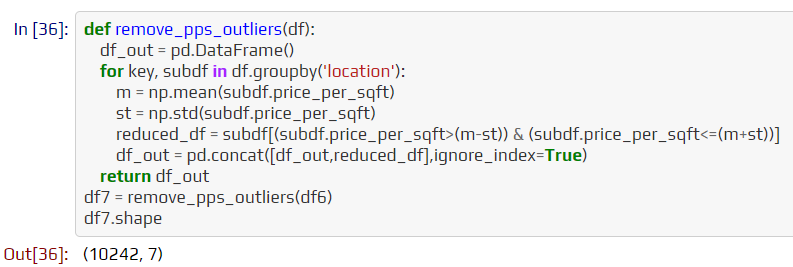
Check the above data points. We have 6 bhk apartments with 1020 sqft. Another one is 8 bhk and the total sqft is 600. These are clear data errors that can be removed safely.

****

**5.1.8 Outlier Removal using Standard Deviation & Mean**

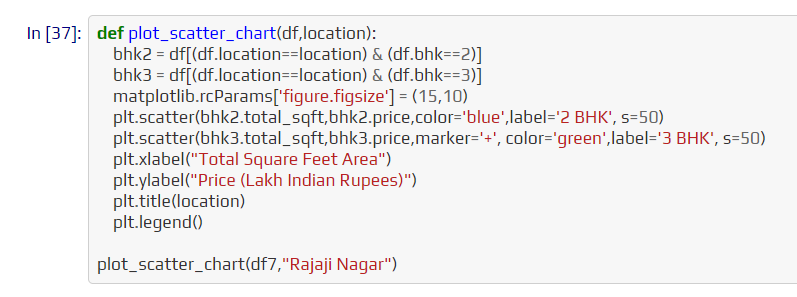
****

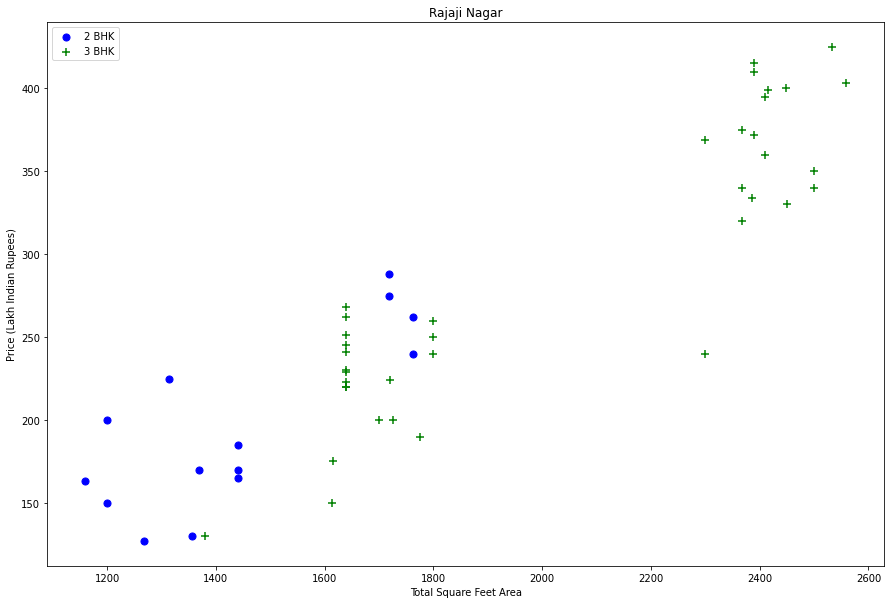
Here we find that the min price per sqft is 267 rs/sqft whereas the max is 12000000, which shows a wide variation in property prices. We should remove outliers per location using mean and one standard deviation.

****

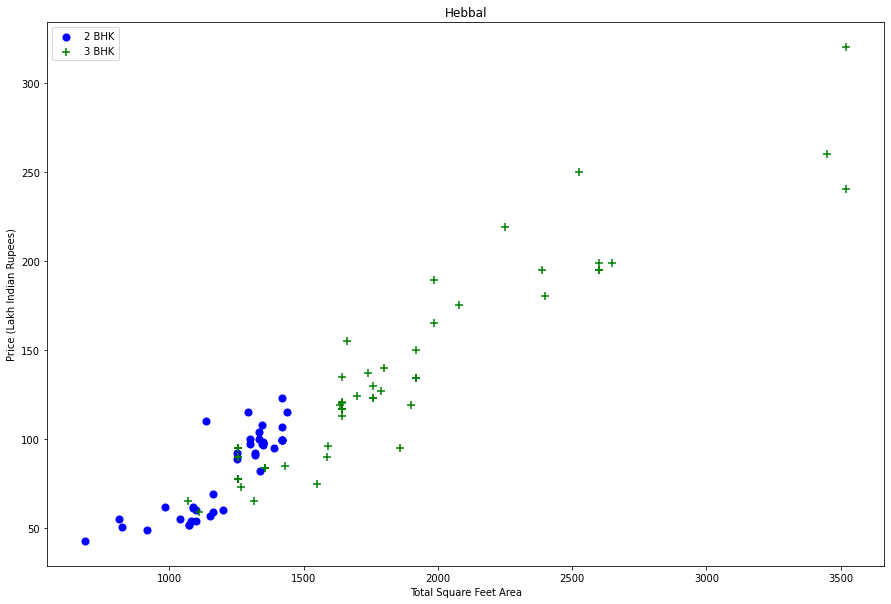
**5.1.9 Data Visualization**

Let's check for a given location how the 2 BHK and 3 BHK property prices look like.

****

****

****

****

We should also remove properties where for the same location, the price of (for example) a 3 bedroom apartment is less than a 2 bedroom apartment (with the same square ft area). What we will do is for a given location, we will build a dictionary of stats per bhk, i.e.

{

'1' : {

'mean': 4000,

'std: 2000,

'count': 34

},

'2' : {

'mean': 4300,

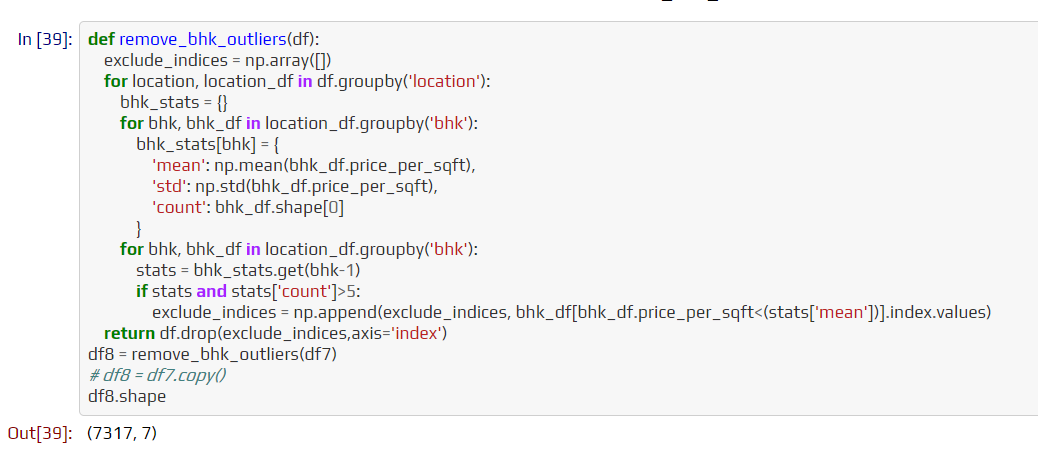
'std: 2300,

'count': 22

},

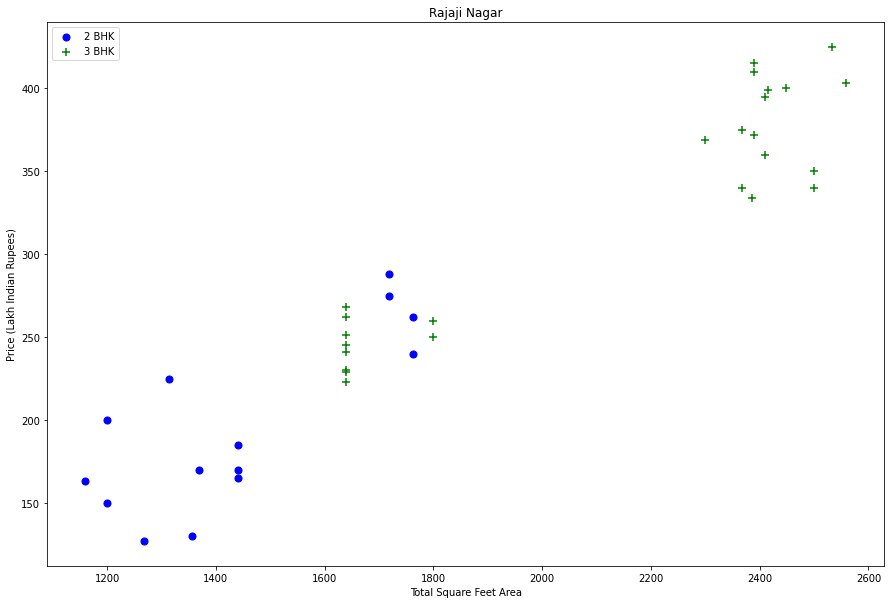
}

Now we can remove those 2 BHK apartments whose price\_per\_sqft is less than the mean price\_per\_sqft of 1 BHK apartment.

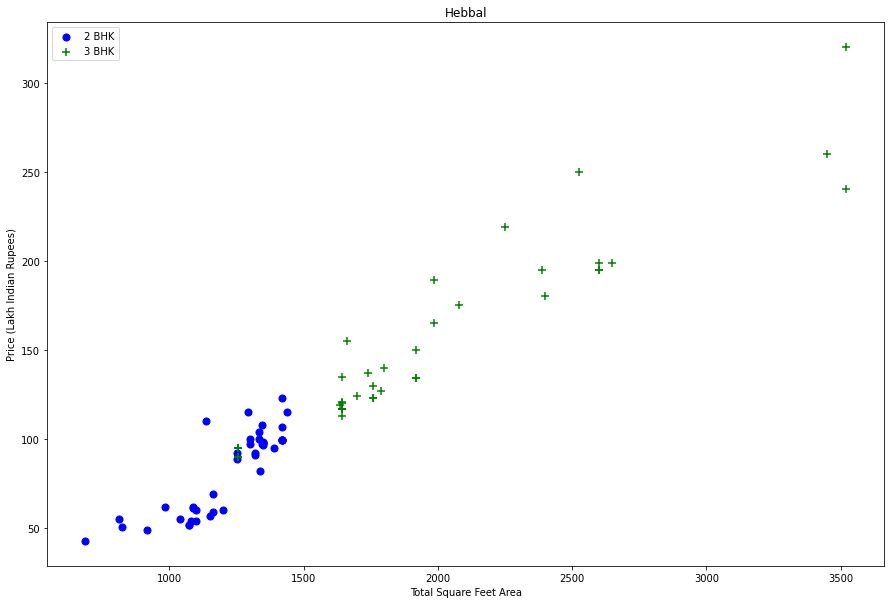
****

Plot the same scatter chart again to visualize price\_per\_sqft for 2 BHK and 3 BHK properties.

****

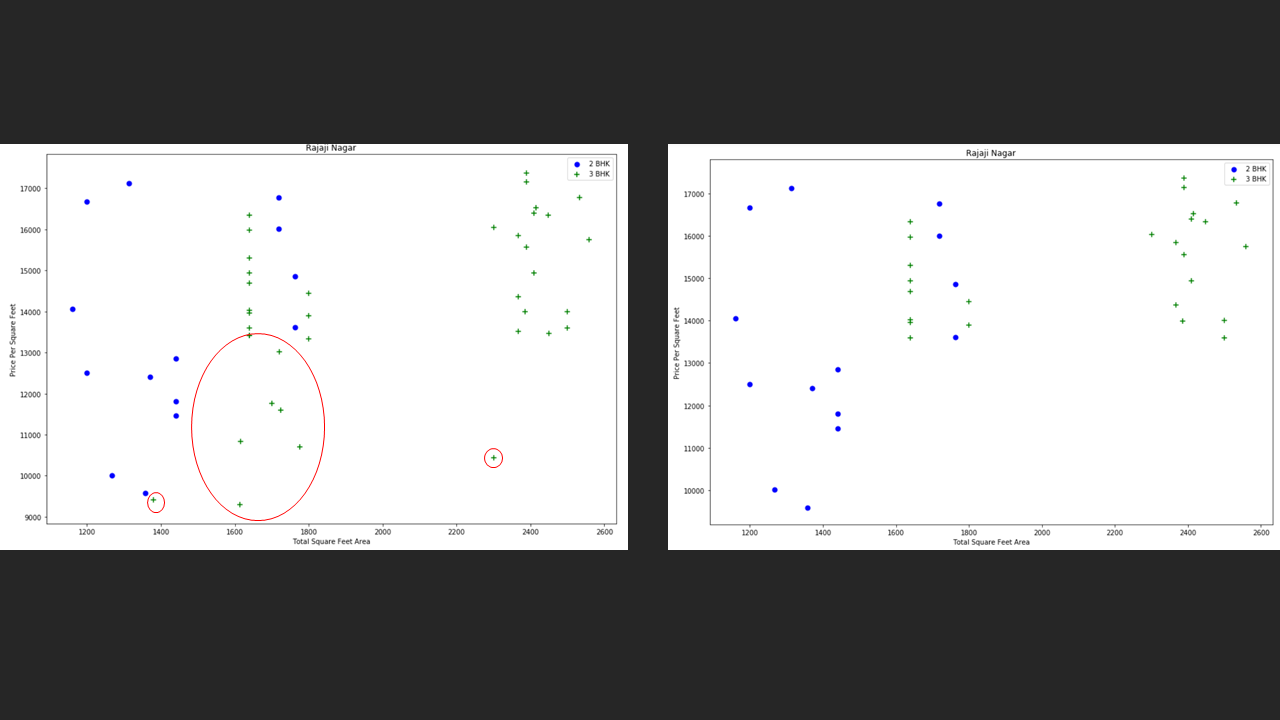
****

****

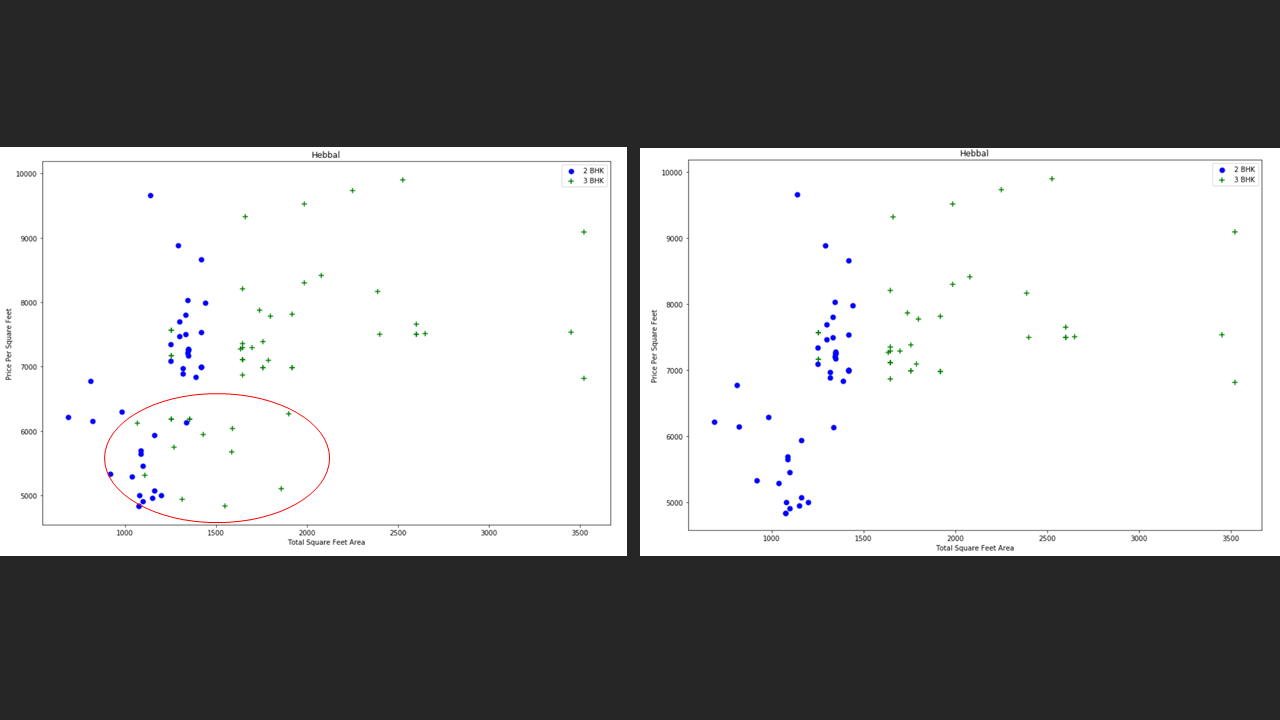
****

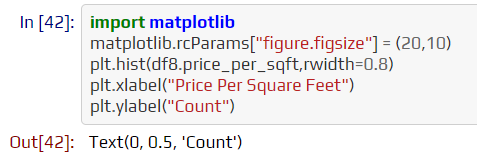
Based on the above charts we can see that data points highlighted in red below are outliers and they are being removed due to the remove\_bhk\_outliers function.

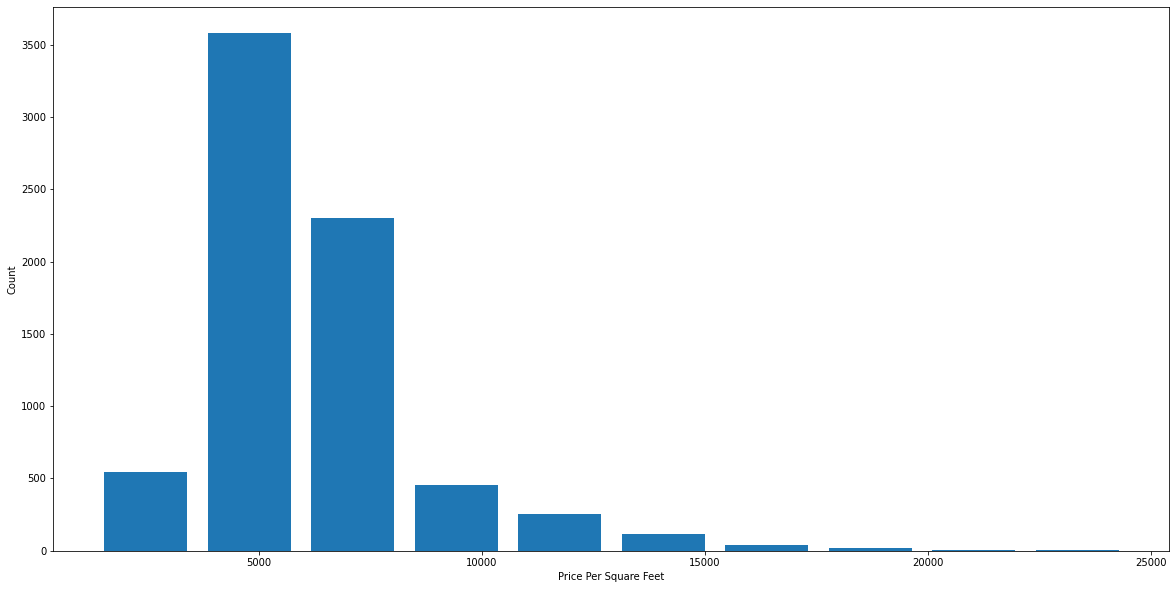
Before and after outlier removal: Rajaji Nagar

****

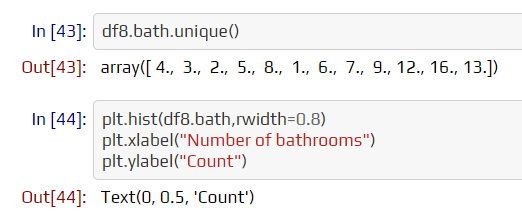
Before and after outlier removal: Hebbal

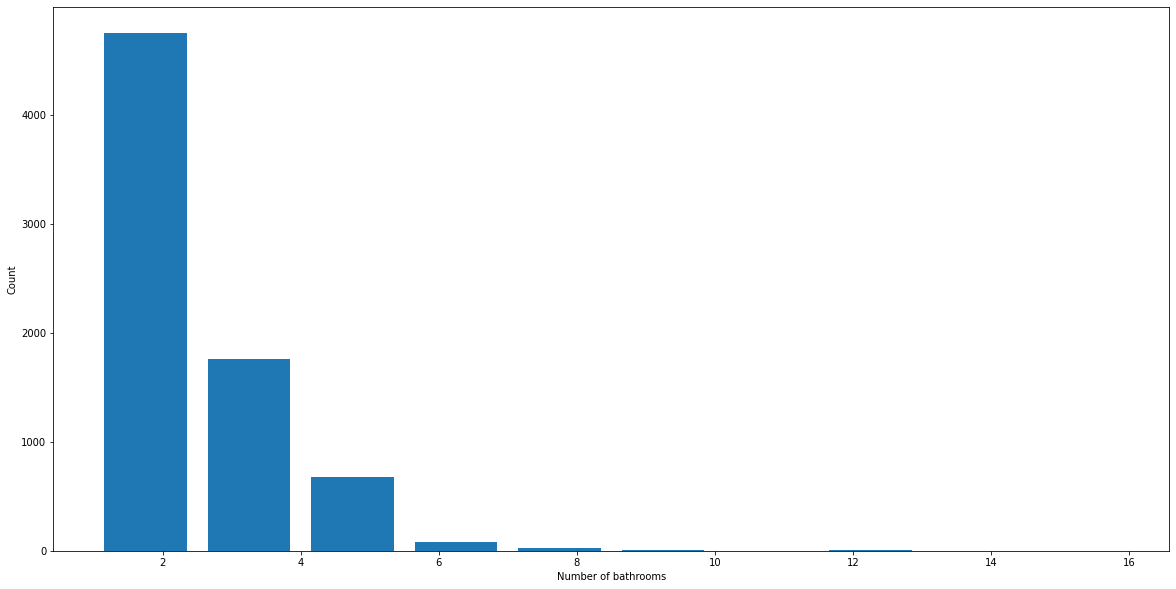


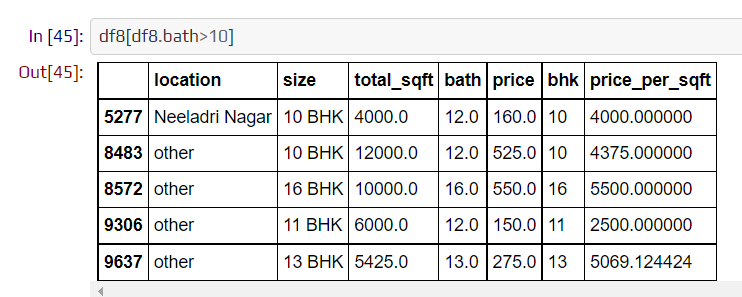




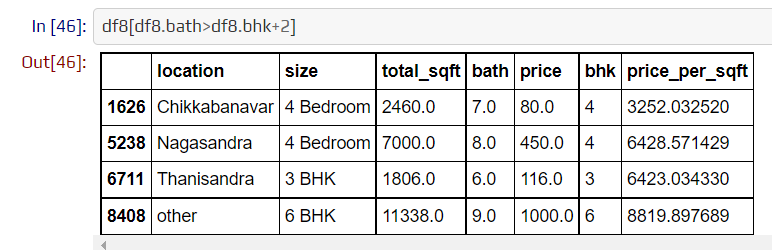
Outlier Removal Using Bathrooms Feature.



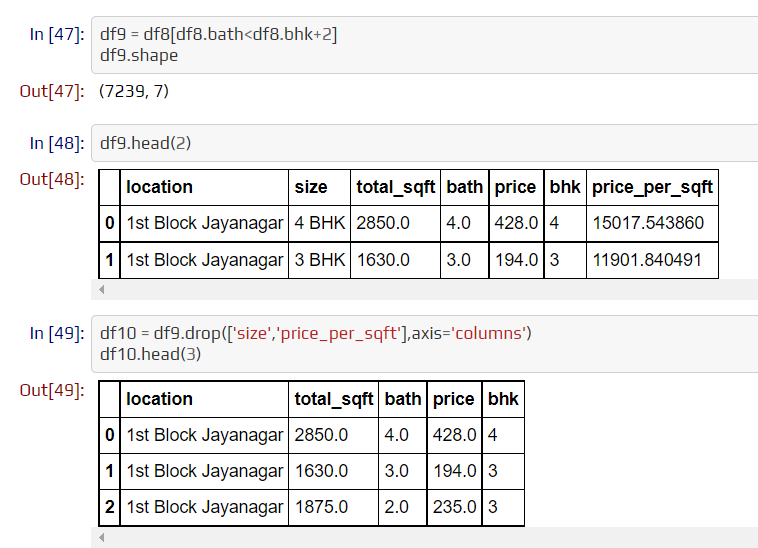




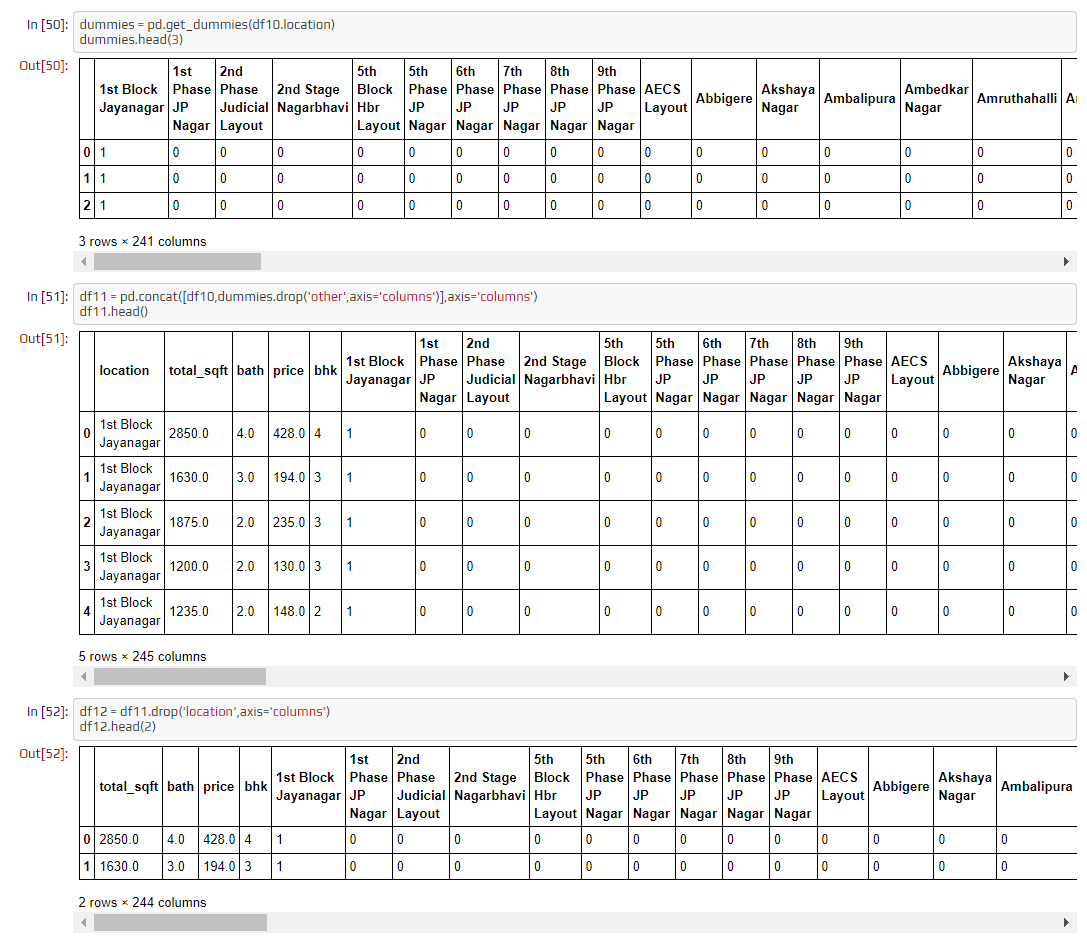
It is unusual to have 2 more bathrooms than several bedrooms in a home.



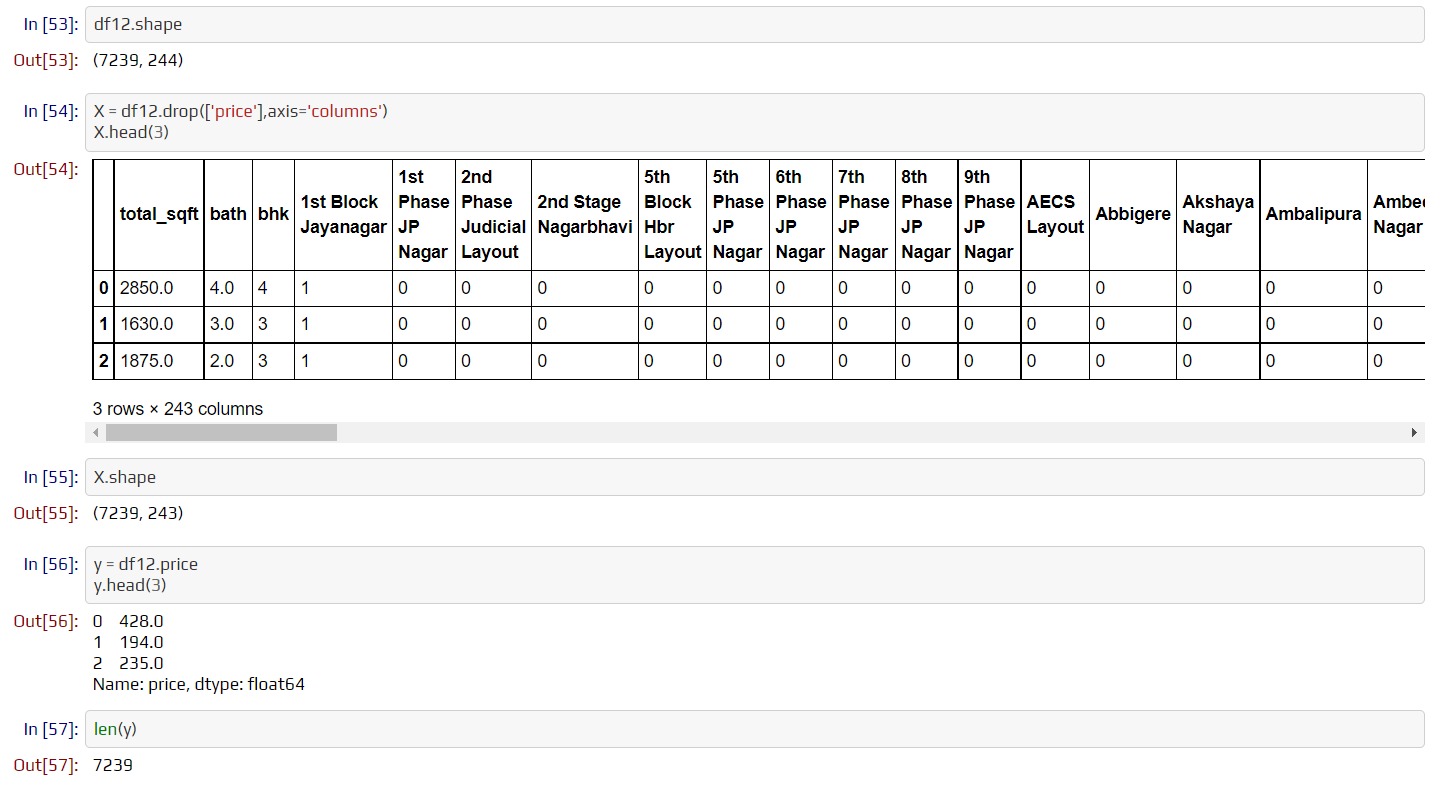
Again the business manager has a conversation with you (i.e. a data scientist) that if you have a 4 bedroom home and even if you have a bathroom in all 4 rooms plus one guest bathroom, you will have a total bath = total bed + 1 max. Anything above that is an outlier or a data error and can be removed.

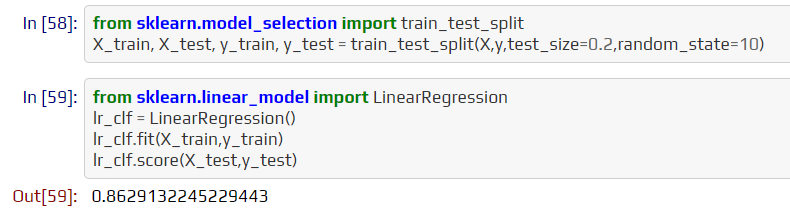


Use One Hot Encoding For Location.

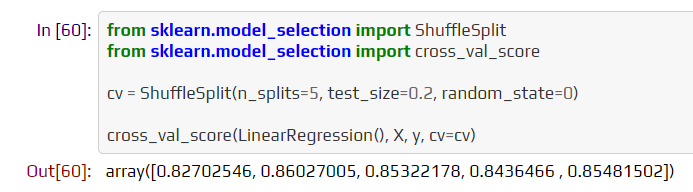


**4.1.10 Building a Model**

****

****

Use K Fold cross-validation to measure the accuracy of our LinearRegression model.

****

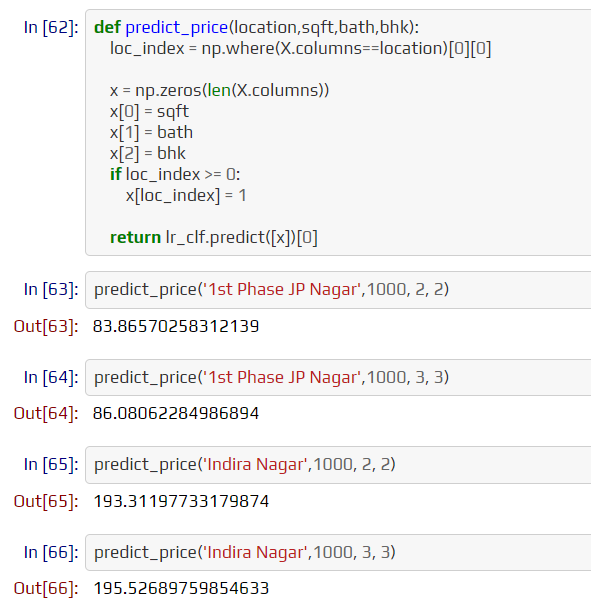
We can see that in 5 iterations we get a score above 80% all the time. This is pretty good but we want to test a few other algorithms for regression to see if we can get an even better score. We will use GridSearchCV for this purpose.

Find the best model using GridSearchCV.

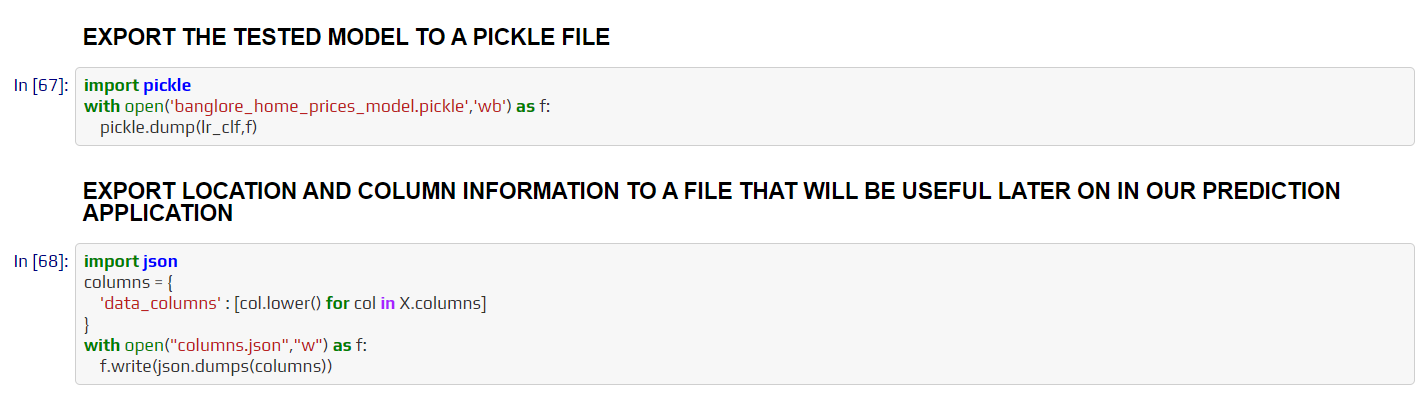


Based on the above results we can say that LinearRegression gives the best score. Hence, we will use that.

**5.1.11 Test the Model for few properties**

****

**5.1.12 Export the tested model to a pickle file**

****

**DEPLOYMENT**

Heroku is a cloud Platform as a container-based Service (PaaS). Heroku is used by developers to launch, manage, and grow contemporary programs.

**6.1 Steps to deploy the model on Heroku**

heroku login

git init

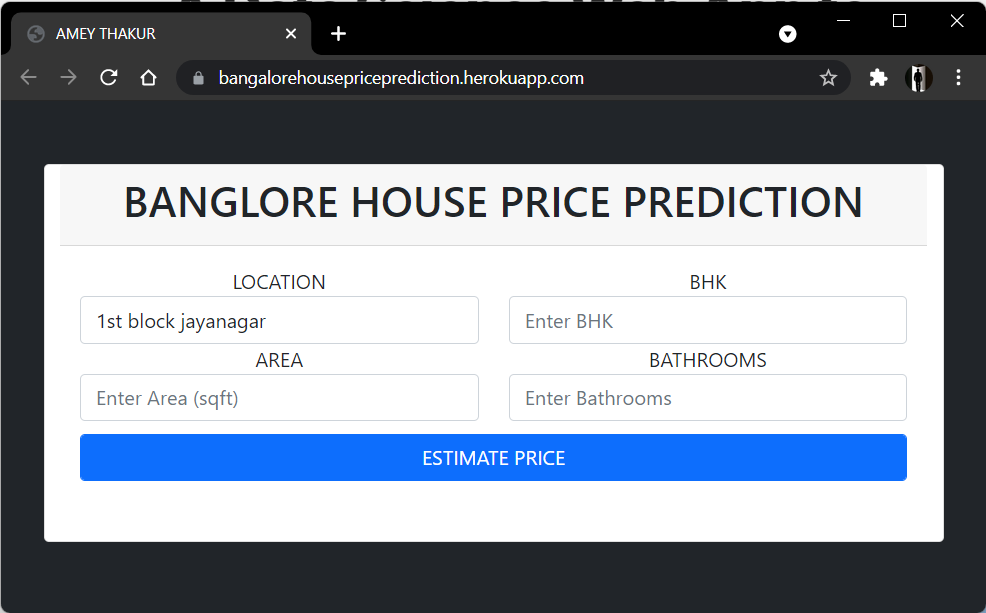
heroku create bangalorehousepriceprediction

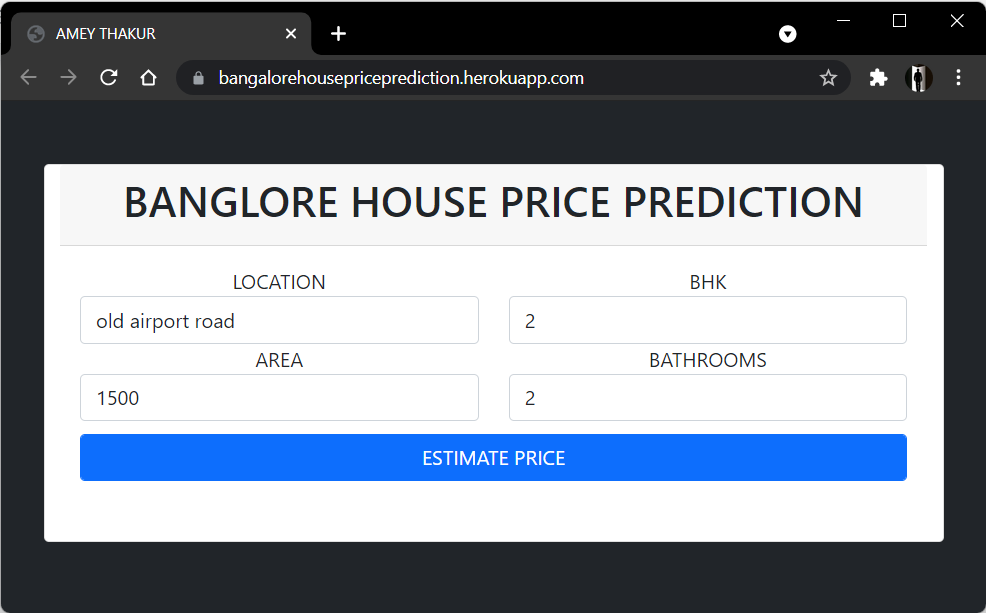
git add .

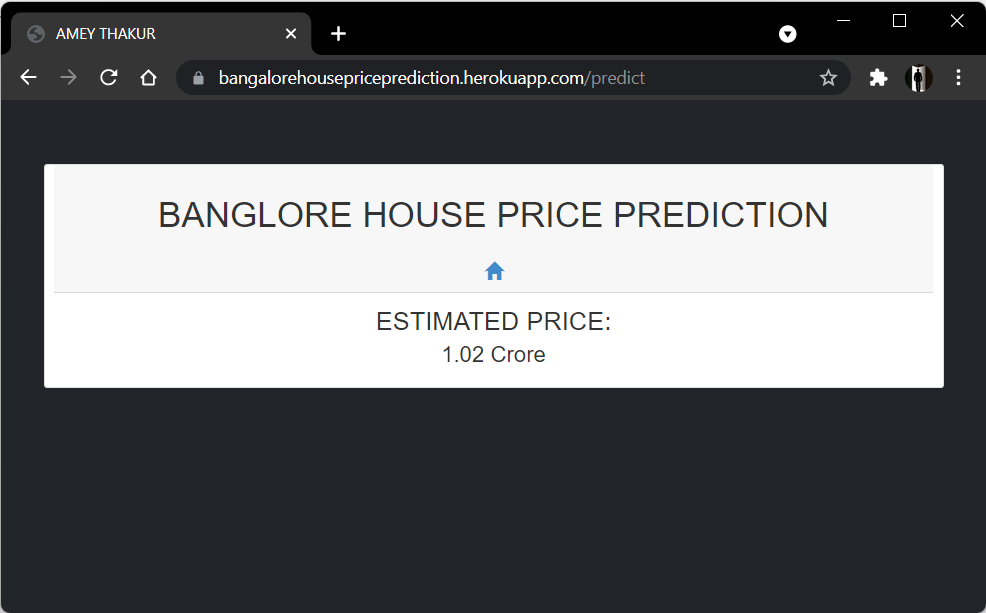
git commit -am "initial commit"

git push heroku master

**RESULTS**







**CONCLUSION**

With several characteristics, the suggested method predicts the property price in Bangalore. We experimented with different Machine Learning algorithms to get the best model. When compared to all other algorithms, the Decision Tree Algorithm achieved the lowest loss and the greatest R-squared. Flask was used to create the website.

Let's see how our project pans out. Open the HTML web page we generated and run the app.py file in the backend. Input the property's square footage, the number of bedrooms, the number of bathrooms, and the location, then click 'ESTIMATE PRICE.' Yay! We forecasted the cost of what may be someone's ideal home.