

HOUSE PRICE PREDICTION USING MACHINE LEARNING

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

MASINENI SARATH KUMAR (Reg. No. 37110440)

MARELLA LIKITH (Reg. No. 37110437)



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI
SALAI, CHENNAI – 600 119**

MARCH - 2021



SATHYABAMA



INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **MASINENI SARATH KUMAR (Reg. No. 37110440)** and **MARELLA LIKITH (Reg. No.37110437)** who carried out the project entitled "**HOUSE PRICE PREDICTION USING MACHINE LEARNING**" under my supervision from **August 2020** to **March 2021**.

Internal Guide

Ms. Ramya.G. Franklin, M.E.,

Head of the Department

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I **Masineni Sarath Kumar** hereby declare that the Project Report entitled “**HOUSE PRICE PREDICTION USING MACHINE LEARNING**” is done by me under the guidance of **Ms. Ramya.G. Franklin, M.E.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. Ramya.G. Franklin, M.E., Assistant Professor**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Predictive models for determining the sale value of houses in cities like Bengaluru continues to be remaining as additional challenging and tough task. The sale value of properties in cities like Bengaluru depends on variety of mutualist factors. Key factors that may have an effect on the value embrace space of the property, location of the property and its amenities. In this research work, Associate in Nursing analytical study has been applied by considering the info set that is still receptive the general public by illustrating the on the market housing properties in machine hackathon platform. the info set has 9 options. during this study, an attempt has been created to construct a prophetical model for evaluating the value supported the factors that have an effect on the value. Modeling explorations apply some regression techniques like multiple rectilinear regression (Least Squares), Lasso and Ridge regression models, support vector regression, and boosting algorithms like Extreme Gradient Boost Regression (XG Boost). Such models are wont to build a prophetical model, and to pick the most effective playing model by playing a comparative analysis on the prophetical errors obtained between these models. Here, the try is to construct a prophetical model for evaluating the value supported factors that affects the value.

TABLE OF CONTENTS

ABSTRACT	v
LIST OF FIGURES	viii

CHAPTER No	TITLE	PAGE No
1.	INTRODUCTION	1
	1.1 INTRODUCTION	1
	1.2 OUTLINE OF PROJECT	2
	1.3 AIM AND PURPOSE	3
2	AIM AND SCOPE	6
	2.1 AIM OF PROJECT	6
	2.1.1 OBJECTIVES	6
	2.2 SCOPE	7
	2.3 ADVANTAGES	9
3	SYSTEM DESIGN & METHODOLOGY	11
	3.1 EXISTING SYSTEM	11
	3.2 PREVIOUS RELATED WORK	12
	3.3 DATA UNDERSTANDING AND PRE-PROCESSING	13
	3.3.1 Data Description	13
	3.3.2 DATA UNDERSTANDING AND BASIC EDA	14
	3.3.3 Data pre-processing	16

	3.4 REGRESSION MODELS	
	AND EVALUATION METRICS USED	17
	3.4.1 BASIC LINEAR MODEL	18
	3.5 python packages	24
4	SOFTWARE DEVELOPMENT METHODOLOGY	28
	4.1 DESCRIPTION OF DIAGRAM	28
	4.2 ARCHITECTURAL DESIGN	31
	4.3 ANALYZE DATA	32
5	RESULTS AND DISCUSSION	35
6	CONCLUSION AND FUTURE WORK	40
	6.1 CONCLUSION	40
	6.2 FUTURE WORK	40
	APPENDIX	41
	A. SOURCE CODE	41
	REFERENCES	51

LIST OF FIGURES

FIGURE No	FIGURE NAME	PAGE No
3.1	Distribution of transformed price	15
3.2	Distribution of price in train data set	14
3.3	price per square feet vs total square feet	16
3.4	Linear regression example	27
4.1	block diagram	28
4.2	System Architecture	31
5.1	price per square feet vs total square feet	35
5.2	count vs price square feet	36
5.3	count vs Number of bathrooms	37
5.4	predictions	37
5.5	House price prediction dashboard	38

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Machine learning is a subfield of artificial intelligence (AI). The objective of AI by and large is to comprehend the construction of information and fit that information into models that can be perceived and used by individuals. In spite of the fact that AI is a field inside software engineering, it contrasts from conventional computational methodologies. In conventional registering, calculations are sets of unequivocally customized guidelines utilized by PCs to figure or issue settle. AI calculations rather take into account PCs to prepare on information data sources and utilize measurable examination to yield esteems that fall inside a particular reach. Along these lines, AI encourages PCs in building models from test information to robotize dynamic cycles dependent on information inputs. Demonstrating utilizes AI calculations, where machine gains from the information and utilizations them to anticipate another information. The most every now and again utilized model for prescient investigation is relapse. AI in software engineering endeavors to tackle issues algorithmically as opposed to absolutely numerically. Hence, it depends on making calculations that grant the machine to learn. In any case, there are two general gatherings in AI which are regulated and solo. Administered is the place where the program gets prepared on pre-decided set to have the option to foresee when another information is given. Solo is the place where the program attempts to discover the relationship and the secret example between the information. A near report was completed with assessment measurements also. When we get a solid match, we can utilize the model to figure financial estimation of that specific lodging property in Bengaluru. The exhibition will be estimated after foreseeing house costs since the forecast in numerous relapse calculations depends on a particular element as well as on an obscure number of properties that bring about the worth to be anticipated. House costs rely upon an individual house detail. Houses have a

variation number of highlights that might not have a similar expense because of its area. For example, a major house may have a greater cost on the off chance that it is situated in attractive rich territory than being put in a helpless area. The

information utilized in the trial will be dealt with by utilizing a blend of pre-preparing strategies to improve the forecast precision. Be that as it may, there are two general gatherings in AI which are administered and unaided. Regulated is the place where the program gets prepared on pre-decided set to have the option to anticipate when another information is given. Unaided is the place where the program attempts to discover the relationship and the secret example between the information.

1.2 OUTLINE OF PROJECT

As we probably are aware, the proposed model for precisely foreseeing future results has applications in financial aspects, business, banking area, medical services industry, web based business, amusement, sports and so on. One such technique used to conjecture house costs depend on numerous elements. In metropolitan urban communities like Bengaluru, the imminent home purchaser considers a few factors like area, size of the land, nearness to parks, schools, emergency clinics, power age offices, and in particular the house cost. Numerous straight relapse is one of the factual procedures for surveying the connection between the (reliant) target variable and a few autonomous factors. Relapse procedures are generally used to construct a model dependent on a few variables to foresee cost. In this investigation, we have made an endeavor to fabricate house value expectation relapse model for informational collection that stays available to people in general in Machine hackathon stage.

We have considered five forecast models, they are standard least squares model, Lasso and Ridge relapse models, SVR model, and XGBoost relapse model and Linear relapse. A relative report was completed with assessment measurements too. When we get a solid match, we can utilize the model to conjecture money related estimation of that specific lodging property in Bengaluru. The presentation will be estimated after anticipating house costs since the expectation in numerous relapse calculations depends on a particular element as well as on an obscure number of traits that bring about the worth to be anticipated. House costs rely upon an individual house detail. Houses have a variation number of highlights that

might not have a similar expense because of its area. For example, a major house may have a more exorbitant cost in the event that it is situated in alluring rich territory

than being set in a helpless area. The information utilized in the examination will be taken care of by utilizing a blend of pre-handling strategies to improve the forecast exactness. This examination plans to investigate the precision of foreseeing house costs when utilizing Multiple direct, Lasso, Ridge, and straight relapse. Hence, the reason for this examination is to extend the information in relapse techniques in AI. AI has been utilized for quite a long time to bring to the table picture acknowledgment, spam location, regular discourse appreciation, item proposals, and clinical judgments. Today, AI calculations can help us upgrade online protection, guarantee public security, and improve clinical results. AI frameworks can likewise make client assistance better and vehicles more secure. One such technique used to figure house costs depend on different variables. In metropolitan urban areas like Bengaluru, the planned home purchaser considers a few factors like area, size of the land, closeness to parks, schools, clinics, power age offices, and in particular the house cost. Various straight relapse is one of the factual methods for evaluating the connection between the (needy) target variable and a few free factors.

1.3 AIM AND PURPOSE

The given datasets ought to be handled to upgrade execution, which is cultivated by distinguishing the essential highlights by applying one of the determination techniques to wipe out the undesirable factors since each house has its exceptional highlights that help to appraise its cost. These highlights might be imparted to all houses, which implies they don't have a similar effect on the house evaluating bringing about mistaken yield. Various Linear Regression (MLR) is a directed strategy used to appraise the connection between one ward variable and more than one free factors. Recognizing the connection and its motivation impact assists with making expectations by utilizing these relations. To assess these connections, the expectation exactness of the model is fundamental; the intricacy of the model is of more interest. The solicitation contains a rundown of highlights, that coordinates with the public dataset's highlights, that is wanted to be accessible when the information is sent. There is no assurance that the information will be accessible in time nor contains the specific mentioned

rundown of highlights. Subsequently, there may be a danger that the entrance will be denied or postponed. Assuming this is the case, the examination will be cultivated dependent on the public dataset.

Additionally, this examination won't cover all relapse calculations; all things considered, it is centered around the picked calculation, beginning from the fundamental relapse strategies to the high level ones.

One such strategy used to estimate house costs depend on different components. Set forth plainly, relapse is an AI device that assists you with making expectations by taking in – from the current measurable information – the connections between your objective boundary and a bunch of different boundaries. As per this definition, a house's cost relies upon boundaries like the quantity of rooms, living region, area, and so forth On the off chance that we apply fake figuring out how to these boundaries we can ascertain house valuations in a given topographical region.

- The thought of relapse is quite straightforward: given sufficient information, you can notice the connection between your objective boundary (the yield) and different boundaries (the information), and afterward apply this relationship capacity to genuine noticed information.
- To show you how relapse calculation functions we'll consider just a single boundary – a home's living zone – to anticipate cost.
- It makes the entire cycle proficient.
- It's coherent to assume that there is a direct connection among territory and cost.

Relapse takes each conceivable incentive for k_0 and k_1 and limits the absolute deviation this is the possibility of relapse more or less. In any case, in actuality, there are different difficulties you need to manage. House costs clearly rely upon various boundaries, and there is no reasonable straight connection between these boundaries. the connections between your objective boundary and a bunch of different boundaries. As per this definition, a house's cost relies upon boundaries like the quantity of rooms, living territory, area, and so forth In the event that we apply fake figuring out how to these boundaries we can ascertain house valuations in a given geological region. The initial step for any sort of AI investigation is gathering the information – which should be substantial. In the

event that you can't ensure the legitimacy of your information, there's no point examining it. You need to focus on the source you take your information from. For my motivations, I've

depended on an information base from one of the biggest land gateways in the Netherlands. Since the housing market in the Netherlands is carefully managed, I didn't need to check its legitimacy.

At first I needed to do some information mining in light of the fact that the necessary information were accessible in different configurations across various sources. In most true activities you'll need to do some information mining too. We will not talk about information mining here, in any case, as it's not actually applicable to our point relapse is an AI instrument that assists you with making expectations by learning.

CHAPTER 2

AIM AND SCOPE

2.1 AIM OF PROJECT

The primary aim of project is to show you how regression algorithm works we'll take the essential point of undertaking is to show you how relapse calculation functions we'll consider just a single boundary – a home's living zone – to foresee cost. It's consistent to assume that there is a direct connection among zone and cost. Set forth plainly, relapse is an AI apparatus that assists you with making expectations by taking in – from the current factual information – the connections between your objective boundary and a bunch of different boundaries. As per this definition, a house's cost relies upon boundaries like the quantity of rooms, living region, area, and so on In the event that we apply fake figuring out how to these boundaries we can compute house valuations in a given geological territory

2.1.1 OBJECTIVES

The purposed model "house value prediction using machine learning" is to apply information pre-handling and readiness methods to acquire clean information to construct AI models ready to foresee house cost dependent on house highlights to investigate and contrast models execution all together with pick the best model. we will create and assess the presentation and the prescient force of a model prepared and tried on information gathered from houses in Boston's rural areas. helpful data of the general public can be recovered from a wide assortment of sources and put away as organized, unstructured and semi-organized arrangements. In the investigation of monetary wonders or social perceptions, progression of creative innovation makes it conceivable to deliberately separate the pertinent data, change them into complex information organizations and constructions, and afterward perform reasonable examinations. On account of these new conditions, customary information preparing and insightful apparatuses will be unable to catch, measure and break down profoundly complex data in the social and financial universes. When we get a solid match, we will utilize this

model to foresee the money related

estimation of a house situated at the Boston's territory. A model like this would be truly significant for a realtor who could utilize the data gave in an everyday schedule.

2.2 SCOPE

Regression is a machine learning apparatus that assists you with making forecasts by taking in – from the current measurable information – the connections between your objective boundary and a bunch of different boundaries. As per this definition, a house's cost relies upon boundaries like the quantity of rooms, living zone, area, and so forth. On the off chance that we apply counterfeit figuring out how to these boundaries we can compute house valuations in a given topographical zone.

The possibility of regression is quite basic: given sufficient information, you can notice the connection between your objective boundary (the yield) and different boundaries (the information), and afterward apply this relationship capacity to genuine noticed data. In the contemporary advanced time, valuable data of the general public can be recovered from a wide assortment of sources and put away as organized, unstructured and semi-organized organizations. In the examination of monetary wonders or social perceptions, progression of creative innovation makes it conceivable to deliberately remove the applicable data, change them into complex information arrangements and constructions, and afterward perform appropriate investigations. In view of these new conditions, conventional information preparing and scientific apparatuses will be unable to catch, measure and dissect exceptionally complex data in the social and financial universes. New procedures have been created in light of the treatment of the epic measure of accessible information. AI is one of the front line strategies that can be utilized to recognize, decipher, and dissect enormously confounded information designs and examples. It permits noteworthy learning and improves model expectations with a methodical contribution of later information. Contemporary exploration on AI is a subset of man-made reasoning (AI) that endeavors to prepare PCs with new information through contribution of information, like writings, pictures, mathematical qualities, etc, and support its cooperation with other PC

organizations. As per Feggella, AI is about 'the study of getting PCs to learn and act like people do, and improve their learning after some time in self-governing design, by taking care of them information and data as perceptions and certifiable cooperations'.

AI can be comprehensively classified into three sorts, in particular administered learning, unaided learning and semi-directed learning. Generally, managed AI calculations expect to recognize a capacity that can give precise out-of-test conjectures. A model prepared through managed learning is supposed to be effective on the off chance that it can make forecasts inside a worthy degree of exactness. Instances of directed learning incorporate straight relapse and backing vector machine.

regression learning is a part of AI that handles unlabelled preparing information. These information don't have any sort of name that can uncover their grouping. When contrasted with administered learning, unaided learning techniques are applied to find the connections between components in a dataset without marks. On account of this nature, machine learning can find covered up structures inside information that may not be effortlessly noticed. To lay it out plainly, it is best embraced if information researchers need to reveal designs that might not have been distinguished hypothetically before the examination. The helpfulness of unaided learning is to show the fundamental construction or dissemination in the information with the end goal of studying the information. Unaided learning issues can be classified into bunching and affiliation (for example find decides that portray a huge segment of information) issues. Instances of solo learning calculations incorporate k-implies for bunching issues, neural organization for examining successions, and direct relapse calculation for affiliation rule learning issues.

Semi-managed learning alludes to the situation when there are a lot of info information (X) and just a portion of the information are marked (Y). For this situation, information researchers can utilize solo learning methods to find and find out about the information structure. Unexpectedly, information researchers can likewise receive regulated learning procedures to make forecasts for the unlabelled information, train the model with administered learning calculation, and make expectations on the test information. In reality, most information investigations fall into this class.

Against this foundation, the instrument of AI is radically not quite the same as regular econometric strategies in friendly and monetary investigations. In econometrics research, monetary models are developed to zero in on various boundaries that address the impacts of progress sought after or potentially supply factors. Scientists need to address for heterogeneity and endeavor to acquire hetero skedasticity- reliable standard mistakes and covariances for the assessed boundaries They likewise need to assess the power of the outcomes by assessing a few elective model determinations. Interestingly, AI moves toward generally put accentuation on the out-of-test expectations, and depend on the information driven model choice to reveal the highlights with the most logical force .These methodologies work with vulnerability related with deficient data and additionally estimation blunders, and mean to show vulnerability through the instruments and strategies from likelihood. Fusing highlights like the quantity of rooms, number of restrooms, home condition, fruition year, absolute duty evaluated esteem, etc, the examination thinks about the outcomes utilizing the strategies for direct relapse, choice tree, inclination boosting, arbitrary backwoods, and backing vector machine. Shockingly, every one of these strategies will in general disparage the Estimate expectation mistake. The reasons that Estimate has effectively given an overall precise lodging value forecast.

2.3 ADVANTAGES

Since lodging cost is unequivocally associated to different factors like area, region, populace, it requires other data separated from HPI to foresee singular lodging cost. There has been a significantly enormous number of papers receiving customary AI ways to deal with anticipate lodging costs precisely, yet they once in a while worry about the exhibition of individual models and disregard the less well known at this point complex models. Therefore, to investigate different effects of highlights on expectation techniques, this paper will apply both conventional and progressed AI ways to deal with research the distinction among a few progressed models. This paper will likewise thoroughly approve numerous methods in model execution on relapse and give an idealistic outcome to lodging value forecast.

- unlike traditional econometrics models, AI calculations don't need our preparation information to be typically circulated. Numerous measurable tests depend with the understanding of ordinariness.

- If the information are not typically circulated, those measurable tests will fall flat and become invalid.
- These measures used to take quite a while previously, yet they would now be able to be immediately finished with the high velocity computational force of present day PCs, and hence the utilization of this procedure today is less exorbitant or tedious.
- The assessment mistakes can be headed to the base level for both preparing and test informational indexes.
- To investigate different effects of highlights on forecast strategies, this paper will apply both conventional and progressed AI ways to deal with examine the distinction among a few progressed models.
- They once in a while worry about the presentation of individual models and disregard the less famous yet complex models.
- The value of solo learning is to demonstrate the hidden construction or appropriation in the information with the end goal of studying the information.
- high-speed computational force of current PCs, and along these lines the utilization of this method today is less exorbitant or tedious.

CHAPTER 3

SYSTEM DESIGN & METHODOLOGY

3.1 EXISTING SYSTEM

Model uses Machine learning calculations, where machine gains from the information and utilizations them to foresee another information. The most as often as possible utilized model for prescient examination is relapse. As we probably are aware, the proposed model for precisely anticipating future results has applications in financial matters, business, banking area, medical services industry, internet business, amusement, sports and so forth. One such strategy used to conjecture house costs depend on numerous variables. In metropolitan urban areas like Bengaluru, the planned home purchaser considers a few factors like area, size of the land, nearness to parks, schools, clinics, power age offices, and above all the house cost. Various straight relapse is one of the factual methods for surveying the connection between the (needy) target variable and a few autonomous factors. Relapse methods are broadly used to assemble a model dependent on a few components to foresee cost. In this investigation, we have made an endeavor to construct house value forecast relapse model for informational index that stays open to the general population in Machine hackathon stage. We have considered five forecast models, they are common least squares model, Lasso and Ridge relapse models, SVR model, and straight relapse model. A near report was done with assessment measurements too. When we get a solid match, we can utilize the model to gauge financial estimation of that specific lodging property in Bengaluru.

Prescient models for deciding the deal cost of houses in urban areas like Bengaluru is as yet staying as really testing and precarious errand. The deal cost of properties in urban areas like Bengaluru relies upon various associated factors. Key factors that may influence the cost incorporate region of the property, area of the property and its conveniences. In this exploration work, a scientific

investigation has been done by considering the informational collection that stays open to people in general by outlining the accessible lodging properties in machine hackathon stage. The

informational collection has nine highlights. In this investigation, an endeavor has been made to build a prescient model for assessing the cost dependent on the elements that influence the cost. Demonstrating investigations apply some relapse methods like numerous direct relapse (Least Squares), Lasso and Ridge relapse models, support vector relapse, and boosting calculations. Such models are utilized to fabricate a prescient model, and to pick the best performing model by playing out a similar investigation on the prescient mistakes acquired between these models. Here, the endeavor is to build a prescient model for assessing the cost dependent on factors that influences the cost.

3.2 PREVIOUS RELATED WORK

Pow, Nissan, Emil Janulewicz, and L. Liu utilized four relapse methods to be specific Linear Regression, Support Vector Machine, K-Nearest Neighbors (KNN) and Random Forest Regression and an outfit approach by joining KNN and Random Forest Technique for anticipating the property's value esteem. The troupe approach anticipated the costs with least blunder of 0.0985 and applying PCA didn't improve the forecast mistake. A few examinations have likewise centered around the assortment of highlights and extraction techniques. Wu, Jiao Yang has analyzed different element determination and highlight extraction calculations joined with Support Vector Regression. A few specialists have created neural organization models to foresee house costs. Limsombunchai, contrasted house evaluating structure and counterfeit neural organization model to anticipate the house costs. The R-Squared worth acquired for Neural Network model was more noteworthy when contrasted with indulgent model and the RMSE estimation of Neural Network model was moderately lower. Henceforth they inferred that Artificial Neural Network performs better when contrasted and Hedonic model. Cebula applies the epicurean value model to anticipate lodging costs in the City of Savannah, Georgia. The log cost of houses has been demonstrated to be emphatically and significantly connected with the quantity of washrooms, rooms, chimneys, carport spaces, stories and the house's all out square feet. thought about the valuing expectation between relapse examination and fake neural organization in anticipating condo's costs. The log cost of houses

has been demonstrated to be emphatically and significantly connected with the quantity of restrooms, rooms, chimneys, carport spaces, stories and the house's absolute square feet. Here, the endeavor is to

develop a prescient model for assessing the cost dependent on factors that influences the cost.

3.3 DATA UNDERSTANDING AND PRE-PROCESSING

Put simply, regression is a machine learning tool that helps you make predictions by learning – from the existing statistical data – the relationships between your target parameter and a set of other parameters. According to this definition, a house's price depends on parameters such as the number of bedrooms, living area, location, etc. If we apply artificial learning to these parameters we can calculate house valuations in a given geographical area.

3.3.1 Data Description

The informational collections are train set and test information considered in the venture is taken from kaggle stage. It comprises of highlights that depict house-property in Bengaluru. There are 9 highlights in both the informational collections. The highlights can be clarified as follows: Area type-describes the area

1. Availability-when it is possessing or when it is ready.
2. Price- Value of the property in lakhs.
3. Size- in BHK or Bedroom (1-10 or more)
4. Society- to which it belongs.
5. Total_sqft - size of the property in sqft.
6. Bath-No of bathrooms
7. Balcony- No of balcony
8. Location – where it is located in Bengaluru

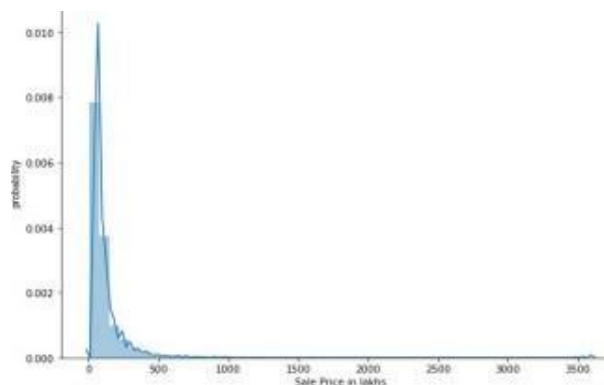
With 9 highlights accessible, we attempt to fabricate relapse models to anticipate house cost. We anticipated the cost of test informational collection with the relapse models based on train informational collection. This dataset incorporates many downright factors (both train and test informational collection) for which we should make faker factors or use mark encoding to change over into mathematical structure. These eventual phony/faker factors since they are placeholders for genuine variable and are made without help from anyone else. Additionally, there are a great deal of invalid qualities present also, so we should

treat them as needs be. The highlights shower, value, gallery are mathematical factors. Highlights like

zone type, total_sqft, area, society, accessibility, and size shows up as clear cut factors..

3.3.2 DATA UNDERSTANDING AND BASIC EDA

The object is to make a model that can appraise lodging costs. We partition the arrangement of information into capacities and target variable. In this segment, we will attempt to comprehend outline of unique informational index, with its unique highlights and afterward we will make an exploratory investigation of the informational index and endeavor to get helpful perceptions. The train informational index comprises of 11200 records with 9 logical factors. In test informational collection, there were around 1480 records with 9 factors. While building relapse models we are frequently needed to change over the unmitigated for example text highlights to its numeric portrayal. The two most regular approaches to do this is to utilize name encoder or one hot encoder. Mark encoding in python can be accomplished by utilizing sklearn library. Name encoder encodes names with a worth somewhere in the range of 0 and n1. On the off chance that a name rehashes, it ascribes a similar worth as recently allotted. One hot encoding alludes to parting the section that contains mathematical all out information to numerous segments relying upon the quantity



of classes present in that segment. Every segment contains "0" or "1" relating to which segment it has been set

Fig 3.1 Distribution of price in train data set

It very well may be seen that value's dissemination is profoundly slanted. The value goes from 8 lakhs to 3600 lakhs. The vast majority of the value lies under 500 lakhs. Kurtosis is a metric that shows whether the informational collection is hefty or light followed contrasted with a typical dissemination. It was seen that the skewness and kurtosis were around 8 and 108 separately. Since the cost has decidedly slanted dispersion; we utilized log change of cost for additional investigation. After log change is applied to value variable, the dispersion was as appeared

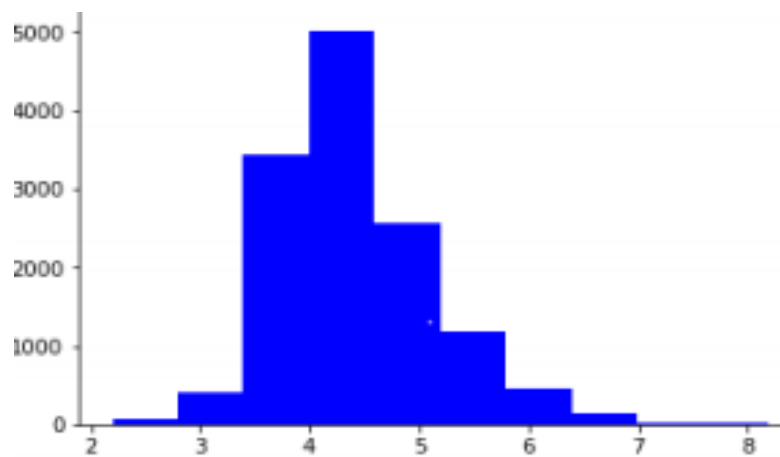


Fig 3.2 Distribution of transformed price

After applying log transformation to the price variable, we observe that kurtosis and skewness were reduced to 0.85628 and 1.34. We considered pairwise scatterplot that will allow us to visualize the pair-wise relationships and correlations between the different features as in Figure The scatter plot helps us see how dispersed the data points are. It is helpful to get a quick overview of how the data is being distributed and whether it includes outliers or not. Furthermore, we can say from the histogram that the price variable (to be noted we applied log transformation for the original price variable) appears to be distributed normally, but contains several outliers.

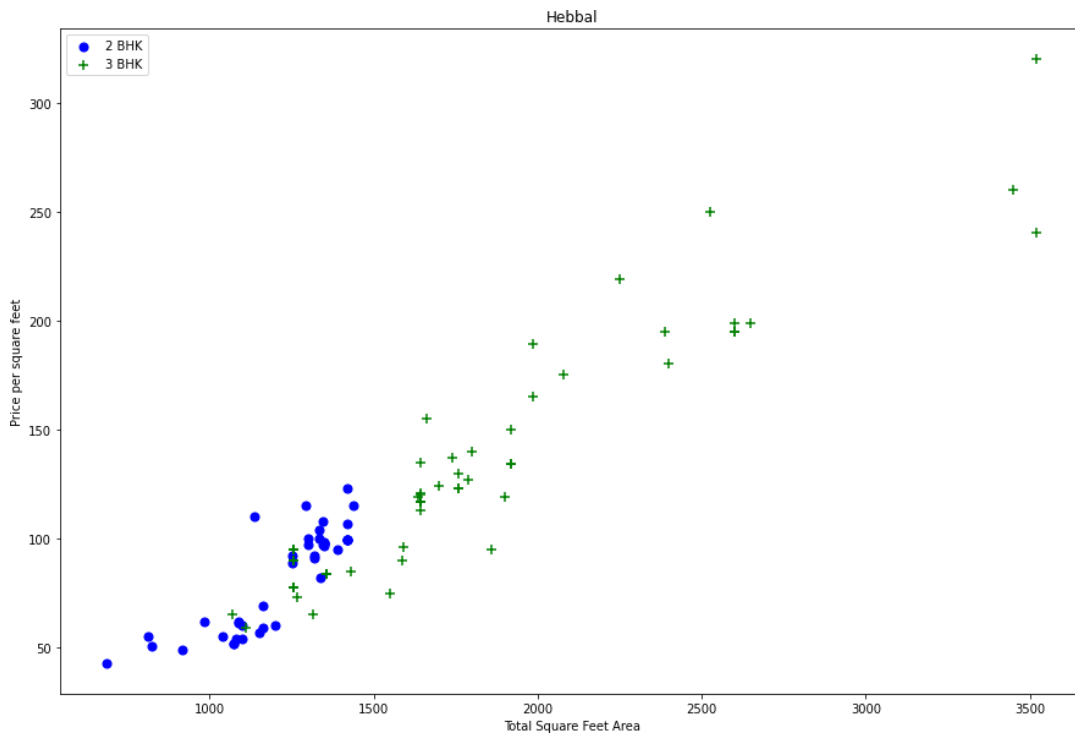


Fig 3.3 price per square feet vs total square feet

3.3.3 Data pre-processing

The overall strides in information pre-preparing are

1. Converting clear cut highlights into mathematical factors to fit direct relapse model.
2. Imputing invalid records with suitable qualities.
3. Scaling of information
4. Split into train – test sets.

The information pre-handling of each element in train and test informational indexes is summed up as beneath:

- Around 41% of society records are absent in the train informational index; around 57% of records are missing in test information. So the component society is dropped from both the informational indexes as it doesn't add a lot to the model.

- There are around 1305 unique areas. One information point area record is absent. We have credited the invalid record with 'others'. Since the component area is downright, we use you Label Encoder to change over all out into mathematical element.
- The invalid qualities present in gallery records around 609 information focuses were attributed with mode (most happening esteem) '2' where invalid qualities in test information which were around 69 have been credited with 2.
- The invalid qualities present in shower records have been attributed with mode (most happening esteem) '2 BHK' in the two sets
- We see that, all absolute sqft records are not in square-feet in both the informational collections. Some of them are in square-yards, sections of land, roost, guntha and grounds. Each information point regarding complete sqft has been changed over into square-feet via doing fundamental changes
- The area_type has four classes: Super developed zone, plot region, cover region and developed zone. We have changed over into sham factors in the two sets.
- The section size has records in BHK, room and RK. The mathematical part connected with BHK and Bedroom has been extricated and two separate highlights BHK and Bedroom have been made. The element size has been barred from information.
- We have gathered the accessibility records into two classes: Ready to Move and Others. Similarly, pre-handling steps in the information test set were performed.

Every one of these information pre-preparing steps have been done in Jupyter note pad python with essential bundles.

3.4 REGRESSION MODELS AND EVALUATION METRICS USED

Straight relapse is perhaps the most notable calculations in measurements and AI. The goal of a direct relapse model is to discover a connection between at least one highlights (free/illustrative/indicator factors) and a persistent objective

variable (subordinate/reaction) variable. On the off chance that there is just one component, the model is basic direct relapse and if there are numerous highlights, the model is different straight relapse.

3.4.1 BASIC LINEAR MODEL

The formulation for multiple regression model

$$\text{is } Y = a_0 + a_1X_1 + a_2X_2 + \dots + a_nX_n$$

The assumptions in the model are:

- The error terms are normally distributed.
- The error terms have constant variance.
- The model carries out a linear relationship between the target variable and the functions.

Here the different relapse models are created by the most un-square methodology (Ordinary Least Squares/OLS). The precision of the planned model is hard to gauge without assessing its yield on both train and test informational collections. This can be accomplished utilizing effectiveness metric or some likeness thereof. It could be by estimating some sort of blunder, fit's integrity, or some other valuable computation. For this investigation, we assessed model's presentation utilizing measurements: the coefficient of assurance, R^2 (Coefficient of Determination), R and RMSE (Root Means Square Error). (Root Means Square Error), RMLSE (Root Mean Squared Logarithmic Error)

RMSE: It can be characterized as the standard example deviation between the anticipated qualities and the noticed ones. It is to be noticed that unit of RMSE is same as reliant variable y . The lower RMSE esteems are characteristic of a superior fit model. On the off chance that the model's essential target is forecast, RMSE is a more grounded measure. 2. R-squared and Adjusted R-squared: The R-square worth gives a proportion of how much the model recreates the real outcomes, in light of the proportion of all out variety of results as clarified in the model.

The higher the R-squared, the better the model fits the information given. The R-squared worth reaches from 0 to 1, addressing the level of a squared connection between's the objective variable's normal and genuine qualities. In any case, if there should be an occurrence of various direct relapses, R-squared worth may increment with expanding highlights despite the fact that the model isn't really

improving. A connected, Adjusted R-squared measurement can be utilized to address this

detriment. This actions the model's decency and punishes the model to utilize more indicators.

RMSLE (Root Mean Squared Logarithmic Error): It is the root mean squared blunder of the logarithmic changed anticipated and log changed real qualities. The blunder term can be detonated to an exceptionally high worth if anomalies are available in the event of RMSE, though in the event of RMLSE the exceptions are radically downsized in this manner so that effect is nullified. We have part the preparation information into subsets of train and test information. We endeavor to assemble straight relapse model utilizing OLS in python utilizing fundamental bundles in particular sk-learn and scikit bundles after information preprocessing and train-test split of informational index. We realize that multi-collinearity is the impact of the various relapse models having related indicators. In straightforward, when informational collection has an enormous number of indicators, it is conceivable that couple of these indicators might be profoundly connected. Presence of high relationship between's free factors is called multi-collinearity. Presence of multi- collinearity can destabilize the model. Aside from connection framework, to check such a relations between factors we use, variety expansion factor (VIF). It estimates the extent of multi-collinearity. It is characterized as follows:

$$VIF = \frac{1}{1-R^2}$$

There are many approaches to handle multicollinearity. One simple technique is to dropping the variables from the model building that is highly correlated with others with the help of pstatistic and Variation Inflation factor. The threshold value for VIF is 5. VIF greater than 5 requires further investigation to assess the impact of multi- collinearity. Taking into account all the above metrics, we attempt to build basic regression model.

Metric	Train set	Test set
R-square	0.418	-2.12
RMSE	0.0912	0.2077
RMSLE	0.02755	0.03493

Fig 3.4 R-square and RMSE values for linear regression model

R-Squared incentive for model created is 0.418 and changed R-squared worth is 0.418. The p esteems and VIF regarding highlights were in allowable reach. We have negative incentive for test set which needs further examination. A prescient model must be straightforward as basic conceivable, however no less complex. Regularization is a strategy used to build an ideally unpredictable model, in other words a model that is just about as basic as conceivable when performing great on preparing information. Through this interaction, we can find some kind of harmony between keeping model less difficult, yet not making it too guileless to be of any utilization. The relapse just attempts to limit the blunder and it doesn't represent model intricacy. A portion of the successful regularization methods bring down the intricacy of the model and forestall overfitting. Lately, numerous specialists have utilized these high level models to deal with multi-collinearity. We at that point arranged a progression of fits utilizing two regularized direct relapse models-Ridge and Lasso relapse models. the model and forestall overfitting. As of late, numerous specialists have utilized these high level models to deal with multi-collinearity. We at that point arranged a progression of fits utilizing two regularized direct relapse models-Ridge and Lasso relapse models.

RIDGE REGRESSION

Ridge regression model is a regularization model, where an extra variable (tuning parameter) is added and optimized to address the effect of multiple variables in linear regression which is usually referred as noise in statistical context. In mathematical form, the model can be expressed as

$$y = bx + e$$

Here, y is the needy variable x alludes to highlights in network structure and b alludes to relapse coefficients and e addresses residuals. In light of this, the factors are normalized by taking away the particular factors and separating them by their standard deviations. The tuning capacity indicated as λ is then appeared as part of regularization in the edge relapse model. In the event that λ 's worth is enormous, the squares ' leftover total gives off an impression of being zero. In the event that it is not exactly the arrangements adjust to least square strategy. λ is discovered utilizing a strategy called cross - approval. Edge relapse diminishes the coefficients to self-assertively low qualities however not to nothing. We will likewise perform matrix search cross-approval to tune the regularization hyper boundary λ . We have picked wide reach for hyper boundary and discovered 0.001

Metric	Train set	Test set
R-square	0.4345	0.4358
RMSE	0.5415	0.5224
RMSLE	0.0410	0.040701

as best worth. The model acquired has a variety of coefficients:

Fig 3.5 R-square and RMSE values for ridge regression model

LASSO REGRESSION

LASSO implies least total shrinkage, and the choice administrator is a LR method that likewise regularizes usefulness. It is indistinguishable from edge relapse, then again, actually it differs in the estimations of regularization. The outright estimations of the amount of relapse coefficients are contemplated. It even sets the coefficients to nothing so it totally decreases the mistakes. So determination of highlights are come about by rope relapse. In the recently referenced edge condition, the part ' e ' has outright qualities rather than squared qualities. It is to be noticed that computationally Lasso relapse strategy is definitely more concentrated than Ridge relapse procedure. We have performed lattice search cross-approval to tune the regularization hyper boundary λ . We have picked wide reach for hyper-boundaries and discovered 0.001 as best worth. The model got has a variety of coefficients:

Cluster ([2.26872602e-02, 3.84815301e-05, 1.88127460e-01, 6.35927974e-02, - 0.00000000e+00, 4.65240926e-01, - 5.40347504e-02, 2.12552087e-01, 1.13374140e-01, 3.49669654e-07])

We see that one of coefficient of highlights has been diminished to nothing. The measurements acquired is summed up

In the recently referenced edge condition, the part ' e ' has outright qualities rather than squared qualities. It is to be noticed that computationally Lasso relapse procedure is definitely more concentrated than Ridge relapse method. We have performed network search cross-approval to tune the regularization hyper boundary λ . We have picked wide reach for hyper-boundaries and discovered 0.001 as best worth. The model got has a variety of coefficients: The total estimations of the amount of relapse coefficients are thought about. It even sets the coefficients to nothing so it totally lessens the mistakes. So choice of highlights are come about by rope relapse.

In basic straight relapse we endeavor to limit the blunder, though in SVR we attempt to fit he mistake inside a specific edge. It is a relapse calculation and utilizations a comparative Support Vector Machines (SVM) procedure for relapse

Analysis. Regression information contains ceaseless genuine numbers. To fit such kind of information, the SVR model approximates the best qualities with a given

edge called ϵ tube (epsilon-tube, ϵ recognizes a cylinder width) with considering the model intricacy and mistake rate appeared.

The model created would now be able to be utilized to make forecasts on test information where the estimation of the objective variable is obscure.

A prescient model must be straightforward as basic conceivable, however no less complex. Regularization is a technique used to build an ideally perplexing model, in other words a model that is just about as straightforward as conceivable when performing admirably on preparing information. Through this cycle, we can find some kind of harmony between keeping model less difficult, yet not making it too gullible to be of any utilization. The relapse just attempts to limit the mistake and it doesn't represent model intricacy. A portion of the compelling regularization methods bring down the intricacy of the model and forestall overfitting.. As of late, numerous scientists have utilized these high level models to deal with multi-collinearity. We at that point arranged a progression of fits utilizing two regularized direct relapse models. - Ridge and Lasso relapse models. To fit such kind of information, the SVR model approximates the best qualities with a given edge called ϵ tube (epsilon-tube, ϵ recognizes a cylinder width) with considering the model intricacy and blunder rate appeared.

The model created would now be able to be utilized to make forecasts on test information where the estimation of the objective variable is obscure. We see that one of coefficient of highlights has been decreased to nothing. The measurements acquired is summed up

In the recently referenced edge condition, the part ' e ' has outright qualities rather than squared qualities. It is to be noticed that computationally Lasso relapse strategy is definitely more concentrated than Ridge relapse method. We have performed framework search cross-approval to tune the regularization hyper boundary λ . We have picked wide reach for hyper-boundaries and discovered 0.001 as best worth. The model got has a variety of coefficients: The total estimations of the amount of relapse coefficients are mulled over. It even sets the coefficients to nothing so it totally diminishes the blunders.

3.5 python

packages Numpy

NumPy (articulated (NUM-py)) is a library for the Python programming language, adding support for huge, multi-dimensional exhibits and networks, alongside an enormous assortment of undeniable level numerical capacities to work on these clusters. The predecessor of NumPy, Numeric, was initially made by Jim Hugunin with commitments from a few different designers. In 2005, Travis Oliphant made NumPy by consolidating highlights of the contending Numarray into Numeric, with extensive modifications. NumPy is open-source programming and has many contributors. NumPy focuses on the CPython reference execution of Python, which is a non-enhancing bytecode mediator. Numerical calculations composed for this form of Python regularly run much more slow than accumulated reciprocals. NumPy tends to the gradualness issue part of the way by giving multidimensional clusters and capacities and administrators that work proficiently on exhibits, requiring reworking some code, for the most part inward circles, utilizing NumPy.

Utilizing NumPy in Python gives usefulness practically identical to matlab since they are both deciphered, and the two of them permit the client to compose quick projects as long as most tasks work on clusters or frameworks rather than scalars. In correlation, matlab brags an enormous number extra tool compartments, outstandingly Simulink, while NumPy is characteristically incorporated with Python, a more current and complete programming language. Additionally, correlative Python bundles are accessible; SciPy is a library that adds more matlab-like usefulness and Matplotlib is a plotting bundle that gives matlab-like plotting usefulness. Inside, both matlab and numpy depend on blas and lapack for productive direct polynomial math calculations.

PANDAS

Pandas is a Python library. Pandas is utilized to examine data. Pandas is a Python library utilized for working with information sets. It has capacities for investigating, cleaning, investigating, and controlling information. The name "Pandas" has a

reference to both "Board Data", and "Python Data Analysis" and was made by Wes McKinney in 2008. Pandas permits us to break down enormous information and make ends dependent on measurable speculations. Pandas can clean muddled

informational collections, and make them intelligible and applicable. Significant information is vital in information science. Pandas offers you responses about the information. Like: Is there a connection between's at least two sections? What is normal worth Max esteem Min esteem Pandas are likewise ready to erase lines that are not applicable, or contains wrong qualities, similar to void or NULL qualities. This is called cleaning the information.

SCIKIT-LEARN

scikit-learn is to a great extent written in python, and utilizes numpy widely for elite direct variable based math and cluster tasks. besides, some center calculations are written in cython to improve execution. support vector machines are executed by a cython covering around libsvm; strategic relapse and straight help vector machines by a comparative covering around liblinear. in such cases, expanding these strategies with python may not be conceivable. scikit-learn coordinates well with numerous other python libraries, for example, matplotlib and plotly for plotting, numpy for exhibit vectorization, pandas dataframes, scipy, and numerous more. scikit-learn (previously scikits.learn and furthermore known as sklearn) is a free programming AI library for the python programming language. it highlights different grouping, relapse and bunching calculations including support vector machines, arbitrary woodlands, inclination boosting, k-implies and dbscan, and is intended to interoperate with the python mathematical and logical libraries numpy and scipy.

MATPLOTLIB

matplotlib is a plotting library for the python programming language and its mathematical arithmetic augmentation numpy. it gives an article situated programming interface to installing plots into applications utilizing universally useful gui tool stash like tkinter, wxpython, qt, or gtk+. there is likewise a procedural "pylab" interface dependent on a state machine (like opengl), intended to intently look like that of matlab, however its utilization is debilitate. scipy utilizes matplotlib.

matplotlib was initially composed by john d. tracker. from that point forward it has

a functioning improvement local area and is conveyed under a bsd-style permit. michael droettboom was designated as matplotlib's lead designer in no time before john tracker's passing in august 2012 and was additionally joined by thomas caswell.

matplotlib 2.0.x backings python forms 2.7 through 3.6. python 3 help began with matplotlib 1.2. matplotlib 1.4 is the last form to help python 2.6. matplotlib has promised not to help python 2 past 2020 by marking the python 3 articulation pyplot is a matplotlib module which gives a matlab-like interface. matplotlib is intended to be just about as usable as matlab, with the capacity to utilize python, and the upside of being free and open-source.

LINEAR REGRESSION

Linear Regression is an AI calculation dependent on directed learning. It plays out a relapse task. Relapse models an objective forecast esteem dependent on free factors. It is for the most part utilized for discovering the connection among factors and guaging. Diverse relapse models contrast dependent on – the sort of connection among needy and free factors, they are thinking about and the quantity of autonomous factors being utilized.

This article will exhibit how to utilize the different Python libraries to execute direct relapse on a given dataset. We will exhibit a twofold direct model as this will be simpler to picture. There are two kinds of directed AI calculations: Regression and order. The previous predicts constant worth yields while the last predicts discrete yields. For example, foreseeing the cost of a house in dollars is a relapse issue while anticipating whether a tumor is threatening or generous is an order issue. we will momentarily examine what direct relapse is and how it tends to be executed utilizing the Python Scikit-Learn library, which is quite possibly the most well known AI libraries for Python. The expression "linearity" in polynomial math alludes to a straight connection between at least two factors. On the off chance that we attract this relationship a two dimensional space (between two factors, for this situation), we get a straight line.

We should consider a situation where we need to decide the straight connection between the quantities of hours an understudy examines and the level of imprints that understudy scores in a test. We need to discover that given the quantity of hours an understudy gets ready for a test, about how high of a score can the understudy accomplish? On the off chance that we plot the free factor (hours) on the x-pivot and ward variable

(percentage) on the y-axis, linear regression gives us a straight line that best fits the data points, as shown in the figure below.

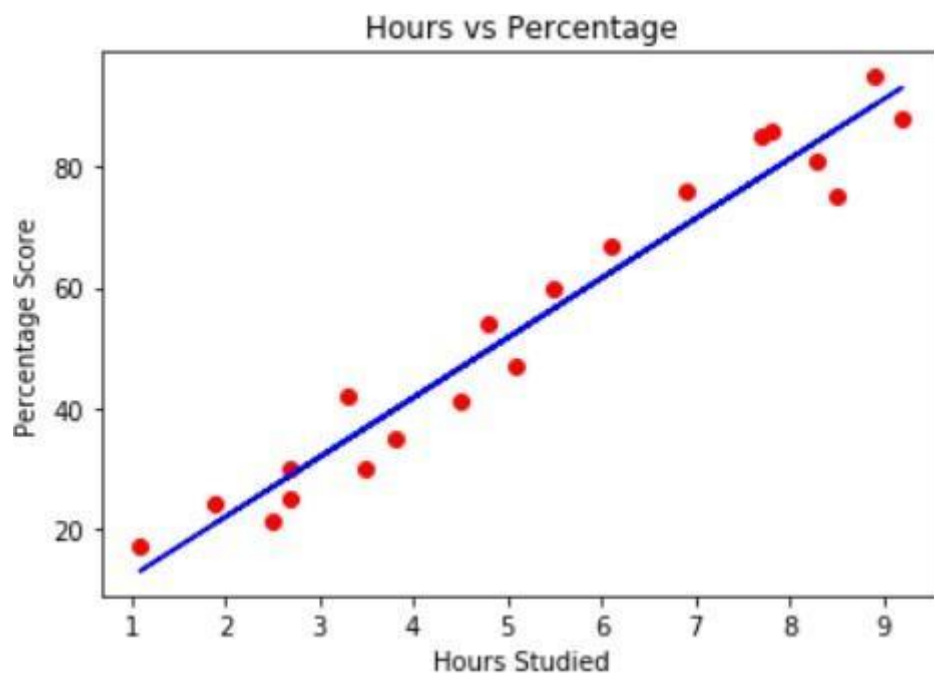


Fig 3.3 Linear regression example

CHAPTER 4

SOFTWARE DEVELOPMENT METHODOLOGY

4.1 DESCRIPTION OF DIAGRAM

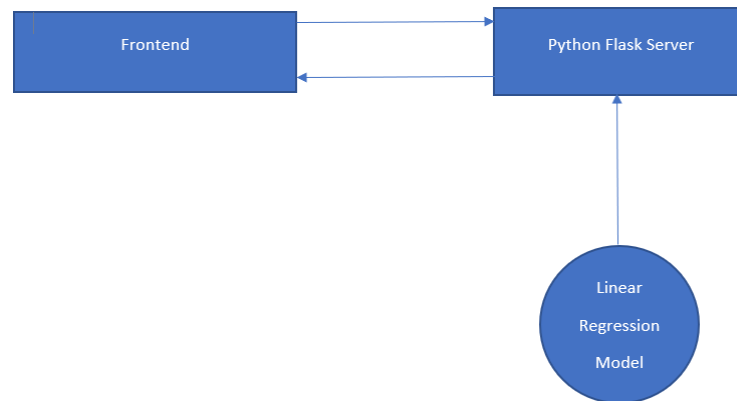


Fig 4.1 block diagram

Web Development and AI have been quite a while energy of mine. It has consistently been a plan to join these two interests and combine them and make a task. At last, I have figured out how to achieve this undertaking by building a web application that predicts the land cost for properties and houses across the city of Bengaluru, India.

The initial step is common information science work where we take an informational index from Kaggle called 'Bengaluru House value data'. We will play out some broad information cleaning work on it to guarantee that it gives exact outcomes during forecast.

This jupyter note pad named 'RealEstatePricePredictor.ipynb' is the place where we play out all the information science related work. As the jupyter journal is plain as day I will momentarily address the ideas that I have executed. Our dataset requires a great deal of work as far as information cleaning. Truth be told, 70% of the scratch pad is about information cleaning where we drop void lines and

eliminate superfluous segments that will not assistance in forecast.

Subsequent stage, Feature Engineering which is the way toward separating valuable and significant data from the dataset that will contribute the most towards an effective expectation.

The last advance is taking care of exceptions. Exceptions are irregularities that cause a colossal measure of harm to information and expectation. There is a great deal of things to comprehend from the dataset consistently to distinguish and eliminate these exceptions.

Once more, these have been clarified in the jupyter scratch pad.

Eventually, the first dataset which had very nearly 13000 lines and 9 sections are decreased to right around 7000 lines and 5 segments.

The last information acquired is exposed to an AI model. We will principally utilize K-cross Validation and GridSearchCV procedure to perform hyper boundary tuning to get the best calculation and boundaries for the model.

Turns out straight relapse model gives the best outcomes for our information with a score above 80% which isn't terrible.

Presently, our model should be traded into a pickle document (Bengaluru_House_Data.pickle) which changes over python objects into a character stream. Additionally, we need to trade the locations(columns) into a json(columns.json) document to have the option to interface with it from the frontend. We will utilize a Flask worker as our backend to have the application locally. In the worker organizer we will set up two documents:

- server.py
- util.py

The server.py document will be answerable for dealing with the courses for getting the area names and anticipating the house cost. It likewise gets the structure information from the front end and feeds it to the util.py. These courses can be tried utilizing Postman application.

The util.py record is the primary cerebrums behind the back end. It has a capacity to stack the JSON and pickle document. This record takes the structure information from server.py and utilizes the model to anticipate the assessed cost of the property.

The front end is comprised of basic HTML, CSS and JavaScript. The client can choose the quantity of square feet zone, BHK, restrooms and area in the structure and hit on 'Gauge' catch to get the assessed cost. The JavaScript document is liable for connecting with both the backend flagon worker courses and the frontend HTML. It gets the structure information filled by the client and considers the capacity that utilizes the forecast model and delivers the assessed cost in lakhs rupees (1 lakh = 100000). Run the server.py document in the backend and open up the HTML website page we made. Information the territory of the property in square feet, the quantity of BHK, the quantity of restrooms and the area and hit 'Gauge'. The last advance is dealing with anomalies. Exceptions are abnormalities that cause a colossal measure of harm to information and forecast. There is a ton of things to comprehend from the dataset coherently to distinguish and eliminate these anomalies. Once more, these have been clarified in the jupyter note pad.

This jupyter note pad named 'RealEstatePricePredictor.ipynb' is the place where we play out all the information science related work. As the jupyter note pad is obvious I will momentarily address the ideas that I have carried out. Our dataset requires a ton of work regarding information cleaning. Truth be told, 70% of the note pad is about information cleaning where we drop void lines and eliminate superfluous sections that will not assistance in forecast. The JavaScript document is liable for collaborating with both the backend flagon worker courses and the frontend HTML. It gets the structure information filled by the client and considers the capacity that utilizes the expectation model and delivers the assessed cost. The server.py record will be answerable for taking care of the courses for bringing the area names and anticipating the house cost. It likewise gets the structure information from the front end and feeds it to the util.py. These courses can be tried utilizing Postman app. The util.py record is the fundamental

minds behind the back end. It has a capacity to stack the JSON and pickle document. This record takes the structure information from server.py and utilizes the model to anticipate the assessed cost of the property. It gets the structure information filled by the client and considers the capacity that utilizes the expectation model and delivers the assessed cost. The server.py

document will be answerable for dealing with the courses for getting the area names and foreseeing the house cost. It likewise gets the structure information from the front end and feeds it to the util.py.

4.2 ARCHITECTURAL DESIGN

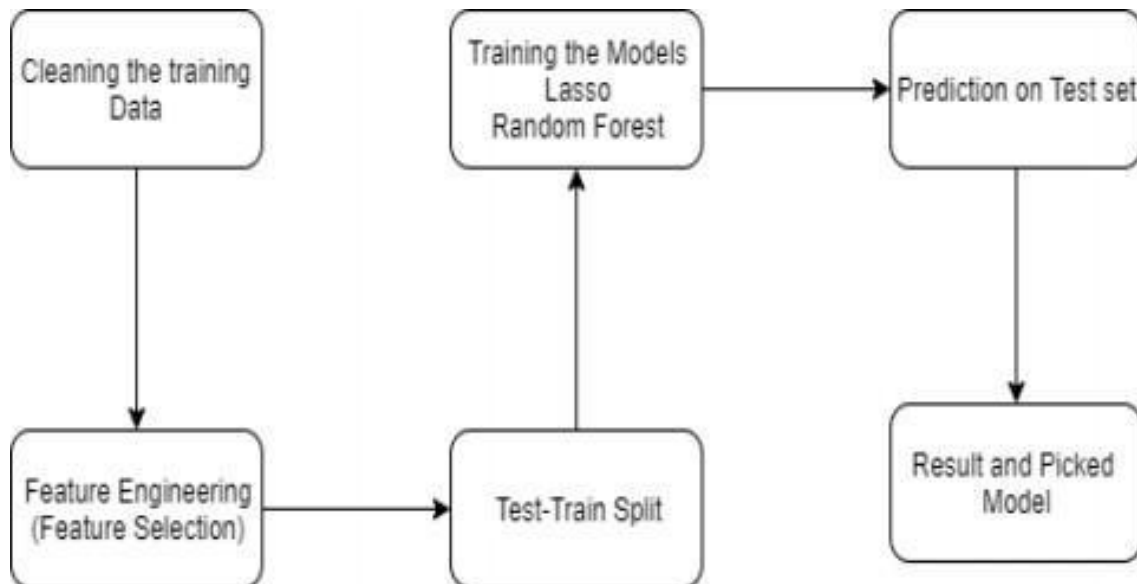


Fig 4.2 System Architecture

Relapse takes each conceivable incentive for k_0 and k_1 and limits the absolute deviation; this is the possibility of relapse basically. In any case, all things considered, there are different difficulties you need to manage. House costs clearly rely upon different boundaries, and there is no reasonable direct connection between these boundaries. Presently I will reveal to you how I utilized relapse calculations to foresee house cost for my pet task.

Gather data

The first step for any kind of machine learning analysis is gathering the data – which must be valid. If you can't guarantee the validity of your data, then there's no point analysing it. You need to pay attention to the source you take your data from.

For my purposes, I've relied on a database from one of the largest real-estate portals in the Netherlands. Since the real estate market in the Netherlands is strictly regulated, I didn't have to check its validity.

Initially I had to do some data mining because the required data were available in various formats across multiple sources. In most real-world projects you'll have to do some data mining as well. We won't discuss data mining here, however, as it's not really relevant to our topic.

For the purpose of this article, let's imagine that all the real-estate data I found were in the format shown below.

Analyze data

Once you've gathered data it's time to analyze it. After parsing the data I got the following records:

Put simply, regression is a machine learning tool that helps you make predictions by learning – from the existing statistical data – the relationships between your target parameter and a set of other parameters. According to this definition, a house's price depends on parameters such as the number of bedrooms, living area, location, etc. If we apply artificial learning to these parameters we can calculate house valuations in a given geographical area.

The data sets are train set and test data considered in the project is taken from kaggle platform. It consists of features that describe house-property in Bengaluru. There are 9 features in both the data sets. The features can be explained as follows:

1. Area type-describes the area
2. Availability-when it is possessing or when it is ready.
3. Price- Value of the property in lakhs.
4. Size- in BHK or Bedroom (1-10 or more)
5. Society- to which it belongs.
6. Total_sqft - size of the property in sqft.
7. Bath-No of bathrooms
8. Balcony- No of balcony
9. Location – where it is located in Bengaluru

With 9 features available, we try to build regression models to predict house price. We predicted the price of test data set with the regression models built on train

data set. This dataset includes quite a few categorical variables (both train and test data set) for which we will need to create dummy variables or use label encoding to

convert into numerical form. These would be fake/dummy variables because they are placeholders for actual variable and are created by ourselves. Also, there are a lot of null values present as well, so we will need to treat them accordingly. The features bath, price, balcony are numerical variables. Features like area type, total_sqft, location, society, availability, and size appears as categorical variables.

correlation between parameters

The assumption that these are all measurable data that affect a home's price. Of course, there may be more parameters that matter as well, such as house condition and location. But these parameters are more subjective and almost impossible to measure, so I ignored them. Now you need to check for strong correlations among given parameters. If there are, then remove one of the parameters. In my dataset there were no strong correlations among values.

Remove outliers from the dataset

Outliers are observation points that are distant from other observations. For example, in my data there was one house with an area of 50 square meters for a price . Such houses may exist on the market for various reasons, but they are not statistically meaningful. I want to make a price estimate based on the market average, and so I won't take such outliers into account. Most regression methods explicitly require outliers be removed from the dataset as they may significantly affect the results. This will return -1 for outliers and 1 for non-outliers. After that you will have non-outlier observations only. Now it's time to start regression analysis.

Choose a regression algorithm

There's more than one approach to do relapse investigation. What we're searching for is the best forecast precision given our information. In any case, how might we check precision? A typical route is to figure a purported r^2 score which is fundamentally a squared distinction between a real and an anticipated value. It's critical to recall that on the off chance that we utilize that equivalent dataset for learning and checking our precision, our model may overfit. This implies it will show brilliant precision on a given dataset yet will totally bomb when

given new data. A basic way to deal with tackle this issue is to part the first dataset into two sections and afterward utilize one for learning and another for testing. This way we will

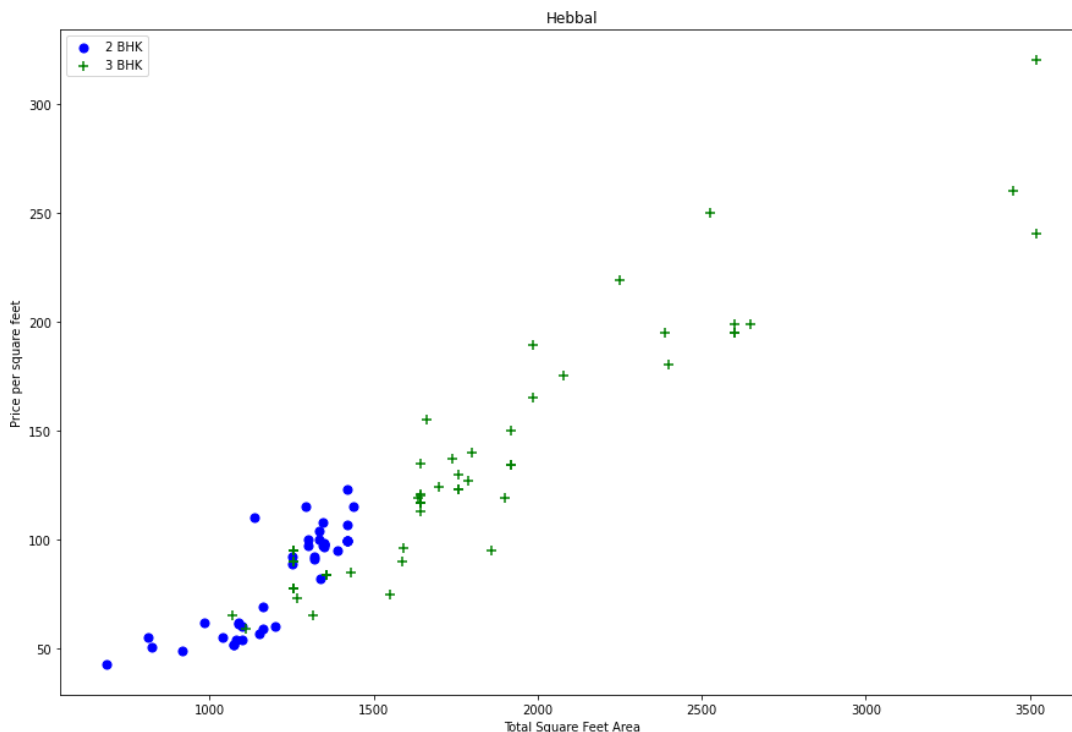
recreate new information for our learning model, and if there is an overfit, we can spot it. We can part our dataset utilizing an extent of 80/20. We'll utilize 80% for preparing and the excess 20% for testing. As should be obvious, the straight relapse showed the best exactness, so we chose to utilize this calculation for creation. Value expectation underway works basically equivalent to in our test code besides there's no compelling reason to figure r^2 and switch models any longer. Now, we can offer reasonable value forecasts. We can analyze the genuine cost of a house with our anticipated cost and notice the deviation. Relapse takes each conceivable incentive for k_0 and k_1 and limits the absolute deviation; this is the possibility of relapse basically. In any case, all things considered, there are different difficulties you need to manage. House costs clearly rely upon numerous boundaries, and there is no reasonable straight connection between these boundaries. Presently I will reveal to you how I utilized relapse calculations to foresee house cost for my pet task. It's critical to recall that on the off chance that we utilize that equivalent dataset for learning and checking our exactness, our model may overfit. This implies it will show fantastic exactness on a given dataset however will totally bomb when given new data. A regular way to deal with take care of this issue is to part the first dataset into two sections and afterward utilize one for learning and another for testing. This way we will recreate new information for our learning model, and if there is an overfit, we can spot it. Such houses may exist available for different reasons, however they are not genuinely significant. I need to make a value gauge dependent available normal, thus I will not consider such exceptions. Most relapse strategies unequivocally require anomalies be taken out from the dataset as they may altogether influence the outcomes. This will return - 1 for exceptions and 1 for non-anomalies. After that you will have non-anomaly perceptions as it were. Set forth plainly, relapse is an AI apparatus that assists you with making forecasts by taking in - from the current measurable information - the connections between your objective boundary and a bunch of different boundaries. As indicated by this definition, a house's cost relies upon boundaries like the quantity of rooms, living region, area, and so forth. On the off chance that we apply fake figuring out how to these boundaries we can compute house valuations in a given geological region. The presumption that these are for

the most part quantifiable information that influence a home's cost. Obviously, there might be more boundaries that matter too, for example, house condition and area. In any case, these boundaries are more abstract and practically difficult to gauge,

CHAPTER 5

RESULTS AND DISCUSSION

First I will import the conditions, that will make this program somewhat simpler to compose. I'm bringing in the AI library sklearn, numpy, and pandas. I will stack the Housing Data Set from sklearn.datasets and print it subsequent to putting away it into the variable dataset by utilizing a capacity/technique called loaddata(). however, first I will import the library sklearn.datasets. I will part the information into autonomous factors (X's) and ward variable (Y) data sets. I will feel free to instate the Linear Regression model, split the information into 67% preparing and 33% testing information, and afterward train the model with the preparation informational collection that contains the autonomous



factors.

Fig 5.1 price per square feet vs total square feet

From fig 5.1 will split the data into independent variables (X's) and dependent variable (Y) data sets. will go ahead and initialize the Linear Regression model, split the data into 67% training and 33% testing data, and then train the model with the training data set that contains the independent variables.

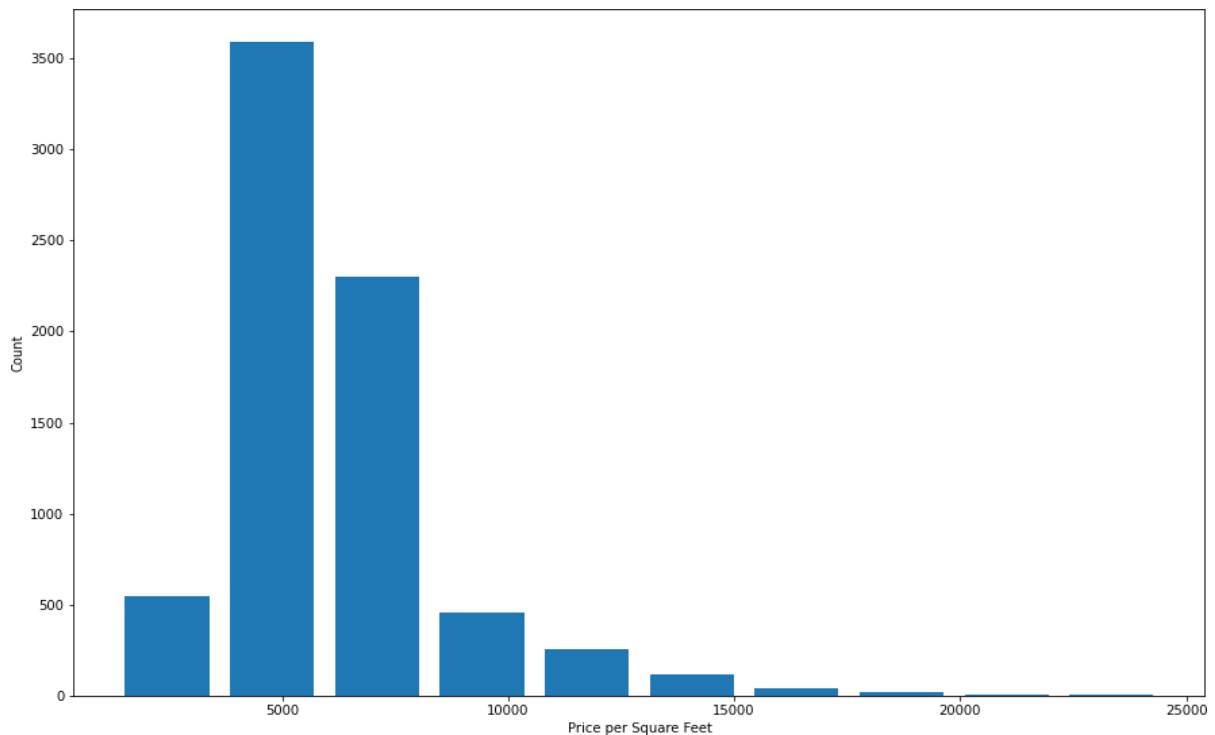


Fig 5.2 count vs price square feet

From fig 5.2 get the assessed coefficients for the straight relapse model. Since we are finished preparing the straight relapse model and taking a gander at the coefficients that depict the direct capacity, how about we print the model's expectations (what it figures the qualities will be for houses) on the test information. I need to realize what was the genuine qualities for that test informational collection, so I will print those qualities to the screen, yet first I will print at any rate one column from the model's forecast, just to make it somewhat simpler to look at information.

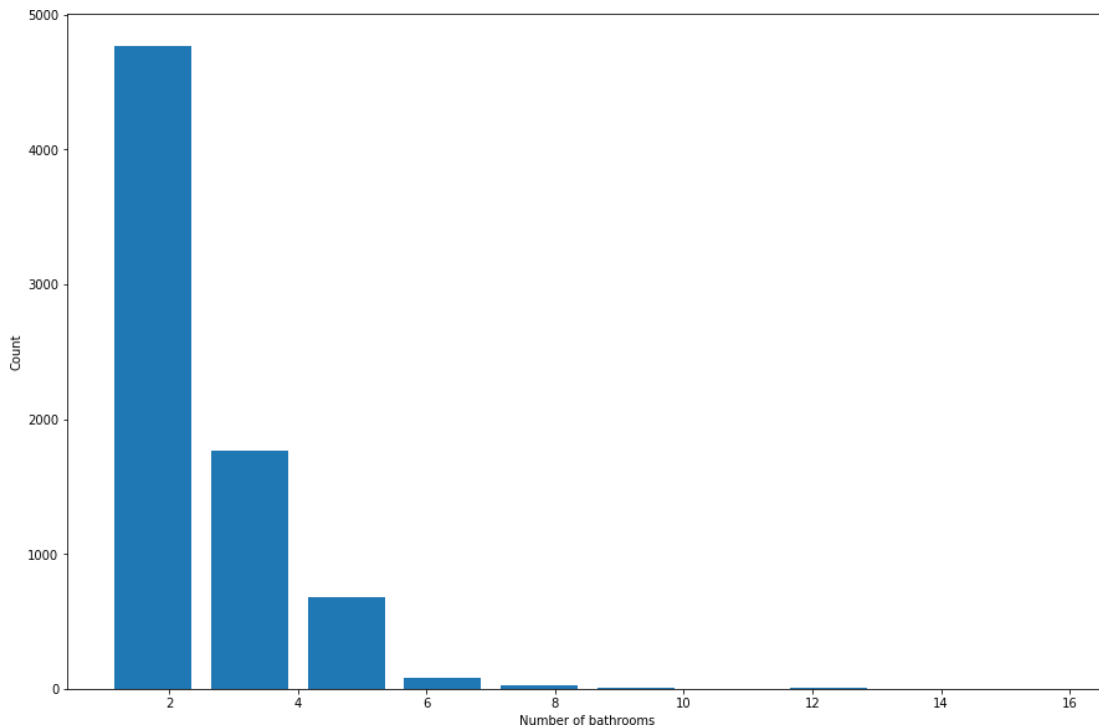


Fig 5.3 count vs Number of bathrooms

From fig 5.3 want to know what was the actual values for that test data set, so I will print those values to the screen, but first I will print at least one row from the model's prediction, just to make it a little easier to compare data.

```
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 O
+ %< %> Run Code
Lets perform our first few predictions.

In [58]: predict_price('1st Phase JP Nagar', 1000, 2, 2)
Out[58]: 83.49904677202828

In [59]: predict_price('1st Phase JP Nagar', 1000, 3, 3)
Out[59]: 86.8051939522959

In [57]: predict_price('Indira Nagar', 1000, 2, 2)
Out[57]: 181.27815484007027

In [58]: predict_price('Whitefield', 1000, 3, 3)
Out[58]: 56.66453527802284

In [59]: predict_price('Bannerghatta Road', 1000, 2, 2)
Out[59]: 49.30029484907186
```

Fig 5.4 predictions

From fig 5.4 will split the data into independent variables (X's) and dependent variable (Y) data sets. I will go ahead and initialize the Linear Regression model, split the data into 67% training and 33% testing data, and then train the model with the training data set that contains the independent variables.

House Price Prediction

Area in Sq. ft

1000

BHK

1 2 3 4 5

Bathroom

1 2 3 4 5

Location

1st block jayanagar ▼

ESTIMATE

196.91 Lakh

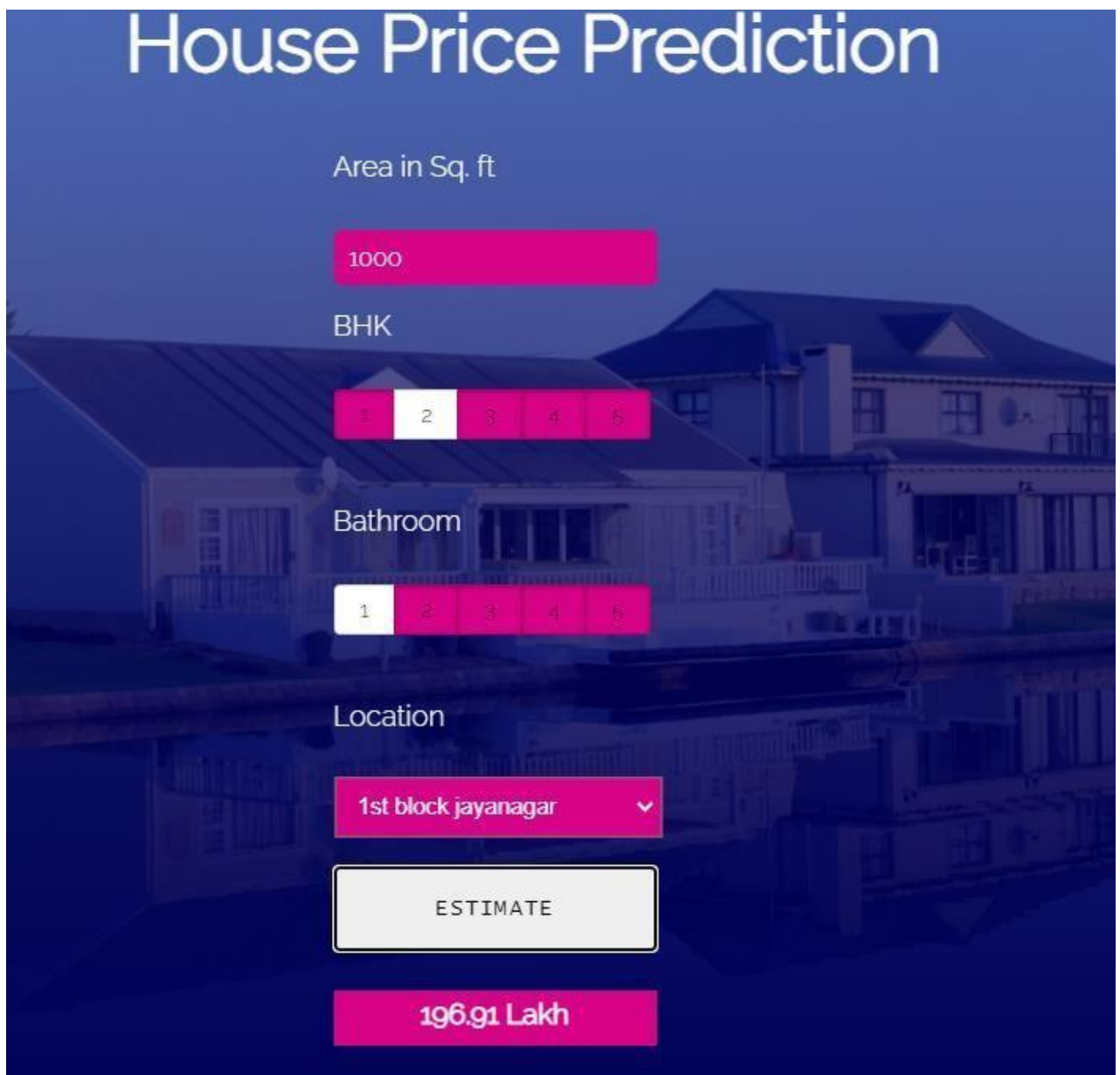


Fig 5.5 House price prediction

we can offer fair price predictions. We can compare the actual price of a house with our predicted price and observe the deviation.

If our model had predicted the exact value as the actual values then the mean squared error function would have returned a value of 0.

The result displays that the approach of the issue needs to be successful, and has the ability to operate predictions that would be comparative with other house price prediction models

This Linear Regression Model which we help the real state agent for estimating the house price. This model will help users to invest in a property without approaching an agent. It also decreases the risk involved in the transaction. Use of regression is done as model because of its convertible and probability methodology on model selection. Although the linear regression model that was created in this post is not perfect, it is able to account for approximately some percentage of the variation in Sale Price of a property, and it also is able to predict the Sale Price within the rupees.

To check the model's performance/accuracy I will use a metric called mean squared error (MSE). This measurement is simple to implement and easy to understand. The MSE is a measure of the quality of an estimator — it is always non-negative, and values closer to zero indicate a better fit.

CHAPTER 6

CONCLUSION AND FUTURE

WORK

6.1 CONCLUSION

An optimal model does not necessarily represent a robust model. A model that An ideal model doesn't really address a vigorous model. A model that oftentimes utilize a learning calculation that isn't reasonable for the given information structure. Once in a while the actual information may be excessively boisterous or it could contain too couple of tests to empower a model to precisely catch the objective variable which suggests that the model remaining parts fit. At the point when we notice the assessment measurements got for cutting edge relapse models, we can say both act along these lines. We can pick possibly one for house value expectation contrasted with essential model. With the assistance of box plots, we can check for anomalies. In the event that present, we can eliminate exceptions and check the model's presentation for development.

6.2 FUTURE WORK

We can build models through advanced techniques namely random forests, neural networks, and particle swarm optimization to improve the accuracy of predictions.

It is necessary to check before deciding whether the built model should or should not be used in a real-world setting. The data has been collected in 2016 and Bengaluru is growing in size and population rapidly. So, it is very much essential to look into the relevancy of data today. The characteristics present in the data set are not sufficient to describe house prices in Bengaluru. The dataset considered is quite limited and there are a lot of features, like the presence of pool or not, parking lot and others, that remain very relevant when considering a house price. The property has to be categorized either as a flat or villa or independent house. Data collected from a big urban city like Bengaluru would not be applicable in a rural city, as for equal value of feature prices, which will be comparatively higher

in the urban area.

APPENDIX

A. SOURCE CODE

MODEL SOURCE CODE

```
import pandas as pd

import numpy as np

from matplotlib import pyplot as plt

%matplotlib

inline import

matplotlib

matplotlib.rcParams["figure.figsize"] = (20, 10)

pd.options.mode.chained_assignment = None # To Disable
'SettingWithCopyWarning'

df1 = pd.read_csv("Bengaluru_House_Data.csv")

df1.head()

df1.shape

df1.groupby('area_type')['area_type'].agg('count')

df2 = df1.drop(['area_type', 'society', 'balcony', 'availability'],
axis='columns') df2.head()

df3 = df2.dropna()

df3.isnull().sum()

df3['size'].unique()

df3['bhk'] = df3['size'].apply(lambda x: int(x.split(' ')[0])) # Make new column called
bhk and extract just the number into them.
```

```
df3 = df3.drop(['size'],  
axis='columns') df3.head()  
df3[df3['bhk']>20]
```

```
def is_float(x):
```

```
    try:
```

```
        float(x)
```

```
    except:
```

```
        return False
```

```
    return True
```

```
df3[~df3['total_sqft'].apply(is_float)].head(10) # Get all those entries with range  
an an entry in 'total_sqft' column
```

```
def convert_sqft_to_num(x):
```

```
    tokens = x.split('-')
```

```
    if len(tokens) == 2:
```

```
        return (float(tokens[0])+float(tokens[1]))/2
```

```
    try:
```

```
        return float(x)
```

```
    except:
```

```
        return None
```

```
df4 = df3.copy()
```

```
df4['total_sqft'] = df4['total_sqft'].apply(convert_sqft_to_num) # Apply the
```

```
function df4.head()
```

```
df4.loc[30]
```

```
df5['price_per_sqft'] =
```

```
df5['price']*100000/df5['total_sqft'] df5.head()
```

```
df5.location = df5.location.apply(lambda x: x.strip())
```

```
location_stats =
```

```
df5.groupby('location')['location'].agg('count').sort_values(ascending=False)
```

```
location_stats
```

```
location_stats_less_than_10 =
```

```
location_stats[location_stats<=10] location_stats_less_than_10
```

```
df5.location = df5.location.apply(lambda x: 'other' if x  
in location_stats_less_than_10 else x)
```

```
len(df5.location.unique())
```

```
df6 = df5[~(df5['total_sqft']/df5['bhk']<300)] # Remove entries whose sqft/bhk is  
less than 300
```

```
df6.head()
```

```
df6.price_per_sqft.describe()
```

```
def remove_pps_outliers(df): # Function to remove outliers for every location
```

```
    df_out = pd.DataFrame()
```

```
    for key, subdf in df.groupby('location'):
```

```
        m = np.mean(subdf.price_per_sqft) # Calculate mean
```

```
        st = np.std(subdf.price_per_sqft) # Calculate SD
```

```
        reduced_df = subdf[(subdf.price_per_sqft>(m-st)) &
```

```
(subdf.price_per_sqft<=(m+st))]
```

```
        df_out = pd.concat([df_out, reduced_df],
```

```
        ignore_index=True) return df_out
```

```
df7 = remove_pps_outliers(df6)
```

```
df7.shape
```

```
def plot_scatter_chart(df, location):
```

```
    bhk2 = df[(df.location==location) &
```

```
(df.bhk==2)] bhk3 = df[(df.location==location)
```

```
& (df.bhk==3)]
```

```
matplotlib.rcParams['figure.figsize'] = (15, 10)
```

```

plt.scatter(bhk2.total_sqft, bhk2.price, color='blue', label='2 BHK', s=50)

plt.scatter(bhk3.total_sqft, bhk3.price, marker='+', color='green', label='3
BHK', s=50)

plt.xlabel('Total Square Feet Area')

plt.ylabel('Price per square feet')

plt.title(location)

plt.legend()


plot_scatter_chart(df7, "Hebbal") # An Example of the issue (Around 1500 sq.ft
area)

def remove_bhk_outliers(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]

                }

    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')

df8 = remove_bhk_outliers(df7)

df8.shape

plt.hist(df8.price_per_sqft,
rwidth=0.8) plt.xlabel('Price per
Square Feet') plt.ylabel('Count')

plt.hist(df8.bath, rwidth=0.8)

plt.xlabel('Number of
bathrooms') plt.ylabel('Count')

dummies =
pd.get_dummies(df10.location)

dummies.head()

df11 = pd.concat([df10, dummies.drop('other', axis="columns")], axis='columns') #
Append df10 and dummies

dataframe df11.head()

df12 = df11.drop('location',
axis='columns') df12.head()

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=10)

from sklearn.linear_model import LinearRegression

lr_clf = LinearRegression() # Linear Regression Model

```

```
lr_clf.fit(X_train, y_train)
```

```
lr_clf.score(X_test, y_test)
```

```
from sklearn.model_selection import ShuffleSplit
```

```

from sklearn.model_selection import cross_val_score

cv = ShuffleSplit(n_splits=5, test_size=0.2, random_state=0) # ShuffleSplit will
randomize the sample

cross_val_score(LinearRegression(), X,y, cv=cv)

from sklearn.model_selection import
GridSearchCV from sklearn.linear_model import
Lasso

from sklearn.tree import DecisionTreeRegressor

# Function that decides between the three most popular algorithms which are
iterated through to find out which one gives the

# best score

def find_best_model_using_gridsearch(X, y):

    algos = {

        'linear_regression': {

            'model': LinearRegression(),

            'params': {

                'normalize': [True, False]

            }

        },

        'lasso': {

            'model':

                Lasso(),

            'params': {

```

```
'alpha': [1, 2],  
'selection': ['random', 'cyclic']  
}
```

```

    },
    'decision_tree': {
        'model': DecisionTreeRegressor(),
        'params': {
            'criterion': ['mse', 'friedman_mse'],
            'splitter': ['best', 'random']
        }
    }
}

```

```

scores = []

cv = ShuffleSplit(n_splits=5, test_size=0.2, random_state=0)

for algo_name, config in algos.items():

    gs = GridSearchCV(config['model'], config['params'],
cv=cv, return_train_score=False)

    gs.fit(X, y)

    scores.append({

        'model': algo_name,

        'best_score': gs.best_score_,

        'best_params':

            gs.best_params_

    })

```

```

return pd.DataFrame(scores, columns = ['model', 'best_score', 'best_params'])

```

```
find_best_model_using_gridsearch(X, y)
```

```

def predict_price(location, sqft, bath, bhk):

    loc_index = np.where(X.columns==location)[0][0]

    x = np.zeros(len(X.columns))

    x[0] = sqft

    x[1] =

    bath x[2]

    = bhk

    if loc_index >= 0:

        x[loc_index] = 1


    return lr_clf.predict([x])[0]

predict_price('1st Phase JP Nagar', 1000, 2, 2)

predict_price('Indira Nagar', 1000, 2,

2) import pickle

with open('Bengaluru_House_Data.pickle', 'wb') as

    f: pickle.dump(lr_clf, f)

import json

columns = {

    'data_columns': [col.lower() for col in X.columns]

}

with open('columns.json', 'w') as

    f: f.write(json.dumps(columns))

```

Server code

```
from flask import Flask, request, jsonify, render_template
```

```
import util
```

```
app = Flask(__name__)
```

```
@app.route('/get_location_names', methods=['GET'])
```

```
def get_location_names():
```

```
    response = jsonify({
```

```
        'locations': util.get_location_names() # Populate locations to the frontend
```

```
    })
```

```
    response.headers.add('Access-Control-Allow-Origin', '*')
```

```
    return response
```

```
@app.route('/predict_home_price', methods=['GET',
```

```
'POST']) def predict_home_price():
```

```
    total_sqft =
```

```
    float(request.form['total_sqft']) location =
```

```
    request.form['location']
```

```
    bhk = int(request.form['bhk'])
```

```
    bath =
```

```
    int(request.form['bath'])
```

```
response = jsonify({
```

```
    # populate the estimated price to the frontend
```

```
    'estimated_price': util.get_estimated_price(location, total_sqft, bhk, bath)
})
```

```
response.headers.add('Access-Control-Allow-Origin', '*')
```

```
return response
```

```
@app.route('/')
```

```
def index():
```

```
    return render_template('index.html')
```

```
if __name__ == "__main__":
```

```
    print("Starting Python Flask Server For Home Price Prediction...")
```

```
    util.load_saved_artifacts()
```

```
    app.run()
```

REFERENCES

- [1] H.L. Harter, Method of Least Squares and some alternatives-Part II. International Static Review. 1972, 43(2), pp. 125-190.
- [2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.
- [3] Lu. Sifei et al, A hybrid regression technique for house prices prediction. In proceedings of IEEE conference on Industrial Engineering and Engineering Management: 2017.
- [4] R. Victor, Machine learning project: Predicting Boston house prices with regression in towards datascience.
- [5] S. Neelam, G. Kiran, Valuation of house prices using predictive techniques, Internal Journal of Advances in Electronics and Computer Sciences: 2018, vol 5, issue-6
- [6] S. Abhishek.: Ridge regression vs Lasso, How these two popular ML Regression techniques work. Analytics India magazine, 2018.
- [7] S. Raheel. Choosing the right encoding method-Label vs One hot encoder. Towards datascience, 2018.
- [8] Raj, J. S., & Ananthi, J. V. (2019). Recurrent Neural Networks and Nonlinear Prediction in Support Vector Machines. Journal of Soft Computing Paradigm (JSCP), 1(01), 33-40.
- [9] Predicting house prices in Bengaluru (Machine Hackathon) <https://www.machinehack.com/course/predicting-house-prices-inbengaluru/>
- [10] Raj, J. S., & Ananthi, J. V. (2019). Recurrent neural networks and nonlinear prediction in support vector machines. Journal of Soft Computing Paradigm (JSCP), 1(01), 33-40.
- [11] Pow, Nissan, Emil Janulewicz, and L. Liu (2014). Applied Machine Learning Project 4 Prediction of real estate property prices in Montréal.
- [12] Wu, Jiao Yang (2017). Housing Price prediction Using Support Vector

Regression.

- [13] Limsombunchai, Visit. 2004. House price prediction: hedonic price model vs. artificial neural network. New Zealand Agricultural and Resource Economics Society Conference.
- [14] Rochard J. Cebula (2009). The Hedonic Pricing Model Applied to the Housing Market of the City of Savannah and Its Savannah Historic Landmark District; *The Review of Regional Studies* 39.1 (2009), pp. 9– 22
- [15] Gu Jirong, Zhu Mingcang, and Jiang Liuguangyan. (2011). Housing price based on genetic algorithm and support vector machine”. In: *Expert Systems with Applications* 38 pp. 3383–3386.
- [16] Danny P. H. Tay and David K. H. Ho. (1992) Artificial Intelligence and the Mass Appraisal of Residential Apartments. In: *Journal of Property Valuation and Investment* 10.2 pp. 525–540.
- [17] Rohan Bafna, Anirudh Dhole, Ankit Jagtap, Asif Kazi, Arbaz Kazi “Prediction of Residential Property Prices – A State of the Art” *International Advanced Research Journal in Science, Engineering and Technology IARJSET*
- [18] Azme Bin Khamis, Nur Khalidah Khalilah Binti Kamarudin., (2014), “Comparative Study on Estimate House Price Using Statistical and Neural Network Model”, *International Journal of Scientific & Technology Research* Vol.3, Issue 12, December 2014, 126-131.
- [19] Calhoun C. A., (2003), “Property Valuation Models and House Price Indexes for The Provinces of Thailand: 1992 2000”, *Housing Finance International*, 17(3): 31 – 41.
- [20] Frew J., Jud G.D., (2003), “Estimating the Value of Apartment Buildings”, *Journal of Real Estate Research* Vol. 25, No. 1, 2003, 77 – 86.

