PROFESSIONAL TRAINING REPORT

at

Sathyabama Institute of Science and Technology

(Deemed to be University)

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

By

RAVIPATI ABHIRAM
REG. NO. 39110840

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU

APRIL 2022

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI

CHENNAI– 600119

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 **RAVIPATI ABHIRAM (Reg. No: 39110840)** who carried out the project entitled "**HOUSE RENT PREDICTION**" under my supervision from February 2022 to April 2022.

Internal Guide

Ms. LAKSHMI PRIYA

Head of the Department

Dr. L. Lakshmanan, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner **External Examiner**

DECLARATION

I, **RAVIPATI ABHIRAM** hereby declare that the project report entitled **HOUSE RENT PREDICTION** done by me under the guidance of **Ms. LAKSHMI PRIYA** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE: 10-04-2022 **R. Abhi Ram**

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. Vigneshwari, 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. LAKSHMI PRIYA** 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.

INTERNSHIP CERTIFICATE

TABLE OF CONTENTS

ABSTRACT

People looking to rent a house tend to be more conservative with their budgets and priorities. There are several factors that affect the rental cost of the house such as location, maintenance cost, property size, etc. This system proposes a performance comparison between two machine learning regression algorithms. The regression algorithms used in this study are Linear Regression and Random Forest. The accuracy of the prediction is evaluated by checking the accuracy and root mean square error score of the training model. The test is performed after applying the required pre-processing methods and splitting the data into two parts. However, one part will be used in training the model and the other in testing phase. Linear Regression gives best score when compared to Random Forest algorithm. So, we used Linear Regression model to predict the house rents. By analyzing previous market trends and price ranges the future prices will be predicted. The functioning of this project involves a website which accepts customer's specifications and then uses the prediction model to predict the house rent. The system will return the predicted house rent to the customer based on the specifications the customer gives. This application will help customers to make their own research and analyze without approaching real-estate brokers when searching to rent a property.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

ABBREVATION	FULL FORM
NumPy	Numerical Python
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION TO HOUSE RENT PREDICTION:

House is the one of the man's basic need. When people are in their starting point of their career they don't have enough money to own a house. Instead they try to rent a house and slowly when they grow in their career and have enough money, they own their dream house. Not only in their early age of career, they try to rent a property even though they have a own property it might be because of their work city being different from their home town where they have the own house. Students also live in a rental property. So, mostly everyone goes through a point where they need to rent a property. When we try to rent a property we have our own preferences. We need to analyze the location, size of the property and maintenance cost etc. The rental prices of houses changes in due time. The cost of the rental property is the main thing which we need to look into when we start thinking to rent a house. Most of the times, we tend to go to a real-estate broker to help us find a rental house based on our preferences. The real-estate broker may or may not give us the exact insights with our preferences. He may give us some limited insights. He also charges us for this entire process. So, we created this House Rent Predictor which can help us analyze the rental costs of the houses located in the city of Hyderabad. We can analyze on own and get more insights than we get from a real-estate broker. For predicting the house rent, we will use machine learning algorithm which takes in the data and predict the value. This project efficiently analyzes previous market trends and price ranges and predicts the rent value. The Back-end is created using Flask Library and front-end is created using Html, CSS and JavaScript. The front-end takes in the values and when the predict button is clicked the values are sent to the backend and the desired output is sent back and displayed on the front-end.

Fig 1.1: House for Rent

1.2 INTRODUCTION TO MACHINE LEARNING:

Machine Learning is a subfield of Artificial Intelligence (AI) that works with algorithms and technologies to extract useful information from data. It allows software applications to become more accurate at predicting outcomes without being explicitly programmed. Machine Learning Algorithms use historical data as input to predict the desired output values. Machine Learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions. It is used in wide variety of applications, such as in speech recognition, computer vision, face recognition, spam detection etc.

1.2.1 Types of Machine Learning:

- Supervised Learning
- Unsupervised Learning
- Semi-supervised Learning
- Reinforcement Learning

Supervised Learning:

In this type, the algorithm is supplied with labeled training data and define the variables they want the algorithm to assess for correlations. Both the input and the output of the algorithm is specified.

Unsupervised Learning:

This type of machine learning involves algorithms that train on unlabeled data. The algorithm scans through data sets looking for any meaningful connection. The data that algorithms train on as well as the predictions or recommendations they output are predetermined.

Semi-Supervised Learning:

This approach to machine learning involves a mix of the two preceding types. The algorithm is mostly fed with labeled training data, but the model is free to explore the data on its own and develop its own understanding of the data set.

Reinforcement Learning:

Reinforcement learning is used to teach a machine to complete a multi-step process for which there are clearly defined rules. Data scientists program an algorithm to complete a task and give it positive or negative clues as it works out how to complete a task. But for the most part, the algorithm decides on its own what steps to take along the way.

1.2.2 Benefits of Machine Learning:

- Easily identifies trends and patterns
- Automation
- Continuous Improvement
- Wide Applications
- Handling multi-dimensional and multi-variety data

1.2.3 Machine Learning Algorithms:

- Linear Regression
- Random Forest

Linear Regression:

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used.

Random Forest:

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

1.3 PYTHON:

Python is an object-oriented programming language, created by Guido van Rossum and released in 1991. Python is primarily a programming language but is used in the field of data due to its versatility of functionalities from a mathematical and statistical perspective. Along with this, its processing speed, accessibility and syntax have made it super popular. It is not only a free software, but it also allows users to develop their own packages and libraries that others can reuse. Python provides a lot of modules for data visualization which offers lot of charts for visualizing the data beautifully.

Machine Learning Libraries used in this Project:

1.2.1 NumPy:

NumPy is a library used with python programming language to convert python arrays into numpy arrays, which are useful for high-performance scientific computing and data analysis.

1.2.2 Pandas:

Pandas is a library written for python programming language to manipulate data. Pandas is useful to read, write, update, delete, and clean the data in datasets for data analysis.

1.2.3 Matplotlib:

Matplotlib is a data visualization library for python programming language used along with numpy. It provides an object-oriented API for embedding plots into applications.

1.2.4 Scikit-learn:

Scikit-learn is a library provided by python for machine learning. Scikit-learn features various classification, regression, and clustering algorithms, including support vector machines, random forests, gradient

boosting, k-means, and DBSCAN, and is designed to work with the Python numerical and scientific libraries NumPy and SciPy.

1.3 FLASK:

Flask is a web application framework written in Python. Flask is used for developing web applications using python, implemented on Werkzeug and Jinja2. It is mostly used to create web applications for machine learning models.

Advantages of Flask:

- Easier to use for simple cases
- Minimal yet powerful platform
- High scalability for simple applications
- Codebase size is relatively smaller
- Higher compatibility with latest technologies

1.4 HTML:

HTML is a standard markup language for creating web pages and web applications. It is widely used language on the web. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

1.5 CSS:

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside

HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting. Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device. The name *cascading* comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable. The CSS specifications are maintained by the World Wide Web Consortium (W3C).

1.6 JavaScript:

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform. Over 97% of websites use JavaScript on the client side for webpage behaviour, often incorporating third-party libraries. JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. JavaScipt is used to provide functionality to the Website.

CHAPTER 2

AIM AND SCOPE OF THE PRESENT INVESTIGATION

1.1 AIM OF THE PROJECT:

To make a model which can give us a good prediction on the rents of house based on different variables. Our aim on this dataset is to achieve an accuracy score of 80% +.

1.2 OBJECTIVE OF THE PROJECT:

Renting a house in a big city like Hyderabad is an hectic process. As the customer has to roam places and has to pay commission to the real estate agent. There is no guarantee that the customer gets insights according to his specifications. So, we are creating this house rent predictor through which the customer can predict the house rents by providing the specifications he/she wants.

1.3 SCOPE OF THE PROJECT:

The main scope of this prediction is to analyze the Hyderabad House Rents dataset using machine learning libraries to make some useful predictions. These analyses will give deeper insight into the house rents of Hyderabad city and helps people to analyze the rents based on different parameters.

We are going to achieve the objective by performing various processes like cleaning, analyzing, feature engineering, outliner detection, one hot encoding the data. Then the data is split into training and testing data sets and apply machine learning algorithms to predict the house rent.

CHAPTER 3

MATERIALS AND METHODS USED

3.1 HARDWARE AND SOFTWARE REQUIREMENTS:

3.1.1 HARDWARE REQUIREMENTS:

- x86 64-bit System
- 4 GB RAM
- 5 GB free disk space

3.1.2 SOFTWARE REQUIREMENTS:

- Python 3.0 or above
- Jupyter Notebook
- Visual Studio Code

3.2 LIBRARIES USED:

- Numpy
- Pandas
- Matplotlib
- Scikit-learn
- Pickle
- Json
- Flask

3.3 MODULES:

- Front-end
- Back-end
- Machine Learning Model

Front-end:

The Front-end module is the interface of the model. It is created using HTML, CSS, JavaScript. The front-end module contains three pages Home page, Predictor page, About page.

Fig 3.1: Home Page

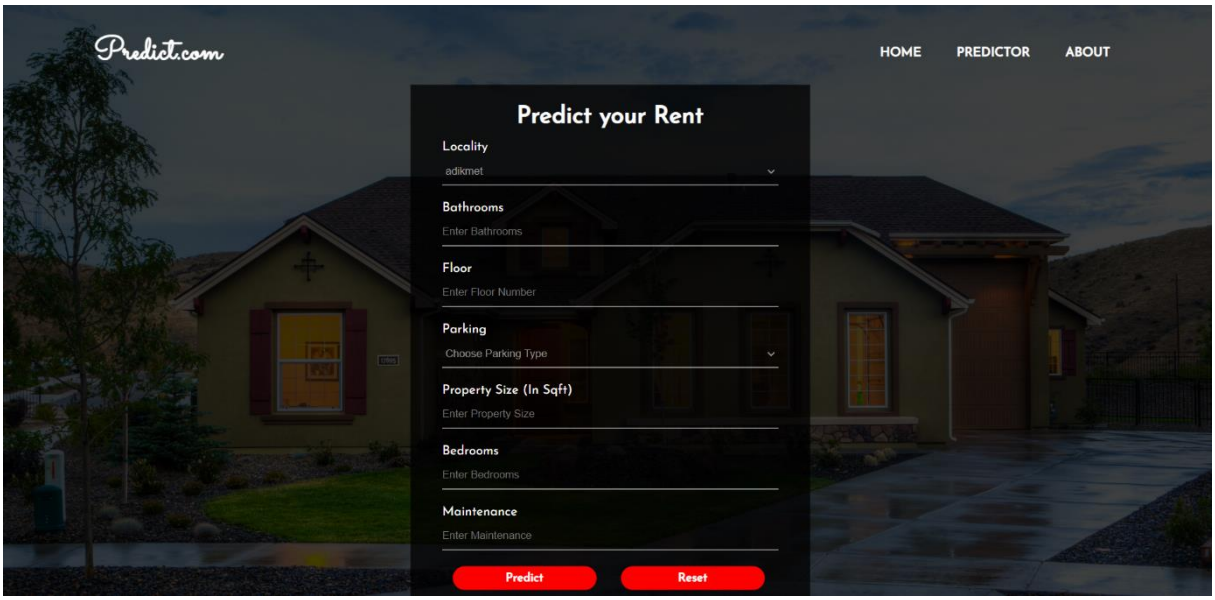

Fig 3.2: Prediction Page

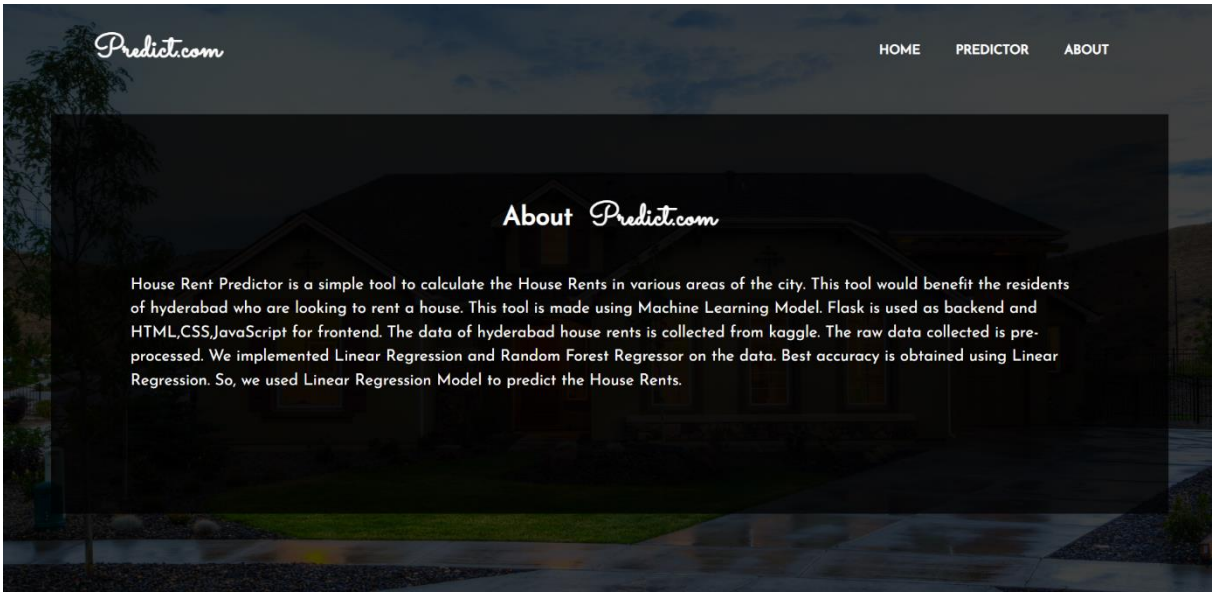

Fig 3.3: About Page

Back-end:

The Back-end is created using Flask framework.

Machine Learning Model:

The machine learning model contains the prediction model that has been trained to predict the house rents.

3.3 METHODOLOGY:

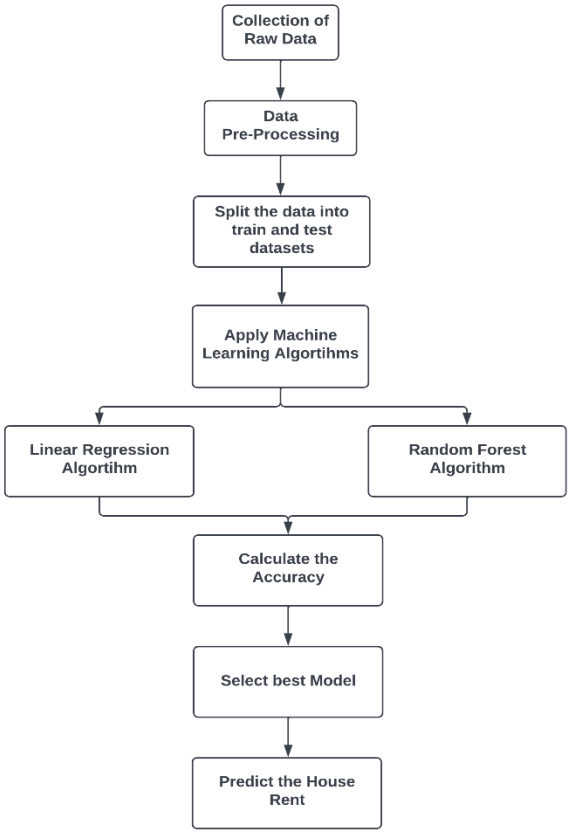

Fig 3.4: House Rent Prediction Methodology

3.3.1 DATA COLLECTION:

The first step of any kind of Machine Learning Project is Data Collection. In order to make any prediction model. We need large amount of data on which we apply various techniques to reach to a particular conclusion. It is advised to take large amount of data because, larger the data greater the accuracy in the result. We have taken the dataset from Kaggle which contains the data of Hyderabad Houses.

3.3.2 DATA PRE-PROCESSING:

The second step after collecting Data is Data Processing. Data

directly obtained from a data source such as dataset is known as Raw data. Raw data can't be used directly for visualization, so the data needs to be processed and converted into useful data. Data-Preprocessing is the process of translating the Raw data into Useful data by conscientiously checking for errors and eliminating redundant, incomplete, or incorrect data. The columns which are not helpful in predicting are removed. If any of the columns contain null values those null values are needed to be omitted or replaced with some valid value like mean of the data in that particular column. After that, feature engineering must performed to transform raw data into features that can be used in supervised learning. Then, Outlier detection must be done on the data. After the outlier detection, One hot encoding must be done to convert categorical variables into a form that could be provided to ML algorithms to perform better.

3.3.3 Training and Testing the Model:

The next step after data pre-processing is creating the ML model for prediction. First, the data must be split into train and test data sets. The train data set contains 80% data and the test dataset 20%. Then it should be fit into the algorithm. We used two algorithms in this project, Linear Regression and Random Forest. We got 86.5 % accuracy with linear regression algorithm and 85.3 % accuracy with random forest algorithm. So, we choose linear regression model to predict the house rents.

CHAPTER 4

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

4.1 RESULTS AND DISCUSSION:

4.1.1 TABLES:

Table 4.1 Hyderabad Houses Dataset

	active	amenities	balconies	bathroom	combineDescription	completeStreetName	deposit	facing	facingDesc	floor	...	p
0	True	{"LIFT":true,"GYM":false,"INTERNET":false,"AC"...	3	3	NaN	Shreya carnation, Block I, NCB Enclave, Gachib...	90000	W	West	3	...	
1	True	{"LIFT":false,"GYM":false,"INTERNET":false,"AC...	1	2	NaN	Inner Ring Rd, near RTO Bandlaguda South Zone	45000	E	East	2	...	
2	True	{"LIFT":true,"GYM":true,"INTERNET":false,"AC":...	3	3	NaN	Rd Number 2, Shirdi Sai Nagar, Manikonda, Hyde...	80000	E	East	0	...	
3	True	{"LIFT":false,"GYM":false,"INTERNET":false,"AC...	1	2	NaN	Plot No. 44, Road No. 1/A, kakatiya colony, LB...	18000	W	West	2	...	
4	True	{"LIFT":true,"GYM":false,"INTERNET":false,"AC"...	2	2	NaN	Madhapur HUDA Techno Enclave, Near MaxCure Su...	80000	E	East	2	...	

5 rows × 36 columns

floor	...	property_size	reactivationSource	rent_amount	sharedAccomodation	shortUrl	swimmingPool	totalFloor	type_bhk	waterSupply	weight
3	...	2200	USER_DASHBOARD	28000	False	http://nobr.kr/sv/9Kvqj	False	5	BHK3	CORP_BORE	NaN
2	...	1200	USER_DASHBOARD	15000	False	http://nobr.kr/sv/j2pD6	False	2	BHK3	BOREWELL	NaN
0	...	1800	MISSED_CALL	16000	False	http://nobr.kr/sv/TXJmj	False	3	BHK3	CORP_BORE	NaN
2	...	750	MISSED_CALL	9000	False	http://nobr.kr/sv/FnCsf	False	2	BHK2	CORP_BORE	NaN
2	...	1250	None	32500	False	http://nobr.kr/sv/BQFRb	False	5	BHK2	CORP_BORE	NaN

Table 4.2: Table obtained after removing unnecessary columns and encoding data

	bathroom	deposit	floor	localityId	maintenanceAmount	parking	property_age	property_size	rent_amount	totalFloor	type_bhk
0	3	90000	3	gachibowli	2000	3	5	2200	28000	5	3.0
1	2	45000	2	chandrayangutta	0	1	1	1200	15000	2	3.0
2	3	80000	0	manikonda	1000	3	0	1800	16000	3	3.0
3	2	18000	2	other	500	3	0	750	9000	2	2.0
4	2	80000	2	hitech_city	2000	3	5	1250	32500	5	2.0

Table 4.3: Creating new column with name maintenance

	bathroom	deposit	floor	localityId	maintenanceAmount	parking	property_age	property_size	rent_amount	totalFloor	type_bhk	maintenance
0	3	90000	3	gachibowli	2000	3	5	2200	28000	5	3.0	2000
1	2	45000	2	chandrayangutta	0	1	1	1200	15000	2	3.0	0
2	3	80000	0	manikonda	1000	3	0	1800	16000	3	3.0	1000
3	2	18000	2	other	500	3	0	750	9000	2	2.0	500
4	2	80000	2	hitech_city	2000	3	5	1250	32500	5	2.0	2000

Table 4.4: Table obtained after removing maintenanceAmount, deposit, property_age, totalFloor columns

	bathroom	floor	localityId	parking	property_size	rent_amount	type_bhk	maintenance
0	3	3	gachibowli	3	2200	28000	3.0	2000
1	2	2	chandrayangutta	1	1200	15000	3.0	0
2	3	0	manikonda	3	1800	16000	3.0	1000
3	2	2	other	3	750	9000	2.0	500
4	2	2	hitech_city	3	1250	32500	2.0	2000

Table 4.5: Added new column rent_per_sqft by calculating

	bathroom	floor	localityId	parking	property_size	rent_amount	type_bhk	maintenance	rent_per_sqft
0	3	3	gachibowli	3	2200	28000	3.0	2000	12.727273
1	2	2	chandrayangutta	1	1200	15000	3.0	0	12.500000
2	3	0	manikonda	3	1800	16000	3.0	1000	8.888889
3	2	2	other	3	750	9000	2.0	500	12.000000
4	2	2	hitech_city	3	1250	32500	2.0	2000	26.000000

Table 4.6: Outlier Detection

	bathroom	floor	localityId	parking	property_size	rent_amount	type_bhk	maintenance	rent_per_sqft
582	1	0	serilingampally	1	100	5000	0.5	0	50.000000
647	1	1	miyapur	1	144	5500	0.5	0	38.194444
706	1	1	other	0	100	4000	0.5	0	40.000000
719	2	6	other	3	0	25000	2.0	2500	inf
734	1	1	other	3	120	10000	0.5	1000	83.333333
...	...	...	...	...	...	...	...	...	...
18845	1	5	hafeezpet	3	500	8000	2.0	720	16.000000
18846	1	0	padmarao_nagar	1	250	5000	1.0	500	20.000000
18937	1	2	ramachandra_puram	1	200	8000	1.0	0	40.000000
18977	2	1	nallagandla	3	400	15000	2.0	550	37.500000
19020	2	1	other	3	0	12000	3.0	0	inf

349 rows × 9 columns

Table 4.7: Table showing data where bathrooms are greater than the bedrooms

	bathroom	floor	localityId	parking	property_size	rent_amount	type_bhk	maintenance	rent_per_sqft
1101	7	0	borabanda	1	3250	40000	5.0	0	12.307692
1212	7	0	east_marredpally	3	6000	65000	5.0	2500	10.833333
1252	2	3	erragadda	1	500	8000	0.5	0	16.000000
1602	8	0	gachibowli	3	4500	100000	5.0	0	22.222222
1736	6	0	gopanapalli	3	2250	32000	4.0	0	14.222222
2495	8	0	himayath_nagar	3	3000	60000	5.0	0	20.000000
2510	2	1	himayath_nagar	1	400	10000	0.5	0	25.000000
3300	6	0	jubilee_hills	3	3400	65000	4.0	5000	19.117647
3484	5	0	karkhana	3	3600	36000	3.0	0	10.000000
3601	2	0	kismatpur	1	300	3500	0.5	500	11.666667
4893	2	4	kothaguda	1	450	9000	0.5	500	20.000000
6669	6	4	manikonda	3	1140	17500	2.0	1500	15.350877
6889	2	3	mehdipatnam	1	375	7000	0.5	500	18.666667
7896	2	1	nallakunta	1	300	3500	0.5	0	11.666667
7977	5	5	nanakram_guda	3	2140	50000	3.0	7000	23.364486
8030	5	3	nanakram_guda	3	2800	58000	3.0	7000	20.714286
8943	7	0	other	1	3600	36000	5.0	2000	10.000000
9246	5	0	other	3	2750	35000	3.0	6000	12.727273
9254	3	2	other	0	400	7000	1.0	100	17.500000
9276	2	0	other	1	400	3500	0.5	0	8.750000
9795	8	0	other	3	3800	40000	5.0	0	10.526316
10206	5	4	other	3	2680	45000	3.0	7000	16.791045
10610	7	0	other	3	6200	55000	4.0	4000	8.870968
10696	2	1	other	3	1150	22000	0.5	0	19.130435
10752	5	10	other	2	3265	60000	3.0	8000	18.376723
11440	5	8	other	3	2800	40000	3.0	7000	14.285714
12426	2	0	quthbullapur	1	300	3000	0.5	0	10.000000
13226	6	0	sainikpuri	3	4000	50000	4.0	0	12.500000
13435	2	0	saroornagar	3	550	5000	0.5	0	9.090909
14472	4	0	upparpally	3	2500	25000	2.0	1000	10.000000

Table 4.8: Table obtained after dropping rent_per_sqft

	bathroom	floor	localityId	parking	property_size	rent_amount	type_bhk	maintenance
0	2	2	adikmet	1	900	10000	2.0	0
1	3	0	adikmet	3	1780	30000	3.0	3560
2	2	2	adikmet	1	680	10000	2.0	0
3	2	3	adikmet	1	1050	12000	2.0	1200
4	1	2	adikmet	1	600	9500	2.0	0

Table 4.9: Table showing dummies

	adikmet	almasguda	amberpet	ameerpet	attapur	badangpet	badesahebguda	balakampet	balanagar	balapur	...	turkayamjal	uppal	upparpally	venkat_n
0	1	0	0	0	0	0	0	0	0	0	...	0	0	0	
1	1	0	0	0	0	0	0	0	0	0	...	0	0	0	
2	1	0	0	0	0	0	0	0	0	0	...	0	0	0	
3	1	0	0	0	0	0	0	0	0	0	...	0	0	0	
4	1	0	0	0	0	0	0	0	0	0	...	0	0	0	

5 rows × 150 columns

alanagar	balapur	...	turkayamjal	uppal	upparpally	venkat_nagar_colony	west_marredpally	whisper_valley	yapral	yella_reddy_guda	yousufguda	zamistanpur
0	0	...	0	0	0	0	0	0	0	0	0	0
0	0	...	0	0	0	0	0	0	0	0	0	0
0	0	...	0	0	0	0	0	0	0	0	0	0
0	0	...	0	0	0	0	0	0	0	0	0	0
0	0	...	0	0	0	0	0	0	0	0	0	0

Table 4.10: Table obtained after dropping localityid column

	bathroom	floor	parking	property_size	rent_amount	type_bhk	maintenance	adikmet	almasguda	amberpet	...	turkayamjal	uppal	upparpally	venkat_nag
0	2	2	1	900	10000	2.0	0	1	0	0	...	0	0	0	
1	3	0	3	1780	30000	3.0	3560	1	0	0	...	0	0	0	
2	2	2	1	680	10000	2.0	0	1	0	0	...	0	0	0	
3	2	3	1	1050	12000	2.0	1200	1	0	0	...	0	0	0	
4	1	2	1	600	9500	2.0	0	1	0	0	...	0	0	0	

5 rows × 156 columns

ısguda	amberpet	...	turkayamjal	uppal	upparpally	venkat_nagar_colony	west_marredpally	whisper_valley	yapral	yella_reddy_guda	yousufguda	zamistanpur
0	0	...	0	0	0	0	0	0	0	0	0	0
0	0	...	0	0	0	0	0	0	0	0	0	0
0	0	...	0	0	0	0	0	0	0	0	0	0
0	0	...	0	0	0	0	0	0	0	0	0	0
0	0	...	0	0	0	0	0	0	0	0	0	0

Table 4.11: Table obtained after dropping rent_amount

	bathroom	floor	parking	property_size	type_bhk	maintenance	adikmet	almasguda	amberpet	ameerpet	...	turkayamjal	uppal	upparpally	venkat_na
0	2	2	1	900	2.0	0	1	0	0	0	...	0	0	0	
1	3	0	3	1780	3.0	3560	1	0	0	0	...	0	0	0	
2	2	2	1	680	2.0	0	1	0	0	0	...	0	0	0	
3	2	3	1	1050	2.0	1200	1	0	0	0	...	0	0	0	
4	1	2	1	600	2.0	0	1	0	0	0	...	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
14814	2	1	3	500	1.0	0	0	0	0	0	...	0	0	0	
14815	2	4	3	1000	2.0	1000	0	0	0	0	...	0	0	0	
14816	1	2	3	720	2.0	0	0	0	0	0	...	0	0	0	
14817	1	2	3	500	1.0	0	0	0	0	0	...	0	0	0	
14818	2	3	1	1010	2.0	1000	0	0	0	0	...	0	0	0	

14789 rows × 155 columns

iberpet	ameerpet	...	turkayamjal	uppal	upparpally	venkat_nagar_colony	west_marredpally	whisper_valley	yapral	yella_reddy_guda	yousufguda	zamistanpur
0	0	...	0	0	0	0	0	0	0	0	0	0
0	0	...	0	0	0	0	0	0	0	0	0	0
0	0	...	0	0	0	0	0	0	0	0	0	0
0	0	...	0	0	0	0	0	0	0	0	0	0
0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...
0	0	...	0	0	0	0	0	0	0	0	0	1
0	0	...	0	0	0	0	0	0	0	0	0	1
0	0	...	0	0	0	0	0	0	0	0	0	1
0	0	...	0	0	0	0	0	0	0	0	0	1
0	0	...	0	0	0	0	0	0	0	0	0	1

4.1.2 FIGURES:

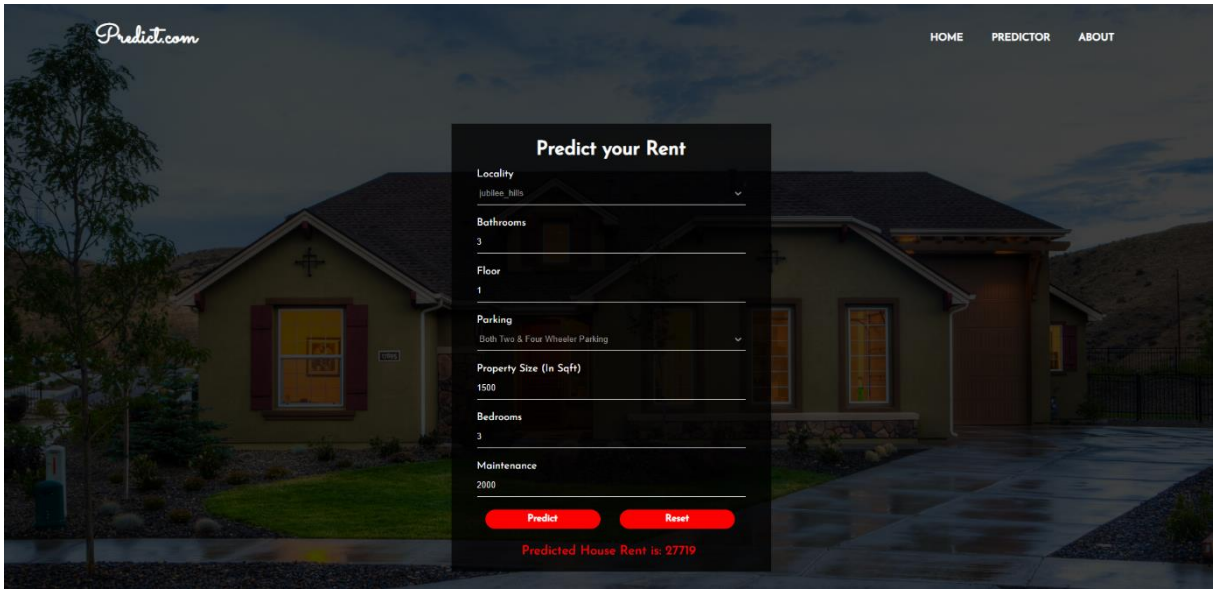

Fig 4.1: Rent Prediction for Jubilee Hills House

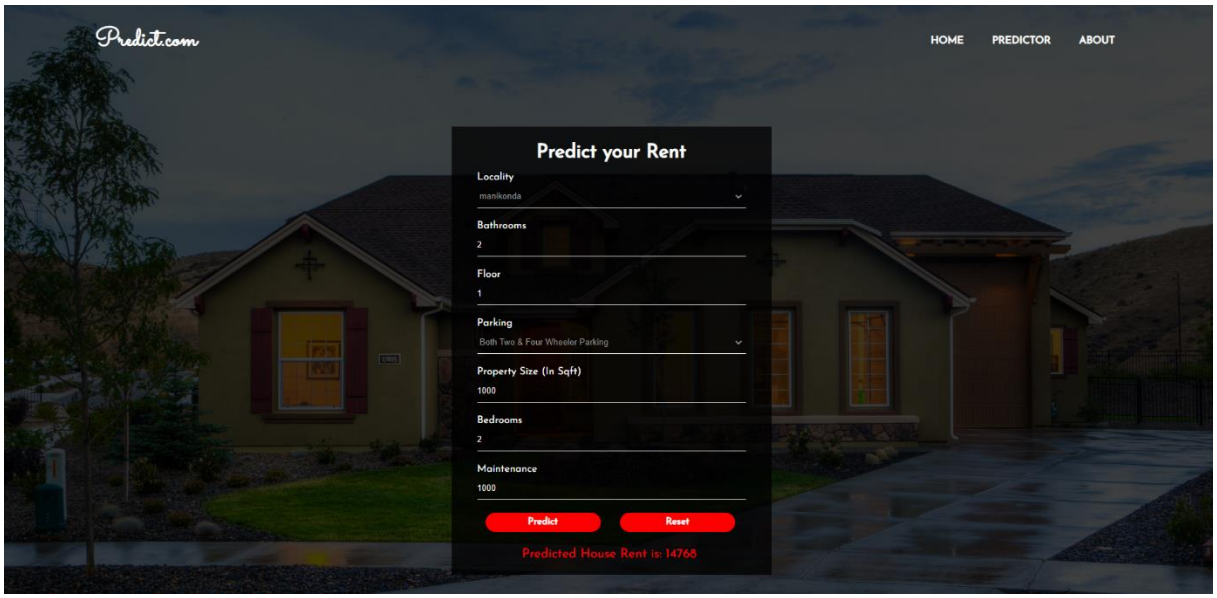

Fig 4.2: Rent Prediction for Manikonda House

4.2: ANALYSIS AND INTERPRETATION OF RESULTS:

Inference from the Analysis:

1. Accuracy of Linear Regression Model: 86.5%
2. Accuracy of Random Forest Model: 85.3%
3. Predicted House Rent for Jubilee Hills area with 3 bathrooms, 1st floor, both 2 & 4 wheeler parking, 1500 Sqft, 3 bedrooms, 2000 maintenance cost is:27,719
4. Predicted House Rent for Manikonda area with 2 bathrooms, 1st floor, both 2 & 4 wheeler parking, 1000 Sqft, 2 bedrooms, 1000 maintenance cost is: 14,768

CHAPTER 5

CONCLUSION AND FUTURE WORK

CONCLUSION:

- We implemented Linear Regression and Random Forest algorithms in this project. On this dataset, Linear Regression performed well.
- As the result of Prediction, we can conclude that Locality of the house and Property Size plays a major role in deciding the house rent.
- Also, increase in number of bedrooms increases the rent of the house.
- Parking and Floor have less impact on the house rent.

FUTURE WORK:

- In this Project, we had applied only Linear Regression and Random Forest Algorithms. We can apply more algorithms and can create a better Predictive Model.
- we have taken only the data of Hyderabad city. In future we can take data of many other cities will which help users from many locations to predict house rents.

REFERENCES

[1] Documentation and User Guides of Modules used in this project:

Python: https://docs.python.org/3.9/

NumPy: https://numpy.org/doc/stable/numpy-user.pdf

Pandas: https://pandas.pydata.org/docs/user_guide/index.html#user-guide

Matplotlib: https://matplotlib.org/stable/tutorials/index.html

Scikit-learn:

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Flask: https://flask.palletsprojects.com/en/2.1.x/

[2] YouTube Tutorials :

Data Visualization Tutorials:

https://www.youtube.com/watch?v=CmorAWRsCAw&list=PLeo1K3hjS3uuASpe-1LjfG5f14Bnozjwy

https://www.youtube.com/watch?v=rN0TREj8G7U&list=PLUcmakntVocWGSKXIsUn1J7Wm9ekpZ87G

https://www.youtube.com/watch?v=E5RjzSK0fvY

https://www.youtube.com/watch?v=ok2s1vV9XW0

https://www.youtube.com/watch?v=Z1RJmh_OqeA

APPENDIX

A. SOURCE CODE:

```
import numpy as np
import pandas as pd
df=pd.read_csv('hyd_house.csv')
df.head()
df.shape
df.columns
df1=df.drop(['amenities','locality','balconies','lift','active','loanAvailable','location','owne
                    rName',

                    'parkingDesc','propertyTitle','propertyType','combineDescription','co
                    mpleteStreetName','facing',

                    'facingDesc','furnishingDesc','gym','id','isMaintenance','weight','wat
                    erSupply','swimmingPool',
                  'shortUrl','sharedAccomodation','reactivationSource'],axis=1)
print(df1.parking.value_counts())
print(df1.bathroom.value_counts())
print(df1.floor.value_counts())
print(df1.type_bhk.value_counts())
print(df1.maintenanceAmount.value_counts())
df1.replace({'parking':{'NONE':0,'TWO_WHEELER':1,'FOUR_WHEELER':2,'BOTH':3
                    }},inplace=True)
df1.replace({'type_bhk':{'RK1':0.5,'BHK1':1,'BHK2':2,'BHK3':3,'BHK4':4,'BHK4PLUS':
                    5}},inplace=True)
df1.replace({'maintenanceAmount':{'None':int(0)}},inplace=True)
df1["localityId"].replace("NOT_FOUND", "other",inplace=True)
df1.head()
```

```
df1.info()
df1.isnull().sum()
df2=df1.copy()
df2['localityId'].value_counts()
df2['maintenance']=df2['maintenanceAmount'].apply(lambda x:int(x))
df2.head()
df3=df2.drop(['maintenanceAmount','deposit','property_age','totalFloor'],axis=1)
df3.head()
df3.info()
df4=df3.copy()
df4['rent_per_sqft']=df4['rent_amount']/df4['property_size']
df4.head()
location_stats=df4.groupby('localityId')['localityId'].agg('count').sort_values(ascending
                    =False)
location_stats
len(location_stats[location_stats<=10])
location_less_than_10=location_stats[location_stats<=10]
location_less_than_10
df4.localityId=df4.localityId.apply(lambda x:'other' if x  in location_less_than_10 else
                    x)
df4['localityId'].value_counts()
location_less_than_10
df4[df4.property_size/df4.type_bhk<300]
df4.shape
df5=df4[~(df4.property_size/df4.type_bhk<300)]
df5.shape
df5.rent_per_sqft.describe()
def remove_pps_outliers (df):
    df_out = pd.DataFrame()
    for key,subdf in df.groupby('localityId'):
        m=np.mean(subdf.rent_per_sqft)
```

```
        st=np.std(subdf.rent_per_sqft)
        reduced_df=subdf[(subdf.rent_per_sqft>(m-st))&(subdf.rent_per_sqft<=(m+st))]
        df_out=pd.concat([df_out,reduced_df],ignore_index=True)
    return df_out
df6=remove_pps_outliers(df5)
df6.shape
import matplotlib as mpl
from matplotlib import pyplot as plt
get_ipython().run_line_magic('matplotlib', 'inline')
mpl.rcParams['figure.figsize']=(20,10)
plt.hist(df6.rent_per_sqft,rwidth=0.8)
plt.xlabel('price per sqft')
plt.ylabel('count')
plt.title('distribution of data')
df6.bathroom.unique()
df6[df6.bathroom>df6.type_bhk+1]
df7=df6[df6.bathroom<=df6.type_bhk+1]
df7.shape
df8=df7.drop('rent_per_sqft',axis=1)
df8.head()
dummies=pd.get_dummies(df8.localityId)
dummies.head()
df9=pd.concat([df8.drop('localityId',axis='columns'),dummies.drop('other',axis='colum
                        ns')],axis='columns')
df9.head()
df9.shape
X=df9.drop('rent_amount',axis='columns')
X
Y=df9['rent_amount']
Y
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=LinearRegression()
lr.fit(X_train,Y_train)
lr.score(X_test,Y_test)
y_pred = lr.predict(X_test)
from sklearn.metrics import r2_score
r2_score(Y_test, y_pred)
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)
cross_val_score(LinearRegression (),X,Y,cv=cv)
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor()
rf.fit(X_train, Y_train)
y_pred = rf.predict(X_test)
from sklearn.metrics import r2_score
r2_score(Y_test, y_pred)
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)
cross_val_score(RandomForestRegressor (),X,Y,cv=cv)
X.columns
def
                    predict_rent(localityId,bathroom,floor,parking,property_size,type_b
                    hk,maintenance):

    loc_index = np.where(X.columns==localityId)[0][0]
    x= np.zeros(len(X.columns))
    x[0]=bathroom
    x[1]=floor
```

```
    x[2]=parking
    x[3]=property_size
    x[4]=type_bhk
    x[5]=maintenance

    if loc_index>=0:
        x[loc_index]=1

    return lr.predict([x])[0]
predict_rent('yousufguda',4,2,2,1000,3,1000)
import pickle
with open('rent_prediction.pickle','wb') as f:
    pickle.dump(lr,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))
```