A
Minor Project Report on
"Chennai House Price Prediction"

In partial fulfillment of requirements for the degree of

Bachelor of Technology (B.Tech.)

in

Computer Science and Engineering

Submitted by

Ms. Shreshtha(170388)

Under the Guidance of

Mr. Siddhanta Kumar Singh

Computer Science and Engineering

SCHOOL OF ENGINEERING AND TECHNOLOGY

Mody University and Science and Technology
Lakshmangarh, Distt. Sikar-332311

December 2020

A C K N O W L E D G E M E N T

I sincerely express my gratitude to my project mentor “Mr. Siddhanta Kumar Singh” for his benevolent guidance in completing the project report on “Chennai House Price Prediction”. His kindness and help have been the source of encouragement for me. I am grateful to him for the guidance, inspiration and constructive suggestions that helped me in the preparation of this project. I would also like to take this golden opportunity to thank Dr. A. Senthil(Dean and HOD(CSE)- SET, Mody University of Science and Technology) for providing us with this opportunity to speak on technical topics of relevance to us and the society at large. I would also like to thank my fellow mates for attending and listening to our deliverance with patience and providing valuable inputs.

Shreshtha

CERTIFICATE

This is to certify that the minor project report entitled “Chennai House Price Prediction” submitted by Ms. Shreshtha , as a partial fulfillment for the requirement of B. Tech.VII Semester examination of the School of Engineering and Technology, Mody University of Science and Technology, Lakshmangarh for the academic session 2019-2020 is an original project work carried out under the supervision and guidance of Mr. Siddhanta Kumar Singh has undergone the requisite duration as prescribed by the institution for the project work.

PROJECT GUIDE:

Approval Code: AUT_20_CSE_F19_03

Name: Mr.Siddhanta Kumar Singh

Date: 27/12/2020

HEAD OF DEPARTMENT

Signature:

Name:Dr.A.Senthil

Date: 27/12/2020

EXAMINER-I:

Name: Dr.Sunil Kumar Jangir

Dept: CSE

EXAMINER-II

Name: Dr.Ajay Kumar Singh

Dept: CSE

ABSTRACT

Machine learning plays a major role from past years in image detection, spam reorganization, normal speech command, product recommendation and medical diagnosis. Present machine learning algorithm helps us in enhancing security alerts, ensuring public safety and improve medical enhancements. Machine learning system also provides better customer service and safer automobile systems. In this project, I made the Machine Learning model for prediction of future housing prices that is generated by machine learning algorithm. For the selection of prediction methods , I compare and explore various prediction methods. More over on other hand housing value indices, the advancement of a housing cost prediction that tend to the advancement of real estate policies schemes. This project utilizes machine learning algorithms that develops housing price prediction models.

I in that point recommend a housing cost prediction model to support a house vender or a real estate agent for better information based on the valuation of house. The evaluations exhibit that Random Forest algorithm, in view of accuracy, reliably outperform alternate models in the execution of housing cost prediction.

Table of Contents

LIST OF FIGURES

Chapter1:Introduction

1. INTRODUCTION

Around the globe, the land area is presented to wide variances in costs on account of existing relationships with endless highlights, either known or obscure. In light of the market and non-market chances, these housing costs can increment or abatement at different rate in a given timeframe. In this project, it is aimed towards building up a Machine Learning model that would recognize better value forecasts for housing properties of Chennai to encourage the purchasers to make commendable interest in the land properties.

The housing sector is the second biggest work supplier with the level of urbanization of 33.54 percent after horticulture area in India and is relied upon to contribute 13 percent of the nation's GDP by 2025. 10 million individuals relocate to urban areas consistently and 35 percent of the populace is in youthful age gathering (15–35 years). Equivocalness among the costs of houses makes it hard for the purchaser to choose their fantasy house thus giving a conclusive lodging value expectation model to profit a purchaser and vender or a real state agent to settle on a superior educated choice.

In this project, an endeavor has been made to foresee conclusive housing price at Chennai, the capital city of Tamil Nadu, India to encourage the purchasers settle on potential choices dependent on the expectations made.

1.1 PRESENT SYSTEM

The current framework isn't dunce proof and has certain downsides. Being a manual framework the potential constraints and loopholes in the current framework is enormous. Some of them are:-

1. Human asset: - The current framework has a lot of manual work from filling a structure to recording a report, conveying proclamation. This expands trouble on laborers yet doesn't yield the outcomes it should.

2. Prickly Job: - In current framework if any adjustment is to be made it builds manual work and blunder is inclined.

3. Error: - As the framework is overseen and kept up by laborers blunders are a portion of the conceivable outcomes.

1.2 PROPOSED SYSTEM

These days, e-schooling and e-learning is profoundly affected. Everything is moving from manual to automated frameworks.

The goal of this project is to anticipate the house costs in order to limit the issues looked by the client. The present strategy is that the client moves toward a real state agent to deal with his/her speculations and recommend reasonable bequests for his ventures. Be that as it may, this strategy is dangerous as the agent may anticipate wrong homes and accordingly prompting loss of the client' ventures. The manual technique which is as of now utilized in the market is out dated and has high danger. To conquer this flaw, there is a requirement for a refreshed and automated framework. Machine Learning algorithms can be used to help investors to invest in an appropriate estate according to their mentioned requirements.

Chapter2: System Design

Machine learning offers a set of algorithms which allow the outcome prediction process to become more accurate without explicitly programming the software applications. ML aims at building algorithms which are capable of applying statistical analysis on received input in order to predict some output, along with being able to update the outputs when new data is made available to it.ML involves processes similar to those used in predictive modeling and data mining- those which require exploring through data in order to identify some patterns and then accordingly adjusting the actions of the program. ML techniques can be broadly categorized as:

1. Supervised Learning- Here, the learning cycle is guided. The accessible dataset is utilized to prepare the fabricated model or machine. When prepared, it can settle on forecasts or take choices when any new information is contribution to it.

2. Unsupervised Learning- Here, the model studies via observations and notes the formations in the statistics. Once a dataset is provided to the model, it automatically learns samples and relationships in the data by creating clusters out of it. For instance, if images of bananas, mangoes, and apples are presented to the model, it creates clusters of the dataset based on some relationships and patterns, and segregates the images into those clusters. So, when new images are fed to the trained model, it can add it to one of the formed clusters.

3. Reinforcement Learning- It alludes to a specialist's capability to interrelate with its general climate and find out about the most ideal result accessible. It goes around with the hit-and-preliminary hypothesis wherein the specialist's either remunerated or punished with a point for each right or wrong answer separately. At that point, premise the positive prize focuses acquired, the model guides itself. When it gets skilled, it gets prepared to foresee the new information given to it.

2.1 SYSTEM FLOWCHART

1. **Collect data**- Here, we research and get information that we use to take care of our machine. The quality and amount of data we get are significant since it will straightforwardly affect how well or severely our model will work. We may have the data in a current information base or we should make it without any preparation. It is likewise basic to utilize the web scratching strategy to consequently gather data from different sources, for example, APIs.

2. **Prepare data** - This is a decent and ideal opportunity to picture our information and check if there are connections between the various features that we acquired. It will be important to cause a determination of features since the ones we pick will straightforwardly affect the execution times and the outcomes. We can likewise lessen measurements by applying PCA if important.

   Also, we should adjust the measure of information we have for each outcome - class-with the goal that it is critical as the learning might be one-sided towards a kind of reaction and when our model attempts to sum up information it will fall flat.

   We likewise separate the information into two gatherings: one for preparing and the other for model assessment which can be partitioned around in a proportion of 80/20 yet it can shift contingent upon the case and the volume of information we have.

   At this stage, we likewise pre-measure our information by normalizing, wiping out duplicates.

3. **Choose the right model** - There are several models that we can choose according to the objective that we might have: we will use algorithms of classification, prediction, linear regression, clustering i.e. k-means or K-Nearest Neighbor, Deep Learning, Neural Networks, Bayesian, etc. There are various models to be used depending on the data we are going to process such as images, sound, text, and numerical values.

4. **Train our machine model** - We need to train the datasets to run easily and see a steady improvement in the forecast rate.

5. **Model evaluation** - We need to check the machine made against our evaluation dataset that contains inputs that the model doesn't have the foggiest idea and confirm the accuracy of our all around prepared model. In the event that the precision is not exactly or equivalent to half, that model won't be helpful since it would resemble flipping a coin to decide. In the event that we arrive at 90% or more, we can have

great trust in the outcomes that the model gives us.

6. **Prediction**- We are now ready to use our Machine Learning model providing results in real-life scenarios.
7. **Deployment-** where business-usable results of the ML process — such as models or insights are deployed to enterprise applications, systems or data stores (for example, for reporting)

2.2 DATASET

Dataset comprises of sale data of the individual private lofts/houses in Chennai. The informational collection comprises of both ceaseless and discrete factors adding up to 7109 records with 19 features.

Data pre-processing, Cleansing, information arrangement and exploratory information investigation are the normal strategies utilized prior to actualizing different ML algorithms.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	PRT_ID	AREA	INT_SQFT	DIST_MAI	N_BEDRO	N_BATHR	N_ROOM	SALE_CON	PARK_FAC	BUILDTYPI	UTILITY_A	STREET	MZZONE	QS_ROOM	QS_BATHF	QS_BEDRC	QS_OVER/	COMMIS	SALES_PRICE
2	P03210	Karapakka	1004	131	1	1	3	AbNormal	Yes	Commerci	AllPub	Paved	A	4	3.9	4.9	4.33	144400	7600000
3	P09411	Anna Nag	1986	26	2	1	5	AbNormal	No	Commerci	AllPub	Gravel	RH	4.9	4.2	2.5	3.765	304049	21717770
4	P01812	Adyar	909	70	1	1	3	AbNormal	Yes	Commerci	ELO	Gravel	RL	4.1	3.8	2.2	3.09	92114	13159200
5	P05346	Velachery	1855	14	3	2	5	Family	No	Others	NoSewr	Paved	I	4.7	3.9	3.6	4.01	77042	9630290
6	P06210	Karapakka	1226	84	1	1	3	AbNormal	Yes	Others	AllPub	Gravel	C	3	2.5	4.1	3.29	74063	7406250
7	P00219	Chrompet	1220	36	2	1	4	Partial	No	Commerci	NoSeWa	No Access	RH	4.5	2.6	3.1	3.32	198316	12394750
8	P09105	Chrompet	1167	137	1	1	3	Partial	No	Other	AllPub	No Access	RL	3.6	2.1	2.5	2.67	33955	8488790
9	P09679	Velachery	1847	176	3	2	5	Family	No	Commerci	AllPub	Gravel	RM	2.4	4.5	2.1	3.26	235204	16800250
10	P03377	Chrompet	771	175	1	1	2	AdjLand	No	Others	NoSewr	Paved	RM	2.9	3.7	4	3.55	33236	8308970
11	P09623	Velachery	1635	74	2	1	4	AbNormal	No	Others	ELO	No Access	I	3.1	3.1	3.3	3.16	121255	8083650
12	P09540	Chrompet	1203	78	2	1	4	AdjLand	Yes	Commerci	AllPub	No Access	RM	4	3.2	4.5	3.83	119504	14938000
13	P07121	Chrompet	1054	143	1	1	3	Partial	No	Others	NoSewr	Gravel	RM	2.2	3.1	3.3	2.89	141746	9449730

Fig 2.2 Dataset

Chapter3: Hardware and Software Details

HARDWARE REQUIREMENTS:

Intel Core i5 8th Gen Processor 1.60 GHz

Installed RAM 12 GB

SOFTWARE REQUIREMENTS:

Windows 10 Operating System

Jupyter Notebook

Python and its Libraries

Chapter4: Implementation Work Details

4.1 REAL LIFE APPLICATIONS

Housing trend patterns are the worry of purchasers and merchants, however it additionally shows the current financial circumstance. There are numerous components which has sway on house costs, for example, quantities of rooms and washrooms. Indeed, even the close by area, an area with an extraordinary openness to parkways, interstates, schools, shopping centers and nearby work openings adds to the ascent in house cost. Manual house forecast becomes troublesome, subsequently created ML model for house value expectation. This project assists with creating a model which can give us a decent house evaluating expectation dependent on different factors.

4.2 DATA IMPLEMENTATION AND PROGRAM EXECUTION

Regression algorithms are ML algorithms used for predicting continuous numerical target values. They are supervised learning models which means they require labeled training examples.

Use-Cases:-

1. Predicting the suitable price for a product based upon size, brand, and location.

2. Predicting the number of sales each day based on store venue, public holidays, day of the week, and the closest competitor in the market.

Below are the common machine learning algorithms that are used for this project:-

1. **Linear Regression** - It endeavors to fit a straight hyperplane to our dataset that is nearest to all information focuses. It is most reasonable when there are direct connections between the factors in the dataset.

   Advantages:

   1. Snappy to figure and can be refreshed effectively with new information.
   2. Generally straightforward and clarify
   3. Regularization methods can be utilized to forestall overfitting.

Disadvantages:

1. Incapable to learn complex connections.

2. Difficult to capture non-linear relationships (without first transforming data which can be complicated).

2. **Decision Trees**: They figure out how to best part the dataset into isolated branches, permitting it to learn non-linear relationships. Decision tree builds regression model in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed**.**
3. **Random Forests** : Random Forests and Gradient Boosted Trees (GBT) are two algorithms that construct numerous individual trees, pooling their expectations. As they utilize an assortment of results to settle on a ultimate conclusion, they are referred to as "Ensemble techniques".
   Firstly, there is the n_estimators hyperparameter , which is just the number of trees the algorithm builds before taking the maximum voting or taking averages of predictions. In general, a higher number of trees increases the performance and makes the predictions more stable, but it also slows down the computation.

Advantages:

1. A single decision tree is fast to train.

2. Robust to noise and missing values.

3. RF performs very well "out-of-the-box".

Disadvantages:

1. Single decision trees are prone to overfitting.
2. Complex trees are hard to interpret.

4. **KNN** - Regression based on k-nearest neighbors.

The target is predicted by local interpolation of the targets associated of the nearest neighbors in the training set.

5. **Ridge Regression** - Ridge Regression is a technique for analyzing multiple regression data that suffer from multicollinearity.
6. **Lasso Regression** - Lasso regression is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters).

Grid Search

To search for the best hyper-parameters for your algorithm and data, grid search cross validation is commonly used.

The various metrics that can be used to evaluate the results of the prediction are given below:

1. Mean Squared Error: MSE or Mean Squared Error is perhaps the most favored measurements for relapse errands. It is basically the normal of the squared contrast between the objective worth and the worth anticipated by the relapse model. As it squares the distinctions, it punishes even a little mistake which prompts over-assessment of how terrible the model is. It is favored more than different measurements since it is differentiable and thus can be advanced better.
2. Root Mean Squared Error: RMSE is the most broadly utilized measurement for relapse errands and is the square foundation of the arrived at the midpoint of squared contrast between the objective worth and the worth anticipated by the model. It is favored more at times on the grounds that the mistakes are first squared prior to averaging which represents a high punishment on huge blunders. This suggests that RMSE is valuable when enormous blunders are undesired.
3. Mean Absolute Error: MAE is the total distinction between the objective worth and the worth anticipated by the model. The MAE is more powerful to anomalies and doesn't punish the blunders as very as mse. MAE is a straight score which implies all the individual contrasts are weighted similarly. It isn't reasonable for applications where you need to give more consideration to the exceptions.

4. Root Mean Squared Logarithmic Error (RMSLE) : This is the metric used for our project and a common metric for regression problems. It is an extension on Mean Squared Error (MSE) that is mainly used when predictions have large deviations, which is the case with this prediction problem. Values range from 0 up to millions and we don't want to punish deviations in prediction as much as with MSE.

Chapter5: Source Code

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

df = pd.read_csv("chennai_house_price_prediction.csv")

df.shape

df.head()

df.describe()

df.describe(include='all')

df.isnull().sum()

df.dtypes

temp = pd.DataFrame(index=df.columns)

temp['data_type'] = df.dtypes

temp['null_count'] = df.isnull().sum()

temp['unique_count'] = df.nunique()

temp

df['SALES_PRICE'].plot.hist(bins = 50)

plt.xlabel('Sales', fontsize=12)

(df['SALES_PRICE'].loc[df['SALES_PRICE']<18000000]).plot.hist(bins=50)
```

```
df['INT_SQFT'].plot.hist(bins = 50)

plt.xlabel('Area in sq feet', fontsize=12)

df['N_BEDROOM'].value_counts()

df['N_BEDROOM'].value_counts()/len(df)*100

df['N_ROOM'].value_counts()

df['N_BATHROOM'].value_counts()/len(df)

df['N_BATHROOM'].value_counts().plot(kind = 'bar')

df['AREA'].value_counts().plot(kind = 'bar')

df['PARK_FACIL'].value_counts().plot(kind = 'bar')

df['PARK_FACIL'].value_counts()

df.drop_duplicates()

df.drop_duplicates(subset=['AREA']).shape

df.shape

df.isnull().sum()

df.dropna(axis=0, how='any')

df.dropna(axis=1, how='any')

df['N_BEDROOM'].mode()

df['N_BEDROOM'].fillna(value = (df['N_BEDROOM'].mode()[0]), inplace=True)

df.loc[df['N_BATHROOM'].isnull()==True]

for i in range(0, len(df)):

    if pd.isnull(df['N_BATHROOM'][i])==True:
```

```
 if (df['N_BEDROOM'][i] == 1.0):
        df['N_BATHROOM'][i] = 1.0
    else:
        df['N_BATHROOM'][i] = 2.0
df[[ 'QS_ROOMS','QS_BATHROOM', 'QS_BEDROOM', 'QS_OVERALL']].head()
temp = (df['QS_ROOMS'] + df['QS_BATHROOM'] + df['QS_BEDROOM'])/3
pd.concat([df['QS_ROOMS'], df['QS_BATHROOM'], df['QS_BEDROOM'], temp], axis=1).head(10)
df.loc[df['QS_OVERALL'].isnull()==True].shape
def fill_na(x):
   return ((x['QS_ROOMS'] + x['QS_BATHROOM'] + x['QS_BEDROOM'])/3)
df['QS_OVERALL'] =  df.apply(lambda x: fill_na(x) if pd.isnull(x['QS_OVERALL']) else x['QS_OVERALL'], axis=1)
df.isnull().sum()
df.dtypes
df = df.astype({'N_BEDROOM': 'object', 'N_ROOM': 'object', 'N_BATHROOM': 'object'})
temp = ['AREA','N_BEDROOM','N_BATHROOM','N_ROOM','SALE_COND','PARK_FACIL','BUILDTYPE','UTILITY_AVAIL','STREET','MZZONE']
for i in temp:
   print('********** Value Count in', i, '**********')
   print(df[i].value_counts())
   print('')
```

```
df['PARK_FACIL'].replace({'Noo':'No'}, inplace = True)

df['PARK_FACIL'].value_counts()

df['AREA'].replace({'TNagar':'T Nagar', 'Adyr': 'Adyar', 'KKNagar': 'KK Nagar',

    'Chrompt': 'Chrompet', 'Chormpet': 'Chrompet','Chrmpet': 'Chrompet','Ana Nagar': 'Anna Nagar', 'Ann Nagar': 'Anna Nagar', 'Karapakam': 'Karapakkam' , 'Velchery': 'Velachery'}, inplace = True)

df['AREA'].value_counts()

df['SALE_COND'].replace({'PartiaLl':'Partial', 'Partiall': 'Partial', 'Adj Land': 'AdjLand', 'Ab Normal': 'AbNormal'}, inplace = True)

df['SALE_COND'].value_counts()

df['BUILDTYPE'].replace({'Comercial':'Commercial', 'Other': 'Others'},inplace = True)

df['UTILITY_AVAIL'].replace({'All Pub':'AllPub'},inplace = True)

df['STREET'].replace({'NoAccess':'No Access', 'Pavd':'Paved'},inplace = True)

df.columns

df.plot.scatter('INT_SQFT','SALES_PRICE')

fig, ax = plt.subplots()

colors = {'Commercial':'red', 'House':'blue', 'Others':'green'}

ax.scatter(df['INT_SQFT'], df['SALES_PRICE'], c=df['BUILDTYPE'].apply(lambda x: colors[x]))

plt.show()

df.pivot_table(values='SALES_PRICE',
index='N_BEDROOM',columns='N_BATHROOM', aggfunc='median')

df.plot.scatter('QS_OVERALL', 'SALES_PRICE')
```

```
fig, axs = plt.subplots(2, 2)

fig.set_figheight(10)

fig.set_figwidth(10)


axs[0, 0].scatter(df['QS_BEDROOM'], df['SALES_PRICE'])

axs[0, 0].set_title('QS_BEDROOM')

axs[0, 1].scatter(df['QS_BATHROOM'], df['SALES_PRICE'])

axs[0, 1].set_title('QS_BATHROOM')

axs[1, 0].scatter(df['QS_ROOMS'], df['SALES_PRICE'])

axs[1, 0].set_title('QS_ROOMS')

axs[1, 1].scatter(df['QS_OVERALL'], df['SALES_PRICE'])

axs[1, 1].set_title('QS_OVERALL')

ax = plt.figure().add_subplot(111)

ax.set_title('Quality score for Houses')

# Create the boxplot

bp = ax.boxplot([df['QS_BEDROOM'], df['QS_ROOMS'], df['QS_BATHROOM'], df['QS_OVERALL']])

df.groupby('BUILDTYPE').SALES_PRICE.median()

temp_df = df.loc[(df['BUILDTYPE']=='Commercial')&(df['AREA']=='Anna Nagar')]

temp_df['SALES_PRICE'].plot.hist(bins=50)

temp_df = df.loc[(df['BUILDTYPE']=='House')&(df['AREA']=='Anna Nagar')]

temp_df['SALES_PRICE'].plot.hist(bins=50)
```

```
df.groupby(['BUILDTYPE', 'PARK_FACIL']).SALES_PRICE.median()

temp = df.groupby(['BUILDTYPE', 'PARK_FACIL']).SALES_PRICE.median()

temp.plot(kind = 'bar', stacked = True)

df.pivot_table(values='SALES_PRICE', index='AREA', aggfunc='median')

temp_df = df.loc[(df['AREA']=='Karapakkam')]

temp_df['SALES_PRICE'].plot.hist(bins=50)

temp_df = df.loc[(df['AREA']=='Anna Nagar')]

temp_df['SALES_PRICE'].plot.hist(bins=50)

df.plot.scatter('DIST_MAINROAD', 'SALES_PRICE')

df.groupby(['STREET']).SALES_PRICE.median()

df.plot.scatter('SALES_PRICE', 'COMMIS')

df[['SALES_PRICE', 'COMMIS']].corr()

df.drop(['PRT_ID'], axis=1, inplace = True)

df = pd.get_dummies(df)

x = df.drop('SALES_PRICE', axis=1)

y= df['SALES_PRICE']

print(x.head(5))

x.shape

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=123)

X_train.shape,X_test.shape,y_train.shape,y_test.shape
```

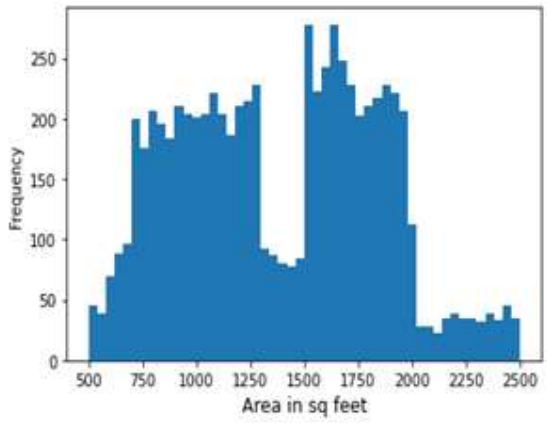
```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

train_scaled = scaler.fit_transform(X_train)

test_scaled = scaler.transform(X_test)

from sklearn.linear_model import LinearRegression

from sklearn.metrics import mean_squared_log_error

from sklearn import model_selection

seed=1

kfold=model_selection.KFold(n_splits=10,random_state=seed)

lreg = LinearRegression()

results=model_selection.cross_val_score(lreg,train_scaled, y_train,scoring='r2',cv=kfold)

results

results.mean()

Adjusted_R2=1-(1-0.9553276768894602)*(7109-1)/(7109-48-1)

Adjusted_R2

from sklearn.tree import DecisionTreeRegressor

from sklearn.ensemble import RandomForestRegressor

rf_model = RandomForestRegressor(n_estimators=50,max_depth=8)

results=model_selection.cross_val_score(rf_model,train_scaled,
y_train,scoring='r2',cv=kfold)

results

results.mean()
```

```
Adjusted_R2=1-(1-0.9646973369177895)*(7109-1)/(7109-48-1)

Adjusted_R2

tree_model = DecisionTreeRegressor(random_state=0,max_depth=5)

seed=6

kfold=model_selection.KFold(n_splits=10)

results=model_selection.cross_val_score(tree_model,train_scaled,
y_train,scoring='r2',cv=kfold)

results

results.mean()

Adjusted_R2=1-(1-0.885424527591583)*(7109-1)/(7109-48-1)

Adjusted_R2

from sklearn.model_selection import GridSearchCV

from sklearn.neighbors import KNeighborsRegressor

import numpy

neighbours=numpy.arange(1,51)

dist_measure=['jaccard','euclidean','manhattan','chebyshev','minkowski']

knn=KNeighborsRegressor()

parameters={"metric":dist_measure,'n_neighbors':neighbours}

GS=GridSearchCV(knn,parameters,cv=10)

GS.fit(train_scaled, y_train)

GS.best_params_

GS.best_score_
```

```
knn=KNeighborsRegressor(metric="manhattan",n_neighbors=9)

results=model_selection.cross_val_score(knn,train_scaled, y_train,scoring='r2',cv=kfold)

results

results.mean()

Adjusted_R2=1-(1-0.9050737417345858)*(7109-1)/(7109-48-1)

Adjusted_R2


from sklearn.linear_model import Ridge

rr = Ridge(alpha=1)

results=model_selection.cross_val_score(rr,train_scaled, y_train,scoring='r2',cv=kfold)

results

results.mean()

Adjusted_R2=1-(1-0.955328183204719)*(7109-1)/(7109-48-1)

Adjusted_R2

from sklearn.linear_model import Lasso

lassoReg = Lasso(alpha=0.3, normalize=True)

results=model_selection.cross_val_score(lassoReg,train_scaled,
y_train,scoring='r2',cv=kfold)

results

results.mean()

Adjusted_R2=1-(1-0.9553277550696831)*(7109-1)/(7109-48-1)

Adjusted_R2
```

Chapter6: Input & Output Screens

When the code gets executed first we get outputs plots and then prediction takes place. These plots help us to understand the correlation between target variable (price) and different predictor variables.



- Most houses have the area between **750 sq feet to 1250 sq feet** or around **1500 sq feet to 2000 sq feet**
- Very less number of houses have area more than 2000 sq feet or less than 750 sq feet

Fig 6.1 Area of house in square feet

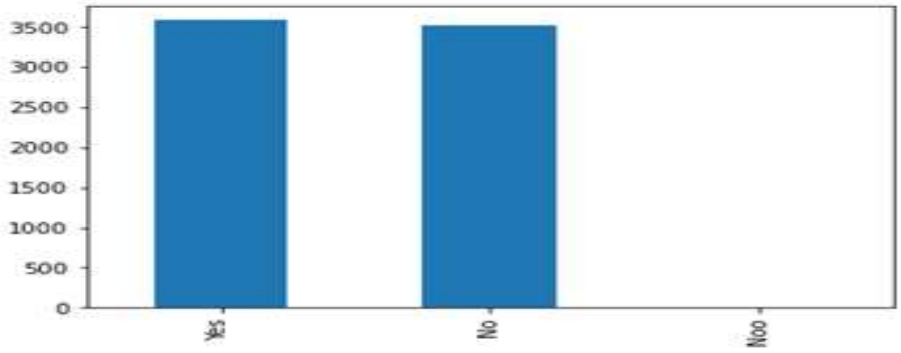


Fig 6.2 Number of houses with parking facility

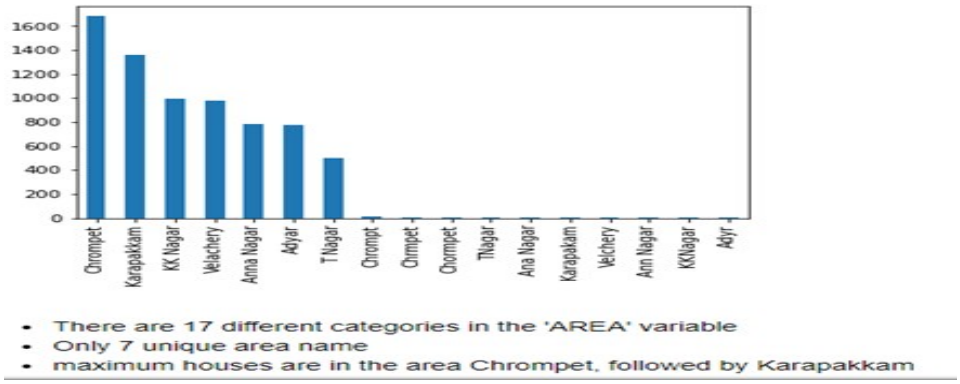


Fig 6.3 Number of houses area wise

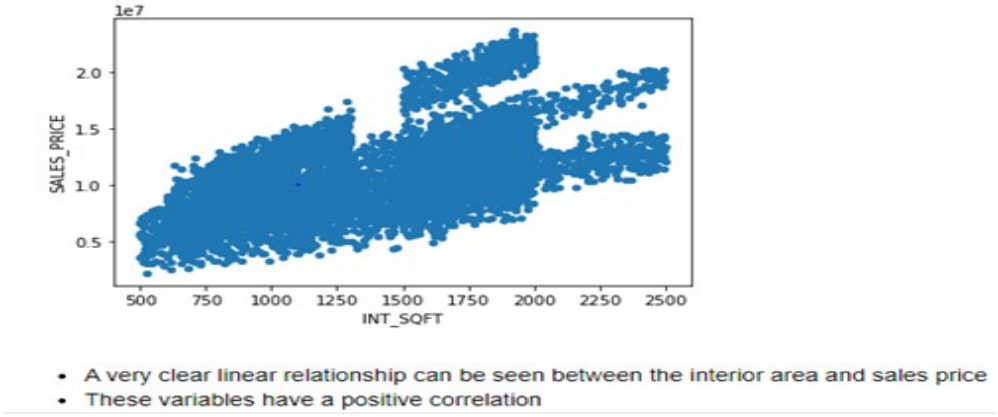


Fig 6.4 Interior area Vs target sales price

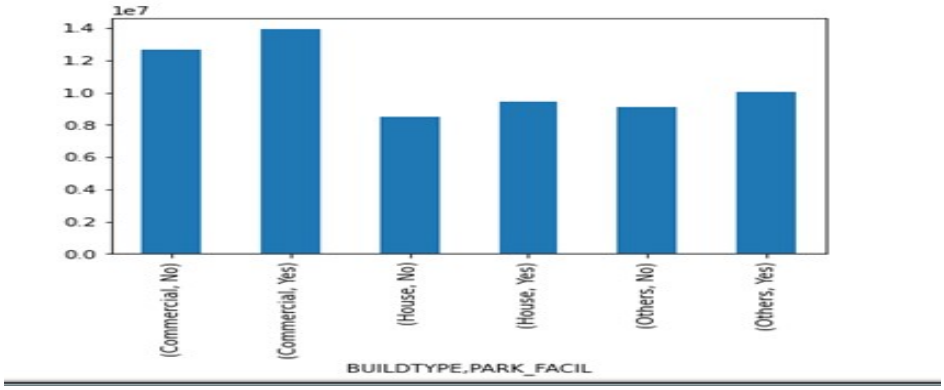


Fig 6.5 Sales Price Vs Building Type and Parking Facility

```
In [66]: # SALE PRICE based on building type

         df.groupby('BUILDTYPE').SALES_PRICE.median()

Out[66]: BUILDTYPE
         Commercial    13356200
         House          8985370
         Others         9637260
         Name: SALES_PRICE, dtype: int64
```

- Houses built for commercial purposes have a considerably higher sale price
- Houses with additional facility should have higher price

Fig 6.6 Sales Price Vs Building Type

```
In [69]: # building type and parking facility

         df.groupby(['BUILDTYPE', 'PARK_FACIL']).SALES_PRICE.median()

Out[69]: BUILDTYPE   PARK_FACIL
         Commercial  No            12692985
                     Yes           13920600
         House       No             8514140
                     Yes            9468150
         Others      No             9104645
                     Yes           10039405
         Name: SALES_PRICE, dtype: int64
```

- For all three categories, houses with park facility have a higher price
- we can use groupby function to generate a plot for better comparison

Fig 6.7 Building Type Vs Parking Facility

```
In [71]: # average price for each area category

         df.pivot_table(values='SALES_PRICE', index='AREA', aggfunc='median')
```

Out[71]:

AREA	SALES_PRICE
Adyar	8878350
Anna Nagar	13727895
Chrompet	9606725
KK Nagar	12146740
Karapakkam	7043125
T Nagar	14049650
Velachery	10494410

- Anna Nagar and T Nagar are comparatively more expensive
- The least priced are among the 7 is karapakkam

Fig 6.8 Area wise House Prices

```
In [75]: df.groupby(['STREET']).SALES_PRICE.median()

Out[75]: STREET
         Gravel       10847225
         No Access     9406050
         Paved        10470070
         Name: SALES_PRICE, dtype: int64
```

- Both gravel and paved roads have approximately same sale price
- Houses marked with 'no access' have a lower sale price

Fig 6.9 Street Type Vs Sales Price

```
In [62]: # sale price of houses wrt number of bedrooms and bathroomms
         df.pivot_table(values='SALES_PRICE', index='N_BEDROOM', columns='N_BATHROOM', aggfunc='median')

Out[62]:
```

N_BEDROOM	N_BATHROOM 1.0	2.0
1.0	9168740.0	NaN
2.0	12129780.0	9125250.0
3.0	NaN	11663490.0
4.0	NaN	13172000.0

Fig 6.10 Sales Price Vs Number of bedrooms and bathrooms

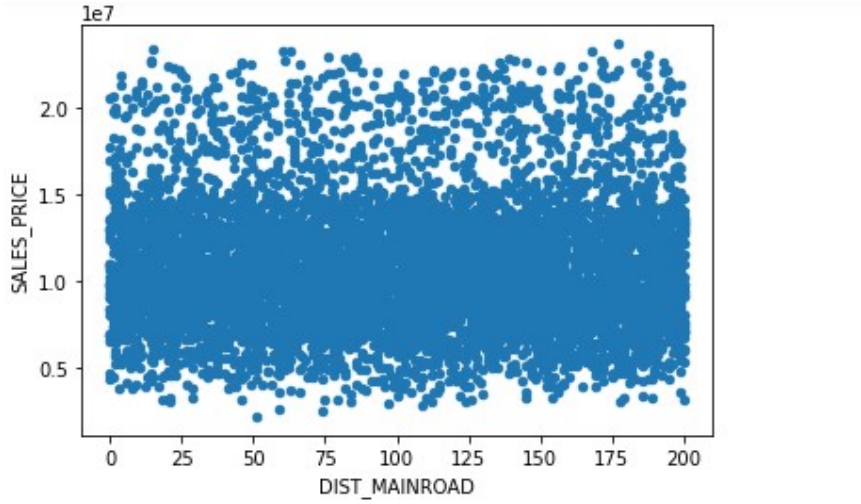


Fig 6.11 Distance from main road Vs Sales Price

Chapter7: System Testing

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic and overfitted models. There are two methods of evaluating models in data science, Hold-Out and Cross-Validation. To avoid overfitting, both methods use a test set (not seen by the model) to evaluate model performance.For Model testing, I have used the following evaluation metrics:-

R-squared (R2) — R-Squared is the proportion of variation in the outcome that is explained by the predictor variables. In multiple regression models, R2 corresponds to the squared correlation between the observed outcome values and the predicted values by the model. The Higher the R-squared, the better the model.

Adjusted R-Squared (R2) — Concerning R2, there is an adjusted version, called Adjusted Rsquared, which adjusts the R2 for having too many variables in the model.A model performing equal to baseline would give R-Squared as 0. Better the model, higher the r2 value. The best model with all correct predictions would give R-Squared as 1. However, on adding new features to the model, the R-Squared value either increases or remains the same. R-Squared does not penalize for adding features that add no value to the model. So an improved version over the R-Squared is the adjusted R-Squared.

```
In [83]: results=model_selection.cross_val_score(rf_model,train_scaled, y_train,scoring='r2',cv=kfold)
         results
Out[83]: array([0.96888995, 0.96406445, 0.9628766 , 0.95643422, 0.96689151,
                0.96477171, 0.96822382, 0.96938638, 0.95860396, 0.96706838])

In [84]: results.mean()
Out[84]: 0.9647210974023064

In [85]: Adjusted_R2=1-(1-0.9646973369177895)*(7109-1)/(7109-48-1)
         Adjusted_R2
Out[85]: 0.9644573188118482
```

Fig 7.1 Best Adjusted R-Squared Score (Random Forest Model)

Chapter8: Conclusion

Given these outputs, we can conclude the model Random Forest is best suited for the dataset. The reason why it's giving better result is that in random forest regressor different trees are created for each instances based on the selected features i.e. n_estimators and hence each tree would give different results. Therefore there exists a high variance across the different trees but ensembling them will cancel each other and hence the accuracy will be high and reliable.

The predictive power of the model chosen for this project would instill confidence among the potential buyer of residential properties in and around Chennai. Hence, there would associated growth in the services sectors concerned whether health, construction, education, consumer products, logistics, infrastructure, etc. for further expansion in the real estate sector whether it is horizontal or vertical expansion.

8.1 LIMITATIONS

Machine learning algorithms produce accurate results wherever there is enough data. Apart from access to data, the advances in computer vision and NLP, it can understand conversations in context and also see and understand images and videos. Data related to Govt. decisions like interest rate changes in housing loan, FDI investment changes. To overcome the effects of volatile attributes like interest rate the model will be re-engineered periodically.

8.2 SCOPE FOR FUTURE WORK

Apart from categorical and numerical features adopted for the present project , sophisticated algorithms making use of image recognition of the flat/houses will easily convince the buyers to make easy decision to finalize their purchase decision so that they can reduce the transaction cost in search for the desired property in the desired locality.

Use of NLP algorithms would make buyers to understand what matters to real estate property in a given pin code /area/ locality and what's unique about a given house from the listings. It would further improve customer interactions during the lending or house buying process for customer service analytics.

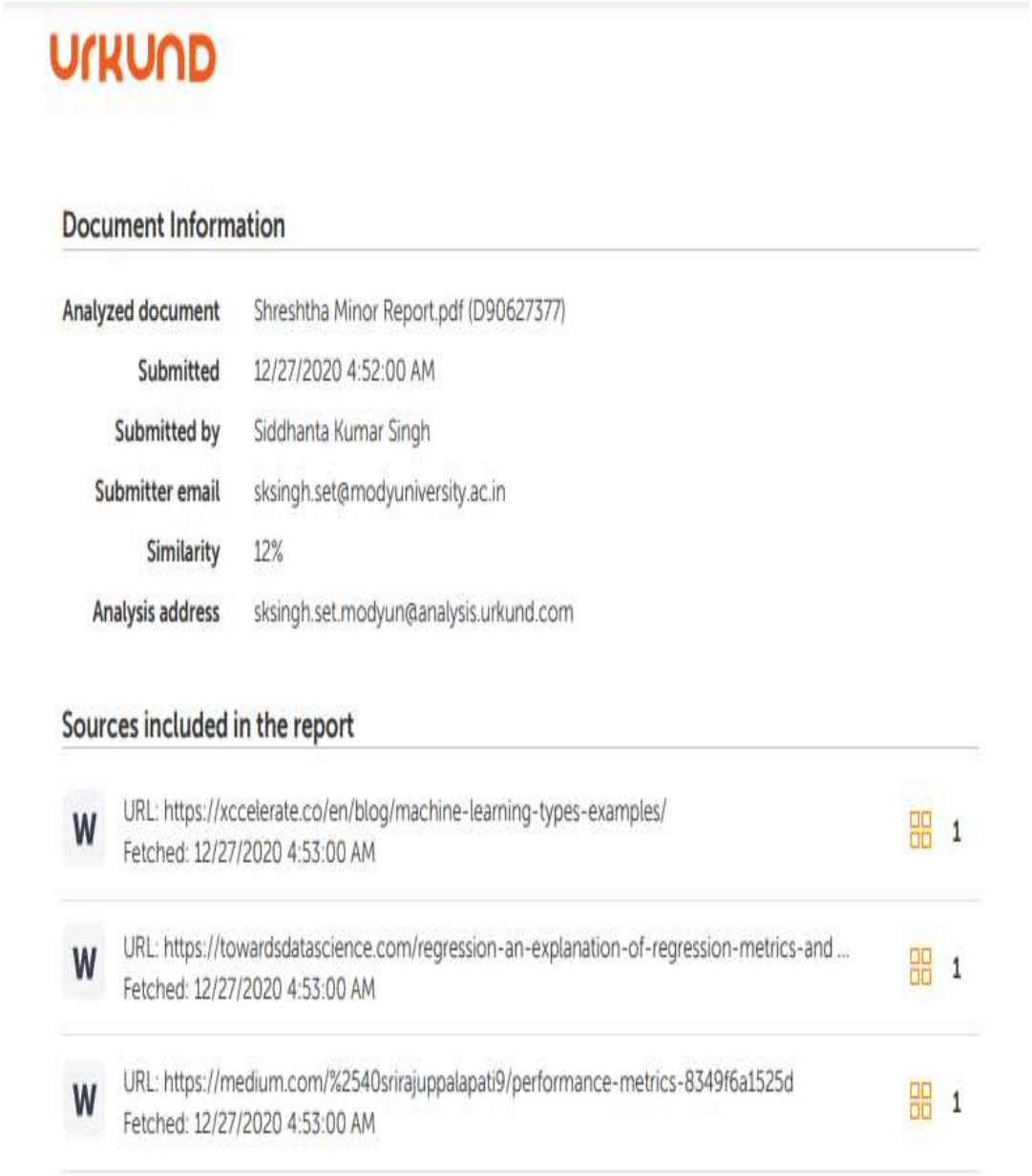
Economic / Political situations like changes in Real Estate Regulation Act (RERA), Unemployment rate, GST, Demonetization, Environmental factors, inflation rate, etc. which are out of human control or any other predictive power which may affect the pricing decisions.

Chapter9: References

- *https://medium.com/@ageitgey/machine-learning-is-fun-80ea3ec3c471*
- *https://www.sas.com/en_us/insights/analytics/machine-learning.html*
- *https://www.wired.co.uk/article/machine-learning-ai-explained*
- *https://deeplearning4j.org/ai-machinelearning-deeplearning*
- *https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/*

Chapter10: Annexures

PLAGIARISM REPORT



URKUND

Document Information

Analyzed document	Shreshtha Minor Report.pdf (D90627377)
Submitted	12/27/2020 4:52:00 AM
Submitted by	Siddhanta Kumar Singh
Submitter email	sksingh.set@modyuniversity.ac.in
Similarity	12%
Analysis address	sksingh.set.modyun@analysis.urkund.com

Sources included in the report

W	URL: https://xccelerate.co/en/blog/machine-learning-types-examples/ Fetched: 12/27/2020 4:53:00 AM	1
W	URL: https://towardsdatascience.com/regression-an-explanation-of-regression-metrics-and ... Fetched: 12/27/2020 4:53:00 AM	1
W	URL: https://medium.com/%2540srirajuppalapati9/performance-metrics-8349f6a1525d Fetched: 12/27/2020 4:53:00 AM	1