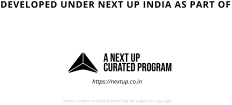
**House Price Detection using Machine Learning**

**Get your dream property estimate now!**

**Q2 - FY 20-21**

****

**HOUSE PRICE PREDICTION**

**Abstract**

Machine Learning plays a major role from past years in image detection, spam reorganization, normal speech recognition, product recommendation, medical diagnosis, and many more. Present machine learning algorithms help us in enhancing security systems, ensuring public safety and improving medical enhancements. Machine Learning also provides better customer service and safer automobile systems. Here, we discuss the prediction of future housing prices that is generated by machine learning algorithms.

**Task Description**

**Description**

Nowadays there are multiple real estate classified websites for house price prediction. However, in each of these websites, we can see a lot of inconsistencies in terms of pricing of a house.

We are predicting the deal price of houses using various machine learning algorithms.

House price predictions are determined by numerous factors such as city,street,floors,bathrooms,bedrooms,year built,living Sq.ft. It uses machine learning algorithms to build the prediction model of houses.

**Problem Statement**

**1. Context**

• The real estate markets, like those in Sydney and Melbourne, present an interesting opportunity for data analysts to analyze and predict where property prices are moving towards. Prediction of property prices is becoming increasingly important and beneficial. Property prices are a good indicator of both the overall market condition and the economic health of a country.

**2. Task Details**

• To identify the factors affecting housing price based on the data. For example, sq. ft., no. of bedrooms, no. of bathrooms, etc. and to predict the price of a house based on the identified factors as input.

• Additionally, how does this data vary for various cab providers across the world, and how does the pricing vary by year, geographical location and locality?

• How would the model vary based on data from Bengaluru?

**3. Datasets**

• House Price Prediction Dataset

• Bengaluru/Karnataka/India House Price Data

**Team**

**Team Member**

* P Vamshi Prasad
* Nese Anil
* Nikitha B S
* Pavithra R S
* Pratistha Vijay
* Sharath Surya S
* Pooja M

**Roles**

|  |  |
| --- | --- |
| **Team Member** | **Roles** |
| Nese Anil | Machine learning |
| P Vamshi Prasad | Web Development, Team Management, UI/UX Design, Documentation. |
| Pooja M | Documentation,ML |
| Sharath Surya S | Web development |
| Pratistha Vijay | Documentation,ML |
| Nikitha B S | Machine learning |
| Pavithra R S | Machine learning |

**Datasets**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Index | Location | BHK | Furnishing | Sq.ft | Yrs | Floor | Price |
| 1 | Bommanahalli | 3 | 1 | 3000 | 1 | 3 | 28000 |
| 2 | Bommanahalli | 3 | 1 | 1650 | 10 | 0 | 18000 |
| 3 | Whitefield | 2 | 1 | 1000 | 5 | 3 | 16400 |
| 4 | Whitefield | 3 | 0 | 1600 | 1 | 9 | 27000 |

Variables are defined as the properties or kinds of characteristics of certain events or objects.

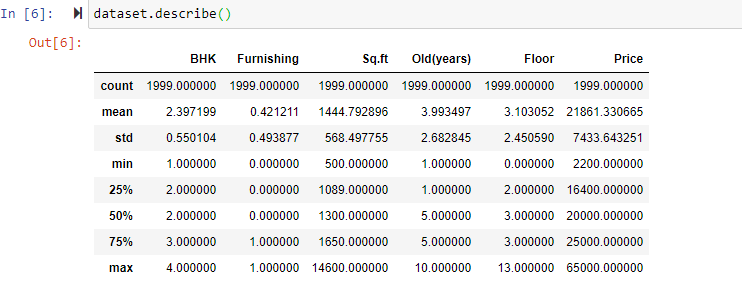
Independent variables are variables that are manipulated or are changed by researchers and whose effects are measured and compared. The independent variables here are location, BHK, furnishing, Sq.ft., years and No. of floors.

The dependent variables refer to that type of variable that measures the effect of the independent variable(s) on the test units. Here, the dependent variable is price.

Attribute types: shows attributes with their respective types.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **attribute** | location | BHK | furnishing | sq.ft | Old | floor | price |
| **type** | object | int | int | int | int | int | int |

The basic statistical measuresare shown in the figure below



**Work Log**

A work log is a report that summarizes how a team member spent his/her time on a project.

[Task Sheet](https://docs.google.com/spreadsheets/d/1fGbeImuR9SNhvgOeFssHix-EwSGoZWSZIXlQrHknuhc/edit#gid=0)

**Implementation**

**Front end:**

For the Front end, the frameworks used are React JS, Bootstrap, and Material Design.

There are 2 screens totally.

* The first screen, i.e., the home screen, is where the inputs of the user is taken, for which the user wants a prediction of the house price.
* The second screen is where we get the predicted value from the python script, and display it with a small message.

**First Screen:**

The first Javascript to be executed is the App.js

In that file, there are 2 routes defined in case of wrong URLs.

<Route exact path="/" component={Home}></Route>

<Route exact path="/result" component={Result}></Route>

**Home Page:**

The first screen uses a background as a PNG with a very less opacity.



The PNG Container is absolute in position.

<img src={back} className="image"></img>

The image CSS is as follows:

.image{

position: absolute;

width: 100vw;

top: 0;

opacity: 10%;

}

The source of the image is imported as:

import back from './assets/mainBackground.png';

On that, a Card is rendered. For this Material Design is used..

It is used as:

<Card className={classes.root} >

</Card>

The root of the classes class is as follows:

root: {

minHeight: 700,

zIndex: 10,

backgroundColor: 'white',

borderRadius: 20,

boxShadow: 'rgba(0, 0, 0, 0.1) 0px 1px 350px, rgba(0, 0, 0, 0.1) 0px 1px 350px'

},

The card is imported by:

import Card from '@material-ui/core/Card';

import CardContent from '@material-ui/core/CardContent';

import Typography from '@material-ui/core/Typography';

Here, those CardContent and Typography are the objects for the content inside the Card, and the Font for the Material Design Library respectively.

Later, Grid is used which is provided by the Material Library for having a responsive design.

It is implemented as:

<Grid container spacing={3} justify="center" style={{ zIndex: 1, }}>

<Grid item xs={9}>

</Grid>

</Grid>

The Grid is imported as:

import Grid from '@material-ui/core/Grid';

Post that, various textFields need to be rendered.

For that, a container is used, set the display to flex, and justified to space-evenly, as such.

<div style={sty}>

</div>

sty = {

display: 'flex', flexDirection: 'column', justifyContent: 'center'

};

The flexDirection can be row or column based upon the device width.

Later the textfields are rendered. These textfields are responsive based upon what device is used.

<TextField id="outlined-basic" label="No. of Bedrooms" variant="outlined" style={textSty1} />

textSty = { margin: 10, width: (window.innerWidth / 4) };

The width is divided by 4 times for the middle textfield, by 6 times, for the first 3 textFields, and by 3 for the last one.

The textFields are imported by the following code:

import { TextField } from '@material-ui/core';

After rendering all the textFields the SUBMIT Button is rendered. It is a simple flat button.

<Button variant="contained" color="primary" style={{ backgroundColor: "#681AFF", color: 'white', fontSize: 20, width: (window.innerWidth / 4) }} disableElevation>SUBMIT</Button>

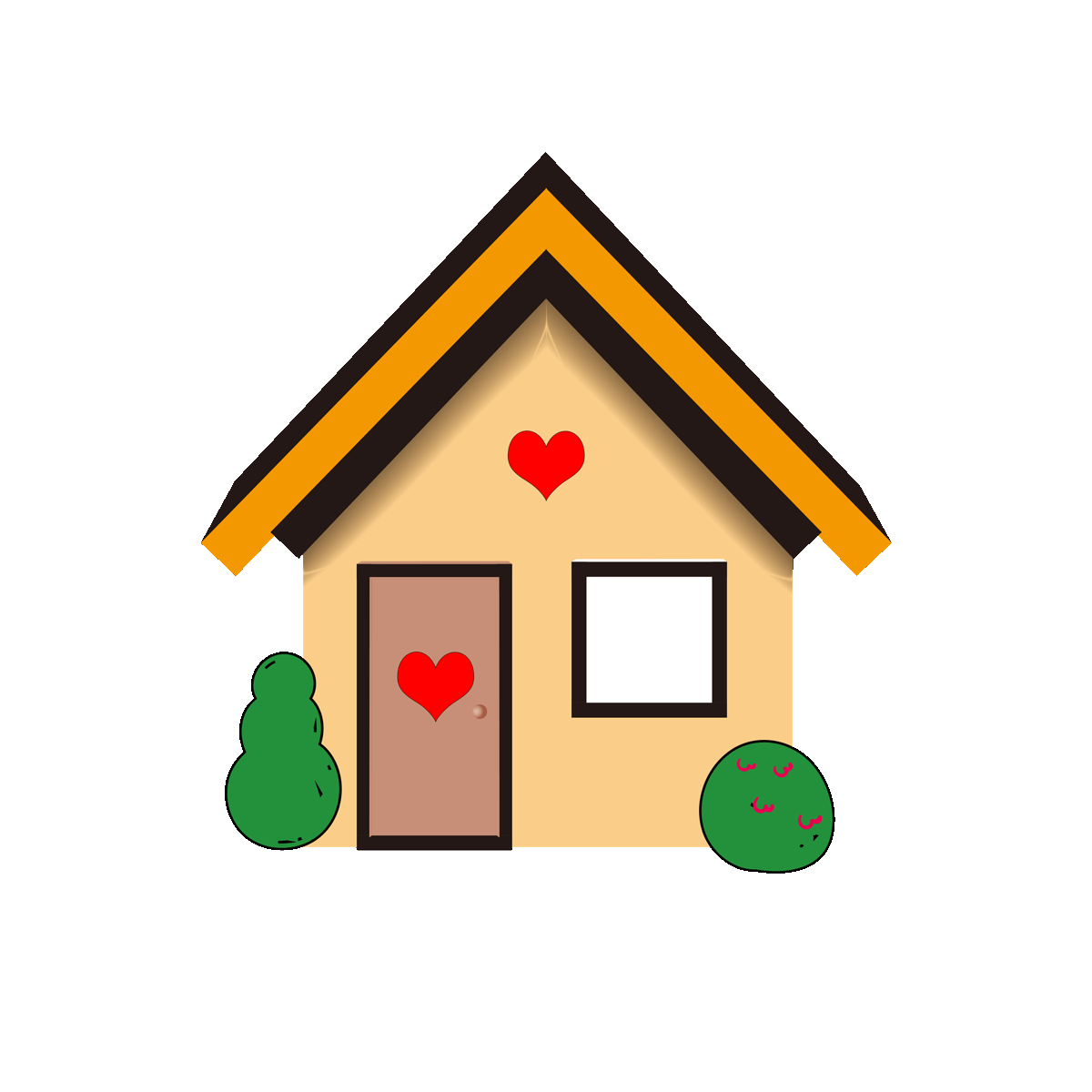
It is imported by:

import Button from "@material-ui/core/Button";

After all the Rendering, the function App is exported as default.

**Result Page:**

The second screen, which is the result screen also uses a PNG with low opacity.



Card is rendered inside a Grid.

<Grid container spacing={3}

justify="center" style={{ zIndex: 1, }}>

<Grid item xs={9}>

<Typography variant="h2" component="h2" style={{ color: "grey" }}>

House Price Detection

</Typography>

<br></br>

<Card className={classes.root} >

</Card><br></br>

</Grid>

</Grid>

Then later our texts is rendered in the CardContent:

<CardContent>

<Typography className={classes.title} color="textSecondary" gutterBottom>

Well, based on Whatever you've given us.... <br></br>We think that your dream property might cost you around<br></br><br></br></Typography>

<Typography style={{ fontSize: 50 }} color="textSecondary" gutterBottom>₹ 987615984<br></br><br></br></Typography>

<Typography className={classes.title} color="textSecondary" gutterBottom>You can click on the button below, so that we can get you in touch with one of our agents.

</Typography><br></br>

<br></br>

<Button

variant="contained"

color="primary"

style={{ backgroundColor: "#681AFF", color: 'white', fontSize: 20, width: (window.innerWidth / 4) }}

disableElevation

>

Real Estate Agent</Button>

</CardContent>

The Card, Grid, and CardContent are imported as follows:

import Grid from '@material-ui/core/Grid';

import CardContent from '@material-ui/core/CardContent';

import Button from "@material-ui/core/Button";

import Typography from '@material-ui/core/Typography';

import Card from '@material-ui/core/Card';

After that, the function is exported as default.

export default Result;

**Index File:**

In the Index file, the App within a BrowserRouter for the Navigation is needed to be wrapped, and then render it into the div of the Index.html with ID of ‘root’.

ReactDOM.render(

<BrowserRouter>

<App />

</BrowserRouter>,

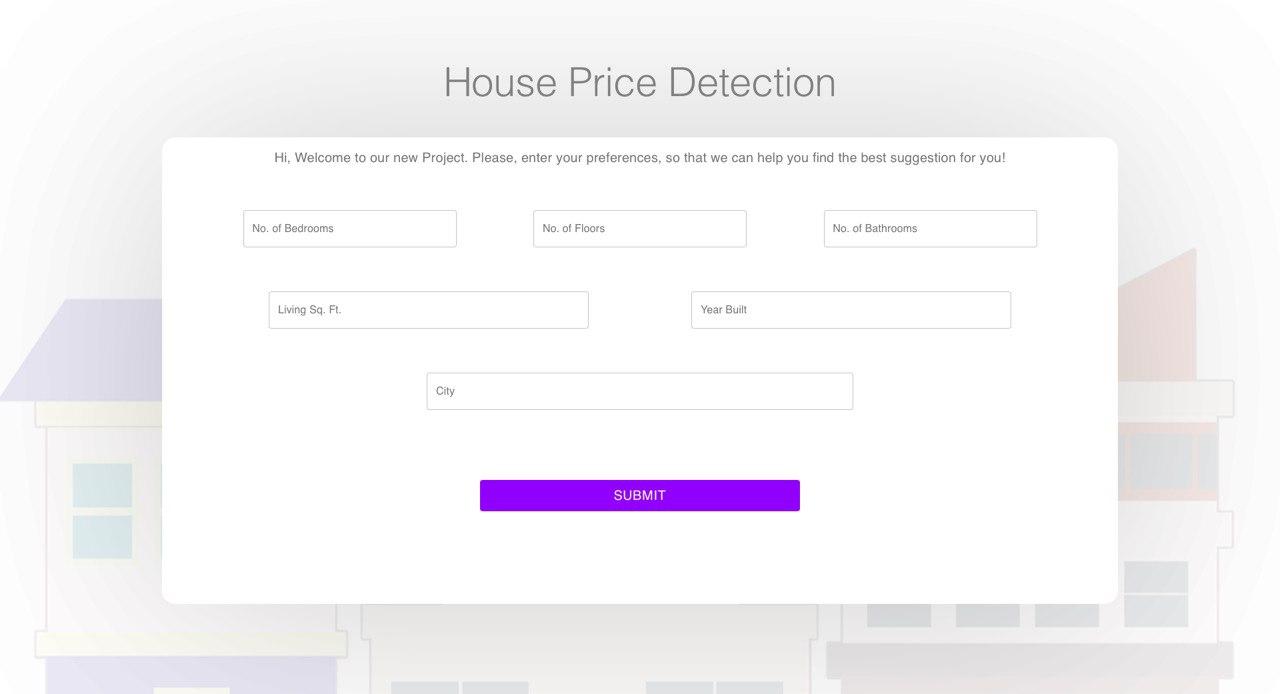
document.getElementById('root')

);

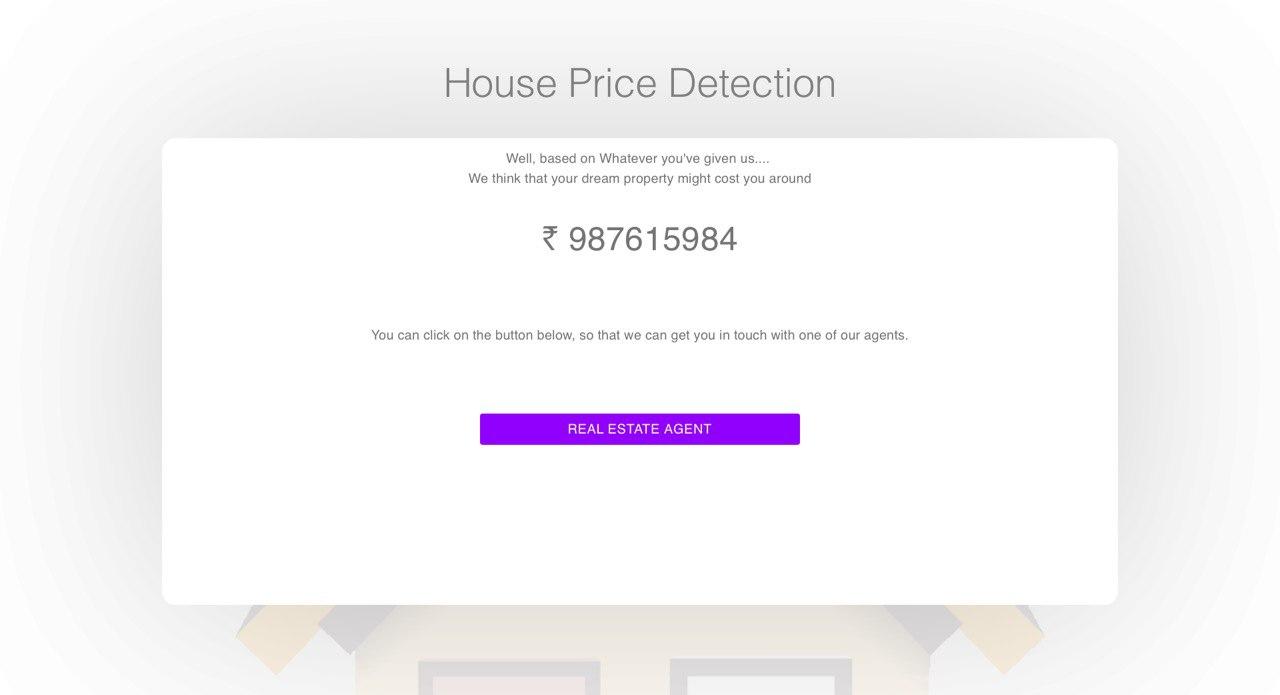
BrowserRouter is imported as such:

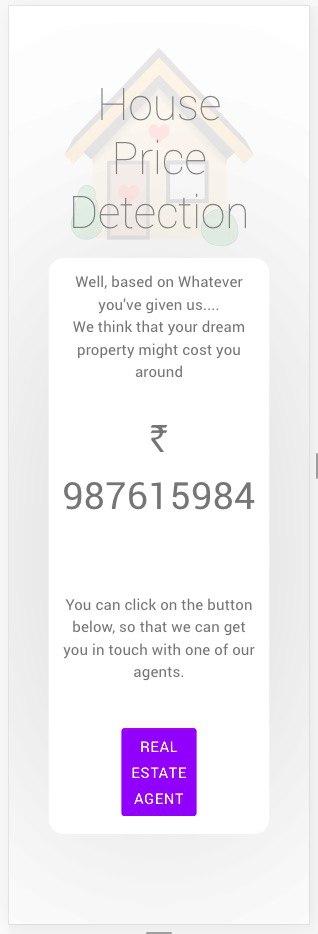
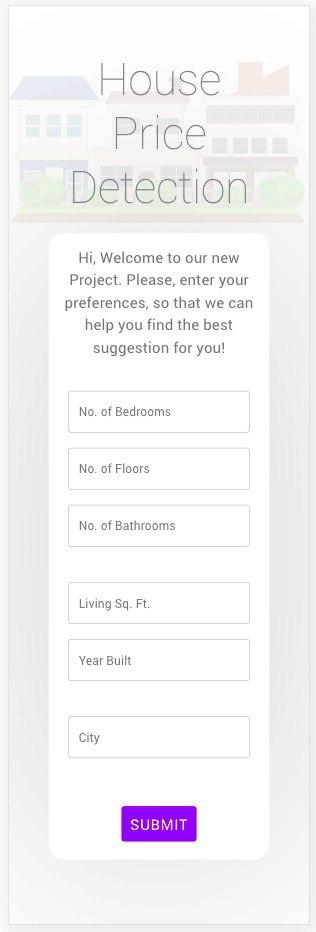
import { BrowserRouter } from 'react-router-dom’

**Screenshots:**

****

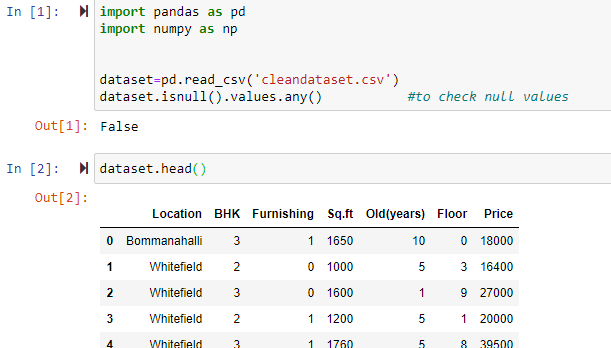
Desktop - Home Page

****Desktop - Result Page

****

Mobile Phone - Home page & Result Page

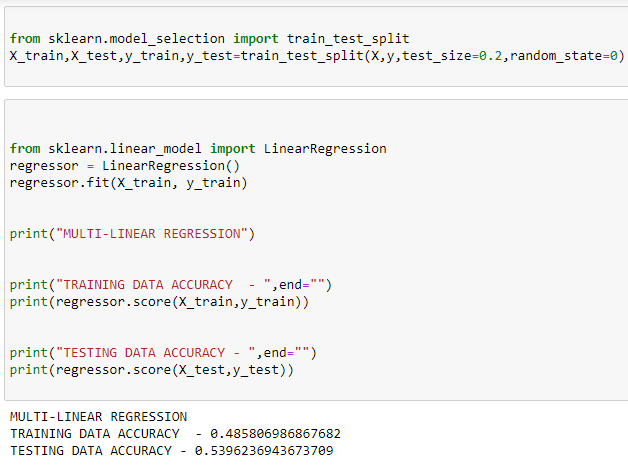
**Back end:**

Libraries and datasets are imported

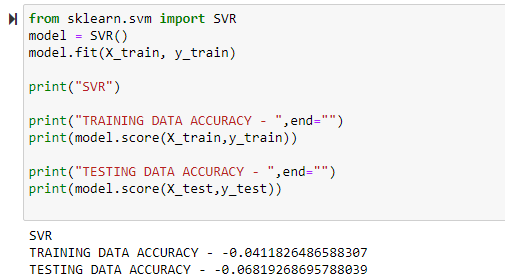
**Starting by importing libraries and reading dataset**

****

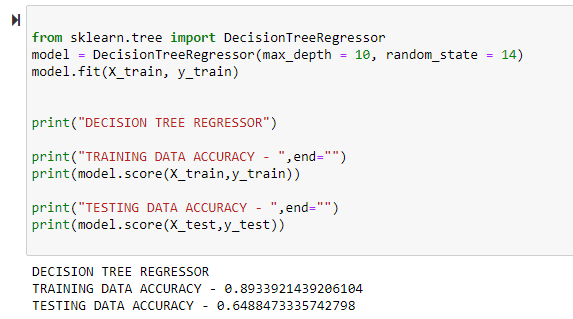
**Knowing more about the dataset**

****

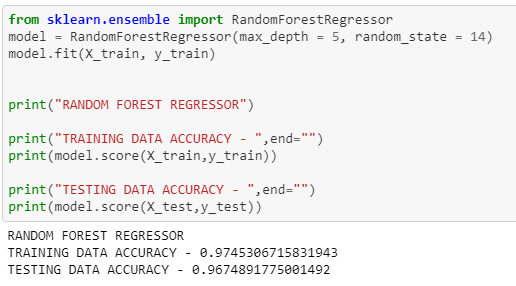
**Linear regression on the data to predict prices**



**SVR on the data to predict prices**



**Decision tree on the data to predict prices**



**Random forest on the data to predict prices**

**Conclusion:**

We built four models and evaluated their performances.

Random Forest is the most accurate model for predicting the house price .It scored an estimated accuracy of 96%.