**Report**

# Introduction

**Project name**

BOSTON HOUSE PRICE PREDICTION

**Project objective**

Predict the House prices (MEDV) based on given features

**Project by,**

**Sathishkumar S**

# Data Description

The Boston Housing Dataset

The Boston Housing Dataset is a derived from information collected by the U.S.

**The Boston housing market is highly competitive, and you want to be the best real estate agent in the area. To compete with your peers, you decide to leverage a few basic machine learning concepts to assist you and a client with finding the best selling price for their home. Luckily, you've come across the Boston Housing dataset which contains aggregated data on various features for houses in Greater Boston communities, including the median value of homes for each of those areas. Your task is to build an optimal model based on a statistical analysis with the tools available. This model will then be used to estimate the best selling price for your clients' homes.**

**CRIM**

per capita crime rate by town

**ZN**

proportion of residential land zoned for lots over 25,000 sq. ft.

**INDUS**

proportion of non-retail business acres per town.

**CHAS**

Charles River dummy variable (1 if tract bounds river; 0 otherwise)

**NOX**

nitric oxides concentration (parts per 10 million)

**RM**

the average number of rooms among homes in the neighbourhood

**AGE**

proportion of owner-occupied units built prior to 1940

**DIS**

weighted distances to five Boston employment centres

**RAD**

Index of accessibility to radial highways

**TAX**

full-value property-tax rate per $10,000

**PTRATIO**

the ratio of students to teachers in primary and secondary schools in the neighbourhood

**B**

1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town

**LSTAT**

the percentage of homeowners in the neighbourhood considered "lower class" (working poor)

**MEDV**

Median value of owner-occupied homes in $1000

# Approach

My approach to the solution of this project is the simple step by step process

First I read the data and understand the variables in data set

After that I do the **EDA**,

* From here I check the features data types, null values and duplicate values. so there is no null values and duplicate values
* After that I want to check there is any outliers and the relationship between the features so for that I use the matplot and seaborn visualization libraries to help of the plot data and so I can visualise the outliers and I can easily understand the relationship between the target and predictor variables.
* I did not remove any outliers because we less number of data

And then I do the data **preprocessing**,

* So we have different columns and different values from different scale so I do the standardization to the input features
* So it’s reduce the scale range between -1 to 1.
* After that I use train test split to split the data into train and test with the sample size of 7 : 3

So finally I do the **model building,**

* So I use some **regression** algorithms and I train the model and test the model with regression test metrics
* I use 6 machine learning algorithms
* I use **gridsearchcv** to **random forest** and **knn** algorithm to find the best parameters

# Visualization

These are the plots I used to visualization the data to get some information’s about the data and gain some insights from it

## Box plot

To the help of box plot we can get the visualization of the outliers

And also with the help of box plot quartile percentage values we can see where the maximum data lies within data itself.

## Distribution plot

It gives us the distribution of the data like normal distribution or not

It also gives the details about data distribution skewness

## Heat map

With the help of heat map, and df.corr( ) we can able to see the relationship between the variables and target variables

So the colour different of the heat map visualization help us understand the relationship better and clear

## Regression plot

Regression plot contain scatter plot and linear line so with the help of this plot we can visualization between the variables

It gives more details about the relationship like it’s negative or positive

# Algorithms

As I mentioned, I have used various regression models on this dataset and they have different accuracy and other performance measures. I have used the following machine learning algorithms on our dataset.

* **Linear regression**
* **Ridge regression**
* **XG boost**
* **Random forest**
* **SVM**
* **KNN**

# Evaluation

These are the four evaluation metrics I used in this regression projects to calculate the accuracy of the models

And **rmse** kind of metrics gives the details about the errors too so we can able to understand the models better.

## **R2 score**

It gives the test score that tell about how well our model works

**R-Squared = 1 – (SSE/SST) =**  **1 − ∑ (y i − y i ^) 2 ∑ (y i − y¯) 2**

## **MaE**

Absolute error refers to **the magnitude of difference between the prediction of an observation and the true value of that observation**.

**MAE = | yi –y^| /n**

## **MSE**

MSE is used **to check how close estimates or forecasts are to actual values**. Lower the MSE, the closer is forecast to actual. This is used as a model evaluation measure for regression models and the lower value indicates a better fit.

**MSE = (yi –y^)^2 /n**

## **RMSE**

The RMSE is the square root of the variance of the residuals. It indicates **the absolute fit of the model to the data–how close the observed data points are to the model's predicted values**. Whereas R-squared is a relative measure of fit, RMSE is an absolute measure of fit.

**RMSE =sqrt ((yi –y^) ^2 /n)**

# comparison

* **I am going to compare the models with each other different type of error metrics**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| models | metrics | MAE | MSE | RMSE | R2\_score |
| Linear regression | | 3.1627098714574102 | 21.51744423117726 | 4.638689926172827 | 0.7112260057484925 |
| Ridge regression | | 3.1618221918632847 | 21.53299465330921 | 4.640365788740065 | 0.7110173119341578 |
| XG boost | | 7.267929130478909 | 83.33742361572888 | 9.12893332299721 | -0.11842653939701919 |
| Random forest | | 2.12 | 9.94 | 3.15 | 0.86 |
| SVM | | 2.90852385528867 | 25.692905365809217 | 5.068816959193655 | 0.6551893976487939 |
| KNN | | 2.5140061405828984 | 16.560463513368706 | 4.069454940574807 | 0.7777509659589257 |

# Result and discussion

* **It’s a small data set we have only 506 data and 14 columns including the target variable**
* **And in the data set we have no missing values and less number of outliers in less number of columns maximum features are correlated with the target variable**
* **So after doing all the EDA and data preprocessing I provide the well good data to the models to train**
* **And finally compare all the models metrics to each other’s we get best r2 score and rmse value**

# Conclusion

After doing the all data cleaning process and visualizations and building the models and Appling all the error metrics to find the best model.

Then compare with the error metrics to other model we get best scores in all the metrics evaluation for RandomForestRegressor

We got,

**MAE : 2.12**

**MSE : 9.94**

**RMSE : 3.15**

**R2\_score : 0.86**

So, we can deploy the RandomForestRegressor model using streamlit and create the web page to predict the

future Prices of the new houses based on the given features

# difficulty faced

* Outlier treatment
* Treat the skewed data into normal distribution
* Find the best models to train
* Finding the best parameters

# Future Work

* House Price prediction, is **important to drive Real Estate efficiency**. As earlier, House prices were determined by calculating the acquiring and selling price in a locality. So this data set contain house details in Boston city
* Therefore, the House Price prediction model is very essential in filling the information gap and improve Real Estate efficiency in Boston city

# References

**Towards data science -** <https://towardsdatascience.com/>

**Sklearn.com -** <https://scikit-learn.org/>

**Kaggle -** <https://www.kaggle.com/>

**GitHub -**