

BOSTON HOUSE PREDICTION REPORT

Author: Kolawole Joseph. E

Introduction

This project work helps to predict the house prices (MEDV) based on given features in the given datasets. Supervised Machine learning algorithms was used. The project work provides technical insights and demonstrates the implementation of house prediction models.

Data Description

The dataset consists of 15 columns and 351 rows with the “label” column as its target column. The Boston Housing Dataset is a derived from information collected by the U.S. Census Service concerning housing in the area of Boston MA. The following describes the dataset columns:

- 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 - average number of rooms per dwelling
- AGE - proportion of owner-occupied units built prior to 1940
- DIS - weighted distances to five Boston employment centers
- RAD - index of accessibility to radial highways
- TAX - full-value property-tax rate per \$10,000
- PTRATIO - pupil-teacher ratio by town
- B - $1000(B_k - 0.63)^2$ where B_k is the proportion of blacks by town
- LSTAT - % lower status of the population
- MEDV - Median value of owner-occupied homes in \$1000's

Approach

For this project work, all necessary Python libraries, modules and dataset were first imported, and then followed by data visualization. Preprocessing and EDA was carried out before the models were built. Evaluation metrics as well such as precision, recall, and f1 score for all the models were also done. The whole code was written more in Jupyter Notebook and then exported to Google Colab

Code and Visualization

The complete code can be found in the link below. However, some of the code snippets and visuals are shown below:

https://drive.google.com/file/d/1jl1vh_mqROdlWaMbJcXWX_YL8Y7jY8Yc/view?usp=sharing

➤ The Code:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Import the Train dataset
df1 = pd.read_csv('Boston_Train.csv')
df1.head()
```

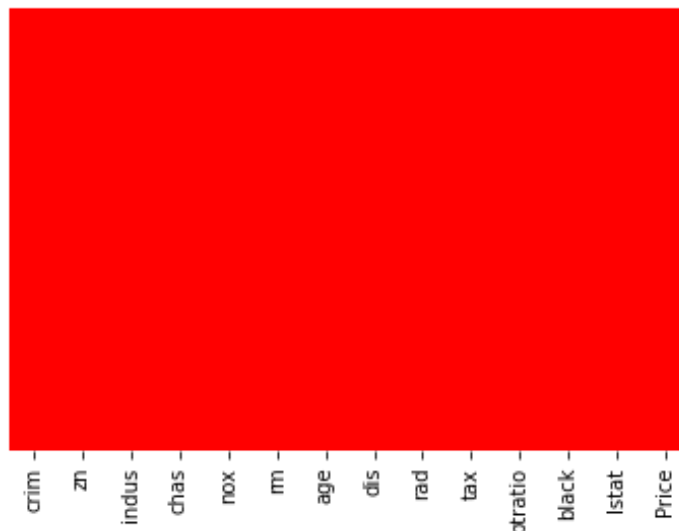
```
# Import the Test dataset
df2 = pd.read_csv('Boston_Test.csv')
df2.head()
```

```
# Lets try to understand which are important feature for this dataset
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
X_train = df1.iloc[:,0:13] #independent columns
y_train= df1.iloc[:,-1] #target column i.e price range
X_test = df2.iloc[:,0:13]
y_test = df2.iloc[:,-1]
```

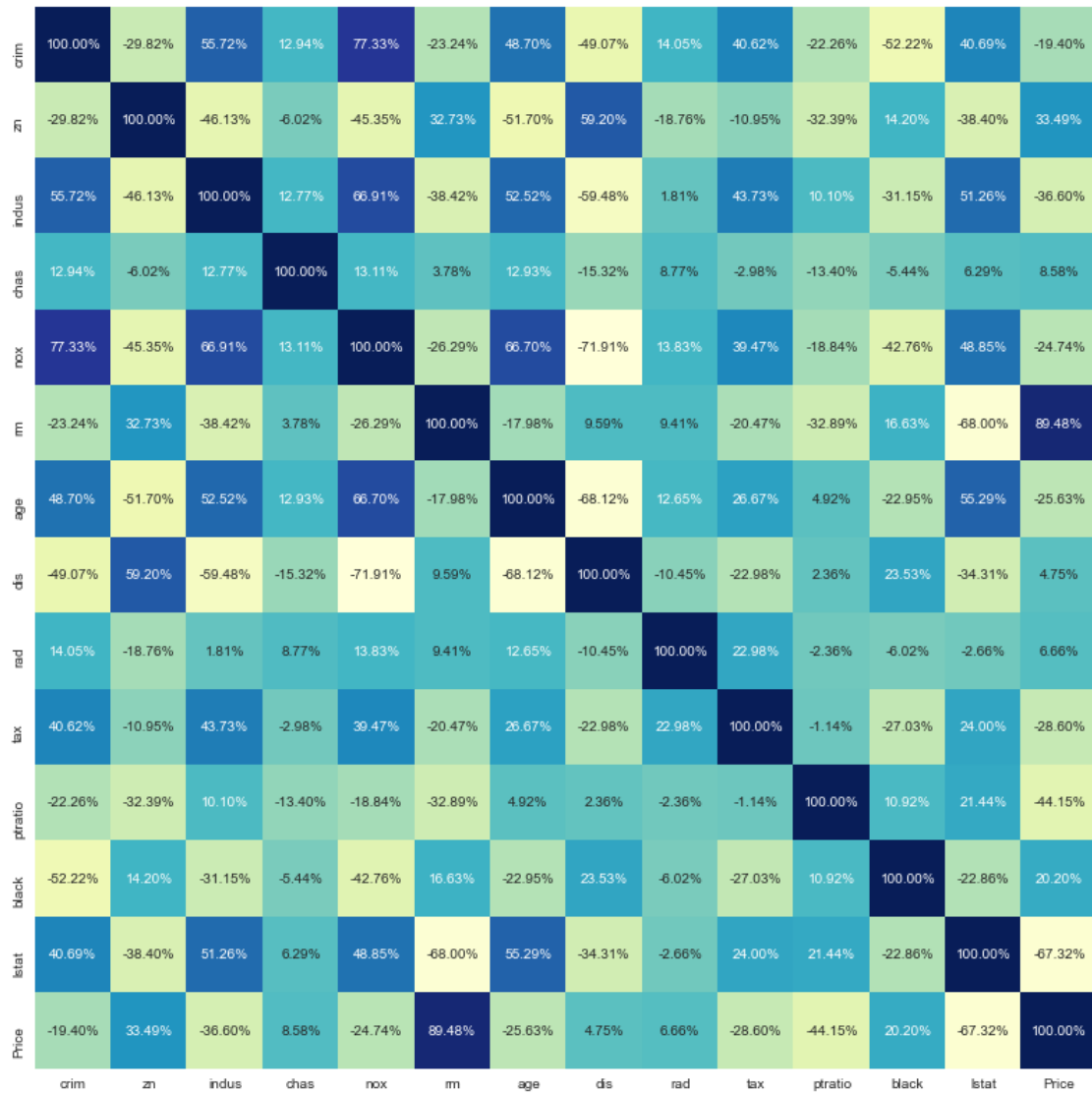
```
from sklearn.ensemble import RandomForestRegressor
reg = RandomForestRegressor()
reg.fit(X_train, y_train)
y_train_pred = reg.predict(X_train)
print("Training Accuracy:", reg.score(X_train, y_train)*100)
reg.fit(X_test, y_test)
y_test_pred = reg.predict(X_test)
print("Testing Accuracy:", reg.score(X_test, y_test)*100)
```

➤ Visualization

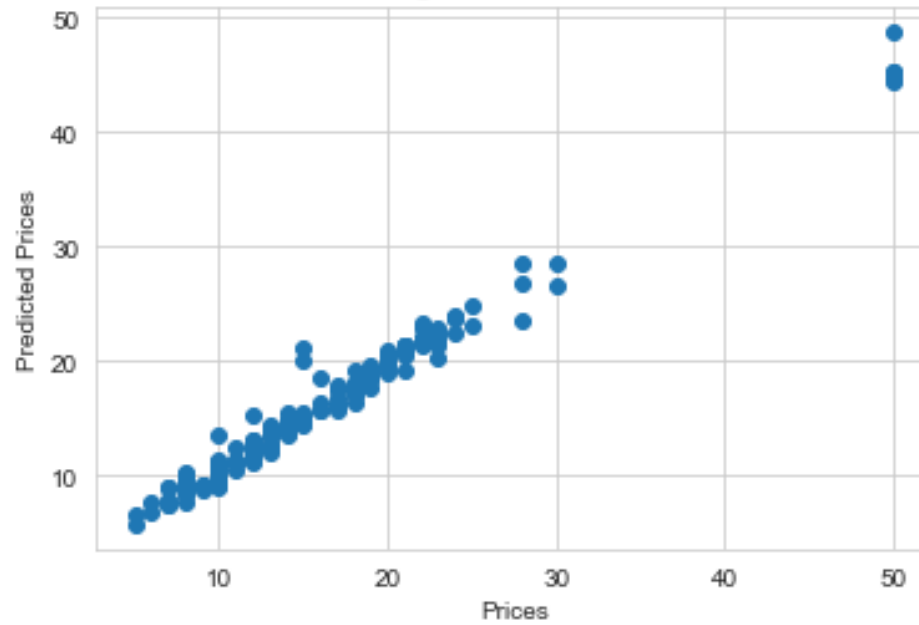
Heatmap showing null values



Heatmap showing correlation between parameters



Prices against Predicted Prices





 **jupyter** Boston House Prediction Project by Joseph K Last Checkpoint: 11/07/2021 (autosaved)




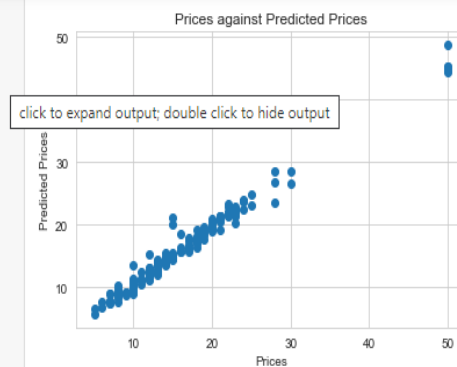
Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted

Python 3

       Run    Markdown



PREDICTION AND FINAL SCORE

*Linear Regression Score:

Model Score: 87.56% Accuracy Training Accuracy: 87.56% Accuracy Testing Accuracy: 68.71% Accuracy

*Random Forest Regressor Score:

Model Score: 99.99% Accuracy Training Accuracy: 98.62% Accuracy Testing Accuracy: 96.92% Accuracy

Algorithms

As earlier mentioned, I have used various classification models on this dataset and they have different accuracy and other performance measures. The following machine learning algorithms were used on the dataset:

1. Random Forest Classifier
2. Linear Regression

Evaluation

The built models were evaluated using the evaluation metrics provided by the scikit-learn package. The main objective in this process is to find the best model for the given case. The evaluation metrics used are the accuracy score metric and r2_score metric.

Result and discussion

The Random Forest classifier has higher accuracy score than Linear Regression. This makes the Random Forest model perform better than Linear Regression

PREDICTION AND FINAL SCORE

*Linear Regression Score:

Model Score: 87.56% Accuracy Training Accuracy: 87.56% Accuracy Testing Accuracy: 68.71% Accuracy

*Random Forest Regression Score:

Model Score: 99.99% Accuracy Training Accuracy: 98.62% Accuracy. Testing Accuracy: 96.92% Accuracy

Conclusion

After EDA, visualizations and machine learning algorithms had been done, it is clearly evident that the Random Forest model performed much better than Linear Regression

Future Work

Given more time and resources, other Machine learning models would be built and tested on the dataset with proper dash boarding and deployment using Streamlit, Flask/Django, and AWS /Azure.

References

SKillVertex – *Boston house prediction Datasets and other study materials*