**Predicting Boston Housing Prices**

**Machine Learning**

**Amandeep Singh**

**ABSTRACT**

In the time of Machine learning and Big Data, all issues can be solved by big data technique. In this assignment forecasting of prices for Boston suburbs houses are done based on actual

Data using various machine learning methods. In this assignment, algorithms used are compared to the predicted results to know the accuracy of the model. In this assignment Mean Value Imputation has been used to clean the data and Linear Multiple Regression algorithm is used to predict the model. And to assess the fitness of the model R-squared method has been used.

Various variables are used to predict the model which are

1. CRIM: Crime rate by town

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

3. INDUS: proportion of non-retail business acres per town

4. CHAS: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)

5. NOX: nitric oxides concentration (parts per 10 million)

6. RM: average number of rooms per dwelling

7. AGE: proportion of owner-occupied units built prior to 1940

8. DIS: weighted distances to five Boston employment centers

9. RAD: index of accessibility to radial highways

10. TAX: full-value property-tax rate per $10,000

11. PTRATIO: pupil-teacher ratio by town

12. B: 1000(Bk −0.63) 2 where Bk is the proportion of blacks by town

13. LSTAT: % lower status of the population

**INTRODUCTION**

In this assignment, prediction of prices of Boston houses will help the researchers in construction domain to built various intelligent systems to achieve good economic and social benefits. If the model by using various machine leaning algorithms can predict the exact or close to exact prices of the houses, then it will be helpful for government to make reasonable urban plans for people.

**CODE**

**#Importing Libraries**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from scipy import stats

%matplotlib inline

**#Reading Input file of Dataset**

boston = pd.read\_excel("Book1.xls")

boston

**#Returning top 10 rows of the dataset**

boston.head(n=10)

boston.info()

**DATA CLEANING**

**#Counting Null records for the dataset**

boston.isnull().sum()

boston.describe()

**#Calculating mean value for the attribute 'medv'**

boston['medv'].mean()

**#Replacing null values with the mean value of the existing records**

boston = boston.fillna(boston['medv'].mean())

**#Correlation and heat map**

correlation = boston.corr()

sns.set\_context("notebook", font\_scale = 1.0, rc = {"lines.linewidth" : 2.5})

plt.figure(figsize=(13, 7))

a = sns.heatmap(correlation,annot = True, fmt = '.2f')

rotx = a.set\_xticklabels(a.get\_xticklabels(), rotation=90)

roty = a.set\_yticklabels(a.get\_yticklabels(), rotation=30)

**Exploratory Data Analysis**

**#Creating a Boxplot**

sns.boxplot(data = boston)

**#Creating a histogram**

boston.hist()

**#Creating Density Plots**

sns.distplot(boston["crim"])

sns.distplot(boston["rm"])

sns.distplot(boston["zn"])

sns.distplot(boston["dis"])

sns.distplot(boston["ptratio"])

plt.figure(figsize=(7,7))

plt.scatter(x='medv',y='crim',data=boston)

plt.xlabel('Mean Housing Price')

plt.ylabel('Crime rate')

plt.title('Crime Vs Price')

plt.figure(figsize=(7,7))

plt.scatter(x='medv',y='nox',data=boston)

plt.xlabel('Mean Housing Price')

plt.ylabel('Notroigen oxide conc.')

plt.title('NOX Vs Price')

plt.figure(figsize=(7,7))

plt.scatter(x='medv',y='rm',data=boston)

plt.xlabel('Mean Housing Price')

plt.ylabel('No of Rooms')

plt.title('Room Vs Price')

plt.figure(figsize=(7,7))

plt.scatter(x='medv',y='age',data=boston)

plt.xlabel('Mean Housing Price')

plt.ylabel('Age')

plt.title('Age Vs Price')

plt.figure(figsize=(7,7))

plt.scatter(x='medv',y='rad',data=boston)

plt.xlabel('Mean Housing Price')

plt.ylabel('Access radial highways')

plt.title('Highways Vs Price')

plt.figure(figsize=(7,7))

plt.scatter(x='medv',y='tax',data=boston)

plt.xlabel('Mean Housing Price')

plt.ylabel('Tax')

plt.title('Tax Vs Price')

**#Removing Outliers**

**#dropping outliers from crim column ie removing values above 9.078**

boston= boston.drop(boston[boston['crim']>9.078].index)

**#dropping outliers from rm ie removing values above 7.661 and below 4.817**

boston = boston.drop(boston[(boston['rm']>7.6615) | (boston['rm']<4.8175)].index)

**#dropping outliers from zn column ie removing values above 31.25**

boston= boston.drop(boston[boston['zn']>31.25].index)

**#Linear Regression**

from sklearn.linear\_model import LinearRegression

lm =LinearRegression()

X = boston.drop('medv',axis=1)

y = boston['medv']

**# Split train - test dataset**

**#Importing Library**

from sklearn.model\_selection import train\_test\_split

**#train\_test\_split**

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=42)

predict = lm.predict(X\_test)

**# Scatter Plot**

plt.scatter(y\_test, predict)

plt.xlabel("Prices")

plt.ylabel("predicted prices")

plt.title("prices vs predicted prices")

**#Mean Squared Error**

from sklearn. metrics import mean\_squared\_error

from math import sqrt

rms = sqrt(mean\_squared\_error(y\_test, predict))

**RESULTS**

By using Linear Regression model in this assignment, we have tried to predict Boston housing prices but to assess the model and to know the fitness of model we introduced (rms) which comes to be **4.22** for our model which shows lot of improvisation can be done in the model to make more accurate predictions of housing prices.

**DISCUSSION**

In this dataset it has been discussed that not all the attributes are affecting the model performance that means there are various attributes which are not required in the dataset and can be excluded.

Other than this, variables like number of rooms and zone majorly affects the model performance which can be cited using correlation and heat map.

**REFERENCES**

<https://www.kaggle.com/datasets>

<https://www.analyticsvidhya.com/>

https://elitedatascience.com/