Building a random forest regression model to predict the housing prices using Boston dataset

Student Name: Shanmugapriya Murugavel

Student No: R00195696

For the module COMP9060 – Applied Machine Learning as part of the Project 2

Master of Science in Data Science and Analytics, Department of Mathematics

DECLARATION:

I hereby certify that this material which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent, that such work has been cited and acknowledged within the text of my work.

I understand that my project documentation may be stored in the library at CIT and may be referenced by others in the future.

Table of Contents

CHAPTER 1:

OBJECTIVE

The main objective of this assignment is to **build a random forest regression model to predict the median house value** for the Boston homes that are present in the given dataset. The dataset provided is the **housing dataset** which originates from the **UCI Machine Learning Repository**. The Boston housing data was **collected in 1978** which contains details of houses in the **Boston city, Massachusetts state of USA**. There are **506 entries** representing aggregated data about **14 home features** from various suburbs of the city. The aim is to estimate the house value based on the information provided.

OVERVIEW:

Pre-processing is initially carried out where the emails from the dataset are **cleaned to get rid of** the **special characters**, numerical values, single letters, **empty and duplicate** values. Further, **an exploratory data analysis is carried** out to educe insights to better understand the housing data. Followed by this, a regression model is designed using the **random forest** methodology to predict the median house value. **Feature engineering** and **Hyperparameter tuning** is also carried out to better understand the model and perform better. In the end, the model is tested, criticised and some future work is explained.

MOTIVATION:

A home price prediction model can be a very useful tool for both sellers and buyers, as it can help them make well-informed decisions. It may assist sellers in determining the average price at which they can list their home for sale, while it may assist buyers in determining the appropriate average price at which to purchase the home.

CHAPTER 2: INTRODUCTION

The accuracy and the predictive ability of the model are accessed and educated. This is later evaluated on the data collected from homes in the Boston suburbs for this project. Only then a good-fitting model, fully trained on this data can be used to make certain predictions like the monetary value. This model would prove to be invaluable for someone like a real estate agent who could make use of such information daily.

OVERVIEW OF THE DATASET:

The Boston Housing dataset is derived from the data collected by the United States Census Bureau on housing in the Boston MA state. The dataset can be described as follows:

- 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 centres
- RAD - index of accessibility to radial highways
- TAX - full-value property-tax rate per $10,000
- PTRATIO - pupil-teacher ratio by town
- B - 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
- LSTAT - % lower status of the population
- MEDV - Median value of owner-occupied homes in $1000's

A new product for the housing market is one possibility for this machine learning model. Property Websites (such as Daft) will use this service to recommend a price to a customer adding a new listing (e.g., apartment for sale) focused on the property's specifics

This can act as a proof of concept (PoC)

Some of the functionality can be as follows:

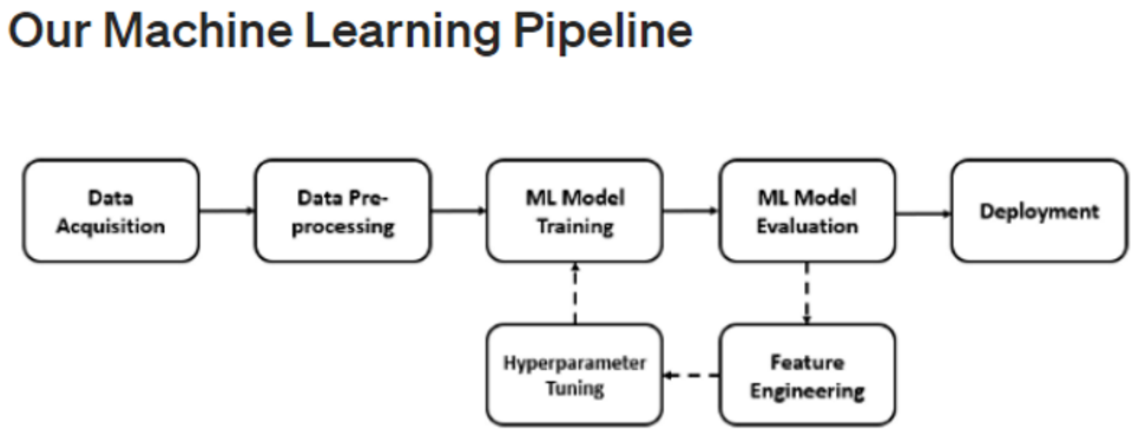
1. Train a model to predict the price based on the Boston Housing dataset

2. A REST API that can make predictions using that model, returning suggested price based on the provided characteristics of a given property.

CHAPTER 3: RESEARCH

The Machine Learning pipeline can be broadly summarised into the following segments:

1. Data Acquisition
2. Data Pre-Processing and Exploratory Data Analysis
3. Creating a Base Model
4. Feature Engineering
5. Hyperparameter Tuning
6. Final Model Training and Evaluation
7. Deployment



Our Machine Learning Pipeline

Fig 1: The above image represents the machine learning pipeline of the model

CHAPTER 4: METHODOLOGY

Data Exploration

In this first segment of the project, a cursory inquiry is performed into Boston housing data then the results were incorporated. An exploratory process is a good way to familiarize with the data and better understand and justify the conclusions. The dataset is divided into feature and target variables to build a working model that can predict the value of houses. 'RM', 'LSTAT', and 'PTRATIO' are the features that provide quantitative details about each data point. The goal and 'MEDV' variables are used to forecast the target variable. These are held in the features and prices sections, respectively.

Implementation: Statistics Calculation

The descriptive statistic of the dataset is first calculated for the Boston housing prices. The *NumPy* library is used to perform the necessary calculations. These statistics will be extremely important later to analyse various prediction results from the constructed model. Calculate the minimum, maximum, mean, median, and standard deviation of 'MEDV', which is stored in prices. Store each calculation in their respective variable.

Hyper Parameter Tuning

A hyper parameter is used to build one using two different methods: GridSearchCV and RandomizedSearchCV. The Boston Housing Dataset, which can be downloaded from Kaggle, will be used in this experiment. The model is constructed with default parameters first, then a hyperparameter tuning approach is used to improve the model's efficiency and compare the results.

Developing The Model

The tools (Python) and techniques (Random Forest) required to develop the model to make a prediction are developed in the second half of the project. Using these methods and techniques to make reliable assessments of each model’s results helps to significantly strengthen the prediction’s trust.

Implementation: Shuffle and Split Data

The Boston Housing dataset is to be split into training and testing subsets for model implementation. The data is generally shuffled before the split is performed on the dataset. This splitting process also eliminates any bias in the dataset's ordering.

For performing this split, *train_test_split* from *sklearn_cross_validation* is used. This splits the dataset features into training and testing sets. Here the split is done like 75% of the data is used for training the model and the remaining 25% is used for testing the model. The *random_state* ensures the results are consistent.

Establishing A Baseline Model

A baseline model is established to compare the refined models on the following criteria:

i) Determining performance metrics using the Mean Absolute Percentage Error (MAPE)

ii) Model Improvement: Implemented through Feature Reduction (reduces runtime without affecting the performance)

iii) Limiting Features: Finding number of features for cumulative importance of 95%

iv) Train and evaluate on important features: Only the important features are considered for training the model and make predictions accordingly on test data

v) Hyper parameter tuning: Examining the default Random Forest to determine parameter

Saving The Best Fit Model

To save the best model that best fits the designed regression algorithm, the pickle function can be used.

CHAPTER 5: EVALUATION

Implementation: Define a Performance Metric

It is difficult to assess a model's consistency without quantifying its success during training and testing. This is usually accomplished using a performance metric, such as measuring error, goodness of fit, or some other useful calculation. The coefficient of determination R^2 is used to measure the success of the model in this project. In regression analysis, the coefficient of determination for a model is a useful statistic since it also reflects how "strong" a model is at making predictions. The percentage of squared correlation between the expected and real values of the target variable is captured by R^2, which ranges from 0 to 1. A model with an R^2 of 0 will never be able to predict the goal variable, while a model with an R^2 of 1 will always be able to do so. Any number between 0 and 1 shows how much of the goal variable can be described by the features using this model. A negative R^2 may also be assigned to a model, indicating that it is no better than a model that naively predicts the target variable's mean.

CHAPTER 6: CONCLUSION AND FUTURE WORK

In the end, the accuracy of the final model has increased to 89.10% from Baseline model performance: accuracy of 88.96971%.

Future work: The pickle file can be deployed in cloud and an application can be created through a REST API.

REFERENCES:

[1] Boston Home Prices Prediction and Evaluation
https://www.ritchieng.com/machine-learning-project-boston-home-prices/

[2] Predicting Housing Prices Using Scikit-Learn's Random Forest Model:
https://towardsdatascience.com/predicting-housing-prices-using-a-scikit-learns-random-forest-model-e736b59d56c5

[3] Guide to Hyperparameters Tuning using GridSearchCV and RandomizedSearchCV
https://analyticsindiamag.com/guide-to-hyperparameters-tuning-using-gridsearchcv-and-randomizedsearchcv/