Introduction

The dataset for this project originates from the UCI Machine Learning Repository. The Boston housing data was collected in 1978 and each of the 506 entries represent aggregated data about 14 features for homes from various suburbs in Boston, Massachusetts. The objective of the Boston Housing Study was to extend previous study (examine the effect of air pollution on housing prices, controlling for the effects of other explanatory variables) with **Random Forest** and ***Gradient boosting*** modeling technique.

summary and problem definition for management

This project as a previous research focuses on ***evaluates the performance and predictive power of a model that has been trained and tested.*** ***Random Forest*** and ***Gradient boosting*** among most effective machine learning algorithms that relay on much smaller assumptions set, capable of fitting complex dataset, do not require feature scaling or centering. Both modeling method might produce better prediction results, with less data preparation efforts and less compute consumption.

measurement and statistical methods

The Python’s *Pandas* and *Numpy* for data handling, and *Scikit* Learn for machine learning and model evaluation metrics. The housing data was presented to us as a CSV file and loaded into the program using Pandas. For the rendering graphs the [graphviz](http://www.graphviz.org/) have been download and installed. All regression modeling methods selected from previous research: ***linear regression, stochastic gradient descent, ridge regression, lasso regression, and elastic net***. Also, **Random Forests** and Gradient Boosting were used. For evaluate the scikit-learn a cross-validation and root mean-squared error (RMSE) as an index of prediction error were used.

Decision Tree also was used as alternative to versions of random forests and gradient boosting.

Exploratory Data Analysis Methods

506 entries represent aggregated data about 14 features for homes from various suburbs in Boston, Massachusetts. The basic statistics information for data set vectors are presented below.



The response variable was presented in two flavors :

* the median value of homes in thousands of dollars
* the log median value of homes in thousands of dollars

|  |  |
| --- | --- |
|  |  |

**Data Visualization and Correlation**

Data Visualization of features, correlation matrix and pair plot could be reviewed in the previous research document.

Overview of Programming Work

In order to compare Random Tree and Gradient bosting results with already used ***Market Linear, Ridge, Lasso, ElasticNet*** new regressors RandomForestRegressor , ExtraTreesRegressor, RradientBoostingRegressor have been added to the regressors array

1. names = ['Linear\_Regression', 'Ridge\_Regression', 'Lasso\_Regression',
2. 'ElasticNet\_Regression',
3. 'RandomForest\_Regression',
4. 'ExtraTrees\_Regression',
5. 'GradientBoosting\_Regressor']

There are also was an effort to estimate hyperparameter values. The most common way to do this is simply make a bunch of models with different settings, evaluate them all on the same validation set, and see which one does best or use automated methods to do this process in Skicit-learn.

Results and Recommendations

**Learning hyperparameters evaluation**

The following graphs for a decision tree model with different maximum depths. Each graph visualizes the learning curves of the model for both training and testing as the size of the training set is increased. The shaded region of a learning curve denotes the uncertainty of that curve (measured as the standard deviation). The model is scored on both the training and testing sets using R2, the coefficient of determination.



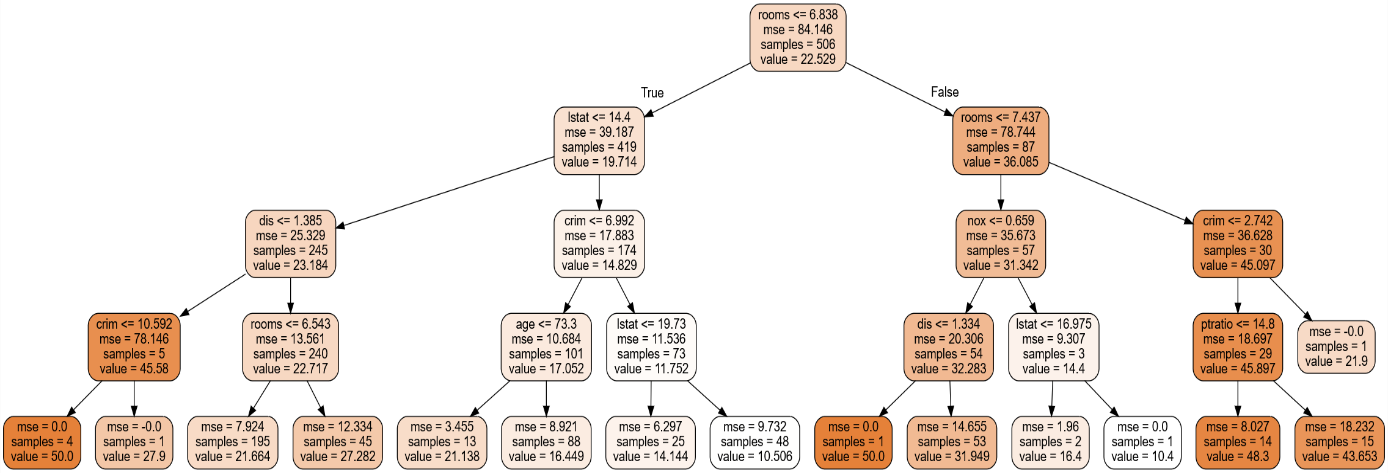
1st and 4th graphs represent biased model and the model suffered from variance respectively. On the first graph with maximum depth 1 training and testing errors converge and are quite high (~0.45). When more training points are added, the scores of the training and testing curves don't improve - no matter how much data we feed it, the model cannot represent the underlying relationship and therefore has systematic high errors.

**Results**

The results from the 10-fold cross-validation in standardized units for selected models depicted in the following grid

|  |  |
| --- | --- |
| **Method** | **Root mean-squared error** |
| Linear\_Regression | 0.56194 |
| Ridge\_Regression | 0.560511 |
| Lasso\_Regression | 0.587381 |
| ElasticNet\_Regression | 0.568084 |
| randomForest\_Regression | 0.440416 |
| ExtraTrees\_Regression | 0.422097 |
| GradientBoosting\_Regressor | 0.230492 |

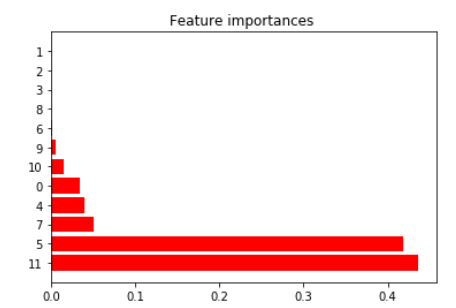
The comparison matrix shows that Gradient boosting with hyperparameter max depth =9 algorithm produces most accurate result. Having said this, this result might be an indication of overfitting. Based on the learning performance and complexity research, the recommendation is to reduce max depth to 3



**Feature importance for the model**

1. For the following columns array(['crim', 'zn', 'indus', 'chas', 'nox', 'rooms', 'age', 'dis', 'rad',
2. 'tax', 'ptratio', 'lstat', 'mv'], dtype=object)

the feature importance feature depicted in the following graph



Recalculated efficiency produces following results.

R^2= 0.78338829102524075

RMSE= 0.3620568426928081