Chongqing University of Science and Technology

Graduation Project (Dissertation)

Title: Housing Prices Prediction Using Machine Learning Algorithm

College: School of Intelligent Technology and Engineering

Major: Computer Science & Technology

Student Name: SOTON ABU RAYHAN

Student Number: 2017490105

Supervisor: Professor Jie Li

Reviewer: Professor Peng Jun

24 / 05 / 2021

Declaration of originality of students' graduation project (dissertation)

I declare with my reputation: the submitted graduation project (dissertation) is the design (research) work and results obtained under the guidance of the supervisor. The project (dissertation) quotes literature, data, drawings and materials from other people have been clearly marked, the conclusions and results in the dissertation are completed independently by myself, and do not include the achievements of others and the use of their materials for obtaining degrees or certificates from Chongqing University of Science and Technology or other educational institutions. The participant who works with me have clearly stated in the paper and expressed gratitude for any contributions to this design (research).

Author: Soton Abu Rayhan

Date: 01/06/2021

Abstract

Machine learning a major role from past years in image detection, spam reorganization, normal speech command, product recommendation and medical diagnosis. Present machine learning algorithm helps us in enhancing security alerts, ensuring public safety and improve medical enhancements. Machine learning system also provides better customer service and safer automobile systems. In the present papers I discuss about the House prices Prediction using machine learning algorithm. For the selection of prediction methods, I look at and investigate different forecast techniques. I use Linear Relapse as our model as a result of its versatile and probabilistic strategy on the model determination.result exhibit that our approach of the issue needs to be successful, and has the ability to process predictions that would be comparative with other house cost prediction models. More over on other hand housing value indices, the advancement of a housing cost prediction that tend to the Ann cement of real estate policies schemes. This study utilizes machine learning algorithms as a research method that develops housing price prediction models. We create a housing cost prediction model in view of machine learning algorithm models for example, XGBoost, Linear Regression, Decision Tree and Random Forest Regression on look at their order precision execution. We in that point recommend a housing prices prediction model to support a house vender or a real estate agent for better information based on the valuation of house. Those assessments show that the rope relapse calculation, considering exactness, dependably beats substitute models in the execution of housing prices prediction.

Keywords: Machine learning algorithm, Random Forest Regression, Decision Tree, Hosing Prices prediction, Multiple linear regression .

摘要

过去几年中，机器学习在图像检测，垃圾邮件重组，正常语音命令，产品推荐和医学诊断中发挥着重要作用。当前的机器学习算法可帮助我们增强安全警报，确保公共安全并改善医疗状况。机器学习系统还提供更好的客户服务和更安全的汽车系统。在本文中，我讨论了使用机器学习算法的房价预测。为了选择预测方法，我比较并探索了各种预测方法。我将线性回归用作我们的模型，因为它在模型选择方面具有适应性和概率性。结果表明，我们的问题解决方法需要成功，并且能够处理可以与其他房屋成本预测模型进行比较的预测。另一方面，住房价值指数的提高，使住房成本预测趋向于房地产政策体系的安定。这项研究利用机器学习算法作为开发住房价格预测模型的研究方法。我们根据机器学习算法模型（例如 XGBoost，线性回归，决策树和随机森林回归）在其订单精度执行上创建房屋价格预测模型。在这一点上，我们建议一种房价预测，以支持房屋供应商或房地产经纪人，以基于房屋的估值获得更好的信息。这些检查表明，在准确性方面，套索回归算法在执行住房价格预测方面可靠地胜过了其他模型

关键词：机器学习算法，随机森林回归，决策树，居屋价格预测，多元线性回归

Contents

List of Figures

1 Introduction

House Price Prediction is generally used to evaluate the movements in housing price. Since housing price is emphatically corresponded to different factors like area, region, populace, it requires other data separated from HPI to anticipate singular Housing prices. There has been a significantly enormous number of papers embracing customary Machine learning ways to deal with anticipate lodging costs precisely, yet they once in a while worry about the presentation of individual models and disregard the less famous yet complex models. Thus, to investigate different effects of highlights on forecast techniques, this paper will apply both conventional and progressed Machine learning ways to deal with examine the distinction among a few progressed models. This paper will likewise completely approve different procedures in model execution on relapse and give a hopeful outcome to Housing value forecast.

I will create and assess the exhibition and the prescient force of a model prepared and tried on information gathered from houses in Boston USA rural areas. The informational index utilized in this venture comes from the UCI Machine Learning Repository. This information was gathered in 1978 and every one of the 506 sections addresses total data about highlights of homes from different rural areas situated in Boston USA.

I will assess the presentation and prescient force of a model that has been prepared and tried on information gathered from homes in rural areas. A model prepared on this information that is viewed as a solid match could then be utilized to make certain expectations about a home specifically, its money related worth. This model would end up being priceless for somebody like a realtor who could utilize such data consistently.

Once we get a good fit, I will use this model to predict the monetary value of a house located at the USA area. A model like this would be very valuable for a real estate agent who could make use of the information provided in a daily basis. Commonly, House Price Index (HPI) is used to measure price changes of residential housing in many countries, such as the US Federal Housing Finance Agency HPI, S&P/Case-Sheller price index, UK National Statistics HPI, UK Land Registry's HPI, UK Halifax HPI, UK Right move HPI and Singapore URA HPI. The HPI is a weighted, repeat- sales index, meaning that it measures average price changes in repeat sales or re-financings on the same properties. This information is obtained by reviewing repeat mortgage transactions on single-family properties whose mortgages have been purchased or sensitized by Fannie Mae or Freddie Mac since January 1975.

With some analytical tools, it allows housing economists to estimate changes in the rates of mortgage defaults, prepayments, and housing affordability in specific geographic areas [1]. Because HPI is a rough indicator calculated from all transactions, it is inefficient to predict the price of a specific house. Many features such as district, age, and the number of floors also need to be considered instead of just the repeat sales in previous decades.

In recent years, due to the growing trend towards Big Data, machine learning has become a vital prediction approach because it can predict house prices more accurately based on their attributes, regardless of the data from previous years. Several studies explored this problem

and proved the capability of the machine learning approach [2],[3],[4]; however, most of them compared the models' performance but did not consider the combination of different machine learning models. S. Lu et al. did conduct an experiment using a hybrid regression technique on forecasting house price data, but it requires intensive parameter tuning to find the optimal solution [5]. Due to the importance of model combination, this paper adopted the Stacked Generalization approach [6],[7], a machine learning ensemble technique, to optimize the predicted values.

The paper is structured as follows: Section 2 illustrates the details of the methodology; Area 3 analyzes the outcomes; and Section 4 discusses the results, draws a conclusion, as well as proposes further potential directions to study the problem. According to a Zillow research in 2016, if a house is priced above its true market valuation, it tends to stay on the market five times longer compared to a house that is well-priced, suggesting a string penalty for overpricing houses [19]. Moreover, the same research suggests that houses that have been on the market for two months can lose 5% of its original listed price. Sabered and Huffman (1993) also supports the theory of a reversed correlation between a house's time on the market and its final sold price [1]. Therefore, the second question of this project is whether my models can get rid of this overestimation problem. The final question of this project is what the most important factors affecting housing prices are. In order to answer the three questions listed above, this project proposes using both the hedonic pricing model and various machine learning algorithms.

2. Literature Survey

Simran's, Macpherson and Zietz (2005) provides a study of 125 papers that use hedonic pricing model to estimate house prices in the past decade [16]. The paper provides a list of 20 attributes that are frequently used to specify hedonic pricing models. This dataset contains 12 attributes on this list. Moreover, Sirmans, Macpherson, and Zietz (2005) also discusses the effects of some variables on housing price. For example, number of bath-rooms is usually positively correlated to the fifinal sale price. Out of 40 times appearing in housing price studies, this attribute has a positive effect 34 times and is statistically signififi- cant 35 times. On average, keeping other variables unchanged, an increase of 1 bathroom leads to 10% to 12% increase in the property's value. Similarly, my paper shows that, based on the dataset of sold houses in fifive counties, the number of bathroom has a statis-tically significant and positive effect on sold price. On average, an increase of 1 bathroom could increase a house's price by $15,787.

Cebula (2009) conducts an examination on the housing prices in the City of Savannah, Georgia using the hedonic pricing model [3]. The paper's data contains 2,888 single-family houses for the period between 2000 and 2005. Cebula (2009) shows that the log price of houses is positively and signifificantly correlated with the number of bathrooms, bedrooms, fifire-places, garage spaces, stories and the total square feet of the house. Additionally, the paper adds three dummy variables, MAY, JUNE, and JULY, to account for seasonable fac-tor with regards to the houses' prices. If the house is sold in May, the variable MAY is set to be equal to 1 and 0 otherwise. The other variables, JUNE and JULY are constructed in a similar fashion. The paper fifinds that the log sale prices of houses are signifificantly and positively correlated with MAY and JULY while JUNE is insignifificant. This implies that houses that are closed in May or July tends to have a higher price. Similar to Cebula (2009), my paper includes sold month of the house as dummy variables. However, these attributes do not appear to be statistically signifificant. Selim (2009) seeks to study the effects of different housing characteristics on housing prices in Turkey using two different methods: hedonic pricing model and Machine learning [15].

The paper's dataset, which was collected from the 2004 Household Budget Survey Data for Turkey, contains 5,741 observations with 46 housing characteristics. For the hedonic pricing model, the author uses the semi-log form, $\ln(P) = \beta x + u$, where P denotes the price of the house, x is the set of independent v ariables and u is the error term. As for the Machine learning model, the paper uses 2 hidden layers, with nine and four nodes for the fifirst and second layer, respectively. The results are consistent with other studies on housing price. The author fifinds that the total number of rooms, the size of the house, the heating systems, appliances such as garbage disposal, garage and pool, etc. have a signifificant and positive effect on the house price. More importantly, Selim fifinds that the machine learning network model has a lower error score than the hedonic model. When the hedonic model's mean squared error is 2.47, the same error measurement by the neural network model is 0.44. Similarly, Tay and Ho (1991/1992) compared the pricing prediction between regression analysis and Machine learning in pre-dicting apartments' prices in Singapore [18]. They found that the neural network model outperforms regression analysis model with a mean absolute error of 3.9%. Jirong, Mingcang, and Liuguangyan (2010) uses support vector machine (SVM) regres-sion to forecast the housing prices in China in between 1993 and 2002 and

in certain dis-trict in Tangshan city in between 2000 to 2002 [9]. The paper utilizes the genetic algorithm to tune the hyper-parameters in the SVM regression model. The error scores for the SVM regression model for both China and a Tangshan City's district are both lower than 4%. This indicates that the SVM regression model perform well in forecasting housing prices in China. In the Singapore's housing market, Fan, Ong and Koh (2006) uses decision tree model study the housing characteristics' effects on prices [6]. The paper concludes that the owners of 2-room to 4-room fl-flats are more concerned with the fl-flats' basic characteristics such as model type and age more than the owners of 5 or-more-room fl-flats. Moreover, owners of executive flats care more about the services characteristics such as the neighborhood location and recreational facilities than basic housing characteristics.

Before the Internet has gone into people's life, research on the relationship between consumer's information demand and housing prices was mainly based on expectations. Case and Shiller (1988) find that people have higher expectations about the increase in housing prices in Boston, where housing prices have been increasing rapidly, by conducting a survey with 886 responses from a mailing list of persons who bought homes in May 1988. These authors conduct the survey approach again in 2003 and conclude that the extreme self-confidence of consumers is the main reason for the rise in housing prices during the studied period. People typically collect all the information they can get before taking their decisions. For participants in real estate markets, they also go through such a decision-making process. Shang and Qiu (2008) conclude that the real estate is a kind of special good and the consumers are very likely to be highly involved, that is, spending a lot of time and effort searching for information. Since search engines have become an important source of information flow, searching the most relevant keywords based on their perceptions of the real estate market is very common for Internet users before they decide. Dong et al. (2014) conclude that searchengines can lower consumers' costs of collecting the housing information.

The framework from search engines data to housing prices and sales volume is shown in Figure 2. 1, elaborating how search engines are involved in the buyer's decision process. Real estate buyers typically go through an information searching stage before making their decisions. When they turn to search engines for information, Baidu and Google records search volumes of the keywords they input and shows them in the Baidu And Google Index. Knowing that real estate purchasing decisions have a direct effect on housing prices and sales volume, therefore, as the reflection of the information searching stage for their final decisions, the Google and Baidu Index data can offer insights on the housing price and sales search engines data to housing prices and sales volume is shown in Figure 1, elaborating how search engines are involved in the buyer's decision process. Real estate buyers typically go through an information searching stage before making their decisions. When they turn to search engines for information, Baidu records web crawlers are engaged with the purchaser's choice interaction. Land purchasers ordinarily go through a data looking through stage prior to settling on their choices. At the point when they go to web crawlers for data, Baidu records search volumes of the catchphrases they information and shows them in the Google and Baidu Index. Realizing that land buying choices directly affect lodging costs and deals volume, in this manner, as the impression of the data scanning stage for their official choices, the

Baidu Index information can offer bits of knowledge on the lodging cost and deals 4 volume. The time lag between the information searching stage and the final decision also.

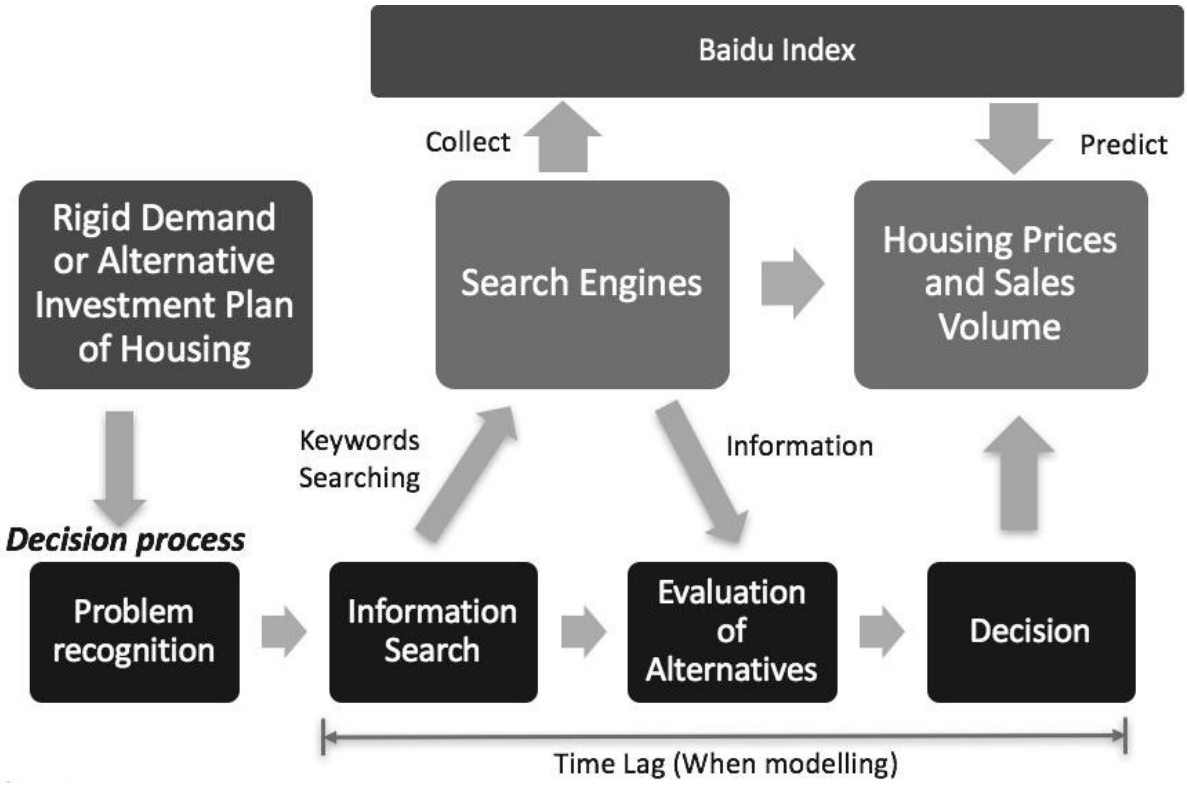


Figure 2.1: Legitimate Framework Diagram for Search Engines Data to Housing Prices and Sales Volume

Therefore, researchers in real estate start recognizing the value of using search engines data and other web search data. Kulkarni et al. (2009) test the relationship between online keyword searches and the housing price index for 20 cities in the United States, where they verify a relationship via the Granger Causality between the keywords and the housing price index. The change of certain keywords' search volume also reflects changes in demand. Pan et al. (2012) establish the link of search engine data to the demand for hotel rooms and it provides additional explanatory power to the prediction model.

They track down that the looked through watchword patterns are early pointers of purchaser's advantage and are appropriate in expectations of exercises like lodging inhabitance, consumption and the participation of occasions. Wu and Brynjolfsson (2015) infer that purchasers can improve their choice interaction utilizing the data in web crawlers. They incorporate the Google Trend's information into the forecast model in the USA housing market, finding that the lodging search list is profoundly corresponded with the home deals. From the perspective of the whole country, the average forecast error after including search data is lower, compared to the prediction model based on traditional indicators. Besides the Google Trend, Park et al. (2015) apply machine learning algorithms to study the price of Fairfax Country, in order to overcome the limitations of

assumptions and estimations of conventional statistical approaches but the study only focuses on a specific region while the location is one of the most important factors in real estate studies.

When the methodology of including search engine series in real estate research started to be used in the Chinese context, researchers also applied the Google Trend's data. However, choosing the most representative data source in the China-based studies become relevant for researchers in real estate, especially after Google's exit of its search engine service in mainland China after 2010. Though a set of literature has documented the predictive power of search engine data on real estate, these studies aim to establish the link between search engine data and housing market trend, where their keywords are determined by experience from experts or recommendations from the search engine. To narrow the range of keywords, these studies rank the keywords based on their correlations with the dependent variables. After having the range of the keywords, individual indexes can be retrieved from the search engine database when entering a single keyword, like an index with the keyword "Housing Price" or an index with the keyword "Housing Price Trend". These individual ones are the typical case for the index data. Other sets of studies choose to group these individual indexes to a composite one before they go to the actual analysis. Although using the most common keyword is the simplest way to reflect a certain category of information, having users who search a few keywords in the same category is even more common. A better reflection of behaviors in search engine trends needs to combine the patterns in some way, namely, like building an index in the stock market, construct a composite index for a category using individual indexes. For example, indexes of "Housing Price" and "Housing price Trend" are two individual indexes but one can combine them into a single composite index called "Housing Price" (pricing status).

They also document that consumers focus most on the pricing information, stating that the strongest sensitivity of Chinese consumers on housing price. Besides the pricing status, consumers are aware of financing, administrative, fiscal, policy control issues, and other general information. These studies reveal the importance of grouping keywords in different categories. Previous literature mainly constructs composite indexes using the correlation-weighed method. The construction of correlation-weighed indexes has a two-stage process. Researchers like Bai et al. (2015), Cao and Mu (2016) select in the first stage a range of keywords based on experience from experts in real estate then run each of the individual indexes with the dependent variables to find their Pearson correlation coefficients. The correlation coefficients usually have positive values as the interest of these studies is to find which keywords are representative of the corresponding category. By eliminating those without having significant correlations, they obtain a keyword set for each category.

In the second stage, they determine the weight of each keyword according to its ranking of the correlation coefficient in the same category before they are plugged into the model. In terms of modelling, past studies preliminarily construct the prediction model to test the predictability, but their model selections vary. And many of them have constraints caused by the limited number of observations from the data. Jiang et al. (2016) apply the Auto-Regressive and Moving Average (ARMA) model. Their research on Shanghai's new apartments' price reveals a positive relationship of the price trend, meaning that the housing price booms together with the search volume and adding the Baidu Index in the model increase the accuracy of 20.8 percent. Pu et al.

(2018) further compare another 6 different models in real estate prediction and propose that the Multiple Linear Regression and Random Forest Model have the average error rate of -0.11% and 0.13% respectively, using the data of Hongshan District in Wuhan city from 01-01-2011 to 31-08-2017. They find that a linear regression model with the Baidu and Google Index data is able to predict the price movement around 10-15 days in advance.

3. Design

3.1 Linear regression

The straightforward direct relapse measurable strategy permits us to sum up and study the relationship

between two ceaseless quantitative factors.

- One variable, meant x, is viewed as the predictor, informative, or free factor.
- The other variable, indicated y, is respected as the reaction, result, or ward variable.

3.2 Random Forest

Random forests or random decision forests choice timber lands are a gathering learning strategy for characterization, relapse, and different undertakings that works by developing a huge number of choice trees at preparing time and yielding the class that is the method of the classes or mean/normal prediction (relapse) of the individual trees. Arbitrary choice woodlands right for choice trees' propensity for over fitting to their preparation set.:587–588 Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

The primary calculation for arbitrary choice woodlands was made in 1995 by Tin Kam Ho utilizing the irregular subspace technique, which, in Ho's plan, is an approach to carry out the "stochastic segregation" way to deal with the order proposed by Eugene Kleinberg.

3.3 Extra Trees Regression

Adding one further advance of randomization yields incredibly randomized trees, or Extra Trees. While like conventional irregular woods in that they are a troupe of individual trees, there are two primary contrasts: first, each tree is prepared utilizing the entire learning test (as opposed to a bootstrap test), and second, the hierarchical parting in the tree student is randomized. Rather than figuring the locally ideal cut-point for each component viable (in view of, e.g., data acquire or the Gini pollutant), an irregular cut-point is chosen.

This worth is chosen from a uniform dissemination inside the element's experimental reach (in the tree's preparation set). At that point, of the multitude of haphazardly produced parts, the split that yields the most noteworthy score is picked to part the hub. Similar to ordinary random forests, the number of randomly selected features to be considered at each node can be specified.

3.4 Diction Tree

A Decision tree is a decision help apparatus that uses a tree-like model of decision and their potential results, including chance occasion results, asset expenses, and utility. It is one approach to show a calculation that just contains restrictive control articulations.

Decision trees are regularly utilized in activities research, explicitly in choice examination, to help recognize a technique destined to arrive at an objective, but on the other hand are a famous instrument in Machine learning.

3.5 Gradient Boosting algorithm.

Gradient boosting is a machine Taking in system to backslide Also course of action issues, that delivers an expectation model in the construction of a gathering from guaranteeing frail forecast models. The precision of a prescient model may be served to two different ways. Perhaps by getting a handle on trademark assembling then again. Toward applying boosting computations straight far. There is critical number boosting computations.

- Gradient Boosting
- XGBoost
- AdaBoost
- Gentle Boost etc.

Each boosting calculation need its own basic math. Additionally, a slight variety might be watched same time applying them.

Boosting computation will be a champion among those The larger part skilled Taking in musings familiar in the last one twenty quite a while. It may have been planned to arrange issues, yet everything it tends to be created should backslide as well. The inspiration to gradient boosting might have been a technique. That combines those outputs about large portions “weak” classifiers to process a capable “committee” a powerless classifier (e. G. Choice tree) will be person whose slip rate is main superior to irregular guessing

4. Materials and Methods

4.1 Data Source

The data source and data composition were described in this chapter. The data source is the house property data, house price data and urban symmetrical traffic data in the main urban area of Boston, America. The spatial projection circulation of information in a versatile, extensive GIS stage (ArcGIS) programming is appeared in Figure 4.1.(a) and (b) The fundamental assortment strategy for the house property information is to catch the data of Lianjia (a house data distributing site), and the assortment time is September 2019. For lodging value information, the fundamental assortment strategy is to acquire data from Anjuke (a lodging data administration stage), and the assortment time is September 2019. The information of metropolitan even fundamental street organization, metropolitan transport, and metro of metropolitan balanced traffic information are gotten primarily through the application programming interface (API) interface of Gaud map, and the information assortment time is August 2019

4.1 (a) 4.1 (b)

Figure 4.1(a) shows the road network data map of Boston City, including urban main roads, auxiliary roads and branch roads. Figure 4.1(b) shows the property data

of houses in the urban area with the road network as the base map. Figure 4.2 (c), and (d) respectively project and mark the bus and metro data of Boston city. The acquisition of these data provides data

Figure 4.2 Data spatial distribution in Boston. (**c**) Bus spatial distribution. (**d**) Metro spatial distribution.

support for the later construction of characteristic indicators. Boston housing attribute data is selected from all the housing attribute data in the main urban area of Boston, including the internal attribute data, location data, etc. In this study, the initial housing attribute data collection amount reached 79,457 pieces of data. The amount of data used in this study was 29,182 pieces after removing the duplicate data, error data and missing information data. It provides a huge data support for the subsequent building of house price forecast model and lays a data foundation for the model research by establishing urban housing data table.

Urban traffic data can be divided into three categories: urban basic road network data table, urban bus data table, and urban metro data table. Urban traffic data has such characteristics of large amount and wide range. In this study, in order to collect data of Boston road network, the number of road network nodes has reached 210,001, the number of bus stations has reached 21,096, and for the metro data, 89 stations of 4 lines are collected. The collection of these data is mainly to provide data support for the establishment of traffic accessibility indicators in the next section for further analysis and research.

4.2 Analysis Framework

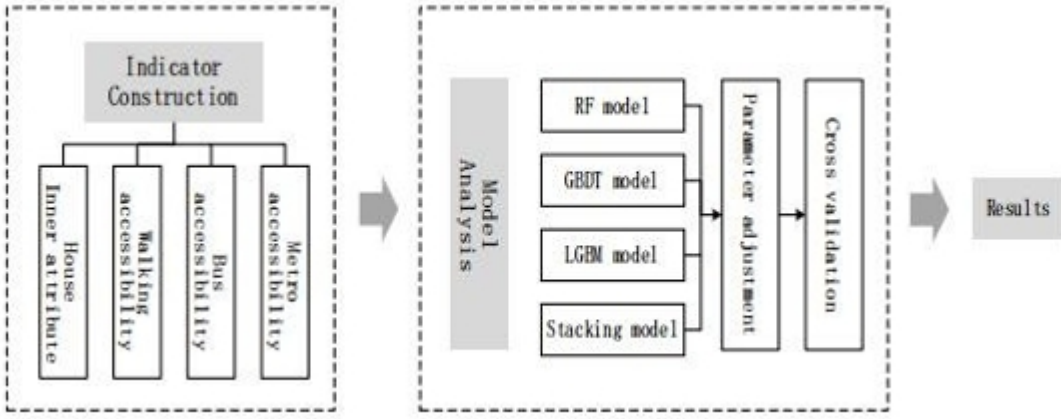


Figure 4.3 Flow chart of Housing price prediction

From the flow chart of house price prediction (Figure 4.3), it can be seen that this method has two advantages: one is to introduce traffic accessibility index, take walking, bus and metro as the carrier of urban spatial network, and analyze the causes of house price in the whole city. The other is to use a variety of machine learning algorithms to build a house price prediction model to ensure the prediction accuracy of the model.

4.3 Walking Accessibility

Walking accessibility reflects the convenience of people walking in the city. It refers to the measurement value of all point-to-point mobile walking in the road network calculated by the space syntax theory when the whole road network is accessible by walking.

First of all, we need to obtain the basic road network map data of Boston city. In order to ensure the connectivity of the map, we need to break the line segment.

Figure 4.4. Topologically connected road network diagram

In this paper, we break the line segment according to the distance of 100 m. In addition, we also break the line at the intersection of the road. In this way, all roads in Boston are connected. Next, the network topology of road network map is carried out in ArcGIS software to ensure that any two points on the map can be connected along the road. Figure 4.4 shows the basic road network and the connected road network after topology in Boston. In Figure 4.4, six points are randomly selected on the road network map to solve the problem, and it is found that the path planning along the map route can be achieved. All lines can be connected after the road network is broken.

Next, we need to convert the map to an axis map. The activity of this progression is finished in Depth map programming (British Space Syntax Ltd), and the changed hub map is appeared in Figure 4.5

Figure 4.5 Axis map of Boston.

After the transformation into the axis map, the integration calculation, that is, the walking accessibility calculation is needed. The calculation of walking accessibility is affected by the search radius. In this study, the radius range is selected as 1000–10,000 m, in which every 200 m is calculated. The calculation formula is as follows:

$$W_i = \frac{n\left[log_2\left(\frac{n+2}{3} - 1\right) + 1\right]}{\sum_{j=1}^{n} d_{ij}} \quad (4.1)$$

In Equation (4.1), Wi represents the walking accessibility of node i, dij represents the shortest path distance, and n represents the number of summary points in the road network. Finally, the axis map is transformed into road network map, which is brought into ArcGIS programming (American ecological frameworks research establishment, Inc.), and the trait table is opened to get the strolling availability under various sweep. Right now, the determined passerby availability list is connected to the street network guide of Boston city, at that point we need to connect it with the lodging information and decide the ideal range to get the person on foot openness of each lodging area. ArcGIS programming will be utilized to relate the qualities of walker availability and lodging information nextly.

Right off the bat, the street network guide of Boston city with strolling availability list is brought into ArcGIS programming, and the point type information or line section type information is changed into pattern surface information by bit thickness investigation work so the openness plane covering the entire guide of Boston city can be acquired. Furthermore, the openness plane is changed into lattice information, and afterward the matrix information is changed into direct information toward get ready for neighbor investigation. At last, the network defining moment information and the house trait information are dissected around there, and the strolling availability files under various radii are identified with the house property information table to get the strolling openness qualities of various radii under each house area.

In the above calculation, there are several groups of walking accessibility values under different search radius, so it is necessary to determine the optimal search radius. In this paper, the Pearson correlation coefficient between house price and walking accessibility is calculated to judge, and the highest coefficient is selected as the best search radius and walking accessibility.

$$p_{(X,Y)} = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{X_i-\overline{X}}{\sigma_X}\right)\left(\frac{Y_i-\overline{Y}}{\sigma_Y}\right) \tag{4.2}$$

In statistics, Pearson correlation coefficient is used to measure the degree of correlation between two groups of variables [32]. The calculation formula is shown in Equation (4.2). Figure 5 shows Pearson correlation coefficient between house price and pedestrian accessibility based on spatial syntax under different radius of pedestrian accessibility.

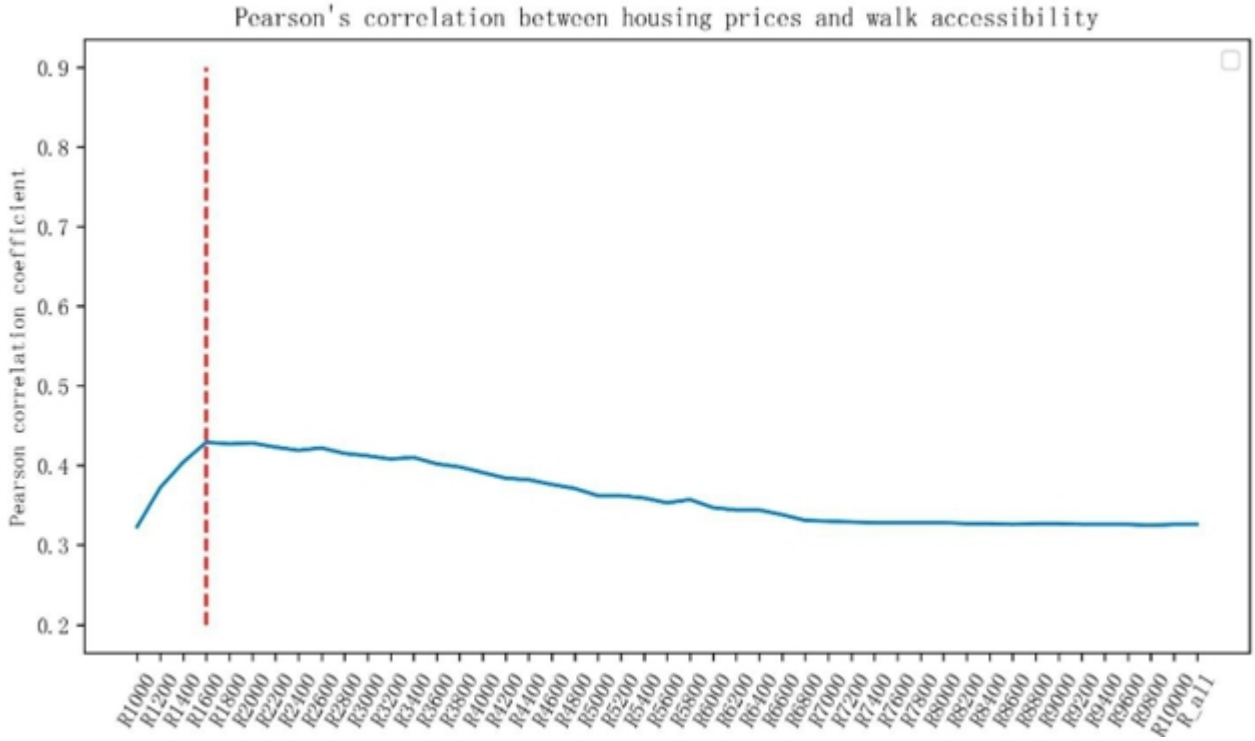


Figure 4.6. Pearson correlation coefficient graph of house price and walk accessibility under different radius.

In Figure 4.6, the openness under the most noteworthy sweep of Pearson relationship coefficient is chosen, and the best quest span for person on foot availability is resolved as R = 1600 m.

Figure 4.7. Axis map of Boston road network.

Figure 4.7 shows the axis diagram under space syntax calculation when R = 1600 m. The pedestrian accessibility under this radius is retained as the optimal pedestrian accessibility feature of the house.

Figure 4.8. Thermal map of walk accessibility.

Figure 4.8 shows the heat distribution map of Boston road network axis and pedestrian accessibility when the search radius is r = 1600 m.

4.4 Bus Accessibility

The Bus accessibility dependent on spatial grammar alludes to the estimation worth of the availability of transport stations in metropolitan street network by consolidating the spatial language structure hypothesis. The estimation strategy for transport openness and strolling accessibility is comparable.

Right off the bat, the street network guide of Boston city and the gathered bus stop information are converged in ArcGIS programming, and the convergence of bus stop and street network is utilized as the street hub to break the street organize and reproduce the organization geography. Besides, change the guide into a hub guide to compute the combination degree under various inquiry sweep. The sweep range is 1000–10,000 m, and compute each 200 m. Then, the pivot map is brought into ArcGIS modified, and afterward the quantity of lines at each bus stop is appointed as the weight.

$$B_i = l_i \frac{m\left[log_2\left(\frac{m+2}{3} - 1\right) + 1\right]}{\sum_{j=1}^{m} s_{ij}} \tag{4.3}$$

At long last, the transport accessibility 3 of each bus stop under various search radius is obtained. The calculation formula of public transport accessibility is shown in Equation (4.3).

In Equation (4.3), Bi is the bus accessibility of node i, sij represents the shortest path distance between two bus stations, and m represents the number of bus stations in the road network. li represents the number of bus lines at station i. In this paper, the definition of bus accessibility, because no specific bus line operation diagram is obtained, can only be replaced by the basic road network, so there is no way to accurately calculate the real route between the station and station, use the shortest distance of the road network to replace

At this time, we get that the bus accessibility under different radius is attached to bus station, and then we need to associate these features with the urban housing features. The association mode is the same as that in the previous section. Firstly, the calculated data of Boston bus station with bus accessibility index is imported into ArcGIS software, and the core density analysis function is applied to convert the point type data into trend surface data, so that the bus accessibility plane covering the whole Boston urban area can be obtained. Secondly, the accessibility plane is transformed into grid data, and then transform the grid data into point data to prepare for neighbor analysis. Finally, the matrix defining moment information and the house trait information are investigated in ArcGIS, and the transport openness records under various radii are identified with the house quality information table to get the transport availability attributes of various radii under each house area

Next, Still need to analyze and determine the bus accessibility under the optimal radius. The Pearson correlation analysis method is still used to calculate the Pearson correlation coefficient between the house price and the bus accessibility under different radius, and the maximum coefficient is taken as the bus accessibility under the optimal radius. Figure 4.9 shows Pearson correlation coefficient of house price and bus accessibility under different radius.

Figure 4.9. Pearson correlation coefficient graph of house price and bus accessibility under different radius.

In Figure 4.9, select the transport openness under the most noteworthy span of Pearson relationship coefficient, and afterward decide the best inquiry sweep of transport availability as R = 2200 m. Consequently, the transport openness under this sweep is held as the ideal transport availability highlight of the house.

Figure 4.10 shows the thermal distribution of bus accessibility in Boston when the search radius is r = 2200 m.

Figure 10. Thermal map of bus accessibility.

4.5 Metro Accessibility

The metro accessibility dependent on spatial punctuation is determined dependent on the Boston tram line guide and station information, joined with spatial linguistic structure hypothesis. The computation cycle is like strolling openness accessibility and bus accessibility.

Firstly, the Boston tram line map is broken at the station, and the availability map is acquired after network geography. Then transform it into an axis map, and the integration degree under different radii is calculated in the Depth map software, the radius range is 1000–10,000 m and the global range and the calculation interval is 200 m once. Next, import the inverse transformation of the axis map with the integration degree calculation into ArcGIS. Figure 4.11 shows the metro axis under the global calculation integration. The calculation results of other radii are similar. Finally, the metro accessibility of each station under different search radius is obtained.

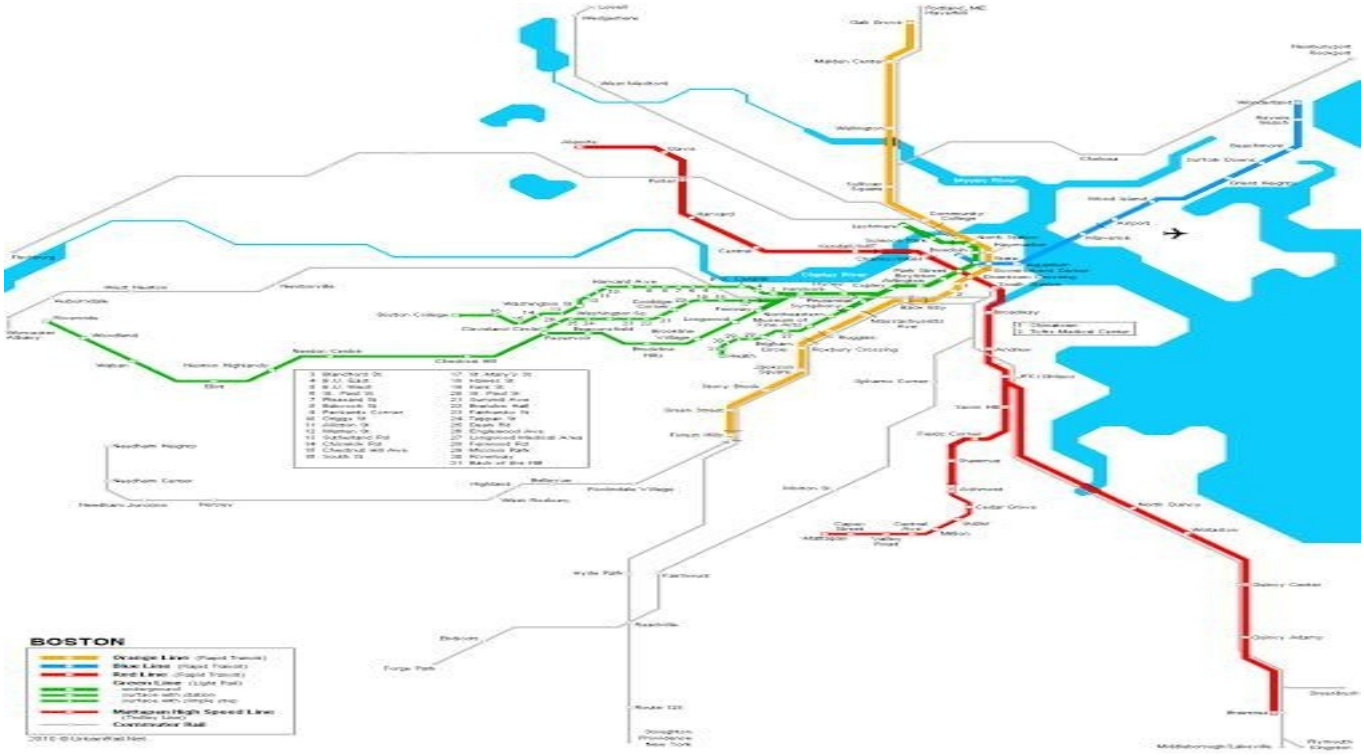


Figure 10. Axis map of metro lines.

As can be found in Figure 4.11, the dispersion of the pivot map doesn't adjust to the availability conveyance of metro lines in emotional experience. The accessibility should be higher at the passenger and exchange stations of the four lines, and the integration degree cannot truly reflect the subway accessibility. It is considered to distinguish the transfer station from the common station and give different weight values. Therefore, after the integration degree is calculated, converted back the axis map to the original map and opened in ArcGIS, the integration degree attribute value of each metro station will be obtained. Assign the weight of the station again, and the weight is the number of transfer lines at the station. The metro accessibility under different radius of subway station is obtained. The calculation formula is shown in Equation (4.4).

$$M_i = C_i \frac{k\left[log_2\left(\frac{k+2}{3}-1\right)+1\right]}{\sum_{j=1}^{k} t_{ij}} \tag{4.4}$$

In Equation (4.4), Mi represents the metro accessibility of node i, tij represents the running distance between two metro stations, and k represents the number of metro stations in the road network. Ci refers to the number of transfer lines at station i.

Like walking accessibility and bus accessibility, metro accessibility also needs to be related to house price characteristics, and the method of association still uses the nearest neighbor analysis function in ArcGIS. Firstly, import the calculated data of Boston public transport station with metro accessibility index into ArcGIS software, and apply the core density analysis function is to convert the point type data into trend surface data, so that the metro accessibility plane covering the whole Boston urban area can be obtained. Secondly, transform the accessibility plane into grid data, and then transform the grid data into point data to prepare for neighbor analysis. Finally, the grid turning point data and the house attribute data are analyzed in ArcGIS, and the metro accessibility indexes under different radii are related to the house attribute data table to get the metro accessibility characteristics of different radii under each house location.

Next, we need to determine the optimal search radius. Figure 4.12 shows Pearson correlation coefficient of house price and metro accessibility under different radii. Select the radius with the largest coefficient as the best radius.

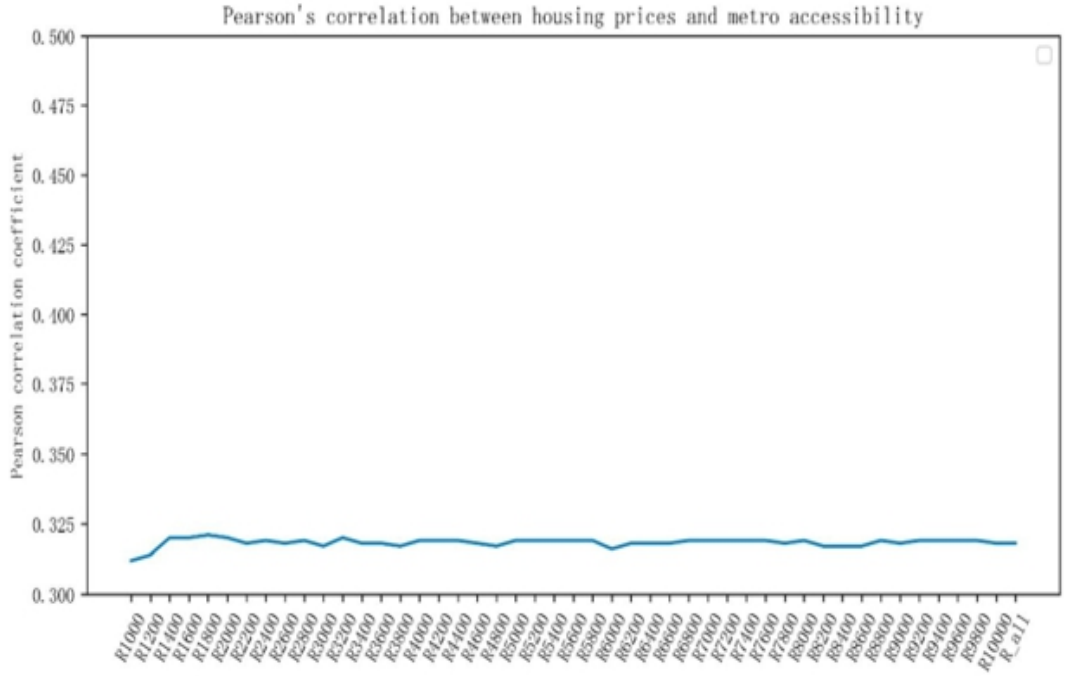


Figure 4.12. Pearson correlation coefficient graph of house price and metro accessibility under different radius.

It tends to be seen from Figure 4.12 that the Pearson connection coefficient of metro openness and house cost is steadily conveyed with low worth, which essentially doesn't change with the difference in range which is on the grounds that there are less tram lines and the dispersion of stretch distance between stations is normal, which isn't just about as perplexing as street organization and bus stop, so the change is little. As of now, Boston has opened four lines, which can't totally cover the entire metropolitan space. A large portion of the houses are far away from the tram station, so Pearson relationship esteem is low. In this paper, the worldwide degree is chosen as the last metro openness list.

5. Analysis Research status

The field of Data Science is rather young, having taken form over the last half century as a discipline distinct from statistics. It is also rapidly growing with many interesting advancements in recent years, most notably within Machine Learning (ML). This has resulted in an increase in media attention as well as funding of AI related businesses and research projects. In 50 years of data Science Donohue comments on the history of data Science and questions whether it is really different from statistics. With regards to Machine Learning, he points to a study he conducted that compared a set of highly-cited and glamorous classifier methods such as Random Forests and k-Nearest neighbor to a simple linear classifier applied on the same problem. The study found that the simpler method did not only perform similarly, but had a lower worst-case regret. This suggests that when benchmarked, more advanced ML algorithms are not necessarily better when put in practice, which highlights the need for making algorithm comparisons. A study using similar techniques was made on predicting the sales price of used cars.

This problem is similar to predicting house prices and arguably simpler because it is dealing with cars, commodities that aren't geographically fixed and are often highly standardized. The methods used were Multiple Linear Regression, k-Nearest Neighbors, Naive Bayes and Decision Trees, including Random Forest. Cross validation was used for finding the optimal hyper parameter's, such as the 'k' for k-NN. The house pricing problem was approached by Palominos et al from the viewpoint of finding investment opportunities. They formulated the regression problem and used several Machine Learning algorithms such as k-Nearest neighbor, variations of neural networks and decision trees. Another study by Oxenstierna investigated it for the purposes of valuation of houses. The data set included 5000 entries. Again, the k-Nearest neighbor method was used as well as Artificial Neural Networks, to minimize the median absolute percentage error of the prediction. The methods performed similarly at around 8-9 % Median Absolute Percentage Error.

5.1 Features Description

The Boston data frame has 506 rows and 14 columns. Each row comprises one data-point and contains details about a plot. Various features affect the pricing of a house.

The Boston housing data set has 14 features, out of which we'll use 13 to train the model. The 14th feature is the price, which we'll use as our target variable. The table gives the list of features included in the data set, along with their respective descriptions.

Features	Description
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	nitrogen 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 mean of distances to five Boston employment centers.
tax	full-value property-tax rate per $10,000
ptratio	pupil-teacher ratio by town.
black	1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town.
lstat	lower status of the population (percent).
medv	median value of owner-occupied homes in $1000s.

Dataset Table

6. Dataset Exploration

I should assess the presentation and prescient force of a model that has been prepared and tried on information gathered from homes in rural areas of Boston, Massachusetts. A model prepared on this information that is viewed as a solid match could then be utilized to make certain expectations about a home specifically, its money related worth. This model would end up being important for somebody like a realtor who could utilize such data consistently.

The dataset for this task starts from the UCI Machine Learning Repository. The Boston lodging information was gathered in 1978 and every one of the 506 sections address accumulated information around 14 highlights for homes from different rural areas in Boston, Massachusetts. For the reasons for this task, the accompanying preprocessing steps have been made to the dataset:

- 16 information focuses have an 'MEDV' worth of 50.0. These information focuses likely contain absent or controlled qualities and has been eliminated.
- 1 information point has an 'RM' worth of 8.78. This information point can be viewed as an exception and has been taken out.
- The highlights 'RM', 'LSTAT', 'PTRATIO', and 'MEDV' are fundamental. The excess non-pertinent highlights have been avoided.
- The component 'MEDV' has been multiplicatively scaled to represent 35 years of market swelling.

Run the code cell beneath to stack the Boston lodging data set, along a couple of the essential Python libraries needed for this venture. You will know the data set stacked effectively if the size of the data set is accounted for.

Boston House Prices Dataset was collected in 1978 and has 506 entries with 14 attributes or features for homes from various suburbs in Boston.

```
Boston Housing Dataset Attribute Information (in order):
        - 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
```

Import modules

In [ ]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
%matplotlib inline
warnings.filterwarnings('ignore')
```

In [2]:
```
# Load the Boston housing dataset
data = pd.read_csv('../input/housing.csv')
prices = data['MEDV']
features = data.drop('MEDV', axis = 1)

data.head()
```

6.1 Implementation: Calculate Statistics

For first coding execution, I ought to figure enlightening measurements about the Boston lodging costs. Since NumPy has effectively been imported for us, utilize this library to play out the vital estimations. These measurements ought to be critical later on to dissect different forecast results from the developed model.

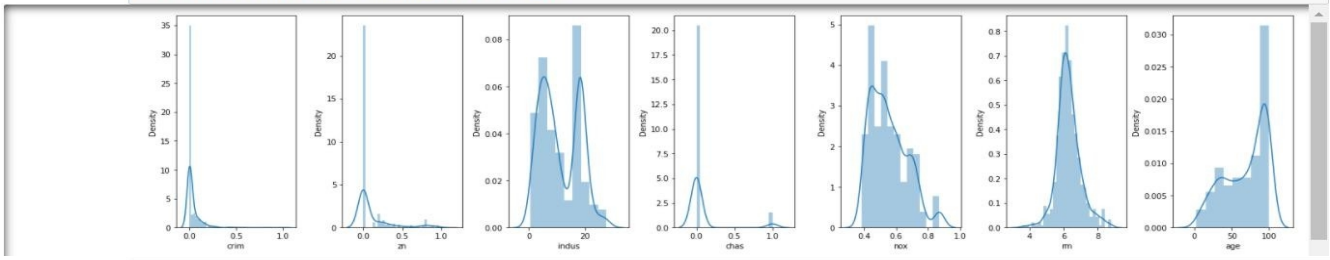
In the code cell below, I should need to implement the following:

- Calculate the minimum, maximum, mean, median, and standard deviation of 'MEDV', which is stored in prices.
- Store each calculation in their respective variable.

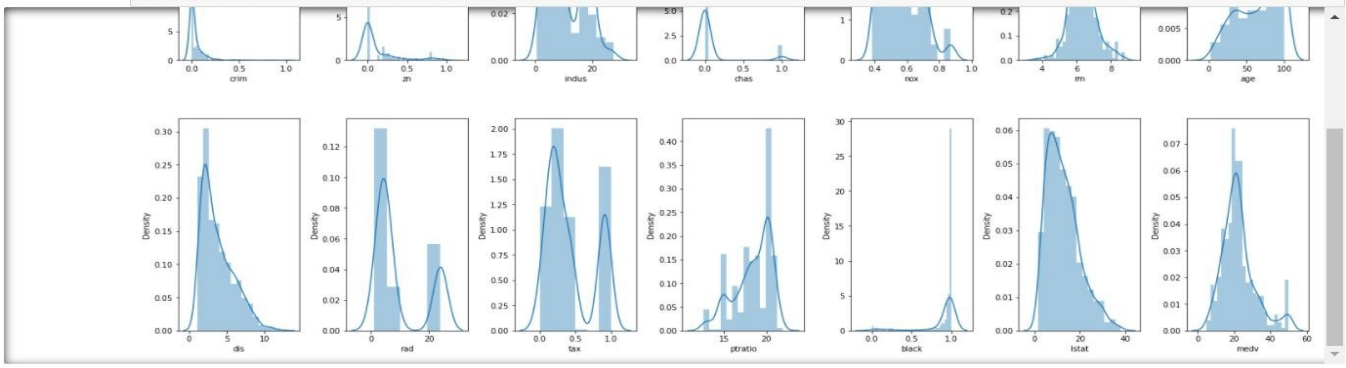
Min-Max Normalization

In [27]:
```
cols = ['crim', 'zn', 'tax', 'black']
for col in cols:
    # find minimum and maximum of that column
    minimum = min(df[col])
    maximum = max(df[col])
    df[col] = (df[col] - minimum) / (maximum - minimum)
```

In [28]:
```
fig, ax = plt.subplots(ncols=7, nrows=2, figsize=(20, 10))
index = 0
ax = ax.flatten()

for col, value in df.items():
    sns.distplot(value, ax=ax[index])
    index += 1
plt.tight_layout(pad=0.5, w_pad=0.7, h_pad=5.0)
```



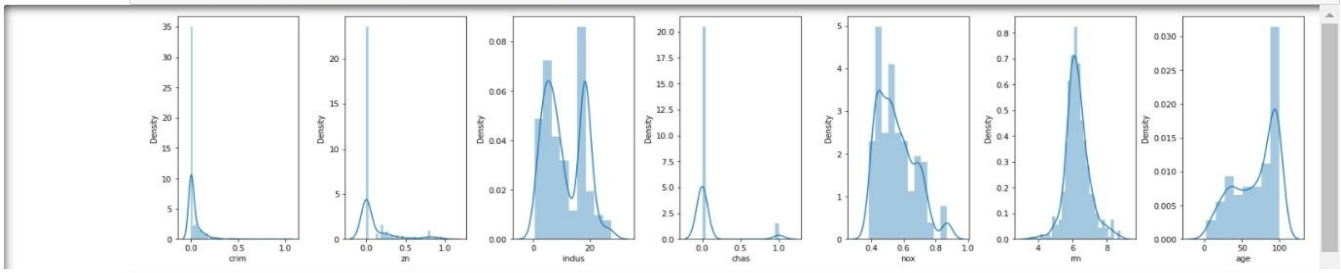
In [28]:
```
fig, ax = plt.subplots(ncols=7, nrows=2, figsize=(20, 10))
index = 0
ax = ax.flatten()

for col, value in df.items():
    sns.distplot(value, ax=ax[index])
    index += 1
plt.tight_layout(pad=0.5, w_pad=0.7, h_pad=5.0)
```



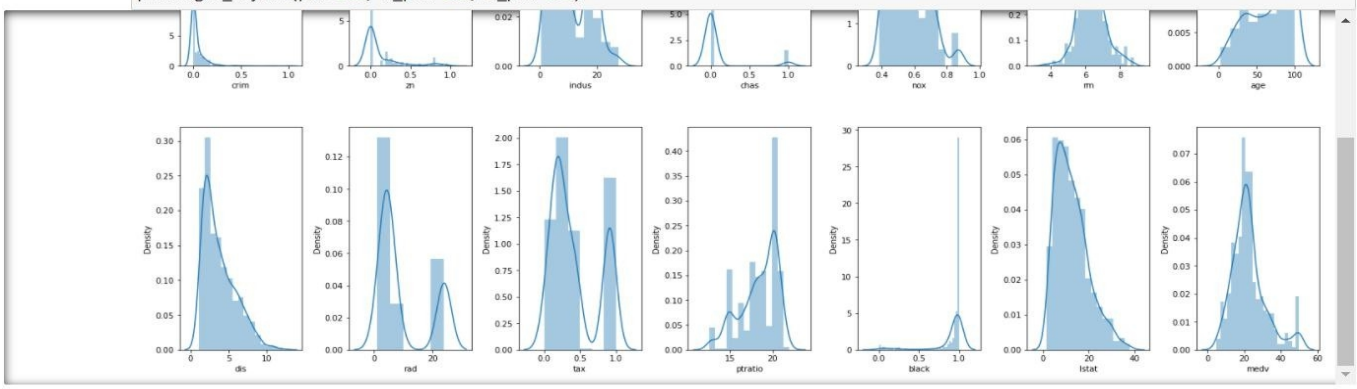
Min-Max Normalization

In [27]:
```
cols = ['crim', 'zn', 'tax', 'black']
for col in cols:
    # find minimum and maximum of that column
    minimum = min(df[col])
    maximum = max(df[col])
    df[col] = (df[col] - minimum) / (maximum - minimum)
```

In [28]:
```
fig, ax = plt.subplots(ncols=7, nrows=2, figsize=(20, 10))
index = 0
ax = ax.flatten()

for col, value in df.items():
    sns.distplot(value, ax=ax[index])
    index += 1
plt.tight_layout(pad=0.5, w_pad=0.7, h_pad=5.0)
```



In [28]:
```
fig, ax = plt.subplots(ncols=7, nrows=2, figsize=(20, 10))
index = 0
ax = ax.flatten()

for col, value in df.items():
    sns.distplot(value, ax=ax[index])
    index += 1
plt.tight_layout(pad=0.5, w_pad=0.7, h_pad=5.0)
```



In [10]:
```
# standardization
from sklearn import preprocessing
scalar = preprocessing.StandardScaler()

# fit our data
scaled_cols = scalar.fit_transform(df[cols])
scaled_cols = pd.DataFrame(scaled_cols, columns=cols)
scaled_cols.head()
```

Out[10]:

	crim	zn	tax	black
0	-0.419782	0.284830	-0.666608	0.441052
1	-0.417339	-0.487722	-0.987329	0.441052
2	-0.417342	-0.487722	-0.987329	0.396427
3	-0.416750	-0.487722	-1.106115	0.416163
4	-0.412482	-0.487722	-1.106115	0.441052

In [11]:
```
for col in cols:
    df[col] = scaled_cols[col]
```

In [12]:
```
fig, ax = plt.subplots(ncols=7, nrows=2, figsize=(20, 10))
index = 0
ax = ax.flatten()

for col, value in df.items():
    sns.distplot(value, ax=ax[index])
    index += 1
plt.tight_layout(pad=0.5, w_pad=0.7, h_pad=5.0)
```

```
-> 1527         return self.transform_path_affine(self.transform_path_non_affine(path))
   1528
   1529     def transform_path_affine(self, path):

~\anaconda3\lib\site-packages\matplotlib\transforms.py in transform_path_affine(self, path)
   1535         ``transform_path_affine(transform_path_non_affine(values))``.
   1536         """
-> 1537         return self.get_affine().transform_path_affine(path)
   1538
   1539     def transform_path_non_affine(self, path):

~\anaconda3\lib\site-packages\matplotlib\transforms.py in get_affine(self)
   2369             return self._b.get_affine()
   2370         else:
-> 2371             return Affine2D(np.dot(self._b.get_affine().get_matrix(),
   2372                                    self._a.get_affine().get_matrix()))
   2373

~\anaconda3\lib\site-packages\matplotlib\transforms.py in get_affine(self)
   2369             return self._b.get_affine()
   2370         else:
```

In my opinion, the value of 'MEDV' will be dependent on these 3 features in the following way:

1) **RM** - The more the worth of RM, the more will be the worth of 'MEDV'. Since it's really clear that with expansion in the quantity of rooms, the cost of the house should increment.

2) **LSTAT** - The more the value of LSTAT, The less should be the value of 'MEDV'. Because with increase in the percentage of "lower class" homeowners in the neighborhood, the crime rate in the neighborhood may increase. Even though LSTAT doesn't have a causal effect on the crime rate in the neighborhood, they are likely to be positively correlated. One more factor is if there are greater percentages of "lower class" homeowners in the neighborhood, then more likely very expensive real estate owners will not build their housing complexes in that region as most of the people will not be able to afford it. So in average, the houses in that region will be cheaper.

3) **PTRATIO** - The lesser the value of PTRATIO, the more will be the value of 'MEDV'. Because if the students to teacher ratio is low, then that means individual students gets much more attention from the students as opposed to a region where this ratio is high. Over there, as the number of students will be much higher than the number of teachers, teachers will not be able to attend to students individually every time and hence this may affect the education of the students. So regions with a low PTRATIO will have higher prices for houses.

6.2 Load Dataset

I should have to isolate the dataset into highlights and the objective variable. The highlights, 'RM', 'LSTAT', and 'PTRATIO', give us quantitative data about every information point. The objective variable, 'MEDV', ought to be the variable I try to foresee. These are put away in highlights and price, individually.

Loading the dataset

In [23]:
```
df = pd.read_csv("Boston Dataset.csv")
df.drop(columns=['Unnamed: 0'], axis=0, inplace=True)
df.head()
```

Out[23]:

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2

In [24]:
```
# statistical info
df.describe()
```

Out[24]:

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.(
mean	3.613524	11.363636	11.136779	0.069170	0.554695	6.284634	68.574901	3.795043	9.549407	408.237154	18.455534	356.674032	12.6
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617	28.148861	2.105710	8.707259	168.537116	2.164946	91.294864	7.1
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.900000	1.129600	1.000000	187.000000	12.600000	0.320000	1.7
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500	45.025000	2.100175	4.000000	279.000000	17.400000	375.377500	6.9
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500	77.500000	3.207450	5.000000	330.000000	19.050000	391.440000	11.3
75%	3.677082	12.500000	18.100000	0.000000	0.624000	6.623500	94.075000	5.188425	24.000000	666.000000	20.200000	396.225000	16.9
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.000000	12.126500	24.000000	711.000000	22.000000	396.900000	37.9

Loading the dataset

In [23]:
```
df = pd.read_csv("Boston Dataset.csv")
df.drop(columns=['Unnamed: 0'], axis=0, inplace=True)
df.head()
```

Out[23]:

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2

In [24]:
```
# statistical info
df.describe()
```

Out[24]:

	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000
11.363636	11.136779	0.069170	0.554695	6.284634	68.574901	3.795043	9.549407	408.237154	18.455534	356.674032	12.653063	22.532806
23.322453	6.860353	0.253994	0.115878	0.702617	28.148861	2.105710	8.707259	168.537116	2.164946	91.294864	7.141062	9.197104
0.000000	0.460000	0.000000	0.385000	3.561000	2.900000	1.129600	1.000000	187.000000	12.600000	0.320000	1.730000	5.000000
0.000000	5.190000	0.000000	0.449000	5.885500	45.025000	2.100175	4.000000	279.000000	17.400000	375.377500	6.950000	17.025000
0.000000	9.690000	0.000000	0.538000	6.208500	77.500000	3.207450	5.000000	330.000000	19.050000	391.440000	11.360000	21.200000
12.500000	18.100000	0.000000	0.624000	6.623500	94.075000	5.188425	24.000000	666.000000	20.200000	396.225000	16.955000	25.000000
100.000000	27.740000	1.000000	0.871000	8.780000	100.000000	12.126500	24.000000	711.000000	22.000000	396.900000	37.970000	50.000000

In [25]:
```
# datatype info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 506 entries, 0 to 505
Data columns (total 14 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   crim     506 non-null    float64
 1   zn       506 non-null    float64
 2   indus    506 non-null    float64
 3   chas     506 non-null    int64
 4   nox      506 non-null    float64
 5   rm       506 non-null    float64
 6   age      506 non-null    float64
 7   dis      506 non-null    float64
 8   rad      506 non-null    int64
 9   tax      506 non-null    int64
 10  ptratio  506 non-null    float64
 11  black    506 non-null    float64
 12  lstat    506 non-null    float64
 13  medv     506 non-null    float64
dtypes: float64(11), int64(3)
memory usage: 55.5 KB
```

Preprocessing the dataset

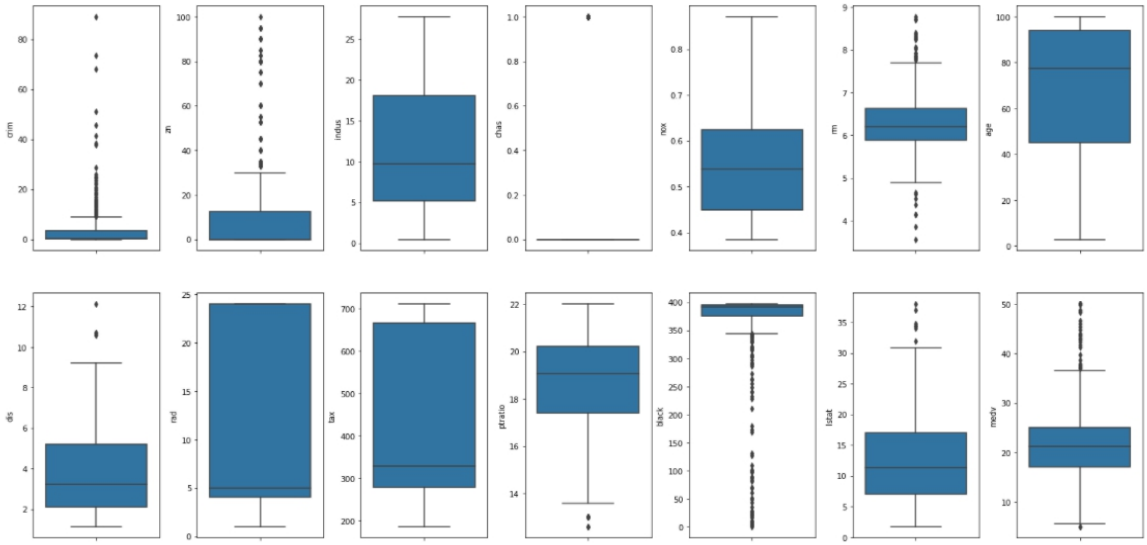
In [5]:
```
# check for null values
df.isnull().sum()
```

Out[5]:
```
crim       0
zn         0
indus      0
chas       0
nox        0
rm         0
age        0
dis        0
rad        0
tax        0
ptratio    0
black      0
lstat      0
medv       0
dtype: int64
```

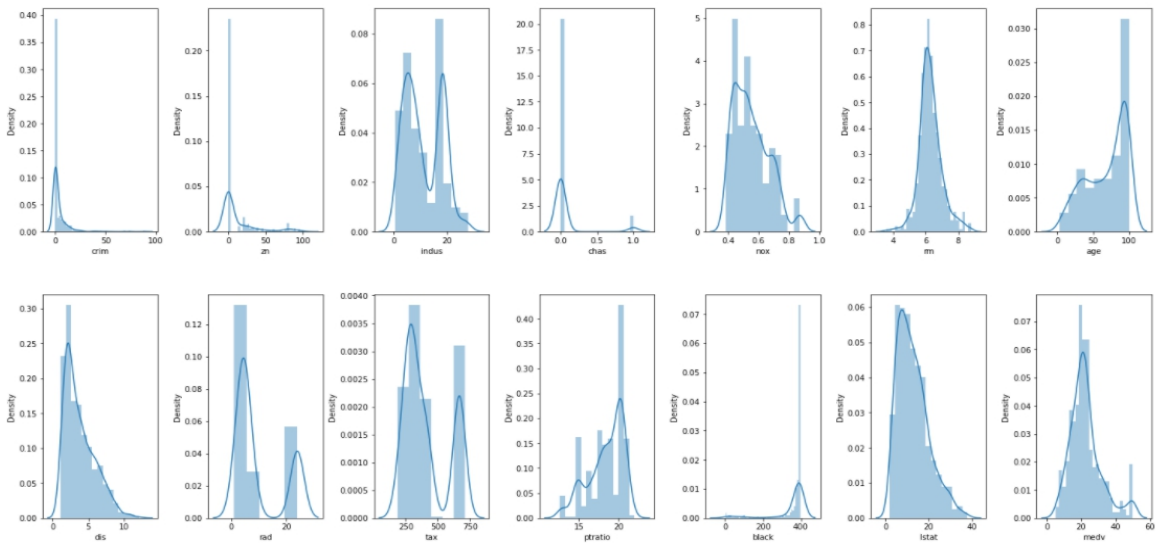
Explore Data

Exploratory Data Analysis

In [26]:
```
# create box plots
fig, ax = plt.subplots(ncols=7, nrows=2, figsize=(20, 10))
index = 0
ax = ax.flatten()

for col, value in df.items():
    sns.boxplot(y=col, data=df, ax=ax[index])
    index += 1
plt.tight_layout(pad=0.5, w_pad=0.7, h_pad=5.0)
```



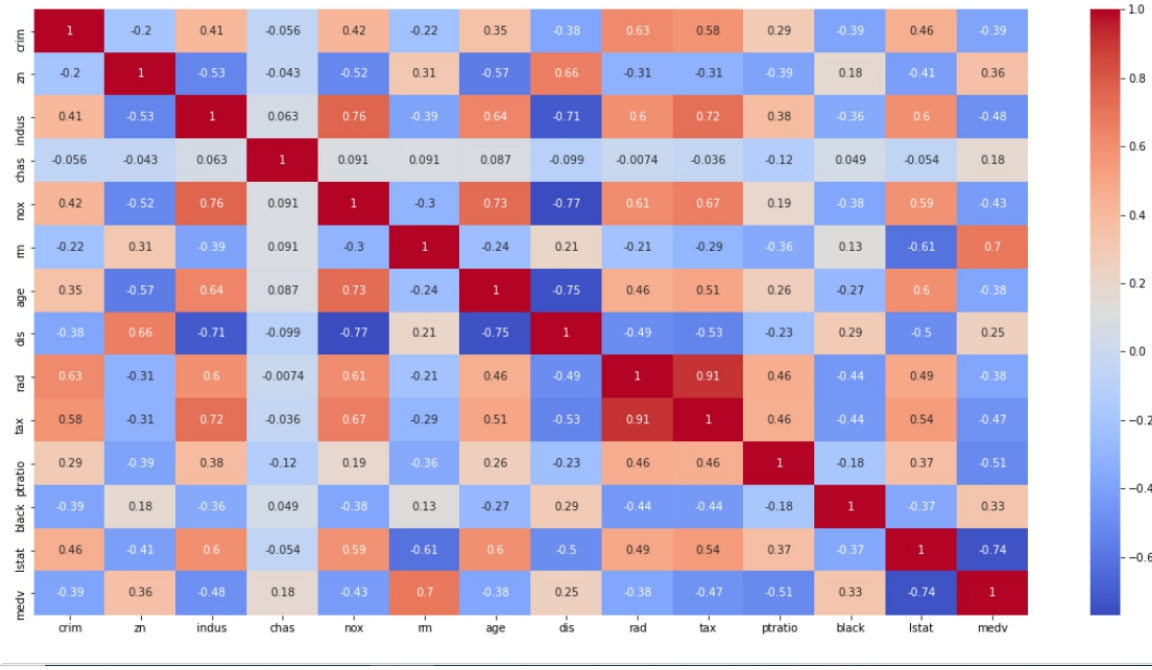
In [7]:
```
# create dist plot
fig, ax = plt.subplots(ncols=7, nrows=2, figsize=(20, 10))
index = 0
ax = ax.flatten()

for col, value in df.items():
    sns.distplot(value, ax=ax[index])
    index += 1
plt.tight_layout(pad=0.5, w_pad=0.7, h_pad=5.0)
```



Coorelation Matrix

```
In [13]: corr = df.corr()
         plt.figure(figsize=(20,10))
         sns.heatmap(corr, annot=True, cmap='coolwarm')
```

```
Out[13]: <AxesSubplot:>
```



Contingency table

This table consists of our 2 categorical variables ‘CHAS’ and ‘RAD’.

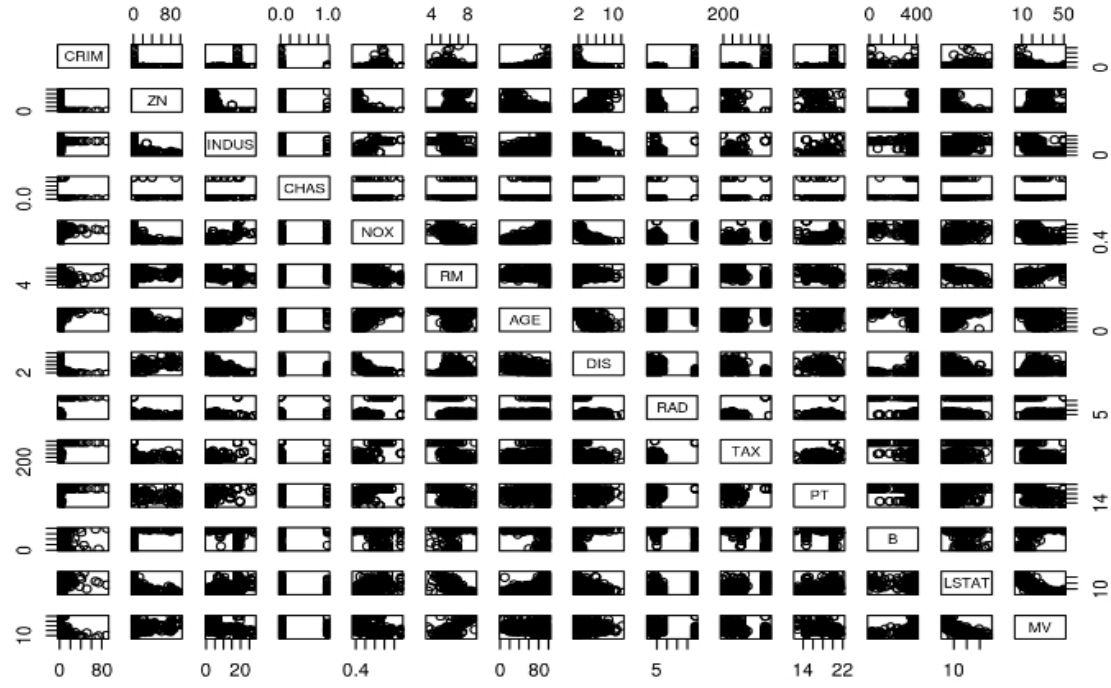
```
ftable(CHAS ~ RAD, data=boston)
```

```
##     CHAS   0   1
## RAD
## 1         19   1
## 2         24   0
## 3         36   2
## 4        102   8
## 5        104  11
## 6         26   0
## 7         17   0
## 8         19   5
## 24       124   8
```

Boxplot

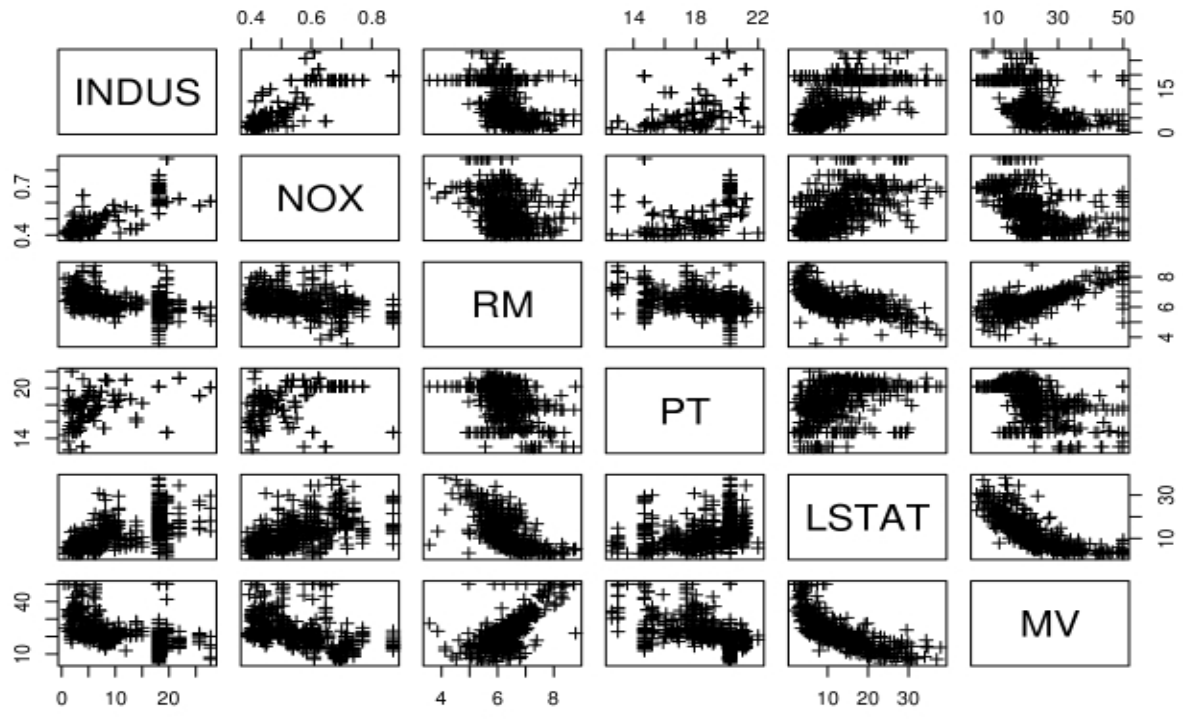
We boxplot all the variables that effect our study of the target variable ’MEDV’00

```
plot(boston)
```



Plotting dependent variables with target variable

```
plot(boston[,c(3,5,6,11,13,14)],pch=3)
```



6.3 Correlation and near zero variance

A few important properties to check now are the correlation of input features with the dependent variable, and to check if any feature has near zero variance (values not varying much within the column)

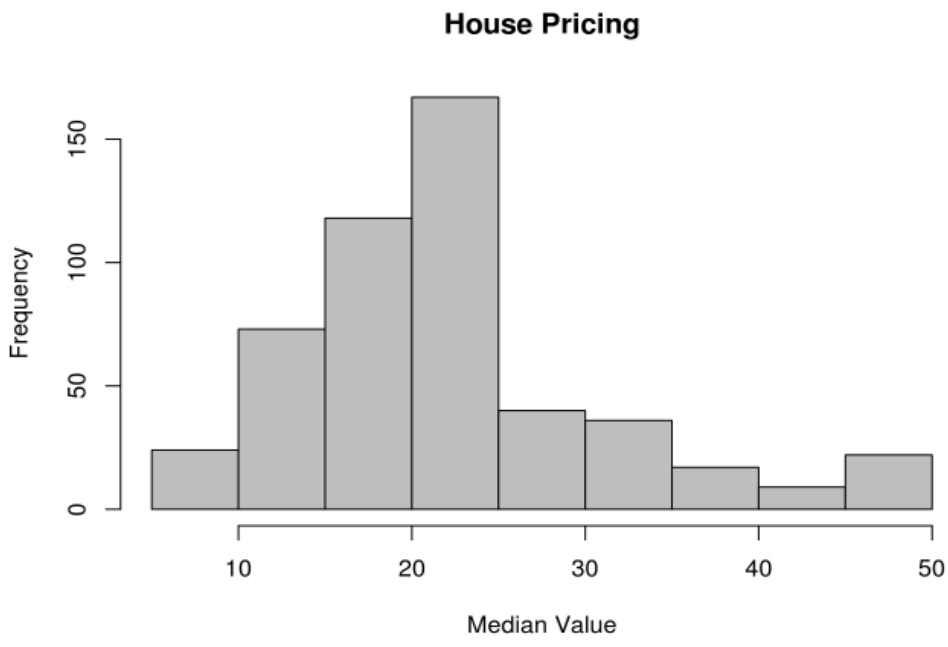
```
cor(boston,boston$MV)
```

```
##             [,1]
## CRIM  -0.3883046
## ZN     0.3604453
## INDUS -0.4837252
## CHAS   0.1752602
## NOX   -0.4273208
## RM     0.6953599
## AGE   -0.3769546
## DIS    0.2499287
## RAD   -0.3816262
## TAX   -0.4685359
## PT    -0.5077867
## B      0.3334608
## LSTAT -0.7376627
## MV     1.0000000
```

I see that the quantity of rooms RM has the most grounded positive relationship with the middle worth of the Housing price, while the level of the lower status populace, LSTAT and the student instructor proportion, PTRATIO, have a solid negative connection. The element with minimal connection to MV is the closeness to Charles River, CHAS.

House Pricing

```
hist(boston$MV,xlab="Median Value", main="House Pricing", col="grey")
```



The right skewed distribution suggests that a log transformation would be appropriate. Similarly, the variables crim, dis, nox, zn are found to be right skewed, making log transformations appropriate. The left skewed distribution of ptratio suggests that squaring it could make for a better fit.

Data Partitioning

We partition the data on a 7/3 ratio as training/test datasets.

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following objects are masked from 'package:psych':
##
##     %+%, alpha
```

```
## Warning: replacing previous import by 'plyr::ddply' when loading 'caret'
```

```
## Warning: replacing previous import by 'tidyr::%>%' when loading 'broom'
```

```
## Warning: replacing previous import by 'tidyr::gather' when loading 'broom'
```

```
## Warning: replacing previous import by 'tidyr::spread' when loading 'broom'
```

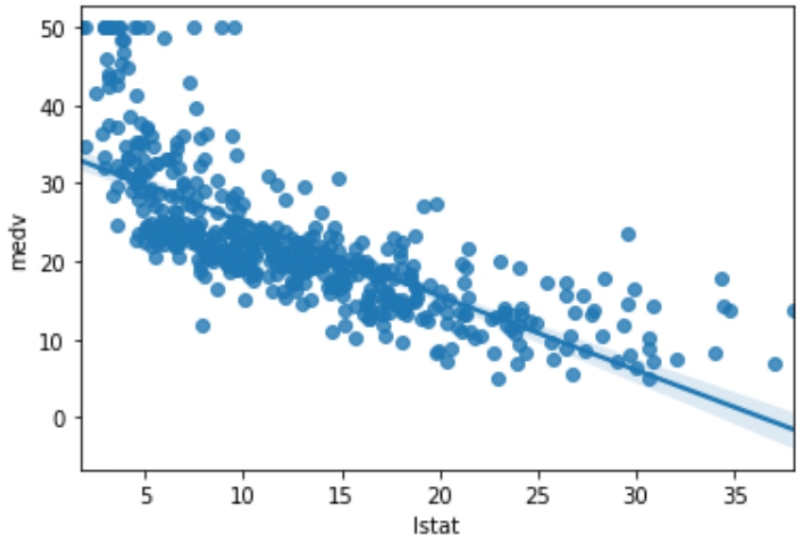
```
## Warning: replacing previous import by 'rlang::!!' when loading 'recipes'
```

```
## Warning: replacing previous import by 'rlang::expr' when loading 'recipes'
```

7.Testing Output

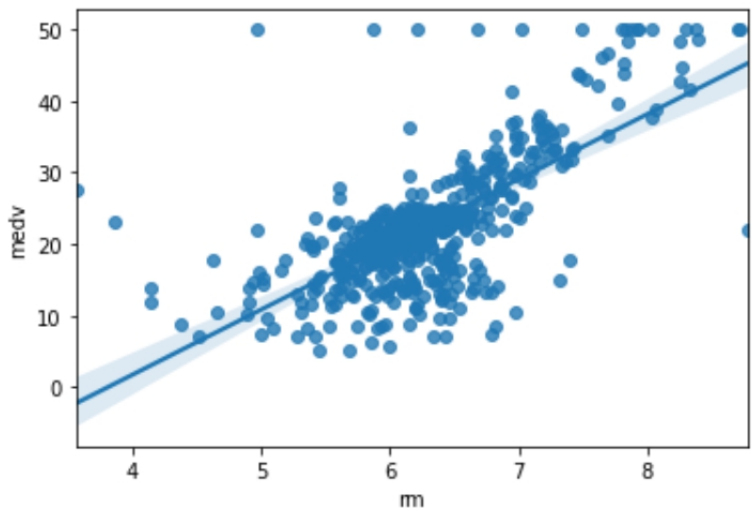
```
In [14]: sns.regplot(y=df['medv'], x=df['lstat'])
```

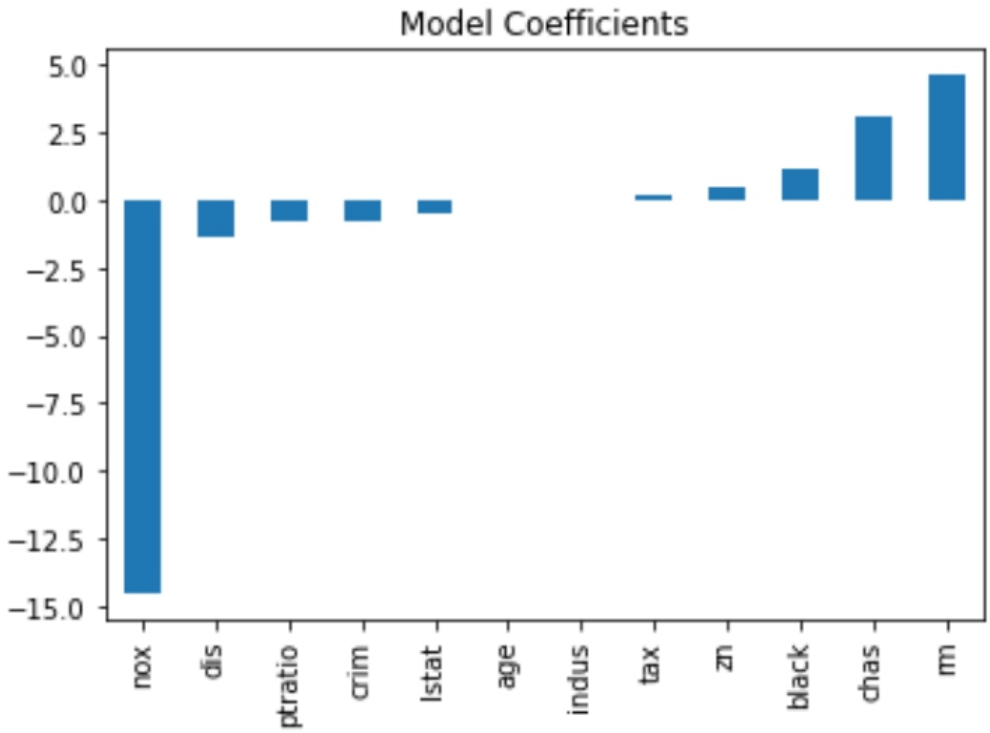
```
Out[14]: <AxesSubplot:xlabel='lstat', ylabel='medv'>
```

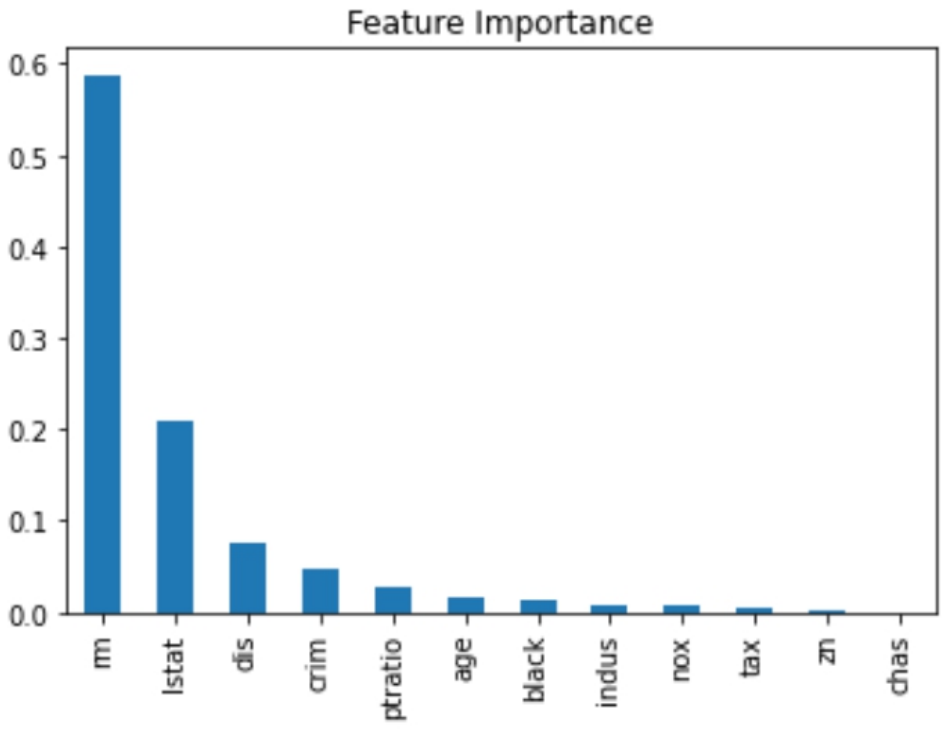


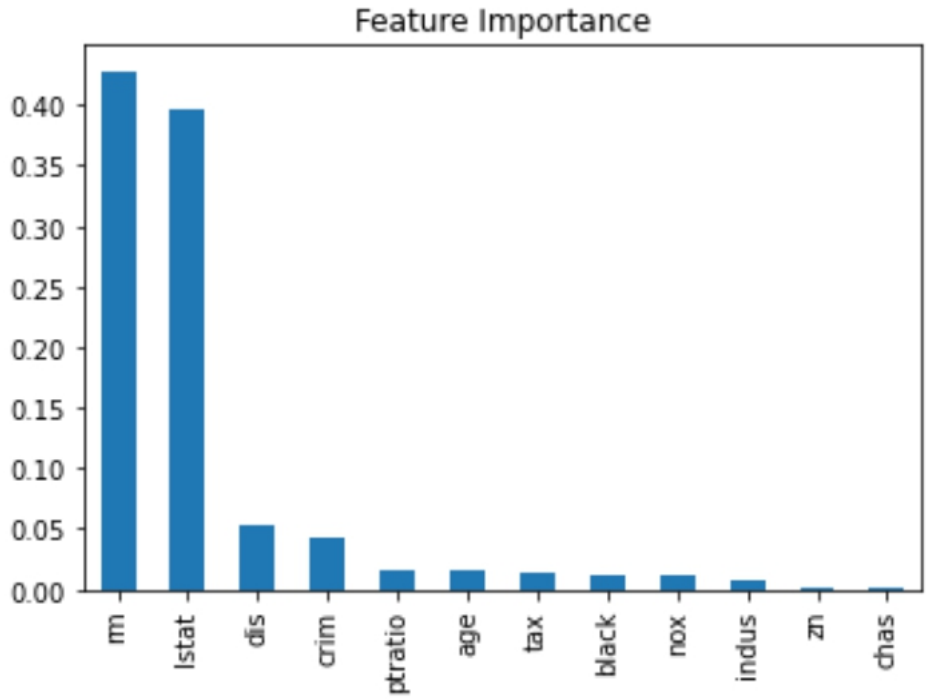

```
In [15]: sns.regplot(y=df['medv'], x=df['rm'])
```

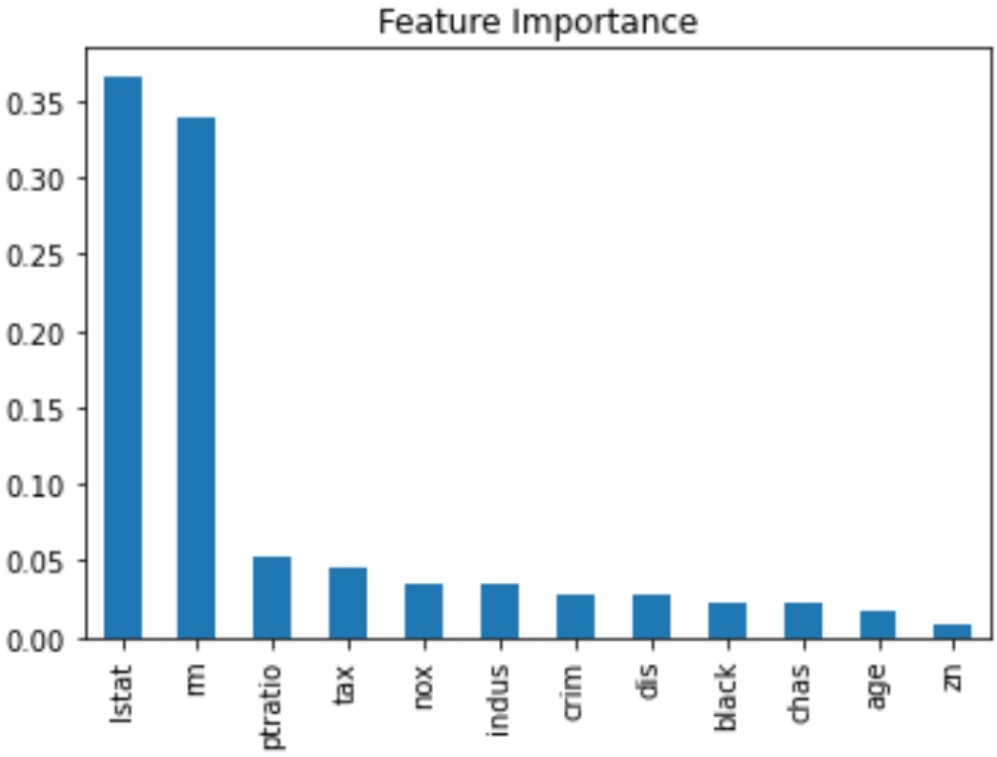
```
Out[15]: <AxesSubplot:xlabel='rm', ylabel='medv'>
```




Model Coefficients
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−12.5
−15.0
nox
dis
ptratio
crim
lstat
age
indus
tax
zn
black
chas
rm


Feature Importance
0.6
0.5
0.4
0.3
0.2
0.1
0.0
rm
lstat
dis
crim
ptratio
age
black
indus
nox
tax
zn
chas


Feature Importance
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
rm
lstat
dis
crim
ptratio
age
tax
black
nox
indus
zn
chas


Feature Importance
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
lstat
rm
ptratio
tax
nox
indus
crim
dis
black
chas
age
zn

8. Conclusion

The main objective of this thesis was to create a housing unit price prediction application to be used in Muscovite. Integrating an existing pricing tool, ARVO, with Muscovite was considered, but the idea was scrapped and replaced by a machine learning solution built as an independent component. The goal of this report was to determine the neighborhood attributes that best explained variation in house pricing. Various statistical techniques were used to eliminate predictors and extraneous observations.

In examining the final model, one finds quite reasonably that house prices are higher in areas with lower crime and lower pupil-teacher ratios. House prices also tend to be higher closer to the Charles River, and houses with more rooms are pricier. This report is interested in the neighborhood attributes of houses, so the number of rooms is not an important predictor. The most interesting factors to consider are nitrogen oxide levels and distance to the main employment centers. On the one hand, people would want to live close to their place of employment. Yet it is reasonable to suggest that pollution levels are higher as one moves closer to these main employment centers. Most importantly, when talking of pollution, it is not just nitrogen oxide levels that are higher, but also noise pollution levels.

The regression model that was fitted shows that higher levels of pollution decrease house prices to a greater extent than distance to employment centers. This suggests that people would prefer to live further away from their place of employment if it meant lower levels of pollution, which is an interesting point to consider. On a concluding note, it is important to note that the data for this report was collected several decades ago. In the years since, there is no doubt that pollution levels have risen and it would be interesting to examine the ways in which that affects house pricing in Boston today.

References

[1] Bai, L., & Yan, X., & Jin, J. (2015). Forecast the Commercial Housing Price Index Based on Search Keywords Attention. Forecasting, 34(4), 65-70.

[2] Cao, X., & Mu, H. (2016). Prediction Research on Commercial Housing Sales Volume Based on Web Search. Construction Economy, 37(2), 73-77.

[3] Phan TD. Housing Price Prediction Using Machine Learning Algorithms: The Case of Melbourne City, Australia. 2018 International Conference on Machine Learning and Data Engineering (ICMLDE) 2018. doi:10.1109/icmlde.2018.00017.

[4] Daniel, W. W., & Cross, C. L. (2018). Biostatistics: a foundation for analysis in the health sciences. Wiley

[5] Fan C, Cui Z, Zhong X. House Prices Prediction with Machine Learning Algorithms. Proceedings of the 2018 10th International Conferenceon Machine Learning and Computing - ICMLC 2018. doi:10.1145/3195106.3195133..

[6] Baldi, P. and Brunak, S. (2002). Bioinformatics: A Machine Learning Approach. Cambridge, MA: MIT Press.This book offers a good coverage of machine learning approaches - especially neural networks and hidden Markov models in bioinformatics.

[7] Mitchell, T. (1997). Machine Learning. New York: Mc Graw-Hill. This is, although a bit dated, an excellent introduction to Machine Learning..

[8] Sherwin Rosen. “Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition”. In: The Journal of Political Economy 82.1 (1974), pp. 34–55.

[9] Hasan Selim. “Determinants of house prices in Turkey: Hedonic regression versus artificial neural network”. In: Expert Systems with Applications 36 (2009), pp. 2843– 2852.

[10] Danny P. H. Tay and David K. H. Ho. “Artificial Intelligence and the Mass Appraisal of Residential Apartments”. In: Journal of Property Valuation and Investment 10.2 (1992), pp. 525–540.

[11]L. Wilkinson, Tree Structured Data Analysis: AID, CHAID and CART, Sawtooth/SYSTAT Joint Software Conference, 1992.

[12] D. Nielsen, Tree Boosting With XGBoost - Why Does XGBoost Win ”Every” Machine Learning Competition?, Norwegian University of Science and Technology

[13] Kotsiantis, Decision trees: A recent overview, Artificial Intelligence Review, Vol. 39, Iss. 4, Apr. 2013, pp. 1–23

[14] Yiuc, Y.; Wong, S.K. The effffects of expected transport improvements on housing prices. Urban Stud. 2005, 42, 113–125.

[15] Dube, J.; Legros, D.; Theriault, M. A Spatial Diffffference-in-diffffferences estimator to evaluateRouwendal, J.; Van, M.R. The effects of highway development on housing prices. Transportation the effffect of change in public mass transit systems on house prices. Transp. Res. Part B 2014, 64, 24–40. [CrossRef]Symmetry 2020, 12, 1329 20 of 21 Levkovich, O.;

[16] Shyr, O.; Andersson, D.E.; Wang, J.; Huang, T.; Liu, O. Where do home buyers pay most for relative transit accessibility? Hong Kong, Taipei and Kaohsiung Compared. Urban. Stud. 2013, 50, 2553–2568. [CrossRef]

[17] Mitra, S.K.; Saphores, J.D.M. The value of transportation accessibility in a least developed country city—The case of Rajshahi City, Bangladesh. Transp. Res. Part A Policy Pract. 2016, 89, 184–200. [CrossRef]

[18] Alonso, W. A reformulation of classical location theory and its relation to rent theory. Pap. Reg. Sci. Assoc. 1967, 19, 22–44. [CrossRef]

[19] Hansen, W.G. How Accessibility shapes land use. J. Am. Plan. Assoc. 1959, 25, 73–76. [CrossRef]

[20] Yue, X.; Xu, J.J.; Zhong, Y. Study on the share ratio between a service provider and two carriers. J. China Univ.Posts Telecommun. 2007, 14, 120–124. [CrossRef]

[21] Shin, K.; Washington, S.; Choi, K. Effects of transportation accessibility on residential property values.Transp. Res. Rec. J. Transp. Res. Board 2007, 1994, 66–73. [CrossRef]

[22] Xinru, L.; Chaoqun, M.; Changjun, L. A Research of Benchmark Town Land Price Based on the Hedonic Price Model. Syst. Eng. 2005, 23, 115–119.

[23] Gao, X.; Asami, Y. Influence of spatial features on land and housing prices. Tsinghua Sci. Technol. 2005, 10,344–353. [CrossRef]

[24] Durganjali, P.; Pujitha, M.V. House Resale Price Prediction Using Classification Algorithms. In Proceedings of the 6th IEEE International Conference on Smart Structures and Systems, ICSSS 2019, Chennai, India, 14–15 March 2019; pp. 1–4.

[25] Qian, D.; Nana, S.; Wei, L. Real estate price prediction based on web search data. Stat. Res. 2014, 31, 81–88.

[26] Bowen, Y.; Buyang, C. Housing price prediction model based on integrated learning. Comput. Knowl. Technol. 2017, 13, 191–194.

[27] Benoit, D. F. & Van den Poel, D. (2017). bayesQR: A Bayesian Approach to Quantile Regression. Journal of Statistical Software, 76 (7).

[28] Breiman, L. (2001). Random Forests. Machine Learning, 45 (1), 5–32.

[29] Haupt, H., Kagerer, K., & Schnurbus, J. (2011). Cross-validating fit and predictive accuracy of nonlinear quantile regressions. Journal of Applied Statistics, 38 (12), 2939 – 2954.

[30] Nguyen, T., Huang, J. Z. & Nguyen, T. (2015). Two-level quantile regression forests for bias correction in range prediction. Machine Learning, 101, 325 – 343.

[31] Meinshausen, N. (2006). Quantile Regression Forests. Journal of Machine Learning Research,7,983–999

[32] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society, Series B, 67:301–320, 2005. DOI: 10.1111/j.1467-9868.2005.00503.x.

[33] Yann LeCun, Leon Bottou, G. Orr, and K. Muller. Efficient BackProp. In G. Orr and Muller K, Eds., Neural Networks: Tricks of the Trade. Springer, 1998a. DOI: 10.1007/3-540-49430-8_2.

[34] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient based learning applied to pattern recognition. Proc. of the IEEE, 86(11):2278–2324, November 1998b.

[35] R. Kohavi and F. Provost, "Glossary of terms," Machine Learning, vol. 30, no. 2–3, pp. 271–274, 1998.

[36] Nilsson N. Learning Machines, McGraw Hill, 1965.

[37] S. Bozinovski "Teaching space: A representation concept for adaptive pattern classification" COINS Technical Report No. 81-28, Computer and Information Science Department, University of Massachusetts at Amherst, MA, 1981.

[38] Garbade, Dr Michael J. (14 September 2018). "Clearing the Confusion: AI vs Machine Learning vs Deep Learning Differences". Medium. Retrieved 28 October 2020.

[39] Annina S, Mahima SD, Ramesh B. An Overview of Machine Learning and its Applications. International Journal of Electrical Sciences & Engineering (IJESE). 2015 January; I(1): 22-24.

[40] Segal MR. Machine Learning Benchmarks and Random Forest Regression. Center for Bioinformatics and Molecular Biostatistics, UC San Francisco. 2004 April.

Acknowledgements

I would to like thank you so much my supervisor Mrs. Jie Li lust for providing her time in guiding me and the knowledge she shared with me, her guidance is the main reason to complete and successfully implement this Thesis Papers.