EAST WEST UNIVERSITY

B. SC. ENGINEERING THESIS

Predicting Real-Estate valuation using Random Forest Regression

Authors:

Armanul Habib Zihan

(2015-1-53-037)

Musharrat Zabin

(2015-2-50-015)

Raihana Taherin

(2015-2-50-025)

Supervisor:

Muhammad Suhail Najeeb

Lecturer

ELECTRONICS AND COMMUNICATIONS ENGINEERING

A thesis submitted in fulfillment of the requirements for the degree of Bachelor of Science in Information and Communication Engineering

June 2020

Letter of Acceptance

This thesis entitled “Predicting Real-Estate Valuation using Random Forest Reegression” submitted by Armanul Habib Zihan (ID: 2015-1-53-037), Musharrat Zabin (ID: 2015-2-50-015) and Raihana Taherin (ID: 2015-2-50-025) to the Electronics and Communication Engineering Department, East West University, Dhaka-1212, Bangladesh is accepted as satisfactory for partial fulfilment of requirements for the degree of Bachelor of Science (B.Sc) Information and Communications Engineering.

_____

Dr. Mohammed Moseeur Rahman

Assistant Professor Chairperson

Department of Electronics and Communications Engineering

East West University

_____

Muhammad Suhail Najeeb

Lecturer

Department of Electronics and Communications Engineering

East West University

Declaration of Authorship

We, Armanul Habib Zihan, Musharrat Zabin and Raihana Taherin, declare that this thesis titled, “Predicting Real-Estate Valuation using Random Forest Regression” supervised by Muhammad Suhail Najeeb and the work presented in it are our own. We confirm that:

This work has done wholly while in candidate for Bachelor of Science in Information and Communication Engineering degree at this University. Where any part of this thesis has done previously by others have clearly cited. Where we have consulted the published work of others, this is always clearly attributed. Where we have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely our own work. We have acknowledged all main sources of help. Where the thesis is based on work done by ourselves jointly with others, we have made clear exactly what was done by others and what we have contributed ourselves.

_____

Armanul Habib Zihan

2015-1-53-037

_____

Musharrat Zabin

2015-2-50-015

_____

Raihana Taherin

2015-2-50-025

Abstract

Prediction models in real estate have a significant role to play in telling the future of the real estate industry. They have a role to play in forecasting, which is important for investors who use the knowledge to make successful decisions.

We propose a methodology using a combination of Machine learning (Random Forests), Graphic Information and different regression models. Examining real estate valuation helps to understand where people tend to live in a city. Predicting real estate valuation can help the urban design and urban politics, as it could help identify what factors have the most impact on property prices. Spot checking algorithms helped us identify a candidate to model our issue and test rapidly different regression models using spot checking. By applying this methods, have found a model that help us predict the house price of unit area in Xindian district, New Taipei, Taiwan.

Acknowledgements

We would like to express our most sincere gratitude to our supervisor Muhammad Suhail Najeeb for his patient guidance, continuous support and advice. He has always been very helpful and inspired us a lot. We would also like to express our gratitude to our honorable Chairperson and other faculty members and staff of our department. Last but not the least we are grateful to our parents who always support and inspired us.

Contents

Dedicated to Our Parents …

CHAPTER 1

INTRODUCTION

1.1 Motivation

To have a house is dream of many. In any case, in this season of development and taking off housing costs, it isn't for each situation easy to find dream home inside the obliged monetary arrangement. In like manner, despite spending plan, there are a couple of various factors that contributes towards finding the right home-zone, basic section, transportation, etc. In such a circumstance, a house cost envisioning system will be helpful for the two buyers and vendors. This assessment intends to anticipate house costs in Xindian District, New Taipei City, Taiwan using backslide examination. The gauge is appeared at by help of various instructive variables, for instance, locale of the property, zone of the house, material used for advancement, age of the property, number of rooms and garages, and so on. This paper clarifies on the introduction of discretionary woodlands figuring for model desire. It moreover nuances the Hyperparameter Tuning the Random Forest in Python gives a nice methodology that we follow here.

The figure of housing costs as they change through time and spot is an unpredictable assignment, particularly when the open data is enormous and apportioned into different data types. Taking everything into account; it was an appropriate choice for the Data Science Challenge at Engineering Education for the Future (EEF). This resistance relied upon envisioning property costs reliant on Machine Learning models. The data for getting ready and testing the models was made by till year 2018 from Taiwan destinations. Each advanced home was accessible to be bought, none for rent. The data characteristics recall delineations for plain substance, dates, geographic headings, photos, similarly as straight out and numerical features.

1.2 Organization of thesis

Since investigating real estate valuation helps to understand where people overlook to live in a city. The higher price or the greater the demand to live in the property. Forecasting real estate valuation can help urban design and policies, as it could help recognize what factors have the most

impact on property prices. Our aim is to predict real estate value, based on several features. For task we have modified the random search and grid search for results. Our objectives are to:

1. Understand the data availability
2. Test rapidly different regression models
3. Assess the best model(s) and improve them
4. Present the results and what could be further addressed.

We have organized the thesis by following manner:

Chapter 2: In this chapter, we have discussed the literature review of this thesis. Then we have discussed about the history and development of Machine Learning.

Chapter 3: This chapter contains necessary information about Linear Regression, Support Vector Regression, Decision Tree Regression and Random Forest Regression.

Chapter 4: This chapter contains the results and dataset and exploratory data analysis.

Chapter 5: This chapter contains the results and discussions about the result by spot check algorithm.

Chapter 6: This chapter contains discussion about fine tuning the processed data.

Chapter 7: This chapter contains discussion about conclusion and future work.

CHAPTER 2

LITERATURE REVIEW

The investigation ashore value pattern is considered to be noteworthy to help the choices in urban arranging. The land framework is an insecure stochastic procedure. Financial specialists' choices depend available patterns to harvest most extreme returns. Designers are intrigued to know the future patterns for their dynamic. So as to precisely appraise property costs and future patterns, enormous measure of information that impacts land cost is required for examination, demonstrating and estimating. The elements that influence the land cost have to be considered and their effect on cost has likewise to be displayed. An investigation of the past information uncovered that the costs show a non-straight trademark. It is surmised that building up a straightforward direct numerical relationship for this time-arrangement information is found not practical for gauging. Thus, it got basic to set up a non-direct model which can very much fit the information trademark to investigate and estimate future patterns. As the land is quick creating division, the investigation and estimate of land costs utilizing numerical demonstrating and other logical procedures is a prompt earnest requirement for dynamic by each one of those concerned. The expansion in populace just as the modern action is ascribed to different elements, the most conspicuous being the ongoing spray in the information area viz. Data Technology (IT) and Information innovation empowered administrations. Interest for land began of demonstrating an upward pattern and lodging and the land movement began blasting. Every single desolate land and paddy fields stopped their reality to clear path for multistore and tall structures. Interests in Real Estate Industry has become essentially high throughout the years and we have seen a non-uniform theme as far as land valuing. The requirement for foreseeing the pattern in land costs was felt by all in the business viz. the Government, the controlling bodies, loaning foundations, the designers and the financial specialists. In the course of the most recent two decades there have been an enormous number of exact examinations breaking down land costs. Kilpatrick demonstrated the value of time-arrangement relapse model which utilized monetary information to give gauge of Central Business District (CBD) land cost in moving business sector. Wilson et al. [3] considered the private property showcase represents a generous extent of UK financial movement. Valuers gauge

property estimations dependent on current offer costs. In this paper, the national lodging exchange information was prepared utilizing Artificial Neural Networks (ANN), which estimates future pattern of the lodging market. Imprint and John built up a relapse model with empty land deals. The model disclosed up to 93% of the market esteems. Wang and Tian [5] utilized the wavelet Neural Network (NN) to conjecture the land value record. This sort of wavelet NN coordinated the value of the wavelet examination and the custom NN. It likewise contrasted the determining result and smoothing technique and the NN estimate. Zhangming gauge the land value record by utilizing the Back Propagation (BP) NN. The BPN utilized the sigmoid capacity. Tinghao utilized the Auto Regressive Integrated Moving Average (ARIMA) model and conveyed the decisive investigation on year information from 1998 to 2006. He utilized the set-up model to make the figure to the land value list of 2007. A decadent relapse on the cost of land proposed that accepted strategy contrasts between political wards have significantly affected land costs between1970 and 1980. Steven and Albert utilized 46,467 private properties spreading over 1999 - 2005 and exhibited that utilizing coordinated sets that comparative with straight decadent valuing models, ANN produce lower dollar evaluating mistakes, had more prominent estimating exactness out-of-test, and extrapolate better from progressively unpredictable evaluating situations. ANN is more qualified to indulgent models that use huge quantities of factors. Sampath Kumar and Santhi contemplated the land value pattern of Sow carpet which is the focal part. They created measurable model utilizing monetary factors and anticipated that the yearly ascent in land cost would be of 17%. Urmila announced that the past patterns were dissected to discover the pace of development or decay and the patterns are utilized in determining. Monetary parameters may be acquainted with figure progressively reasonable relationship. A portion of different strategies they Mansural Bhuiyan and Mohammad Al Hasan 2016 use is relapse, profound figuring out how to take in the idea of models from the past outcomes (the property/land which were auctions off already which are utilized as preparing data).There are various models utilized, for example, straight model information utilizing just one component, multivariate model, utilizing a few highlights as its info and polynomial model utilizing the contribution as cubed or squared and consequently determined the root mean squared blunder (RMS esteem) for the model.

2.1 Requirement for Real Estate Value Prediction

• While our country proceeds with development pattern and the development business falls behind interest, costs will keep on increasing while loan fees knock upward.

• Securing speculation property now with exhaustive due tirelessness and watch venture pay off throughout the following not many years.

2.2. Points of interest of Machine Learning Over Real Estate Value Prediction

• Trends and Patterns Are Identified effortlessly

• Machine Learning Improves Over Time

• Machine Learning Lets You Adapt Without Human Intervention

• Enables Automation

2.3 Writing Survey

AI is a type of man-made consciousness which form accessible PCs with the effectiveness to be prepared without being veraciously customized. AI enthusiasm on the augmentations of PC programs which is able enough to adjust when unprotected to modern information. AI calculations are comprehensively ordered into three divisions, specifically; Supervised learning, Unsupervised learning and Reinforcement learning.

Directed learning is a learning wherein we educate or train the machine utilizing information which is very much named that implies a few information is now labeled with right answer. From that point onward, machine is furnished with new arrangement of models so regulated learning calculation investigations the preparation information and produces a right result from marked information. Unaided learning is the preparation of machine utilizing data that is neither arranged nor marked and permitting the calculation to follow up on that data without direction. Here the errand of machine is to gather unsorted data as per likenesses, examples and contrasts with no

earlier preparing of information. In contrast to, directed learning, no educator is given that implies no preparation will be given to the machine. Consequently, machine is confined to locate the concealed structure in unlabeled information by our-self.

Fortification learning is a region of Machine Learning. Fortification. It is tied in with making appropriate move to augment award in a specific circumstance. It is utilized by different programming and machines to locate the most ideal conduct or way it should take in a particular circumstance. Support taking in varies from the managed learning in a route that in regulated learning the preparation information has the appropriate response key with it so the model is prepared with the right answer itself while in fortification learning, there is no answer yet the support specialist chooses what to do to play out the given assignment. Without preparing dataset, it will undoubtedly gain from its experience.

AI has numerous application's out of which one of the applications is forecast of land. The land showcase is one of the most serious as far as valuing and same will in general be change essentially dependent on heaps of factor, gauging property cost is a significant modules in dynamic for both the purchasers and financial specialists in supporting spending designation, discovering property discovering tricks and deciding reasonable strategies thus it gets one of the prime fields to apply the ideas of AI to enhance and foresee the costs with high precision. The investigation ashore value pattern is felt critical to help the choices in urban arranging. The land framework is an unsteady stochastic procedure. Financial specialists' choices depend available patterns to procure most extreme returns. Engineers are intrigued to know the future patterns for their dynamic. To precisely evaluate property costs and future patterns, huge measure of information that impacts land cost is required for investigation, displaying and estimating.

The variables that influence the land cost have to be considered and their effect on cost has additionally to be displayed. An examination of the past information is to be thought of. It is construed that building up a basic direct numerical relationship for this time-arrangement information is found not suitable for gauging. Subsequently it got basic to set up a non-direct model which can very much fit the information trademark to investigate and gauge future patterns. As the land is quick creating segment, the investigation and conjecture of land costs utilizing

numerical displaying and other logical procedures is a prompt pressing requirement for dynamic by every one of those concerned.

The expansion in populace just as the modern movement is ascribed to different variables, the most conspicuous being the ongoing spray in the information division viz. Data Technology (IT) and Information innovation empowered administrations. Interest for land began of indicating an upward pattern and lodging and the land movement began blasting. Every single desolate land and paddy fields stopped their reality to clear path for multistore and elevated structures. Ventures began pouring in Real domain Industry and there was no uniform example in the land cost throughout the years. The requirement for anticipating the pattern in land costs was felt by all in the business viz. the Government, the controlling bodies, loaning foundations, the engineers and the financial specialists.

In this manner, in this paper, we present different significant highlights to utilize while foreseeing lodging costs with great exactness. We can utilize relapse models, utilizing different highlights to have lower Residual Sum of Squares mistake. While utilizing highlights in a relapse model some component designing is required for better forecast. Regularly a lot of highlights numerous relapses or polynomial relapse (applying a different arrangement of forces in the highlights) is utilized for improving model fit. For these models are required to be vulnerable towards over fitting edge relapse is utilized to decrease it. Along these lines, it coordinates to the best utilization of relapse models notwithstanding different procedures to streamline the outcome.

2.3.1 Previous works and studies

In the course of the most recent two decades there have been countless observational investigations breaking down land costs. Kilpatrick demonstrated the helpfulness of time-arrangement relapse model which utilized financial information to give conjecture of Central Business District (CBD) land cost in moving business sector. Wilson et al contemplated the private property showcase represents a generous extent of UK financial movement. Valuers gauge property estimations dependent on current offer costs. In this paper, the national lodging exchange information was prepared utilizing Artificial Neural Networks (ANN), which gauges future pattern of the lodging market. Imprint and John built up a relapse model with empty land deals. The model disclosed up

to 93% of the market esteems. Wang and Tian utilized the wavelet Neural Network (NN) to estimate the land value list. This sort of wavelet NN coordinated the value of the wavelet examination and the convention NN. It likewise contrasted the determining result and smoothing strategy and the NN estimate.

Zhangming determined the land value record by utilizing the Back Propagation (BP) NN. The BPN utilized the sigmoid capacity. Tinghao utilized the Auto Regressive Integrated Moving Average (ARIMA) model and conveyed the illustrative investigation on year information from 1998 to 2006. He utilized the built-up model to make the figure to the land value list of 2007. An epicurean relapse on the cost of land recommended that accepted arrangement contrasts between political purviews have significantly affected land costs between1970 and 1980. Steven and Albert utilized 46,467 private properties spreading over 1999 - 2005 and exhibited that utilizing coordinated sets that comparative with straight epicurean valuing models, ANN produce lower dollar estimating mistakes, had more prominent evaluating accuracy out-of-test, and extrapolate better from increasingly unpredictable estimating situations. ANN is more qualified to gluttonous models that use enormous quantities of factors. Sampath Kumar and Santhi considered the land value pattern of Sow carpet which is the focal part. They created measurable model utilizing monetary factors and anticipated that the yearly ascent in land cost would be of 17%.

Urmila revealed that the past patterns were broke down to find out the pace of development or decay and the patterns are utilized in determining. Financial parameters may be acquainted with figure increasingly reasonable relationship. A portion of different strategies they Mansural Bhuiyan and Mohammad Al Hasan 2016 use is relapse, profound figuring out how to take in the idea of models from the past outcomes (the property/land which were auctions off beforehand which are utilized as preparing data).There are various models utilized, for example, straight model information utilizing just one element, multivariate model, utilizing a few highlights as its info and polynomial model utilizing the contribution as cubed or squared and subsequently determined the root mean squared mistake (RMS esteem) for the model.

CHAPTER 3

THEORETICAL CONCEPTS OF DIFFERENT MACHINE LEARNING MODELS

Regression analysis is both probably the most seasoned part of measurements, with least-squares investigation having been proposed route in 1805, and likewise one of the most up to date territories, as the AI procedures being overwhelmingly examined today. Of course, at that point, there is a huge writing regarding the matter. All things considered, at that point, why compose one more relapse book? Numerous books are out there as of now, with titles utilizing words like relapse, classification, predictive examination, AI, etc. They are composed by creators whom I significantly respect, and whose work I myself have discovered valuable. However, I didn't feel that any current books canvassed the material in a way that saliently gave understanding to the rehearsing information investigator. Simply incorporating models with genuine information isn't sufficient to really tell the story such that will be helpful practically speaking. Barely any books go a lot past introducing the recipes and strategies, and hence the hapless expert is to a great extent left to his/her own gadgets. Too little is said as far as what the ideas truly mean from a commonsense perspective, what should be possible with respect to the inescapable blemishes of our models, which procedures are an excess of the subject of \hype, etc. This book means to cure this vast deceit. It builds up the material in a way that is accurately expressed at this point consistently keeps up as its top need | obtaining from a book title of the late Leo Breiman | \a see toward applications."

3.1 Big Data:

Nowadays there is a lot of discussion about Big Data. Despite the fact that it is a long way from the case that most information nowadays is Big Data, then again it is valid that things today are surely very different from the times of \your father's relapse book." Maybe the most sensational of these progressions is the development of informational collections with huge quantities of indicator factors p, as a small amount of n, the number of perceptions. Without a doubt, for certain informational collections p >> n, an incredibly testing circumstance. Part 9, Dimension Reduction, covers not just ordinary" issues of variable determination, yet additionally this significant more current sort of issue, for which numerous arrangements have been proposed.

3.2 A remark on the field of AI:

Notice ought to be made of the way that this current book's title incorporates both the word relapse and the expression AI. At the point when China's Deng Xiaoping was tested on his then-questionable arrangement of acquainting entrepreneur thoughts with China's economy, he broadly stated, Black feline, white feline, it doesn't make a difference as long as it gets mice." Statisticians also, AI clients should notice, and this book draws upon both fields, which at center are not so much different from one another anyway. My own view is that AI (ML) comprises of the improvement of relapse models with the Prediction objective. Normally nonparametric techniques are utilized. Classification models are more typical than those for anticipating consistent factors, and usually multiple classes are included, at times a large number of classes. All things considered; however, it's still relapse examination, including the contingent mean of Y given X (decreasing to P(Y = 1jX) in the classification setting). One frequently asserted qualification among measurements and ML is that the previous depends on the idea of an example from a populace though the last mentioned is concerned uniquely with the substance of the information itself. Be that as it may, this difference is more seen than genuine. Cross-approval is integral to ML techniques, and since that approach is expected to quantify how well one's model sums up past our own information, obviously ML individuals do think as far as tests all things considered. Along these lines, toward the day's end, we as a whole are doing relapse investigation, and this book takes this perspective.

3.3 Intended audience:

This book is focused on both rehearsing experts and use in the study hall. Some insignificant foundation is required (see beneath), yet a few peruses will have some foundation in certain parts of the inclusion of the book. The book plans to both available and significant to such decent variety of readership, following the old counsel of Samuel Johnson that a creator should make the new natural and the recognizable new."

3.4 Regression Models:

We have tried to analyze the concepts of four machine learning models.

These are-

1. Linear Regression
2. Support Vector Regression
3. Decision Tree Regression
4. Random Forest Regression

After that, we will eventually show with what model we worked and why.

3.4.1 Linear Regression Models

In this chapter we go into the details of linear models. Let's first set some

notation, to be used here and in the succeeding chapters.

3.4.1.1 Notation

Leave Y alone our reaction variable, and let X = (X(1);X(2); :::;X(p)))0 indicate the vector of our p indicator factors. Utilizing our weight/stature/age baseball player model from Chapter 1 as our running model here, we would have p = 2, and Y , X(1) and X(2) would be weight, tallness and age, individually. Our example comprises of n information focuses, X1;X2; :::;Xn, each a p-component indicator vector, and Y1; Y2; :::; Yn, related scalars. In the baseball model, n was 1015.

So, again using the baseball player example, the height, age and weight of the third player would be X (1)3, X (2)3 and Y3, respectively. And just one more piece of notation: We sometimes will need to augment a vector with a 1 element at the top, such as we did in. Our notation for this will consist of a tilde above the symbol.

Random- vs. Fixed-X Cases:

We will for the most part believe the Xi and Yi to be irregular examples from a few populace, so that (X1; Y1); :::; (Xn; Yn) are free and indistinguishably disseminated as per the populace. This is an irregular X setting, implying that both the Xi and Yi are irregular. There are a few

circumstances in which the X-values are fixed by configuration, known as a fixed-X setting. This may be the situation in science inquire about, in which we choose ahead of time to perform tests at specific levels of centralization of certain synthetic concoctions.

Least-Squares Estimation

Linear regression analysis is sometimes called least-squares estimation.

It will be crucial to always keep in mind the distinction between population values and their sample estimates, especially when we discuss overfitting in detail.

3.4.1.2 Matrix Formulations

Utilization of grid documentation in straight relapse investigation enormously compactifies also, clarifies the introduction. You may find this requires a time of alteration at first, however it will be certainly justified regardless of the effort.

Using Matrix Operations to Minimize

Keep in mind, we will set b to whatever estimation of b limits. In this way we need to take the subordinate of that articulation regarding b, and set the result to 0. There is a hypothesis of grid subordinates, not secured here, however the fact of the matter is that the subsidiary of regarding b can be appeared to be.

(Intuitively, this is the multivariate analog of

$$\frac{\partial}{\partial b}(d-ab)^2 = -2(d-ab)a$$

for scalar d, a, b.)

Setting this to 0, we have

$$A'D = A'Ab$$

Solving for b we have our answer:

$$\widehat{\beta} = (A'A)^{-1}A'D$$

Statistical Inference

The lm yield is vigorously centered around factual deduction shaping confidence interims and performing significance tests and the first thing you may see is each one of those marks: The assessments of the capture, the stature coefficient and the age coefficient all are set apart with three stars, showing a p-estimation of under 0.001. The trial of the theory H0 : 1 = 0 (2.24) would be resoundingly dismissed, and one could state, "Stature has a significant impact on weight." to be expected by any means, however the finding for age may be additional fascinating, in that we anticipate that competitors should keep t, even as they age. We could shape a confidence interim for 2, for example, by including and taking away 1.96 occasions the related standard mistake, which is 0.1257 in this case. Our subsequent CI would be about (0.66,1.16), demonstrating that the normal player gains somewhere in the range of 0.66 and 1.16 points every year; even baseball players put on weight after some time.

Assumption:

But where did this come from? Surely there must be some assumptions underlying these statistical inference procedures. What are those assumptions? The classical ones, to which the reader may have some prior exposure, are-

Normality: The assumption is that, conditional on the vector of predictor variables X, the response variable Y has a normal distribution. In the weight/height/age example, this would mean, for instance, that within the subpopulation of all baseball players of height 72 and age 25, weight is normally distributed.

3.4.1.3 Unbiasedness and Consistency:

We will begin by discussing two properties of the least-squares estimator.

Unbiased

One of the central concepts in the early development of statistics was unbiasedness. As you'll see, to some degree it is only historical baggage, but on the other hand it does become quite relevant in some contexts here. To explain the concept, say we are estimating some population value, using an estimator based on our sample. we get a new value of. So, some samples will yield a that

overestimates, while in other samples will come out too low. The pioneers of statistics believed that a nice property for to have would be that on average, i.e., averaged over all possible samples, comes out “just right". This seems like a reasonable criterion for an estimator to have, and sure enough, our least-squares estimator has that property. Note that since this is a vector equation, the unbiasedness is meant for the individual components.

Bias as an Issue/Nonissue:

Arguably the pioneers of statistics shouldn't have placed so much emphasis on unbiasedness. Most statistical estimators have some degree of bias, though it is usually small and goes to 0 as the sample size n grows. Other than least-squares, none of the regression function estimators in common use is unbiased. Indeed, there is a common estimator, learned in elementary statistics courses, that is arguably wrongly heralded as unbiased. The “natural" sample-analog divisor would be n, not n1. Using our “correspondence" notation.

But as the reader may be aware, use of n as the divisor would result in a biased estimate. The divisor is then a “fudge factor" that can be shown to produce an unbiased estimator. Yet even that is illusory. While it is true that S2 is an unbiased estimate of, we actually don't have much use for S2. Instead, we use S, as in the familiar t-statistic, for inference on means.

Thus, one should indeed not be obsessive in pursuing unbiasedness. However, the issue of bias does play an important role in various aspects of regression analysis. It will arise often in this book, including in the present chapter.

Consistent

In contrast to unbiasedness, which has argued above may not be a general goodness criterion for an estimator, there is a more basic property that we would insist that almost any estimator to have, consistency: As the sample size n goes to infinity, then the sample estimate goes to. This is not a very strong property, but it is a minimal one. It is shown that the least-squares estimator is indeed a consistent estimator.

3.4.1.4 Collective Predictive Strength

The R^2 quantity in the output measure of how well our model predicts Y. Yet, just as a sample quantity, estimates the population quantity, one would reason that the R^2 value printed out by must estimate a population quantity too. In this section, we'll make that concept precise, and deal with a troubling bias problem.

Definition of R^2

R^2 is the squared sample correlation between the actual response values and the predicted values.

The latter is the average squared prediction error in the population, whose sample analog is the average squared error in our sample. In other words, using our “correspondence" notation from before.

As a sample estimate of the population, the quantity R^2 would appear to be a very useful measure of the collective predictive ability. However, the story is not so simple, and curiously, the problem is actually bias.

Bias Issues

R^2 can be shown to be biased upward, not surprising in light of the fact that we are predicting on the same data that we had used to calculate. In the extreme, we could polynomial in a single predictor, with the curve passing through each data point, producing $R^2 = 1$, even though our ability to predict future data would likely be very weak. The bias can be severe if p is a substantial portion of n. (In the above polynomial example, we would have p, even though we started with $p = 1$.) This is the overfitting problem mentioned in the last chapter,

and to be treated in depth in a later chapter.

These results are not encouraging at all! The R^2 values are typically around 0.7, rather than 0.5 as they should be. In other words, R^2 is typically giving us much too rosy a picture as to the predictive strength. Of course, it should be kept in mind that I deliberately chose a setting which produced substantial overfitting 8 predictors for only 25 data points, which is probably too many predictors. Running the simulation with $n = 250$ should show much better behavior. This is indeed much

better. Note, though, that the upward bias is still evident, with values more typically above 0.5 than below it. Note too that R^2 seems to have large variance, even in the case of n = 250. Thus, in samples in which p=n is large, we should not take our sample's value of R^2 overly seriously.

Adjusted- R^2

The adjusted-R^2 statistic is aimed at serving as a less biased version of the ordinary R^2. Its derivation is actually quite simple, though note that we do need to assume homoscedasticity. Under the latter assumption. Then the numerator is biased, which we know fixed by using the factor instead of 1=n. Similarly, we know that the denominator will be unbiased if we divide instead if 1=n. Those changes do NOT make unbiased; the ratio of two unbiased estimators is generally biased. However, the hope is that this new version of R^2, called adjusted R^2, will have less bias than the original. We can explore this using the same simulation code as above. We simply change the line.

The “Leaving-One-Out Method"

Our theme here has been assessing the predictive ability of our model, with the approach described so far being the R^2 measure. But recall that we have another measure introduced the concept of cross-validation for assessing predictive ability. We will now look at a variant of that method. First, a quick review of cross-validation: Say we have n observations in our data set. With cross-validation, we randomly partition the data into a training set and a validation set, of k and n □ k observations, respectively. We fit our model to the training set, and use the result to predict in the validation set, and then see how well those predictions turned out.

Clearly there is an issue of the choice of k. If k is large, our validation set will be too small to obtain an accurate estimate of predictive ability. That is not a problem if k is small, but then we have a subtler problem: We are getting an estimate of strength of our model when constructed on k observations, but in the end, we wish to use all n observations. One solution is the Leaving One Out Method (LOOM). Apply the training/validation process to all possible partitions. The name alludes to the fact that LOOM repeatedly omits one observation, predicting it from fitting the model to the remaining observation. This gives us \the best of both worlds": We have n validation points, the best possible, i.e., nearly full-sized.

There is an added benefit that the same code to implement this method can be used to implement the jackknife. The latter is a resampling technique. To see what it does, let's look at a more general technique called the bootstrap, which is a method to empirically compute standard errors. Say we wish to determine the standard error of an estimator. We repeatedly take random samples of size k, with replacement, from our data set, and calculate on each of them. The resulting values form a sample from the distribution of b_ (for sample size k). One can compute standard errors from this sample in various ways, e.g. simply by finding their standard deviation. The jackknife does this, and thus we can again approximate the sampling distribution of our estimator based on n data points.

3.4.2 Support Vector Regression

Support Vector Machines (SVM) are learning machines implementing the structural risk minimization inductive principle to obtain good generalization on a limited number of learning patterns. Structural risk minimization (SRM) involves simultaneous attempt to minimize the empirical risk and the VC (Vapnik– Chervonenkis) dimension. The theory has originally been developed by Vapnik and his co-workers on a basis of a separable bipartition problem at the AT & T Bell Laboratories. SVM implements a learning algorithm, useful for recognizing subtle patterns in complex data sets. The algorithm performs discriminative classification

learning by example to predict the classifications of previously unseen data.

The VC dimension of a set of functions is the size of the largest data set due to that the set of functions can scatter. Let us consider a set of function $F = \{f(X, W)\}$ that map points from Rn into the set {0, 1} or {-1, 1}. These are called *indicator* functions that map data points into one of two classes. If one considers Q points in Rn, each of these can be assigned (called labelling) a class of 0 or 1 randomly. Now, Q points can be labeled in 2Q different ways. For example, for three points in the plane R^2, the eight possible labeling are shown in Fig. 1.

For the eight possible labeling the threshold logic neuron (TLN) can correctly separate or classify all eight configurations as shown in Fig 1. This is achieved by carefully placing the hyperplane to have the correct orientation such that all points to be classified as a +1 lie on the positive side of

the hyperplane indicated by a small arrow. Now, it can be said that the VC dimension of the set of oriented straight lines in R^2 is three.

The SV(Support Vector) calculation is a nonlinear speculation of the summed up Portrait calculation created in Russia in the sixties. VC hypothesis has been created in the course of the most recent three decades by Vapnik, Chervonenkis and others. This hypothesis describes properties of learning machines which empower them to adequately sum up the inconspicuous information. In its current structure, the SV machine has been created AT and T Bell Laboratories by Vapnik and associates. Beginning work has concentrated on OCR (optical character acknowledgment). Inside brief period, SV classifiers have gotten serious with the best accessible frameworks for both OCR and item acknowledgment assignments. Burges distributed an extensive instructional exercise on SV classifiers. Superb exhibitions have been gotten in relapse and time arrangement forecast applications.

Measurable Learning Theory has given a powerful structure to arrangement and relapse undertakings including highlights. Bolster Vector Machines (SVM) are legitimately gotten from this structure and they work by taking care of an obliged quadratic issue where the raised target work for minimization is given by the mix of a misfortune work with a regularization term (the standard of the loads). While the regularization term is straightforwardly connected, through a hypothesis, to the VC-measurement of the speculation space, and in this way completely defended,the misfortune work is for the most part (heuristically) picked based on the job that needs to be done. Conventional/measurable relapse methods are frequently expressed as the procedures inferring a capacity f(x) that has minimal deviation among anticipated and tentatively watched reactions for all preparation models. One of the primary qualities of Support Vector Regression (SVR) is that as opposed to limiting the watched preparing blunder, SVR endeavors to limit the summed-up mistake bound in order to accomplish summed up execution. This speculation mistake bound is the blend of the preparation blunder and a regularization term that controls the multifaceted nature of the theory space.

Bolster vector machine (SVM) has been first presented by Vapnik. There are two primary classifications for help vector machines: bolster vector order (SVC) and bolster vector relapse (SVR). SVM is a learning framework utilizing a high dimensional component space. It yields

expectation works that are developed a subset of help vectors. SVM can sum up confounded dim level structures with just a not many help vectors and subsequently gives another component to picture pressure. A variant of a SVM for relapse has been proposed in 1997 by Vapnik, Steven Golowich, and Alex Smola. This strategy is called bolster vector relapse (SVR). The model created by help vector order just relies upon a subset of the preparation information, in light of the fact that the cost work for building the model couldn't care less about preparing focuses that lie past the edge. Similarly, the model created by SVR just relies upon a subset of the preparation information, on the grounds that the cost work for building the model overlooks any preparation information that is close (inside a limit ε) to the model forecast.

Bolster Vector Regression (SVR) is the most widely recognized application type of SVMs. A review of the essential thoughts basic help vector (SV) machines for relapse and capacity estimation has been surrendered. Moreover, it has incorporated an outline of at present utilized calculations for preparing SVMs, covering both the quadratic (or arched) programming part and propelled techniques for managing enormous datasets. At last, a few adjustments and augmentations have been applied to the standard SV calculation. It has talked about the part of regularization and limit control from a SV perspective. The objective has been to discover a capacity f(x) that has all things considered ε deviation from the really acquired targets yi for all the preparation information and simultaneously as level as could be expected under the circumstances.

Akaike Information Criteria (AIC) and Bayesian Information Criteria (BIC)

$$AIC(d) = R_{emp}(d) + \frac{2d}{n}\hat{\sigma}^2$$

$$BIC(d) = R_{emp}(d) + (\ln n)\frac{d}{n}\hat{\sigma}^2$$

In AIC and BIC, d is the quantity of free parameters (of a straight estimator) and denotes the standard deviation of added substance clamor in standard relapse detailing under general setting for prescient learning for example y = g (x) + where is i.i.d. (free and indistinguishably dispersed) zero mean arbitrary blunder (clamor), x is a multidimensional info and y is a scalar yield. Both AIC and BIC are determined utilizing asymptotic examination (for example enormous example size). Also, AIC expect that right model has a place with the arrangement of potential models. Practically speaking, be that as it may, AIC and BIC are frequently utilized when these presumptions don't hold. When utilizing AIC or BIC for commonsense model determination, one is confronted with following two issues: (1) estimation and importance of (obscure) clamor difference, (2) estimation of model multifaceted nature. A third model choice technique in some cases utilized practically speaking depends on Structural Risk Minimization (SRM) which gives an exceptionally broad and amazing system for model multifaceted nature control. Under SRM, a lot of potential models frames a settled structure so every component (of this structure) speaks to a lot of models of fixed unpredictability. Thus, a structure gives regular requesting of potential models as indicated by their multifaceted nature. Model choice adds up to picking an ideal component of a structure utilizing VC speculation limits. For relapse issues, the accompanying VC – bound has been utilized:

$$R(h) \le R_{emp}(h)\left(1-\sqrt{p-p\ln p+\frac{\ln n}{2n}}\right)^{-1}$$

where p = h/n and h are a proportion of model intricacy (called VC-measurement). The model intricacy has been evaluated precisely and the direct estimators initially looked at and utilized in rough heuristic appraisals of model multifaceted nature for k-closest neighbor relapse where exact assessments of model unpredictability are not known.

In help vector machines, time multifaceted nature shows up observationally to locally develop directly with the quantity of models, while speculation execution can be improved. Non-direct arrangement and capacity guess is a significant subject of enthusiasm with ceaselessly developing examination zones. Estimation procedures dependent on regularization and piece techniques assume a significant job. Bolster vector machines are a group of information examination

calculations, in light of raised quadratic programming. Their utilization has been exhibited in grouping, relapse and bunching issues. Bolster vector relapse (SVR) fits a persistent esteemed capacity to information such that shares a considerable lot of the benefits of help vector machines order. Most calculations for SVR necessitate that the preparation tests be conveyed in a solitary group.

A tool compartment LS-SVMlab for Matlab has been given first usage for various LS-SVM related calculations. Most capacities can deal with datasets up to 20,000 information focuses or more. LS-SVMlabs interface for Matlab comprises of an essential rendition for apprentices just as a further developed variant with programs for multi-class encoding procedures and a Bayesian structure. The Matlab tool compartment is worked around a quick LS-SVM preparing and reproduction calculation. The relating calls can be utilized for arrangement just as capacity estimation. Added substance models have been portrayed dependent on least square help vector machines (LS-SVM) which are fit for taking care of higher dimensional information for relapse just as characterization undertakings. Expansions of LS-SVMs towards strength, scantiness and weighted variants, just as various strategies for tuning of hyper-parameters have been incorporated. Points of interest of utilizing part savvy LS-SVMs incorporate the productive estimation of added substance models as for old style practice, interpretability of the evaluated model, open doors towards structure discovery and the association with existing measurable procedures.

Following a gradual help vector grouping calculation, an exact on-line bolster vector relapse (AOSVR) has been created. AOSVR has proficiently refreshed a prepared SVR work at whatever point an example has been added to or expelled from the preparation set. The refreshed SVR work has been indistinguishable from that delivered by a bunch calculation.

SVR has been explored as a substitute strategy for approximating complex building examinations. The computationally productive hypothesis behind SVR has been explored and SVR approximations have been thought about against the distinctive meta-displaying procedures utilizing a testbed of 26 building investigation capacities. SVR has accomplished more precise and more strong capacity approximations than the meta-displaying methods. To conquer the colossal time and computational expenses of running complex designing codes, an estimation of the intricate investigation code known as "metamodels" has been portrayed. Numerically, if the

contributions to the real PC investigation are provided in vector x, and the yields from the examination in vector y, the genuine computational code assesses:

$y = f(x)$ where $f(x)$ is a complex engineering analysis function.

The computationally efficient metamodel approximation is:

$\hat{y} = g(x)$ such that $y = \hat{y} + \varepsilon$ where includes both approximation and random errors.

Lin and Weng have proposed a basic methodology for probabilistic forecast reasonable for the standard SVR and they have begun with producing out-of-test residuals by cross approval (CV), and afterward fitted the residuals by basic parametric models like Gaussian and Laplace. Bolster Vector Machines have risen as a groundbreaking multivariate demonstrating method for grouping just as relapse purposes.

Center Vector Machine (CVM) calculation abuses the "estimation" in the structure of SVM usage. The ideal arrangement is approximated by an iterative methodology. Ordinarily, the halting model uses either the accuracy of the Lagrange multipliers or the duality hole. The CVM calculation has an asymptotic time unpredictability that is straight in m, m being the quantity of preparing designs and a space multifaceted nature that is free of m. Materialness of the CVM relies upon the accompanying conditions: (1) the bit k fulfills k(x,x) = a consistent; and (2) the QP (quadratic programming) of the part technique is of an uncommon structure. There is no direct term in the QP's goal. The double target of SVR contains a direct term and isn't of the necessary structure. An improvement of the CVM permitting an increasingly broad QP plan lifts the condition on the part [27]. The resultant Core Vector Regression (CVR) calculation can be utilized with any straight/nonlinear bits and can acquire roughly ideal arrangements. The resultant CVR technique acquires the effortlessness of CVM, and has little asymptotic reality complexities. Tentatively, it is as exact as existing SVR usage, however is a lot quicker and creates far less help vectors (and consequently quicker testing) on huge informational indexes. This expansion can likewise be utilized for scaling up other part techniques, for example, positioning SVM, SVMs in imbalanced learning issues, and SVMs with reliant and organized yields.

3.4.3 Decision Tree Regression

Decision trees follow their sources to the period of the early improvement of put down accounts. This history delineates a significant quality of trees: extraordinarily interpretable outcomes which have an instinctive tree-like presentation which, thusly, improves understanding and the spread of results. The computational starting points of choice trees—once in a while called characterization trees or relapse trees—are models of organic and subjective procedures. This normal legacy drives integral improvements of both factual choice trees and trees intended for AI. The unfurling and dynamic explanation of the different highlights of trees all through their initial history in the late twentieth century is talked about alongside the significant related reference focuses and capable creators. Measurable methodologies, for example, a speculation testing and different resampling approaches, have coevolved alongside AI executions. This had brought about particularly versatile choice tree apparatuses, suitable for different factual and AI assignments, across different degrees of estimation, with changing degrees of information quality. Trees are strong within the sight of missing information and offer various methods of consolidating missing information in the subsequent models. In spite of the fact that trees are ground-breaking, they are likewise adaptable and simple to utilize techniques. This guarantees the creation of great outcomes that require barely any suspicions to convey. The treatment closes with a conversation of the most present advancements which keep on depending on the cooperative energies and cross-preparation among measurable and AI people group. Current improvements with the development of different trees and the different resampling approaches that are utilized are examined.

3.4.3.1 Basic Algorithm

Calculation shows the conventional depiction of choice tree enlistment. The choice tree acceptance calculation starts with a vacant tree. Since this calculation is a recursive calculation, it needs to have the end condition. On the off chance that the calculation doesn't end, it proceeds by actuating a root hub that considers the whole dataset for developing the tree. For the root hub, the calculation forms every datum in the dataset D by repeating over every single accessible trait and picking the quality with best data hypothetical measures. Further clarify the figuring of the rules. After the calculations settles on abest, it makes a split-hub that utilizes abest to test every datum in the dataset. This testing procedure instigates sub-dataset Dv. The calculation proceeds by recursively

preparing each sub-dataset in Dv. This recursive call produces sub-tree Treev. The calculation at that point connects Treev. Into its relating branch in the comparing split-hub.

3.4.3.2 Techniques in Decision Tree Induction

Sadly, data addition and increase proportion are insufficient to manufacture a choice tree that suits creation settings. One case of notable choice tree execution is. This discloses further procedures to make the calculation appropriate for creation settings. Proposes tree pruning method to abstain from overfitting by diminishing the quantity of hubs in the tree. Normally plays out this strategy in a solitary base up go after the tree is completely developed. It presents negative pruning that assesses the mistake pace of a hub dependent on the evaluated blunders of its sub-branches. On the off chance that the assessed mistake of a hub is not as much as its sub-branches' blunder, at that point cynical pruning utilizes the hub to supplant its sub-branches.

The following method is in term of ceaseless properties dealing with. Ceaseless ascribes require the calculation to pick limit esteems to decide the quantity of parts. To deal with this necessity, uses just the data gain procedure in picking the limit. For picking the property, it despite everything utilizes the data gain and the increase proportion methods by and large. It additionally has a few prospects in dealing with missing characteristic qualities during learning and testing the tree. There are three situations when it needs to deal with the missing qualities appropriately, they are:

1. When contrasting qualities with split and a few properties have missing qualities.
2. After it parts a hub into a few branches, a preparation datum with missing qualities
3. shows up into the split hub and the split hub can't connect the datum with any

   of its branches.

4. When it endeavors to arrange a testing datum with missing qualities, yet it cannot connect the datum with any of the accessible branches in a split hub.

It presents a coding plan in and to manage each of the previously mentioned situations. Instances of the coding plan are: to overlook preparing occasions with missing qualities and substitute the missing qualities with the most widely recognized an incentive for ostensible properties or with the mean of the known qualities for the numeric characteristics.

Very Fast Decision Tree (VFDT) Induction

We quickly depicted in segment Very Fast Decision Tree (VFDT) which is the pioneer of spilling choice tree acceptance. VFDT satisfies the essential necessities in taking care of information streams in an effective manner. The past spilling choice tree calculations that presented before VFDT doesn't have this trademark. VFDT is one of the central ideas in our work, thus this area further talks about the calculation. This area alludes the subsequent model from VFDT as Hoeffding tree and the acceptance calculation as Hoeffding tree enlistment. We allude to information as cases, subsequently we additionally allude datum as a solitary occasion. Besides, this segment alludes the entire usage of VFDT as VFDT.

Calculation shows the conventional portrayal of Hoeffding tree enlistment. During the learning stage, VFDT begins Hoeffding tree with just a solitary hub. For each preparation case E that shows up into the tree, VFDT summons Hoeffding tree acceptance calculation. The calculations begins by arranging the occasion into a leaf l. This leaf is a learning leaf, along these lines the calculation needs to refresh the adequate measurement in l. For this situation, the adequate measurement is the class dispersion for each property estimation. The calculations additionally increase the quantity of examples (nl) seen at lea l dependent on E's weight. One occurrence isn't sufficiently critical to develop the tree, accordingly the calculation just develops the tree each specific number of cases (nmin). The calculation doesn't develop the trees if all the information seen at l have a place with a similar class. Here likewise demonstrates these two conditions to conclude whether to develop or not to develop the tree.

In this calculation, developing the tree implies endeavoring to part the hub. To play out the split, the calculation repeats through each trait and ascertain the comparing data hypothetical standards Gl(Xi) . It likewise processes the data hypothetical standards for no-split situation (X;). The creators of VFDT alludes this incorporation of no-split situation with term pre-pruning. The calculation at that point picks the best (Xa) and the subsequent best (Xb)) traits dependent on the models. Utilizing these picked properties, the calculation registers the Hoeffding bound to decide if it needs to part the hub or not. Additionally demonstrates the total condition to part the hub. On the off chance that the best property is the no-split situation (X;), at that point the calculation doesn't play out the split. The calculation utilizes tiebreaking component to deal with the situation

where the distinction of data gain among Xa and Xb) is exceptionally little. On the off chance that the calculation parts the hub, at that point it replaces the leaf with a split hub and it makes the comparing leaves dependent on the best characteristic.

What makes Hoeffding bound alluring is its capacity to give similar outcomes in any case the likelihood appropriation producing the perceptions. This engaging quality accompanies one disadvantage which is distinctive number of perceptions to arrive at specific qualities relying upon the likelihood conveyances of the information. VFDT has no end condition since it is a gushing calculation. The tree may develop boundlessly and this negates one of the necessities for calculation for gushing setting (require constrained measure of memory). To fulfill the necessity of constrained memory utilization, the creators of VFDT present hub restricting method. This procedure figures the guarantee for every dynamic learning leaf l. A guarantee of a functioning learning leaf in VFDT is characterized as an expected upper-bound of the blunder decrease accomplished by keeping the leaf dynamic. In view of the guarantee, the calculation may decide to deactivate leaves with low guarantee when the tree arrives at as far as possible. In spite of the fact that the leaves are dormant, VFDT still screens the guarantee for each inert leaf. The explanation is that VFDT may enact the dormant leaves when their guarantees are higher than right now dynamic leaves' guarantees. Other than the hub restricting procedure, the VFDT creators present additionally poor-credit expulsion strategy to lessen VFDT memory utilization. VFDT evacuates the qualities that doesn't look encouraging while at the same time parting, consequently the measurement related with the expelled trait can be erased from the memory.

3.4.4 Random Forest Regression

Irregular woodlands are a plan proposed by Leo Breiman in the 2000's for building an indicator troupe with a lot of choice trees that develop in haphazardly chose subspaces of information. In spite of developing interest and viable use, there has been little investigation of the measurable properties of arbitrary woodlands, and little is thought about the numerical powers driving the calculation. In this paper, we offer an inside and out investigation of an arbitrary timberlands model proposed by Breiman (2004), which is near the first calculation. We appear specifically that the

technique is steady and adjusts to sparsity, as in its pace of union relies just upon the quantity of solid highlights and not on what number of commotion factors are available.

In a progression of papers and specialized reports, Breiman (1996, 2000, 2001, 2004) showed that significant gains in characterization and relapse precision can be accomplished by utilizing outfits of trees, where each tree in the group is developed as per an arbitrary parameter. Last expectations are acquired by collecting over the troupe. As the base constituents of the group are tree-organized indicators, and since every one of these trees is developed utilizing an infusion of haphazardness, these methods are designated "arbitrary woods."

Breiman's thoughts were definitively impacted by the early work of Amit and Geman (1997) on geometric element determination, the arbitrary subspace strategy for Ho (1998) and the irregular split choice methodology of Dietterich (2000). As featured by different exact investigations (see for example Breiman, 2001; Svetnik et al., 2003; Diaz-Uriarte and de Andr'es, 2006; Genuer et al., 2008, 2010), irregular woodlands have developed as genuine contenders to cutting edge techniques, for example, boosting (Freund and Shapire, 1996) and bolster vector machines (Shawe-Taylor and Cristianini, 2004). They are quick and simple to execute, produce profoundly precise forecasts and can deal with an enormous number of info factors without overfitting. Truth be told, they are viewed as one of the most precise universally useful learning procedures accessible. The review by Genuer et al. (2008) may give the peruser down to earth rules and a decent beginning stage for understanding the technique.

In Breiman's methodology, each tree in the assortment is shaped by first choosing aimlessly, at every hub, a little gathering of info facilitates (likewise called highlights or factors from now on) to part on and, furthermore, by computing the best part dependent on these highlights in the preparation set. The tree is developed utilizing CART approach (Breiman et al., 1984) to most extreme size, without pruning. This subspace randomization plot is mixed with sacking (Breiman, 1996; B¨uhlmann and Yu, 2002; Buja and Stuetzle, 2006; Biau et al., 2010) to resample, with substitution, the preparation informational collection each time another individual tree is developed. Despite the fact that the component seems basic, it includes a wide range of main

thrusts which make it hard to break down. Truth be told, its numerical properties stay to date to a great extent obscure and, up to now, most hypothetical examinations have focused on detached parts or adapted variants of the calculation. Fascinating endeavors with regards to this heading are by Lin and Jeon (2006), who set up an association between irregular woods and versatile closest neighbor techniques (see likewise Biau and Devroye, 2010, for additional outcomes); Meinshausen (2006), who examines the consistency of arbitrary timberlands with regards to restrictive quantile forecast; and Biau et al. (2008), who offer consistency hypotheses for different streamlined variants of irregular woods and other randomized troupe indicators. By and by, the factual system of "valid" irregular woodlands isn't yet completely comprehended is still under dynamic examination.

In the current paper, we go above and beyond into irregular woodlands by working out and cementing the properties of a model recommended by Breiman (2004). In spite of the fact that this model is as yet straightforward contrasted with the "valid" calculation, it is by and by nearer to reality than some other plan we know about. The short draft of Breiman (2004) is basically founded on instinct and scientific heuristics, some of them are flawed and make the archive hard to peruse and comprehend. Be that as it may, the thoughts introduced by Breiman merit explaining and creating, and they will fill in as a beginning stage for our investigation.

Before we formalize the model, a few definitions are all together. All through the report, we guess that we are given a preparation test Dn = {(X1,Y1), . . . , (Xn,Yn)} of i.i.d. [0,1]d ×R-esteemed irregular factors (d _2) with a similar appropriation as a free conventional pair (X,Y) fulfilling EY2 < □. The space [0,1]d is outfitted with the standard Euclidean measurement. For fixed x 2 [0,1]d, we will likely gauge the relapse work r(x) = E[Y|X = x] utilizing the information Dn. In this regard, we state that a relapse work gauge rn is predictable if E[rn(X)−r(X)]2 !0 as n!□. The principle message of this paper is that Breiman's technique is steady and adjusts to sparsity, as in its pace of intermingling relies just upon the quantity of solid highlights and not on what number of commotion factors are available.

3.4.4.1 Random Forest Model

Officially, an irregular timberland is an indicator comprising of an assortment of randomized base relapse trees. These arbitrary trees are joined to shape the accumulated relapse gauge

where denotes desire concerning the arbitrary parameter, restrictively on X and the informational index Dn. In the accompanying, to help documentation a bit, we will overlook the reliance of the appraisals in the example and compose. Note that, practically speaking, the above desire is assessed by Monte Carlo, that is, by producing M (normally enormous) irregular trees, and taking the normal of the individual results (this strategy is legitimized by the law of huge numbers, see the reference section in Breiman, 2001). The randomizing variable is used to decide how the progressive cuts are performed when assembling the individual trees, for example, choice of the organize to part and position of the split. In the model we have as a main priority, the variable is thought to be free of X and the preparation test Dn. This prohibits specifically any bootstrapping or resampling step in the preparation set. This likewise precludes any information subordinate system to fabricate the trees, for example, looking for ideal parts by advancing some basis on the real perceptions. Nonetheless, we permit to be founded on a subsequent example, free of, yet conveyed as, Dn. This significant issue will be completely examined.

In view of these alerts, we will accept that every individual irregular tree is built in the accompanying manner. All hubs of the tree are related with rectangular cells to such an extent that at each progression of the development of the tree, the assortment of cells related with the leaves of the tree. The foundation of the tree is itself. The accompanying strategy is then rehashed occasions, where is the base-2 logarithm, the roof work and a deterministic parameter, fixed already by the client, and conceivably relying upon n.

Each randomized tree yields the normal over all Yi for which the relating vectors Xi fall in a similar cell of the arbitrary parcel as X. At the end of the day, letting be the rectangular cell of the arbitrary parcel containing X.

Let us currently offer some broad comments about this irregular woodland model. As a matter of first importance, we note that, by development, every individual tree has precisely terminal hubs, and each leaf has Lebesgue measure. In this way, if X has uniform dissemination on, there will be

on normal about perceptions per terminal hub. Specifically, the decision incites few cases in the last leaves, as per the possibility that the single trees ought not be pruned. Next, we see that, during the development of the tree, at every hub, every applicant facilitate might be picked with likelihood. This suggests specifically. In spite of the fact that we don't exact for the second the manner in which these probabilities are produced, we stress that they might be prompted by a subsequent example. This incorporates the circumstance where, at every hub, arbitrariness is presented by choosing aimlessly (with or without substitution) a little gathering of info highlights to part on, and deciding to cut the phone along the organize—inside this gathering—which most abatements some exact rule assessed on the additional example. This plan is near what the first irregular woods calculation does, the basic distinction being that the last calculation utilizes the genuine informational collection to ascertain the best parts. This point will be appropriately talked about. At last, the prerequisite that the parts are constantly accomplished at the center of the cell sides is fundamentally specialized, and it could in the long run be supplanted by a progressively included arbitrary instrument—in light of the subsequent example—at the cost of a considerably more convoluted examination. The archive is composed as follows. We demonstrate that the arbitrary backwoods relapse gauge $\bar{}$rn is reliable and talk about its pace of assembly. As a striking outcome, we appear under a sparsity structure that the pace of combination relies just upon the quantity of dynamic (or solid) factors and not on the element of the encompassing space. This component is especially attractive in high-dimensional relapse, when the quantity of factors can be a lot bigger than the example size, and may clarify why arbitrary backwoods can deal with an enormous number of information factors without overfitting. Area 3 is given to a conversation, and a little reenactment study is introduced in. For lucidity, proofs are deferred to.

3.4.4.2 Analysis

Throughout the document, we denote by $Nn(\mathbf{X},\Theta)$ the number of data points falling in the same cell

as **X**, that is,

$$N_n(\mathbf{X},\Theta) = \sum_{i=1}^{n} \mathbf{1}_{[\mathbf{X}_i \in A_n(\mathbf{X},\Theta)]}.$$

We start the analysis with the following simple theorem, which shows that the random forests estimate

¯*rn* is consistent.

Theorem 1 *Assume that the distribution of* **X** *has support on* [0,1]*d. Then the random forests estimate* ¯*rn is consistent whenever pn j* log*kn* !∞ *for all j* = 1, . . . ,*d and kn*/*n*!0 *as n*!∞.

Theorem 1 mainly serves as an illustration of how the consistency problem of random forests predictors may be attacked. It encompasses, in particular, the situation where, at each node, the coordinate to split is chosen uniformly at random over the *d* candidates. In this "purely random" model, independently of *n* and *j*, and consistency is ensured as long as. This is however a radically simplified version of the random forests used in practice, which does not explain the good performance of the algorithm. To achieve this goal, a more indepth analysis is needed. There is empirical evidence that many signals in high-dimensional spaces admit a sparse representation.

As an example, wavelet coefficients of images often exhibit exponential decay, and a relatively small subset of all wavelet coefficients allows for a good approximation of the original image. Such signals have few non-zero coefficients and can therefore be described as sparse in the signal domain (see for instance Bruckstein et al., 2009). Similarly, recent advances in highthroughput technologies—such as array comparative genomic hybridization—indicate that, despite the huge dimensionality of problems, only a small number of genes may play a role in determining the outcome and be required to create good predictors (van't Veer et al., 2002, for instance). Sparse estimation is playing an increasingly important role in the statistics and machine learning communities, and several methods have recently been developed in both fields, which rely upon the notion of sparsity (e.g., penalty methods like the Lasso and Dantzig selector, see Tibshirani, 1996; Cand`es and Tao, 2005; Bunea et al., 2007; Bickel et al., 2009, and the references therein).

3.4.4.3 Discussion

The outcomes which have been gotten depend on proper conduct of the likelihood. We review that these groupings ought to be and comply with the imperatives, where the pattern to 0 as n keeps an eye on endlessness. At the end of the day, at each progression of the development of the individual trees, the irregular system should follow and specially cut the solid directions. In this progressively casual segment, we quickly talk about an irregular instrument for instigating such likelihood groupings.

Assume, to begin with a fanciful situation, that we definitely realize which directions are solid, and which are most certainly not. In this perfect case, the irregular determination methodology depicted in the presentation might be effectively made increasingly exact as follows. A positive whole number conceivably relying upon fixed heretofore and the accompanying parting plan is iteratively rehashed at every hub of the tree:

1. Select at arbitrary, with substitution, Mn up-and-comer directions to part on.
2. If the determination is all feeble, at that point pick one at arbitrary to part on. On the off chance that there is more than one in number variable chosen, pick one aimlessly and cut.

Inside this system, it is anything but difficult to see that each facilitate in S will be cut with the "perfect" likelihood

$$p_n^\star = \frac{1}{S}\left[1 - \left(1 - \frac{S}{d}\right)^{M_n}\right]$$

Though this is an idealized model, it already gives some information about the choice of the parameter

Mn, which, in accordance with the results of Section 2 (Corollary 6), should satisfy

$$\left(1 - \frac{S}{d}\right)^{M_n} \log n \to 0 \quad \text{as } n \to \infty.$$

This is true as soon as

$$M_n \to \infty \quad \text{and} \quad \frac{M_n}{\log n} \to \infty \quad \text{as } n \to \infty.$$

This outcome is predictable with the general exact seeing that doesn't require as exceptionally huge (see, for instance, Breiman, 2001), yet not with the broad conviction that ought not rely upon. Note additionally that on the off chance that the highlights are picked indiscriminately without substitution, at that point things are significantly increasingly basic since, for this situation, sufficiently enormous.

By and by, we have just an obscure thought regarding the size and substance of the set S. Be that as it may, to go around this issue, we may utilize the perceptions of a free second set so as to copy the perfect split likelihood.

To illustrate this mechanism, suppose—to keep things simple—that the model is linear, that is,

$$Y = \sum_{j \in \mathcal{S}} a_j X^{(j)} + \varepsilon,$$

where $\mathbf{X} = (X(1), \ldots, X(d))$ is uniformly distributed over $[0,1]d$, the aj are non-zero real numbers, and ε is a zero-mean random noise, which is assumed to be independent of $\mathbf{X}$ and with finite variance. Note that, in accordance with our sparsity assumption, $r(\mathbf{X}) = \sum j2S\, ajX(j)$ depends on $\mathbf{X}S$ only. Assume now that we have done some splitting and arrived at a current set of terminal nodes. Consider any of these nodes, say $A = \prod dj\ =1\ Aj$, fix a coordinate $j\ 2\ \{1, \ldots, d\}$, and look at the weighted conditional variance $\mathsf{V}[Y|X(j)\ 2\ Aj]\mathsf{P}(X(j)\ 2\ Aj)$. It is a simple exercise to prove that if $\mathbf{X}$ is uniform and $j\ 2\ S$, then the split on the j-th side which most decreases the weighted conditional variance is at the midpoint of the node, with a variance decrease equal to $a2j\ /16 > 0$. On the other hand, if $j\ 2W$, the decrease of the variance is always 0, whatever the location of the split. On the

down to earth side, the restrictive differences are obviously obscure, yet they might be evaluated by supplanting the hypothetical amounts by their individual example gauges (as in the CART

method, see Breiman, 2001, Chapter 8, for a careful conversation) assessed on the second example.

This recommends the accompanying method, at every hub of the tree:

1. Select at irregular, with substitution, Mn competitor directions to part on.
2. For every one of the Mn chose arranges, figure the best split, that is, the part which most abatements the inside hub whole of squares on the subsequent example.
3. Select one variable at arbitrary among the directions which yield the best inside hub aggregate of squares diminishes, and cut.

This technique is in reality near what the irregular backwoods calculation does. The fundamental contrast is that we expected to have close by a subsequent example, while the first calculation plays out the hunt of the ideal cuts on the first perceptions. This point is significant, since the utilization of an additional example saves the autonomy of □□the arbitrary component and the preparation test. We don't know whether our outcomes are still obvious if □□depends on as in the CART calculation, yet the examination doesn't give off an impression of being straightforward. Note likewise that, at stage 3, an edge (or a test system, as recommended in Amaratunga et al., 2008) could be utilized to pick among the hugest factors, though the real calculation just chooses the best one. Indeed, contingent upon the unique situation and the genuine cut determination strategy, the useful probabilities may comply with the requirements. This ought not influence the aftereffects of the article.

This observational randomization plot prompts confound probabilities of cuts which, this time, shift at every hub of each tree and are not effectively managable to investigation. All things considered, seeing that the normal number of cases per terminal hub is about, it might be construed by the law of enormous numbers that every factor in S will be cut with likelihood

$$p_{nj} \approx \frac{1}{S}\left[1-\left(1-\frac{S}{d}\right)^{M_n}\right](1+\zeta_{nj}),$$

where $\zeta n\, j$ is of the order $O(kn/n)$, a quantity which anyway goes fast to 0 as n tends to infinity. Put differently, for j 2 S,

$$p_{nj} \approx \frac{1}{S}(1+\xi_{nj}),$$

where $\xi n\, j$ goes to 0 and satisfies the constraint. 0 as n tends to infinity, provided. This is intelligent with the prerequisites of Corollary 6. We understand anyway this is a harsh methodology, and that progressively hypothetical work is required here to completely comprehend the components engaged with CART and Breiman's unique randomization process.

It is likewise critical that irregular timberlands utilize the purported out-of-sack tests (i.e., the bootstrapped information which are not used to fit the trees) to develop a variable significance basis, which gauges the forecast quality of each component (see, e.g., Genuer et al., 2010). To the extent we know, there is to date no methodical scientific investigation of this basis. It is our conviction that such an examination would extraordinarily profit by the sparsity perspective created in the current paper, yet is shockingly much past its degree. In conclusion, it would likewise be fascinating to work out and stretch out our outcomes to the setting of unaided learning of trees. A decent course to follow with this regard is given by the techniques laid out in Section 5.5 of Amit and Geman (1997).

CHAPTER 4

DATASET

The source of data is the UCI Machine Learning Repository. The donor of the dataset is Pr. Yeh, I-Cheng. This dataset deals with real estate located in Sindian district, in New Taipei city in Taiwan. There are 414 instances, which have 7 attributes:

Designation	Attribute	Unit	Value for example
X1	Transaction date	year and month	2013.250=2013 March, 2013.500=2013 June, etc.
X2	House age	year	32
X3	Distance to the nearest MRT station	meter	84.87882
X4	Number of convenience stores in the living circle on foot	Integer	10
X5	Geographic coordinate: latitude	degree	24.98298
X6	Geographic coordinate: longitude	degree	121.54024
y	house price of unit area	10,000 New Taiwan Dollar/Ping, where Ping is a local unit, 1 Ping = 3.3 meter squared	37.9

4.1 1 Attributes of dataset

	A	B	C	D	E	F	G	H
1	No	X1 transaction date	X2 house age	X3 distance to the nearest MRT	X4 number of convenience store	X5 latitude	X6 longitude	Y house price of unit area
2	1	2012.917	32	84.87882	10	24.98298	121.54024	37.9
3	2	2012.917	19.5	306.5947	9	24.98034	121.53951	42.2
4	3	2013.583	13.3	561.9845	5	24.98746	121.54391	47.3
5	4	2013.500	13.3	561.9845	5	24.98746	121.54391	54.8
6	5	2012.833	5	390.5684	5	24.97937	121.54245	43.1
7	6	2012.667	7.1	2175.03	3	24.96305	121.51254	32.1
8	7	2012.667	34.5	623.4731	7	24.97933	121.53642	40.3
9	8	2013.417	20.3	287.6025	6	24.98042	121.54228	46.7
10	9	2013.500	31.7	5512.038	1	24.95095	121.48458	18.8
11	10	2013.417	17.9	1783.18	3	24.96731	121.51486	22.1
12	11	2013.083	34.8	405.2134	1	24.97349	121.53372	41.4
13	12	2013.333	6.3	90.45606	9	24.97433	121.5431	58.1
14	13	2012.917	13	492.2313	5	24.96515	121.53737	39.3
15	14	2012.667	20.4	2469.645	4	24.96108	121.51046	23.8
16	15	2013.500	13.2	1164.838	4	24.99156	121.53406	34.3
17	16	2013.583	35.7	579.2083	2	24.9824	121.54619	50.5
18	17	2013.250	0	292.9978	6	24.97744	121.54458	70.1
19	18	2012.750	17.7	350.8515	1	24.97544	121.53119	37.4
20	19	2013.417	16.9	368.1363	8	24.9675	121.54451	42.3
21	20	2012.667	1.5	23.38284	7	24.96772	121.54102	47.7
22	21	2013.417	4.5	2275.877	3	24.96314	121.51151	29.3
23	22	2013.417	10.5	279.1726	7	24.97528	121.54541	51.6
24	23	2012.917	14.7	1360.139	1	24.95204	121.54842	24.6

4.1 2 2 excel file of dataset

4.2 The distribution of the continuous variables:

First of all, we import libraries and dataset and check the distribution. When dealing with a set of data, often the first thing you'll want to do is get a sense for how the variables are distributed. Python is a great language for doing data analysis, primarily because of the fantastic ecosystem of data-centric Python packages. Pandas is one of those packages and makes importing and analyzing data much easier.

Pandas describe() is used to view some basic statistical details like percentile, mean, std etc. of a data frame or a series of numeric values.

Out[3]:

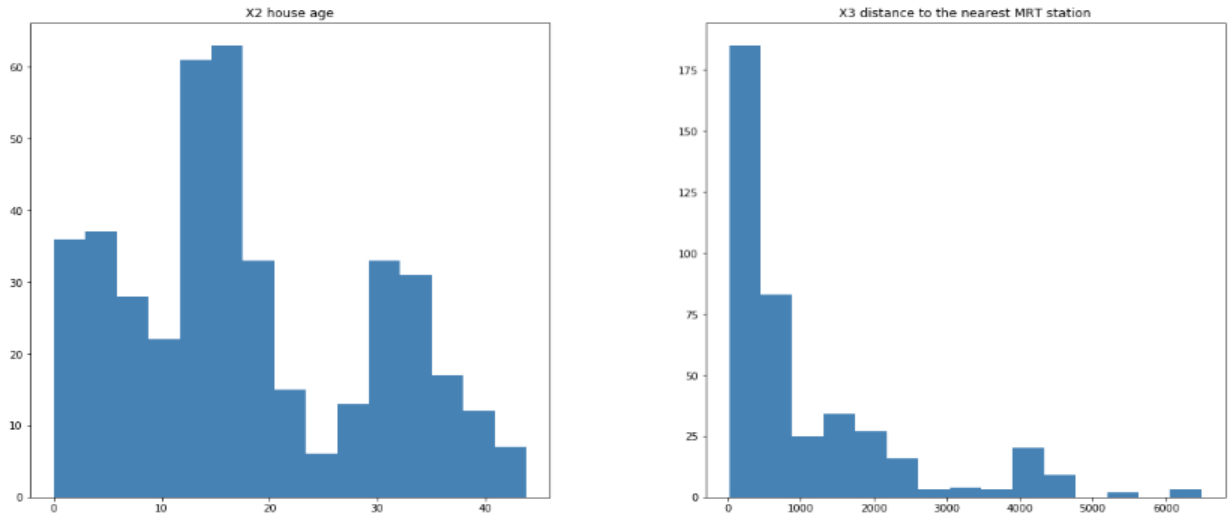
	No	X1 transaction date	X2 house age	X3 distance to the nearest MRT station	X4 number of convenience stores	X5 latitude	X6 longitude	Y house price of unit area
count	414.000000	414.000000	414.000000	414.000000	414.000000	414.000000	414.000000	414.000000
mean	207.500000	2013.148953	17.712560	1083.885689	4.094203	24.969030	121.533361	37.980193
std	119.655756	0.281995	11.392485	1262.109595	2.945562	0.012410	0.015347	13.606488
min	1.000000	2012.666667	0.000000	23.382840	0.000000	24.932070	121.473530	7.600000
25%	104.250000	2012.916667	9.025000	289.324800	1.000000	24.963000	121.528085	27.700000
50%	207.500000	2013.166667	16.100000	492.231300	4.000000	24.971100	121.538630	38.450000
75%	310.750000	2013.416667	28.150000	1454.279000	6.000000	24.977455	121.543305	46.600000
max	414.000000	2013.583333	43.800000	6488.021000	10.000000	25.014590	121.566270	117.500000

4.2 1 Distribution of Dataset

Histograms

When exploring a dataset, you'll often want to get a quick understanding of the distribution of certain numerical variables within it. A common way of visualizing the distribution of a single numerical variable is by using a histogram. A histogram divides the values within a numerical variable into "bins", and counts the number of observations that fall into each bin. By visualizing these binned counts in a columnar fashion, we can obtain a very immediate and intuitive sense of the distribution of values within a variable.

This recipe will show you how to go about creating a histogram using Python. Specifically, you'll be using pandas hist() method, which is simply a wrapper for the matplotlib pyplot API.

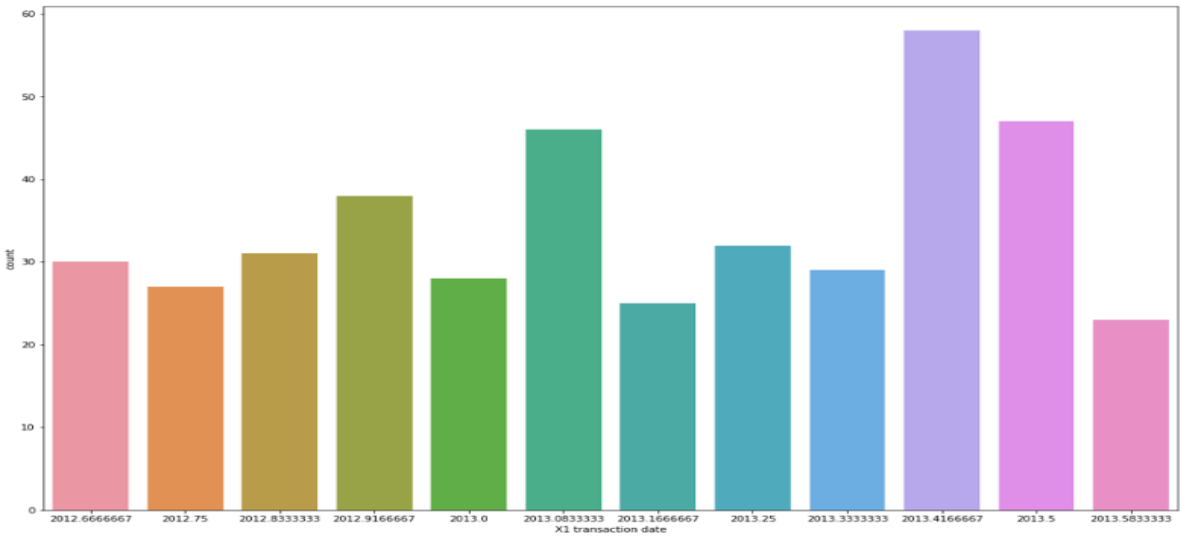


4.2 2 histrogram of house age and distance to the nearest MRT station

- **House age varies from 0 to 43 years, with 3 peaks: 12-18 years, 0-6 years, 30-36 years.**
- **Most houses are located within 1 km of the nearest MRT station.**

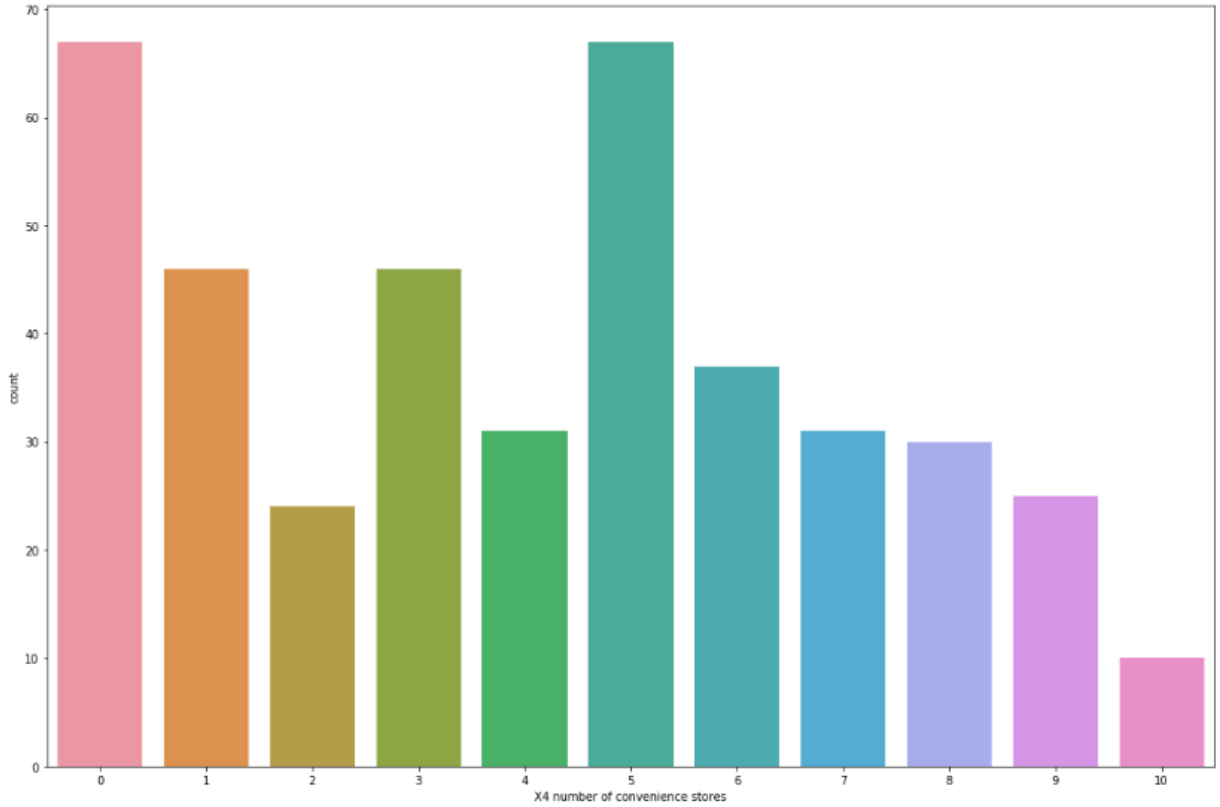
Bar graphs

Bar graphs are useful for displaying relationships between categorical data and at least one numerical variable. **seaborn.countplot** is a barplot where the dependent variable is the number of instances of each instance of the independent variable.



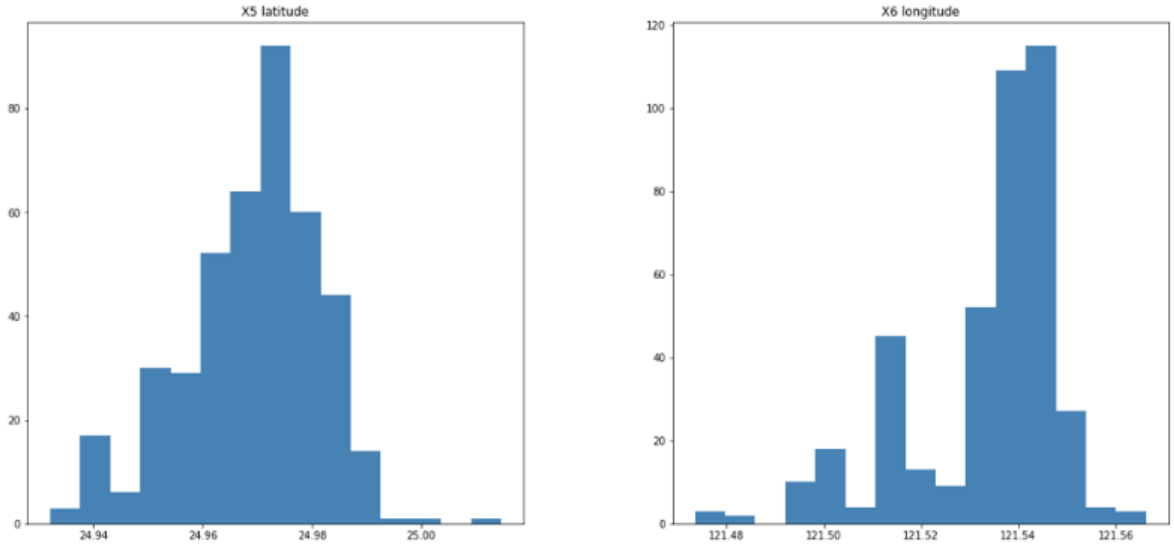
4.2 3 Graph of number of transaction had peak

- ***We observed that the number of transactions had a peak in May 2013, and was higher than the average on 3 other months: November 2012, January 2013 and June 2013.***



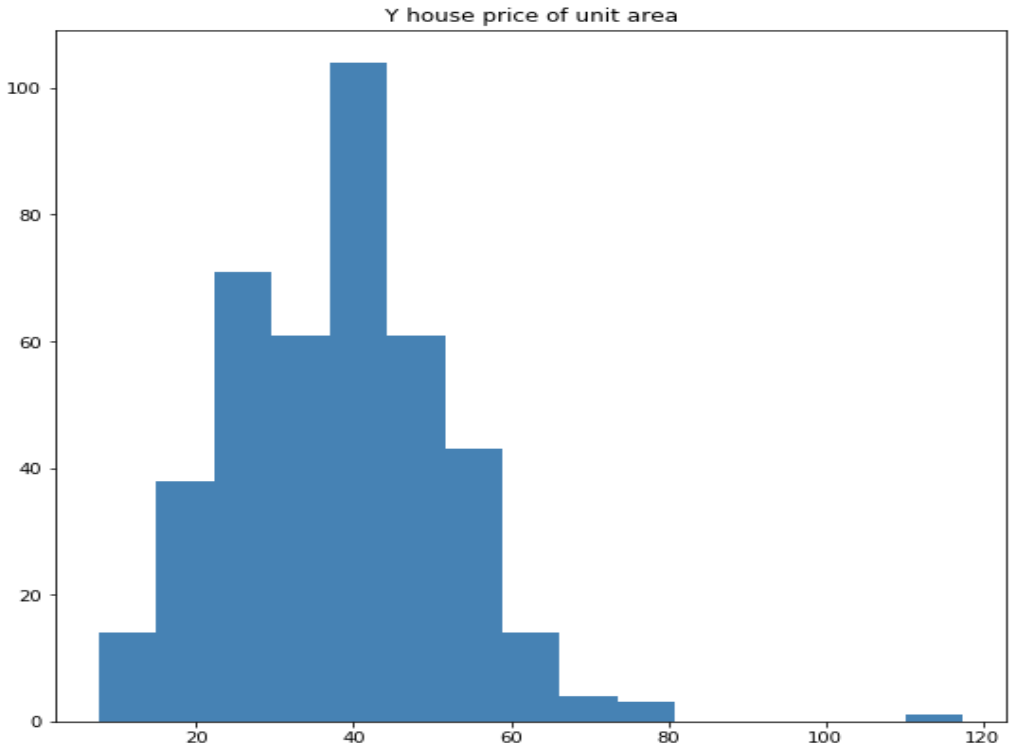
4.2 4 Bar graph of numberofconvenience stores

- **Approximately 1 house in 6 does not have any convenience store in the living circle on foot. These houses are probably located in residential areas.**
- **Half of the houses has between 1 and 6 convenience stores in the living circle on foot.**



4.2 5 visualise the distribution of latitude and longitude

- **Most of the houses are located in the East central part of the area within the extreme values of latitude and longitude of this dataset. It'll be more obvious when we visualize the geographical distribution on a map!**



4.2 6 visualise the distribution of Y house price of unit area

We observe the following on the distribution of house prices of unit area:

- **House price of unit are varies greatly from 76,000 New Taiwan Dollar/Ping to more than 1 million New Taiwan Dollar/Ping.**
- **More than half of house prices of unit area are between 250,000 and 500,000 New Taiwan Dollar/Ping.**
- **There are 8 houses for which the house price of unit area is more than 650,000 New Taiwan Dollar/Ping, with one being 1,115,000 New Taiwan Dollar/Ping (extreme value).**

```
Out[9]: 270    117.5
        220     78.3
        312     78.0
        166     73.6
        105     71.0
        16      70.1
        379     69.7
        389     67.7
        413     63.9
        361     63.3
        Name: Y house price of unit area, dtype: float64
```

4.2 7 sort house price of unit area

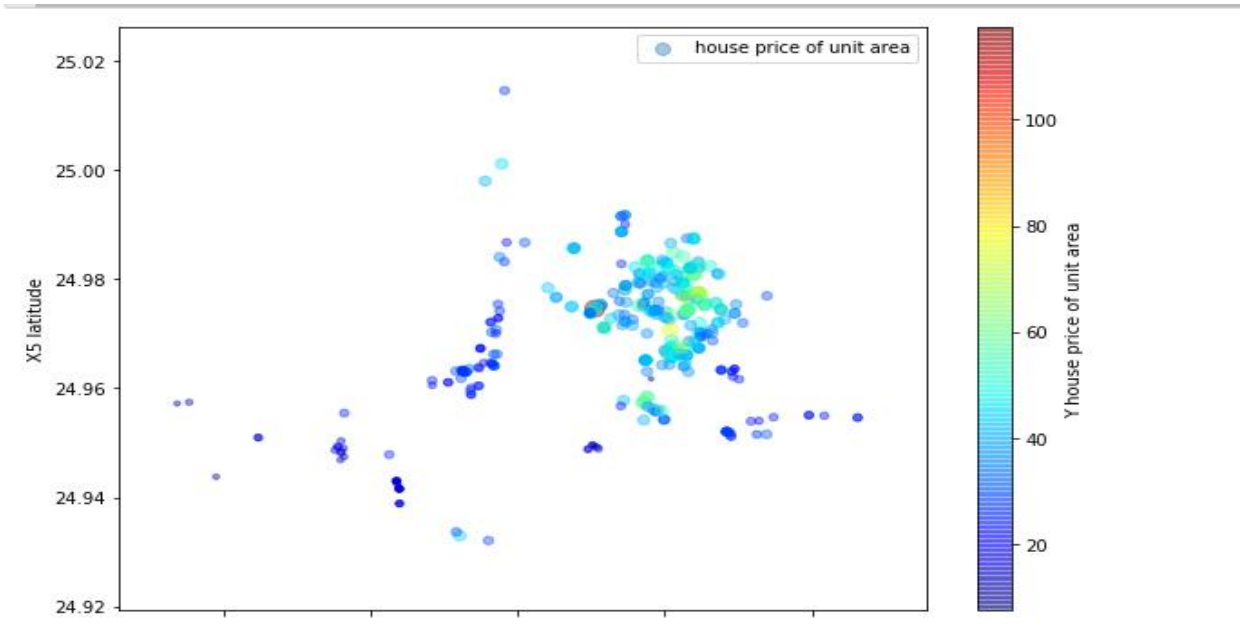
- **In our dataset, the mean house price of unit area is 379,802 TWD/ping and the median is 384,500 TWD/ping, which is a little higher than the figure reported for the whole New Taipei city. Highest prices in our dataset are more in line with the average home price in Taipei, even though they're located in New Taipei city.**
- **House price of unit area are not evenly distributed in the Xindian district, with lowest values observed in the Western part, and highest values observed in the North eastern part.**
- **Their house prices of unit area are not extreme values. However, we might exclude them from the analysis, as the combination of their features might be quite specific and different from those observed in Xindian district.**

4.3 Geographical distribution of the variables

When we mention geographical data it crosses to our mind the coordinates of a data point which are: Longitude and Latitude. This is true, they are just the X and Y coordinates for a specific point on the map.

Scatter Plots

Scatterplot displays the value of **2** sets of data on 2 dimensions. Each dot represents an observation. The position on the X (horizontal) and Y (vertical) axis represents the values of the 2 variables. It is really useful to study the relationship between both variables. It is common to provide even more information using colors or shapes (to show groups, or a third variable). It is also possible to map another variable to the size of each dot, what makes a bubble plot. If you have many dots and struggle with overplotting, consider using 2D density plot.



4.3 1 visualize the geographical distribution of the house prices of unit area: X5 latitude and X6 longitude

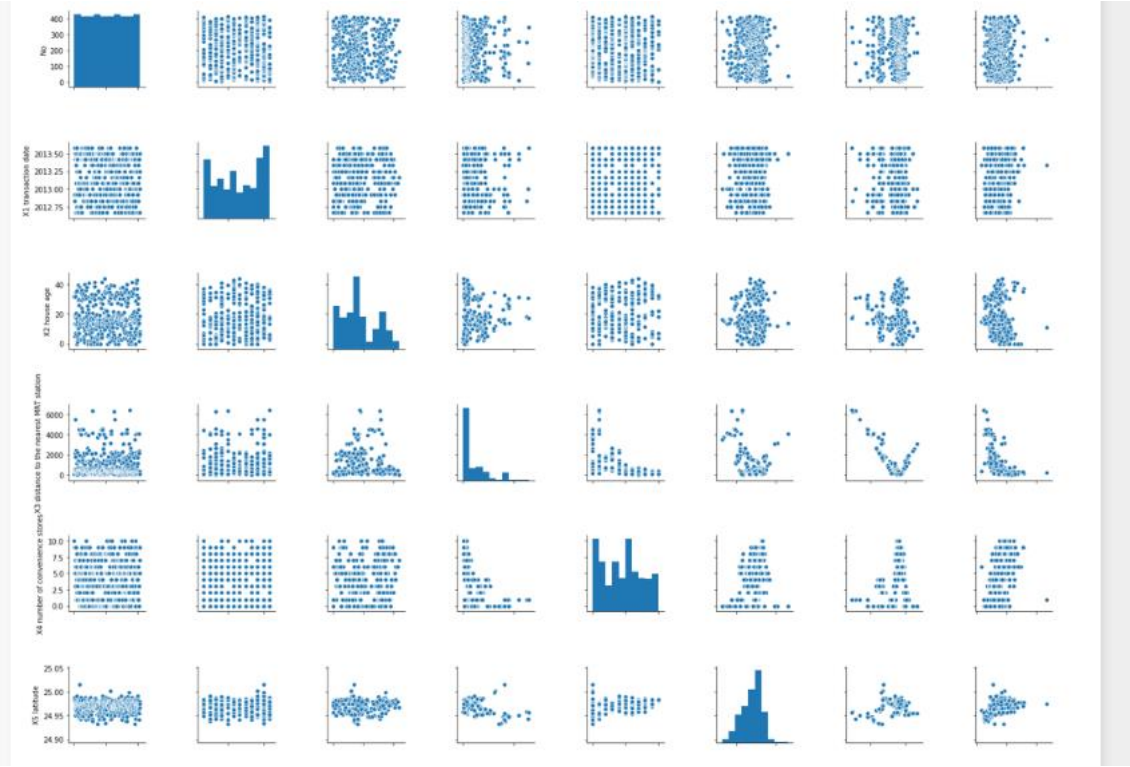
- House price of unit area are not evenly distributed in the Xindian district, with lowest values observed in the Western part, and highest values observed in the North eastern part.

4.4 Outliers

We would like to identify obvious outliers in our dataset. Based on the distribution of the Y house price of unit area, we've identify properties with very high prices that could be outliers.

By default, this function will create a grid of Axes such that each numeric variable in data will by shared in the y-axis across a single row and in the x-axis across a single column. The diagonal Axes are treated differently, drawing a plot to show the univariate distribution of the data for the variable in that column.

It is also possible to show a subset of variables or plot different variables on the rows and columns. This is a high-level interface for **PairGrid** that is intended to make it easy to draw a few common styles. You should use **PairGrid** directly if you need more flexibility.



4.4 1 outliners of dataset

- The property with the highest house price of unit (more than 1 million TWD/Ping) seems to be an outlier. We might exclude it from our analysis (if so, we'll mention it). When looking at Google Street view, we didn't notice anything special about this property: There might be an error in the value reported. Or the property is very tiny?

4.5 Correlation

It can be useful in data analysis and modeling to better understand the relationships between variables. The statistical relationship between two variables is referred to as their correlation.

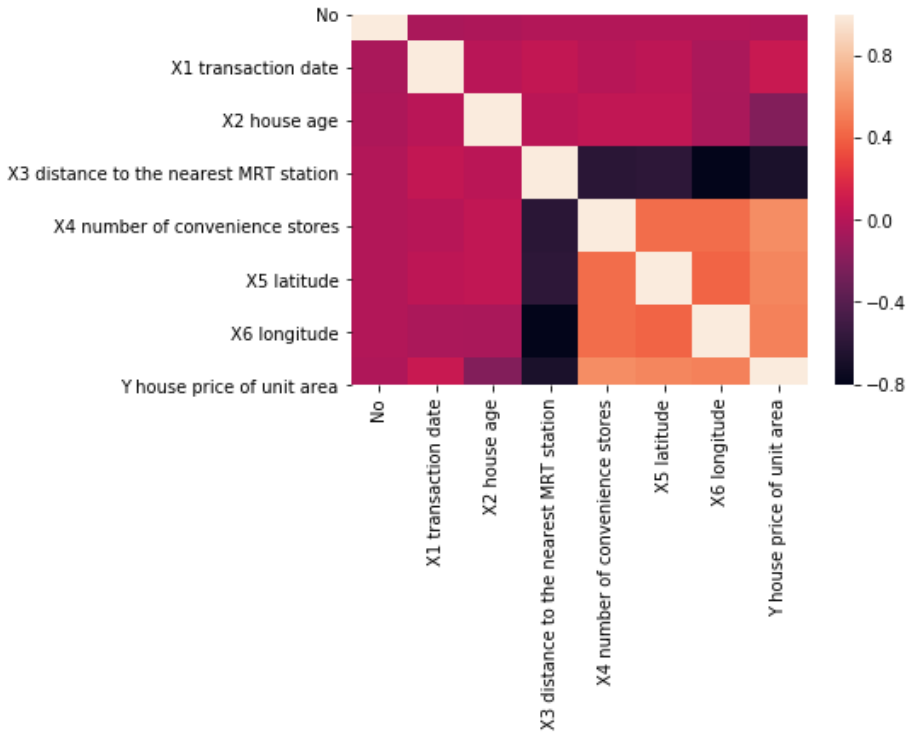
A correlation could be positive, meaning both variables move in the same direction, or negative, meaning that when one variable's value increases, the other variables' values decrease. Correlation can also be neutral or zero, meaning that the variables are unrelated.

- **Positive Correlation**: both variables change in the same direction.
- **Neutral Correlation:** No relationship in the change of the variables.
- **Negative Correlation:** variables change in opposite directions.

The performance of some algorithms can deteriorate if two or more variables are tightly related, called multicollinearity. An example is linear regression, where one of the offending correlated variables should be removed in order to improve the skill of the model.

We may also be interested in the correlation between inputs variables with the output variable in order provide insight into which variables may or may not be relevant as input for developing a model.

The structure of the relationship may be known, e.g. it may be linear, or we may have no idea whether a relationship exists between two variables or what structure it may take. Depending what is known about the relationship and the distribution of the variables, different correlation scores can be calculated.



4.5 1 correlations of dataset

X3 distance to the nearest MRT station seems to be quite correlated to:

- **X4 number of convenience stores (-0.60)**
- **X5 latitude (-0.60)**
- **X6 longitude (-0.80)**
- **Y house price of unit area (-0.67)**

CHAPTER 5

SPOT-CHECKING ALGORITHM

5.1 Spot checking algorithm

Spot-checking algorithms are associated with a quick evaluation of a number of different algorithms related to our machine-learning problem, so that we know which algorithms to focus on and what to get rid of.

Spot-checking algorithms is developed for applied machine learning. When we want to solve a new problem, first we have to know which type or class of algorithms is good at picking for our problem and which are not. Spot-checking algorithms is designed to quickly provide a first set of results on a new predictive modeling problem.

5.2 Why spot check algorithm is necessary?

Unlike grid searching and other types of algorithm settings that seek the optimal algorithm or the optimal configuration, spot checking algorithm is designed to quickly evaluate a variety of algorithms and obtain an approximate initial result. This first cut result may be used to get an idea of the fact that a problem or its representation is indeed predictable and, in this case, the types of algorithms that could merit further study of the problem.

Spot-checking is an approach to help overcome the “hard problem” of applied machine learning and encourage you to clearly think about the higher-order search problem being performed in any machine-learning project.

5.3 Benefits of Spot-Checking Algorithms

There are 3 key benefits of spot-checking algorithms on your machine learning problems

5.3.1 Speed: we can spend a lot of time playing with different algorithms, setting parameters and thinking about algorithms that will work well with your problem. A single spot-check experience can save hours, days and even weeks in advance.

5.3.2 Purpose: There is a tendency to go with those who have worked with you before. We choose our preferred algorithm (or algorithms) and apply it to all the problems we see. The power of machine learning is that there are many different ways to deal with a particular problem. The sample control experience allows you to automatically and objectively explore these algorithms to determine the nature of the problem so that you can focus your attention.so, allowing you to discover what might work well for a problem rather than going with what you used last time.

5.3.3 Results: spot check algorithms can quickly provide you with current results. You can find a reasonably good solution in the first field experience. Or, you can quickly discover that your dataset does not provide enough structure for a general algorithm to work. Spot checks will give you the results you need, and you can decide to go ahead and optimize a specific model or track the performance of the problem

Finally, the result of the spot checking check is the starting point. It suggest where to focus on the problem, not the optimal algorithm. This process is designed to break away from typical thought and analysis and instead focus on results.

5.4 Spot-Checking Framework in Python.

There are four parts to the framework that we applied to develop our code [figure 5.3]

- Define Models
- Evaluate Models
- Load Dataset
- Summarize Results

5.5 Define Models

The models defined will be specific to the type predictive modeling problem, e.g. classification or regression.

We defined the functions to develop a reusable framework to spot check algorithms

We have called this function *get_models()*. It will return a dictionary of model names mapped to scikit-learn model object. The function will also take a dictionary as an optional argument; if not provided, a new dictionary is created and populated. If a dictionary is provided, models are added to it. This is to add flexibility if you would like to have multiple functions for defining models, or add a large number of models of a specific type with different configurations. [Figure 5.5]

5.6 Evaluate Models

The scikit-learn library provides the ability to pipeline models during evaluation. This allows the data to be transformed prior to being used to fit a model, and this is done in a correct way such that the transforms are prepared on the training data and applied to the test data.

We can define a function that prepares a given model prior to evaluation to allow specific transforms to be used during the spot checking process. They will be performed in a blanket way to all models. This can be useful to perform operations such as standardization, normalization, and feature selection.

We will define a function named *make_pipeline()* that takes a defined model and returns a pipeline. This function can be expanded to add other transforms, or simplified to return the provided model with no transforms. [Figure 5.6.1]

Now we need to evaluate a prepared model.We will use a standard of evaluating models using k-fold cross-validation. The evaluation of each defined model will result in a list of results We will define a function named *evaluate_model()* that will take the data, a defined model, a number of folds, and a performance metric used to evaluate the results. It will return the list of scores.

The function calls *make_pipeline()* for the defined model to prepare any data transforms required, then calls the cross_val_score() scikit-learn function. Importantly, the *n_jobs* argument is set to -1 to allow the model evaluations to occur in parallel, harnessing as many cores as you have available on your hardware. [Figure 5.6.2]

We do not care about exceptions or warnings when spot checking. We only want to know what does work and what works well. Therefore, we can trap exceptions and ignore all warnings when

evaluating each model. The function named *robust_evaluate_model()* implements this behavior. The *evaluate_model()* is called in a way that traps exceptions and ignores warnings. If an exception occurs and no result was possible for a given model, a *none* result is returned. [Figure 5.6.3]

We will define a function named *evaluate_models()* that takes the dictionary of models as an argument and returns a dictionary of model names to lists of results. The number of folds in the cross-validation process can be specified by an optional argument that defaults to 10. The metric calculated on the predictions from the model can also be specified by an optional argument and defaults to classification accuracy.

We provide some verbose output, summarizing the mean and standard deviation of each model after it was evaluated. This is helpful if the spot checking process on your dataset takes minutes to hours. [Figure 5.6.4]

5.7 Results from Spot Checking Test

We only want to know what algorithms performed well. So finally, we can evaluate the results. Two useful ways to summarize the results are:

Line summaries of the mean and standard deviation of the top 10 performing algorithms.

Box and whisker plots of the top 10 performing algorithms.

The line summaries are quick and precise, although assume a well behaving Gaussian distribution, which may not be reasonable.

The box and whisker plots assume no distribution and provide a visual way to directly compare the distribution of scores across models in terms of median performance and spread of scores.

We will define a function named *summarize_results()* that takes the dictionary of results, prints the summary of results, and creates a boxplot image that is saved to file. The function takes an argument to specify if the evaluation score is maximizing, which by default is *True*. The number of results to summarize can also be provided as an optional parameter, which defaults to 10.

The function first orders the scores before printing the summary and creating the box and whisker plot. [Figure 5.7]

Rank	Algorithm	Metric Cnmse	Standard Deviation
1	Random Forest Regressor	Score=-59.515	(+/- 38.819)
2	Linear Regression	Score=-79.008	(+/- 43.856)
3	Decision Tree Regressor	Score=-92.809	(+/- 31.322)
4	SVR	Score=-103.466	(+/- 47.211)

Table 1 Neg mean squared error

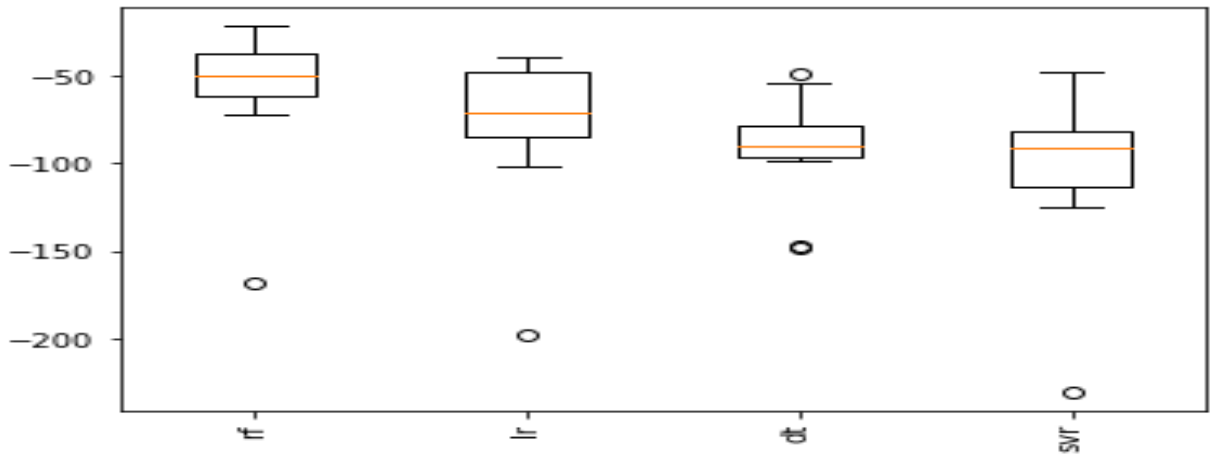


Figure 1 neg_mean_squared_error'

Rank	Algorithm	Metric Cnmse	Standard Deviation
1	Random Forest Regressor	Score=0.693	(+/- 0.124)
2	Linear Regression	Score=0.583	(+/- 0.133)
3	Decision Tree Regressor	Score=0.496	(+/- 0.161)
4	SVR	Score=0.448	(+/- 0.124)

Table 2 R2 Score

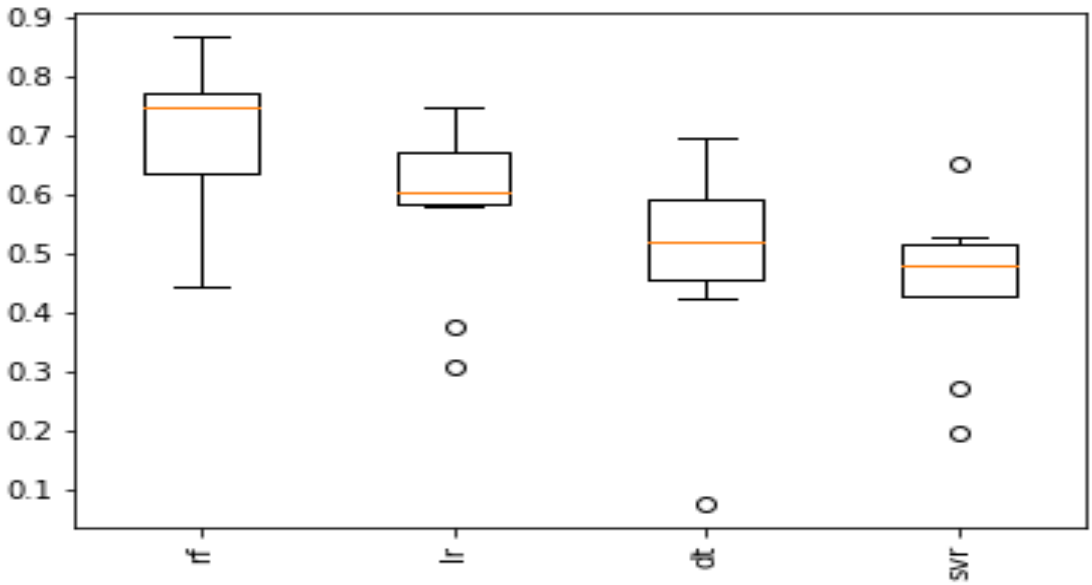


Figure 2 R2 Score

Finally we can see the random forest algorithm has the best performance, based on the mean square error and R2 metrics. We'll then try to fine tune this model to further improve its performance.

CHAPTER 6

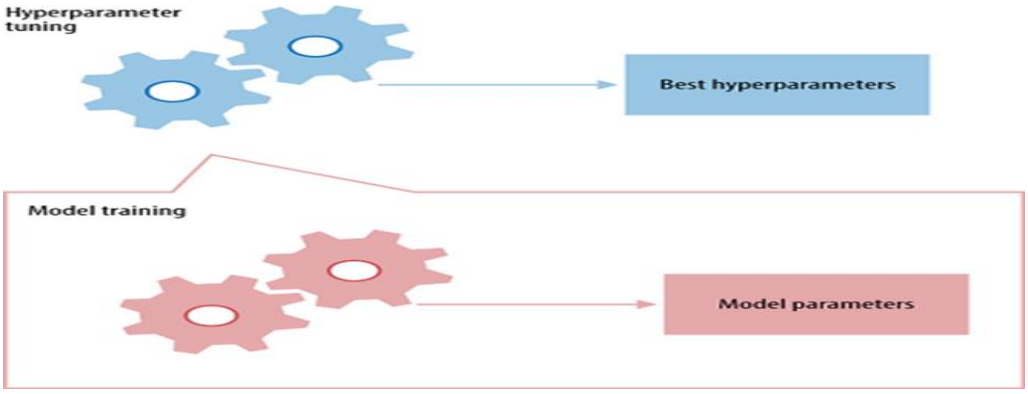
FINE TUNING THE MODELS

6.1 Concept Related to the Hyper parameter Tuning

So we've built a random forest model to solve our machine learning problem but we're not too impressed by the results. However, we're not very dazzled by the outcomes. . Assembling more information and highlight building for the most part has the best result as far as time contributed versus improved execution, yet when we have depleted all information sources; it is an ideal opportunity to proceed onward to show hyper parameter tuning. This post will focus on optimizing the random forest model in Python using Scikit-Learn tools.

6.1.1 Hyperparameter Tuning

Optimizing hyperparameters for machine learning is a key advance in making precise expectations. Hyperparameters characterize attributes of the model that can affect model exactness and computational productivity. They are ordinarily set preceding fitting the model to the information. In contrast, Parameters are estimates during the training process that allow the model to adapt to the data. Hyperparameters are often improved through trial and error. Multiple models suitable for different parameter values and compare their performance. Hyperparameters are regularly upgraded through experimentation; different models are fit with an assortment of hyperparameter values, and their exhibition is analyzed.



6.1. 1 Hyperparameter Tuning

6.1.2 Why we use Hyperparameter tuning?

The most ideal approach to consider hyperparameters resembles the settings of a calculation that can be changed in accordance with advance execution, similarly as we may turn the handles of an AM radio to get a reasonable sign

Hyperparameter tuning depends more on test results than hypothesis, and consequently the best technique to decide the ideal settings is to attempt a wide range of mixes assess the exhibition of each model. Be that as it may, assessing each model just on the preparation set can prompt one of the most principal issues in machine learning: overfitting.

Scikit-Learn actualizes a lot of reasonable default hyperparameters for all models, yet these are not destined to be ideal for an issue. The best hyperparameters are generally difficult to decide early, and tuning a model is the place AI abandons a science into experimentation based designing.

6.1.3 Most important hyperparameters of Random Forest

n_estimators = n of trees

max_features = max number of features considered for splitting a node

max_depth = max number of levels in each decision tree

min_samples_split = min number of data points placed in a node before the node is split

min_samples_leaf = min number of data points allowed in a leaf node

Bootstrap = method for sampling data points (with or without replacement)

Hyperparameter	Meaning	Default or recommended values
n_estimators	Same as num.trees	10 (lot of difference!)
max_features	Same as mtry in ranger of R	auto
criterion	Same as importance in ranger of R	Gini
Max_depth	Same as max.depth in ranger of R.	Default is no limit and 5 is recommended
min_samples_leaf	Same as min.node.size . Leaf node is same as terminal node.	
bootstrap	Sampling with replacement (bootstrapping) or use whole training data	True (random sampling with replacement takes place)
No parameter for Sampsize	The sub-sample size is always the same as the original input sample size	
Class_weight	If class imbalance needs to be accounted. Use "none" unless you want to deal with class imbalance if there, at this stage. But recommended would be to use a separate technique like SMOTE before running the model.	"None"
oob_score	Whether to use out-of-bag samples to estimate the generalization accuracy. Another meta-parameter.	False
More leaf parameters: max_leaf_nodes, min_weight_fraction_leaf Other split parameters: (min_samples_split, min_impurity_decrease, min_impurity_split)	Would put them as meta-parameters and would not tweak them because they are to limit your trees further, which is not essential. Unless, you have restrictions to apply on trees and how trees are built, need not use them.	

If we optimize the model for the training data, then our model will score very well on the training set, but will not be able to generalize to new data, such as in a test set. When a model performs highly on the training set but poorly on the test set, this is known as overfitting, or essentially creating a model that knows the training set very well but cannot be applied to new problems. It's

like a student who has memorized the simple problems in the textbook but has no idea how to apply concepts in the messy real world.

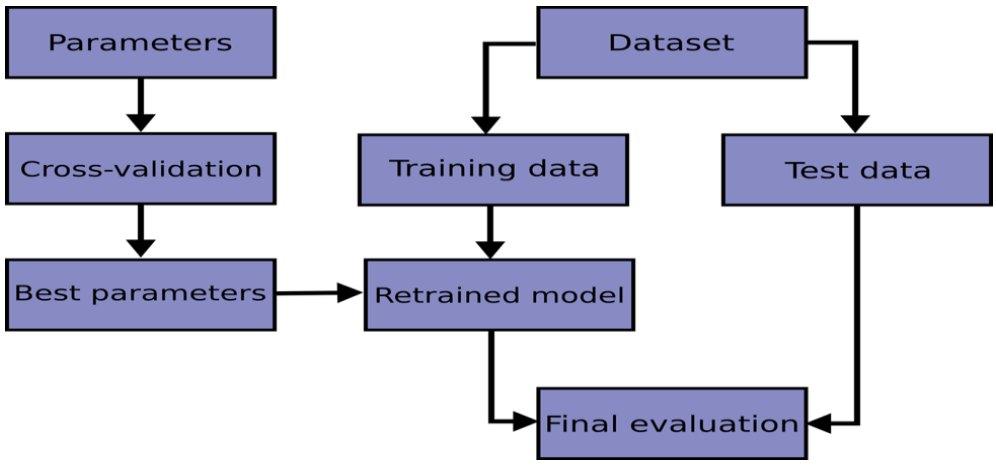
An over fit model may look impressive on the training set, but will be useless in a real application. Therefore, the standard procedure for hyperparameter optimization accounts for overfitting through cross validation.

6.1.4 What is Cross Validation?

Cross Validation is a technique which involves reserving a particular sample of a dataset on which you do not train the model. Later, you test your model on this sample before finalizing it.

Here are the steps involved in cross validation:

1. You *reserve* a sample data set
2. Train the model using the remaining part of the dataset
3. Use the reserve sample of the test (validation) set. This will help you in gauging the effectiveness of your model's performance. If your model delivers a positive result on validation data, go ahead with the current model. It rocks!



6.1.4 Cross Validation Chart

6.1.5 Why We Use k-fold Cross Validation

1. We should train the model on a large portion of the dataset. Otherwise we'll fail to read and recognize the underlying trend in the data. This will eventually result in a higher bias
2. We also need a good ratio of testing data points. As we have seen above, less amount of data points can lead to a variance error while testing the effectiveness of the model
3. We should iterate on the training and testing process multiple times. We should change the train and test dataset distribution. This helps in validating the model effectiveness properly

6.1.6 Working Process Of k-fold Cross Validation

1. Randomly split your entire dataset into k"folds"
2. For each k-fold in your dataset, build your model on k – 1 folds of the dataset. Then, test the model to check the effectiveness for *kth* fold
3. Record the error you see on each of the predictions
4. Repeat this until each of the k-folds has served as the test set
5. The average of your k recorded errors is called the cross-validation error and will serve as your performance metric for the model

6.1.7 Regression Metrics

In this section will review 3 of the most common metrics for evaluating predictions on regression machine learning problems:

1. **Mean Absolute Error.**
2. **Mean Squared Error.**
3. **R^2.**

6.1.7.1. Mean Absolute Error

The Mean Absolute Error (or MAE) is the average of the absolute differences between predictions and actual values. It gives an idea of how wrong the predictions were.

The measure gives an idea of the magnitude of the error, but no idea of the direction (e.g. over or under predicting).

6.1.7.2. Mean Squared Error

The Mean Squared Error (or MSE) is much like the mean absolute error in that it provides a gross idea of the magnitude of error.

Taking the square root of the mean squared error converts the units back to the original units of the output variable and can be meaningful for description and presentation. This is called the Root Mean Squared Error (or RMSE).

6.1.7.3. R^2 Metric

The R^2 (or R Squared) metric provides an indication of the goodness of fit of a set of predictions to the actual values. In statistical literature, this measure is called the coefficient of determination. This is a value between 0 and 1 for no-fit and perfect fit respectively

6.1.8 Search Technique

In random forests, there are a number of hyperparameters available. Although we can get good results without any changes to these parameters, there are some parameters which have great impact on the output of our classifier or regressor. But we do not want to manually search and test for the optimal values of these hyperparameters. Such a trial and error method may take a lot of time.

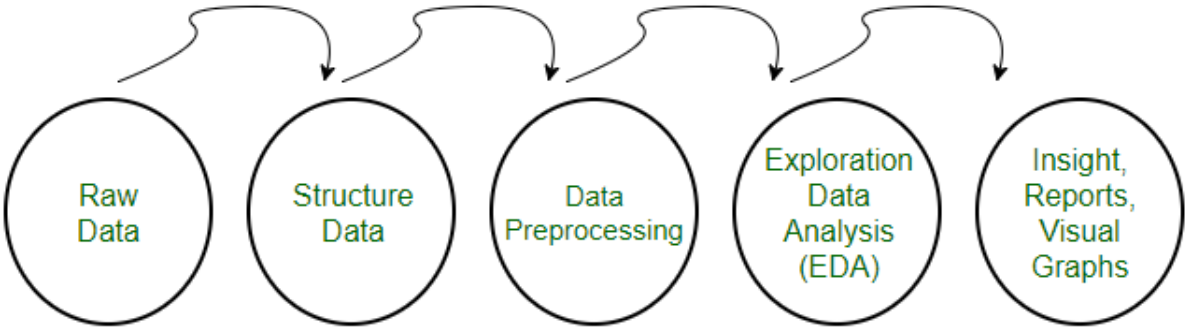
Therefore, we have methods like **RandomizedSearchCV** and **GridSearchCV** which help us to fine-tune the hyperparameters by providing us with the best values. On top of that, we can implement these using Scikit-Learn as well.

6.2 Methodology

6.2.1 Data Preprocessing

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. For achieving better results from the applied model in Machine Learning projects the format of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format, for example, Random Forest algorithm does not support null values, therefore to execute random forest algorithm null values have to be managed from the original raw data set.

Another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in one data set, and best out of them is chosen.



6.2.1 Data Preprocessing

6.2.1.1 Train Test Split

Goal in machine learning is to build a model that generalizes well to the new data. Hence the dataset is split into the Train dataset and the Test dataset during supervised machine learning experiment. Test dataset serves as a proxy for new data. Evaluation of a trained machine learning model and optimization of the hyperparameters is performed using k-fold cross validation on Train dataset only. Test dataset (also known as hold-out set) is not used in training of models and hence can be used under predict_model function to evaluate metrics and determine if the model has over-fitted the data.

6.2.1.2 Standardization

Standardization is a transformation that centers the data by removing the mean value of each feature and then scale it by dividing (non-constant) features by their standard deviation. After standardizing data the mean will be zero and the standard deviation one.

Standardization can drastically improve the performance of models. For instance, many elements used in the objective function of a learning assume that all features are centered on zero and have variance in the same order. If a feature has a variance that is orders of magnitude larger than others, it might dominate the objective function and make the estimator unable to learn from other features correctly as expected.

Depending on your needs and data, sklearn provides a bunch of scalers: StandardScaler, MinMaxScaler, MaxAbsScaler and RobustScaler.

6.2.1.3 Normalization

Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to rescale the values of numeric columns in the dataset without distorting differences in the ranges of values or losing information. This can be achieved using *normalize* parameter within setup. There are several methods available for normalization, by default it uses 'zscore' to normalize the data, which can be changed using *normalize_method* parameter within setup.

6.2.2 Baseline Algorithm

Before applying the Randomized Search for our data, first, we can check how a baseline algorithm performs without any parameter tuning. This will give us a good idea of whether our model is performing better or worse with parameter tuning

6.2.2.1 Parameters of Baseline Model:

bootstrap	True
criterion	mse
max_depth	None
max_features	auto
max_leaf_nodes	None
min_impurity_decrease	0.0
min_impurity_split	None
min_samples_leaf	1
min_samples_split	2
min_weight_fraction_leaf	0.0
n_estimators	10
oob_score	False

random_state	None
n_jobs	None

The Parameters are used in Baseline Model are below:

6.2.2 1 Parameters of Baseline Model

6.2.2.2 Result for Baseline Model

	tanning	Test
MAE	-5.030 (+/-0.939)	5.207
MSE	-64.689 (+/-59.342)	64.513
R^2	0.655 (+/-0.161)	0.621

6.2.2 2 Result for Baseline Model

6.2.3 Random Search Cross Validation in Scikit-Learn

Usually, we have a vague idea of the best hyperparameters and for that reason the satisfactory method to narrow our search is to evaluate a wide range of values for each hyperparameter. Using Scikit-Learn's RandomizedSearchCV method, we will outline a grid of hyperparameter ranges, and randomly sample from the grid, acting K-Fold CV with every mixture of values.

6.2.3 .1 The Random Grid

Random Search sets up a grid of hyperparameter values and chooses arbitrary blends to prepare the model and score. This allows you to explicitly control the number of parameter combinations

that are attempted. The quantity of search emphases is set dependent on schedule or assets. Scikit Learn offers the RandomizedSearchCV work for this procedure.

bootstrap	True, False
max_depth	10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None
max_features	auto, sqrt
min_samples_leaf	1,2,4
min_samples_split	2, 5, 10
n_estimators	200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000

6.2.3.1 The random grid

6.2.3.2 Best parameters from Random search:

bootstrap	True
max_depth	110
max_features	Sqrt
min_samples_leaf	1
min_samples_split	2,
n_estimators	1000

6.2.3 1 Best parameters from Random search

6.2.3.3 Result for Best Model from Random Search:

	tanning	Test
MAE	-5.280 (+/-1.113)	4.563
MSE	-65.284 (+/-55.941)	51.341
R^2	0.704 (+/-0.137)	0.699

6.2.3.3 Result for Best Model from Random Search

6.2.4 Grid Search with Cross Validation

Random search allowed us to narrow down the variety for every hyperparameter. Now that we realize where to pay attention our search, we are able to explicitly specify every aggregate of settings to try. We do that with GridSearchCV, a technique that, in preference to sampling randomly from a distribution, evaluates all combinations we define.To execute the Grid Search calculation we have to import GridSearchCV class from the sklearn.model_selection library.

The initial step you have to perform is to make a dictionary reference of the considerable number of parameters and their relating set of qualities that you need to test for best execution. The name of the dictionary reference things compares to the parameter

The Grid Search algorithm can be very slow, owing to the potentially huge number of combinations to test. Furthermore, cross validation further increases the execution time and complexity

6.2.3.1 Grid Search

Grid Search can be thought of as a comprehensive quest for choosing a model. In Grid Search, the data scientist sets up a grid of hyperparameter values and for every blend, prepares a model and scores on the testing information. In this methodology, each blend of hyperparameter values is attempted which can be exceptionally wasteful. For instance, scanning 20 distinctive parameter esteems for every one of 4 parameters will require 160,000 preliminaries of cross- validation. This Scikit Learn offers the GridSearchCV capacity to disentangle the procedure, it would be an incredibly exorbitant execution both in processing force and time.

6.2.4.2 Best parameters from grid search:

bootstrap	True
max_depth	120
max_features	sqrt
min_samples_leaf	1,
min_samples_split	3
n_estimators	900

6.2.4.2 Best parameters from grid search

6.2.4.3 Result for Best Model from grid search:

	tanning	Test
MAE	-4.994 (+/-1.043)	4.587
MSE	-65.844 (+/-61.745)	51.736
R^2	0.691 (+/-0.107)	0.696

6.2.4.3 Result for Best Model from grid search

CHAPTER 7

CONCLUSION & FUTURE WORKS

We have defined several models with various features and various model complexities. There is a need to use a mix of these models a linear model gives a high bias (under fit) whereas a high model complexity-based model gives a high variance (overfit). Data Scientist tends to overfit their models which can be reduced by ridge regression and LASSO The study reveals that economic factors influence land price more than the social factors. The interaction of the selected factors (X) on land price (Y) is analyzed. It is found that four factors viz. GLV (84%), silver price per gram (92%), population (86%) and cost of crude oil (88%) have more positive effect on land price. The outcome of this study can be used in annual revision of guideline value of land which may add more revenue to the State Government while land transaction is made. This study will support the policy makers to relook the movement of the identified factors to have control on rise in the land price and stabilize it. Since there is a greater need for good long-term data analysis about land price, general land market behavior and spatial development, the results produced in this research may be of great use for Government and non-Government agencies which involve in land administration.

The objective of this research was to empirically compare the predictive power of the OLS hedonic model with a random forest model for predicting apartment prices in Ljubljana, for the period between 2008 and 2013. Before OLS modelling was performed, the initial set of 36 predicting variables was transformed into a Principal Components Analysis feature space in order to avoid immanent multicollinearity between variables. The 10 extracted PCAs were analyzed by component loadings and clustering of initial explanatory variables in the component space in order to obtain an interpretation and semantic description in the original feature space.

Analogous to the OLS model, the random forest (RF) and out of the bag (OOB) permuting error estimate was adopted to select 10 of the most important predicting variables that were used for RF modelling. We discovered a relatively high rate of equivalent semantic relationships (approximately 70%) between the set of interpretations of the top ten PCAs with the set of top ten ranked predictors selected by RF. The commonly applied adjustment of prices over time for the

sales data was purposely skipped in order to examine the sensitivity of the predictive models to the influence of time variability. Hence, the time period with the greatest change of prices in Slovenia, the six consecutive years between 2008 and 2013 that showed a 28% decline, was chosen for this research. The OLS model did not account for the time specific variable, “date of transaction”, which represents the basic information for adjustment of prices over time. However, the “date of transaction” variable was considered strongly by RF and was determined to be the third most influential variable and is effectively used for time adjustment. All performance measure—R2 values (0.23 for OLS and 0.57 for RF), the sales ratios (1.04 for OLS and 1.02 for RF), the MAPE (17% for OLS and 7% for RF) and the COD (17% for OLS and 7% for RF)—revealed significantly better results with random forest. The low R2 value for the OLS model indicated that non-linear modelling was required.

Visual inspection of the differences between the sales ratios (SR) of the OLS and RF predictions

showed that the models perform similarly at identical locations. Both methods underestimated the higher prices of apartments (SR < 1) and overestimated the lower prices of apartments (SR > 1). However, we found that the RF predictions were closer to actual prices than the OLS predictions by combining results of kernel density for the differences of average of sales ratios between OLS and RF (SR (OLS)-SR (RF)) for the apartments in the buildings and the results of the Hot Spot Analysis. In addition, negative values of differences between the average SR (sales ratios of RF are larger than OLS sales ratios) were located at the spots where elite groups of condominium buildings are raised. Apartments in these specific locations would be under-valued using OLS predictions. In contrast, RF captures their differences due to amenities attributed to them.

Finally, the entire analysis of the spatial distribution of sales ratios for both methods has revealed that the random forest algorithm could provide better detection of the variability in apartment values and predicts them more effectively than multiple regression in complex urban forms like the city.

The Enriched RF functions admirably with numeric highlights, as it can infer rules not just contingent upon the estimation of a quality yet additionally on its essence or nonattendance. In any case, it can't deal with crude picture or content information, then again, can speak to different sorts

of information through the installing layers. Be that as it may, it didn't deal with numeric properties just as the irregular woodland, and needed to depend on more information types to edge it out. Joining the two strategies yielded the best outcome, joining the various qualities of each approach.

As future work, include determination calculations can likewise be utilized to use the preparing speed for the models. Another method which we intend to apply on this dataset is feeble managed learning with pseudo-naming to expand the quantity of preparing information occurrences for profound learning.

Appendix

FIGURE 5.3

```
In [1]: import pandas as pd
        import warnings
        from numpy import mean
        from numpy import std
        from matplotlib import pyplot

        from sklearn.preprocessing import StandardScaler
        from sklearn.preprocessing import MinMaxScaler
        from sklearn.pipeline import Pipeline

        # insert libraries for the required regression algorithms
        from sklearn.linear_model import LinearRegression
        from sklearn.svm import SVR
        from sklearn.tree import DecisionTreeRegressor
        from sklearn.ensemble import RandomForestRegressor

        from sklearn.model_selection import cross_val_score
```

FIGURE 5.5

```
In [2]: # create a dict of standard models to evaluate {name:object}
        def get_models(models=dict()):
            # linear models
            models['lr'] = LinearRegression()
            models['svr'] = SVR(kernel = 'rbf')
            models['dt'] = DecisionTreeRegressor()
            models['rf'] = RandomForestRegressor()
            return models
```

FIGURE 5.6.1

```
In [3]:
# create a feature preparation pipeline for a model
def make_pipeline(model):
    steps = list()
    # standardization
    steps.append(('standardize', StandardScaler()))
    # normalization
    steps.append(('normalize', MinMaxScaler()))
    # the model
    steps.append(('model', model))
    # create pipeline
    pipeline = Pipeline(steps=steps)
    return pipeline
```

FIGURE 5.6.2

```
In [4]:
# evaluate a single model
def evaluate_model(X, y, model, folds, metric):
    # create the pipeline
    pipeline = make_pipeline(model)
    # evaluate model
    scores = cross_val_score(pipeline, X, y, scoring=metric, cv=folds, n_jobs=-1)
    return scores
```

FIGURE 5.6.3

```
In [5]:
# evaluate a model and try to trap errors and and hide warnings
def robust_evaluate_model(X, y, model, folds, metric):
    scores = None
    try:
        with warnings.catch_warnings():
            warnings.filterwarnings("ignore")
            scores = evaluate_model(X, y, model, folds, metric)
    except:
        scores = None
    return scores
```

FIGURE 5.6.4

```
In [6]:
# evaluate a dict of models {name:object}, returns {name:score}
def evaluate_models(X, y, models, folds=10, metric='accuracy'):
    results = dict()
    for name, model in models.items():
        # evaluate the model
        scores = robust_evaluate_model(X, y, model, folds, metric)
        # show process
        if scores is not None:
            # store a result
            results[name] = scores
            mean_score, std_score = mean(scores), std(scores)
            print('>%s: %.3f (+/-%.3f)' % (name, mean_score, std_score))
        else:
            print('>%s: error' % name)
    return results
```

FIGURE 5.7

In [7]:

```
# print and plot the top n results
def summarize_results(results, maximize=True, top_n=10):
    # check for no results
    if len(results) == 0:
        print('no results')
        return
    # determine how many results to summarize
    n = min(top_n, len(results))
    # create a list of (name, mean(scores)) tuples
    mean_scores = [(k,mean(v)) for k,v in results.items()]
    # sort tuples by mean score
    mean_scores = sorted(mean_scores, key=lambda x: x[1])
    # reverse for descending order (e.g. for metric)
    if maximize:
        mean_scores = list(reversed(mean_scores))
    # retrieve the top n for summarization
    names = [x[0] for x in mean_scores[:n]]
    scores = [results[x[0]] for x in mean_scores[:n]]
    # print the top n
    print()
    for i in range(n):
        name = names[i]
        mean_score, std_score = mean(results[name]), std(results[name])
        print('Rank=%d, Name=%s, Score=%.3f (+/- %.3f)' % (i+1, name, mean_score, std_score))

    # boxplot for the top n
    pyplot.boxplot(scores, labels=names)
    _, labels = pyplot.xticks()
    pyplot.setp(labels, rotation=90)
    pyplot.savefig('spotcheck.png')
```

FIGURE 6.1.8

In [19]:

```
# Define a function for model evaluation using cross validation
def evaluate_model_cross_validation(name, model, X_train, y_train, folds = 10):

    from sklearn.model_selection import cross_val_score

    # Cross Validation Regression MAE
    metric='neg_mean_absolute_error'
    scores = cross_val_score(regressor, X_train, y_train, scoring=metric, cv=folds, n_jobs=-1)
    mean_score, std_score = np.mean(scores), np.std(scores)
    print('>%s - training - MAE: %.3f (+/-%.3f)' % (name, mean_score, std_score))


    # Cross Validation Regression MSE
    metric='neg_mean_squared_error'
    scores = cross_val_score(regressor, X_train, y_train, scoring=metric, cv=folds, n_jobs=-1)
    mean_score, std_score = np.mean(scores), np.std(scores)
    print('>%s - training - MSE: %.3f (+/-%.3f)' % (name, mean_score, std_score))

    # Cross Validation Regression R^2
    metric='r2'
    scores = cross_val_score(regressor, X_train, y_train, scoring=metric, cv=folds, n_jobs=-1)
    mean_score, std_score = np.mean(scores), np.std(scores)
    print('>%s - training - R^2: %.3f (+/-%.3f)' % (name, mean_score, std_score))
```

Reference:

[1] M. Praveena, V. Jaiganesh, International Journal of Computer Applications (0975 – 8887) Volume 169 – No.8, July 2017.

[2] Sampathkumar et al. / Procedia Computer Science 57 (2015)112 – 121.

[3] Kilpatrick, J.A Factors Influencing CBD Land Prices. Journal of Real Estate; 2000, 25: 28-29.

[4] Wilson, I.D., Paris, S.D, Ware, J.A., & Jenkins, D.H. Residential Property Price Time Series Forecasting With Neural Networks.Journal of Knowledge-Based Systems; 2002, 15: 335-341.

[5] Mark, A.S., & John, W.B. Estimating Price Paths for Residential Real Estate. Journal of Real Estate Research; 2003: 25, 277–300.

[6] Wang, J., & Tian, P. Real Estate Price Indices Forecast by Using Wavelet Neural Network, Computer Simulation, 2005:2.

[7] Zhangming, H. Research on Forecasting Real Estate Price Index Based on Neural Networks. Journal of the Graduates Sun Yat Sen University, 2006;27.

[8] Tinghao,. Real Estate Price Index Based on ARMA Model, Statistics and Decision; 2007, 7.

[9] David, E.D., & Paavo, M. Urban Development and Land Markets in Chennai, India. International Real Estate Review; 2008, 11: 142165.

[10] Steven, P., & Albert, B.F. Neural Network Hedonic Pricing Models in Mass Real Estate Appraisal. Journal of Real Estate Research; 2009, 31: 147-164.

[11]Sampathkumar.V and Helen Santhi.M. Artificial Neural Network Modeling of Land Price at Sowcarpet in Chennai City, International Journal of Computer Science & Emerging Technologies; 2010, 1:44–49. Available at http:// download. excelingtech.co. uk/Journal/IJCSET%20V1%284%29.pdf

[12] Urmila, S. Module No. 3, Ports Logistics and Connectivity with Inland Container Depot, Institute of Rail Transport, Delhi; 2010, 61.

[13] Mansural Bhuiyan and Mohammad Al Hasan (2016) “Waiting to be Sold: Prediction of TimeDependent House Selling Probability” IEEE International Conference on Data Science and Advanced Analytics pp468-477

[14] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani, The elements of statistical learning, 1. Springer, 2009, vol. 2

[15] C. Cortes and V. Vapnik, “Support-vector networks,” Mach. Learn., vol. 20, no. 3, pp. 273–297, Sep. 1995, ISSN: 0885-6125. DOI: 10.1023/A:1022627411411.

[16] V.N.Vapnik,TheNatureofStatisticalLearningTheory. New York, NY, USA: Springer-Verlag New York, Inc., 1995, ISBN: 0-387-94559-8.

[17] A. J. Smola and B. Sch¨olkopf, “A tutorial on support vector regression,” Statistics and Computing, vol. 14, no. 3, pp. 199–222, Aug. 2004, ISSN: 0960-3174. DOI:

10.1023/B:STCO.0000035301.49549.88. [Online]. Available: http://dx.doi.org/10.1023/B:STCO.0000035301.49549.88.

[18] E. Fix and J. L. Hodges Jr, “Discriminatory analysisnonparametric discrimination: consistency properties,” DTIC Document, Tech. Rep., 1951.