STATISTICAL LEARNING PROJECT

SUPERVISED LEARNING-DECISION TRESS AND LINEAR REGRESSION

AMANPREET SINGH
942091

Contents

Chapter 1

Introduction

1.1 What is Decision Tree

A Decision Tree is a supervised learning predictive model that uses a set of binary rules to calculate a target value.It is used for either classification (categorical target variable) or regression (continuous target variable). Hence, it is also known as CART (Classification Regression Trees). Some real-life applications of Decision Trees include: Credit scoring models in which the criteria that cause an applicant to be rejected need to be clearly documented and free from bias Marketing studies of customer behavior such as satisfaction or churn, which will be shared with management or advertising agencies

1.2 Structure of a Decision Tree

Decision trees have three main parts:

- Root Node: The node that performs the first split. The most important Variable
- Terminal Nodes/Leaves :Nodes that predict the outcome. The next variable effecting decision
- Branches: arrows connecting nodes, showing the flow from question to answer.

The root node is the starting point of the tree, and both root and terminal nodes contain questions or criteria to be answered. Each node typically has two or more nodes extending from it. For example, if the question in the first node requires a "yes" or "no" answer, there will be one leaf node for a "yes" response, and another node for "no."

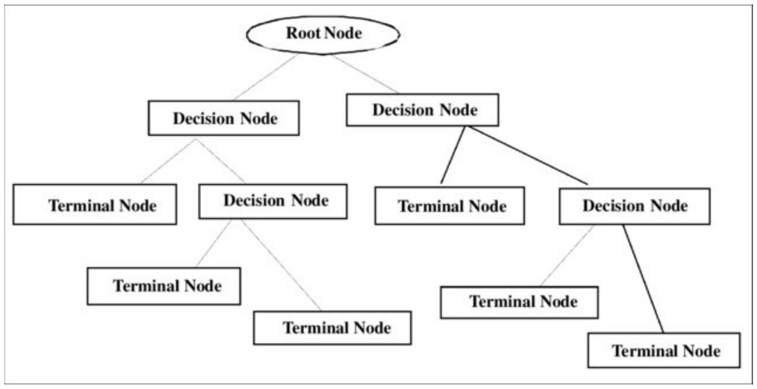


Figure 1.1: Decision Tree Nodes

Chapter 2

Introduction

2.1 The Algorithm behind Decision Trees.

The algorithm of the decision tree models works by repeatedly partitioning the data into multiple sub-spaces so that the outcomes in each final sub-space is as homogeneous as possible. This approach is technically called recursive partitioning. The produced result consists of a set of rules used for predicting the outcome variable, which can be either: a continuous variable, for regression trees a categorical variable, for classification trees The decision rules generated by the CART (Classification Regression Trees) predictive model are generally visualized as a binary tree.

CART tries to split this data into subsets so that each subset is as pure or homogeneous as possible. If a new observation fell into any of the subsets, it would now be decided by the majority of the observations in that particular subset.

2.2 Let us now see how a Decision Tree algorithm generates a TREE.

To Predict Red Or Grey The first split tests whether the variable x is less than 60. If yes, the model says to predict red, and if no, the model moves on to the next split. Then, the second split checks whether or not the variable y is less than 20. If no, the model says to predict gray, but if yes, the model moves on to the next split. The third split checks whether or not the variable x is less than 85. If yes, then the model says to predict red, and if no, the model says to predict grey.

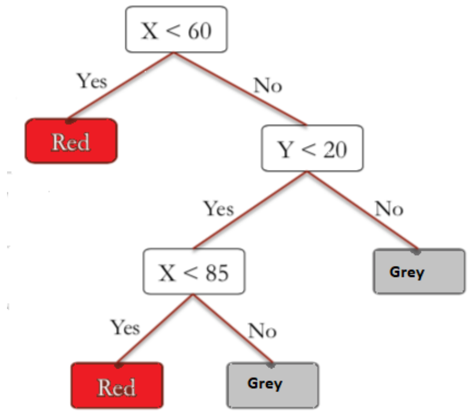


Figure 2.1: Decision Tree Nodes

Chapter 3

Introduction

3.1 Predictions from Decision Trees

In the above example,we discussed Classification trees i.e when the output is a factor/category:red or grey.Trees can also be used for regression where the output at each leaf of the tree is no longer a category, but a number. They are called Regression Trees.

3.2 Classification Trees:

With Classification Trees, we report the average outcome at each leaf of our tree. However, Instead of just taking the majority outcome to be the prediction, we can compute the percentage of data in a subset of each type of outcome.

3.3 Advantages of Decision Trees

- It is quite interpret-able and easy to understand.
- It can also be used to identify the most significant variables in data-set

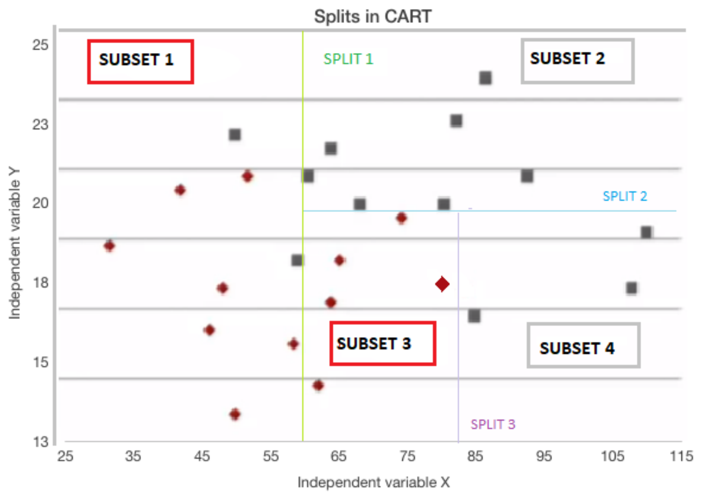


Figure 3.1: Splits in CART

Chapter 4

Introduction

4.1 Measures Used for Split

There are different ways to control how many splits are generated.

- Gini Index: It is the measure of inequality of distribution. It says if we select two items from a population at random then they must be of same class and probability for this is 1 if population is pure.It works with categorical target variable “Success” or “Failure”.It performs only Binary splits,Lower the value of Gini, higher the homogeneity.CART uses Gini method to create binary splits.Process to calculate Gini Measure:

$$Gini = 1 - \sum_j p_j^2$$

- **Entropy**

  Entropy is a way to measure impurity.Less impure node require less information to describe them and more impure node require more information. If the sample is completely homogeneous, then the entropy is zero and if the sample is an equally divided one, it has entropy of one.

$$Entropy = -\sum_j p_j \log_2 p_j$$

- **Information Gain** : Information Gain is simply a mathematical way to capture the amount of information one gains(or reduction in randomness) by picking a particular attribute.In a decision algorithm, we start at the tree root and split the data on the feature that results in the largest information gain (IG). In other words, IG tells us how important a given attribute is.The Information Gain (IG) can be defined as follows:

$$IG(D_p) = I(D_p) - \frac{N_{left}}{N_p} I(D_{left}) - \frac{N_{right}}{N_p} I(D_{right})$$

Chapter 5

Call Housing Data

I worked on the Call-housing dataset provided by Prof Cesa Bianchi.This data shows various variable that may or maynot be effect the prices of houses in a certain area.I explored the relationship between prices of houses with the aid of Decision trees and Regression tress and tried to built a model initially of how prices vary by location across a region.

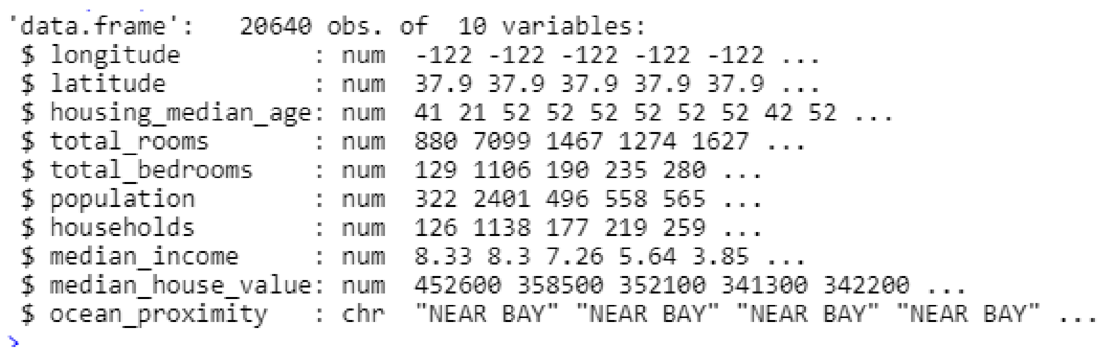
longitude and latitude = Location of the areas of the particular houses
housing_median_age= Median age of houses in that Area
`total_rooms = Total Rooms of the houses in that area`
total_bedrooms= Total bedrooms out of those total tooms
population = Population living in that Area
households = Number of households in that area
median_income = Median Income of the household living in the area
median_house_value:= Median House Values in that area
ocean_proximity = Houses Proximity from Ocean Classified into

< 1H OCEAN
NEAR BAY
INLAND
ISLAND
NEAR OCEAN

Chapter 6

Decision Tree with Data on R

6.1 Analyzing the Data



```
'data.frame':   20640 obs. of  10 variables:
 $ longitude         : num  -122 -122 -122 -122 -122 ...
 $ latitude          : num  37.9 37.9 37.9 37.9 37.9 ...
 $ housing_median_age: num  41 21 52 52 52 52 52 52 42 52 ...
 $ total_rooms       : num  880 7099 1467 1274 1627 ...
 $ total_bedrooms    : num  129 1106 190 235 280 ...
 $ population        : num  322 2401 496 558 565 ...
 $ households        : num  126 1138 177 219 259 ...
 $ median_income     : num  8.33 8.3 7.26 5.64 3.85 ...
 $ median_house_value: num  452600 358500 352100 341300 342200 ...
 $ ocean_proximity   : chr  "NEAR BAY" "NEAR BAY" "NEAR BAY" "NEAR BAY" ...
>
```

Figure 6.1: Appendix Ref 1

There are 20640 observations corresponding to 20640 census tracts. I am interested in building a model of how prices vary by location across a region. So, let's first see how the points are laid out.

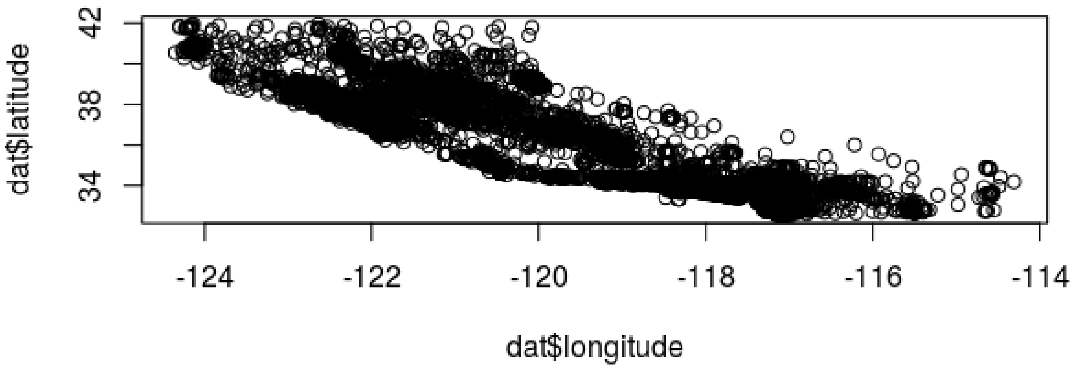


Figure 6.2: Appendix Ref 2

Now lets plot Different Region types on the Map.As from Data study and real estate knowledge we know that Houses near oceans tend to be more expensive and on islands being more exclusive tend to be high Luxury.

< 1H OCEAN =Purple

NEAR BAY =Blue

INLAND =Black

ISLAND =Green

NEAR OCEAN=Pink

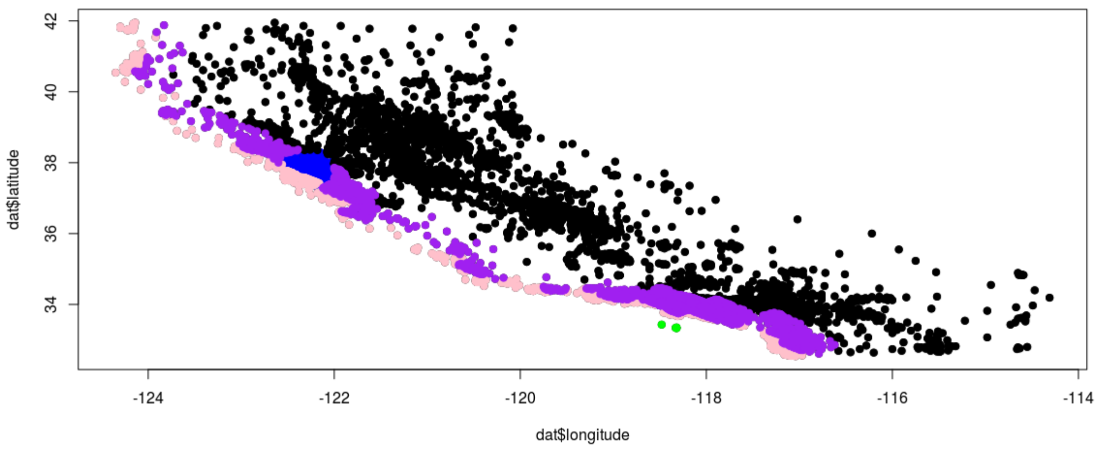
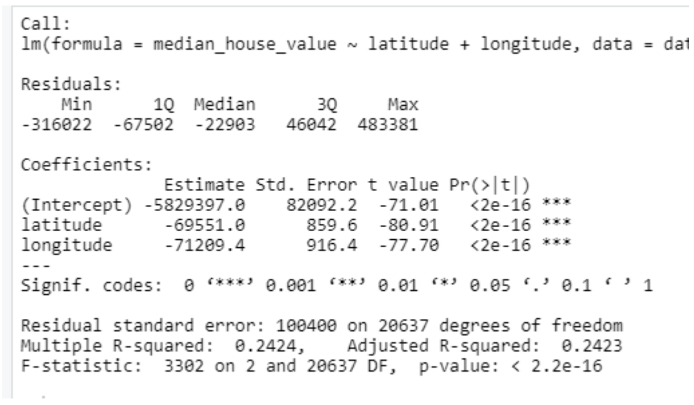


Figure 6.3: Appendix Ref 3

6.1.1 2. Applying Linear Regression to the problem

Since, this is a regression problem as the target value to be predicted is continuous(house price), it is but natural to look up to Linear Regression algorithm to solve the problem. As seen in the last figure that the house prices were distributed over the area in an interesting way according to ocean proximity, certainly not the kind of linear way and I feel Linear Regression is not going to work very well here.



```
Call:
lm(formula = median_house_value ~ latitude + longitude, data = dat

Residuals:
    Min      1Q  Median      3Q     Max
-316022  -67502  -22903   46042  483381

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -5829397.0    82092.2  -71.01   <2e-16 ***
latitude      -69551.0      859.6  -80.91   <2e-16 ***
longitude     -71209.4      916.4  -77.70   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 100400 on 20637 degrees of freedom
Multiple R-squared:  0.2424,    Adjusted R-squared:  0.2423
F-statistic:  3302 on 2 and 20637 DF,  p-value: < 2.2e-16
```

Figure 6.4: Appendix Ref 4

- R-squared is around 0.24, which is not great.
- The latitude is not significant, which means the north-south location differences aren't going to be really used at all. This also seems unlikely.
- Longitude is significant, but negative which means that as we go towards the east house prices decrease linearly, which is also unlikely.

To see how this linear regression model looks on a plot.So I can plot the census tracts again and then plot the above-median house prices with bright red dots. The red dots will tell the actual positions in data where houses are costly. We shall then test the same fact with Linear Regression predictions using blue sign,

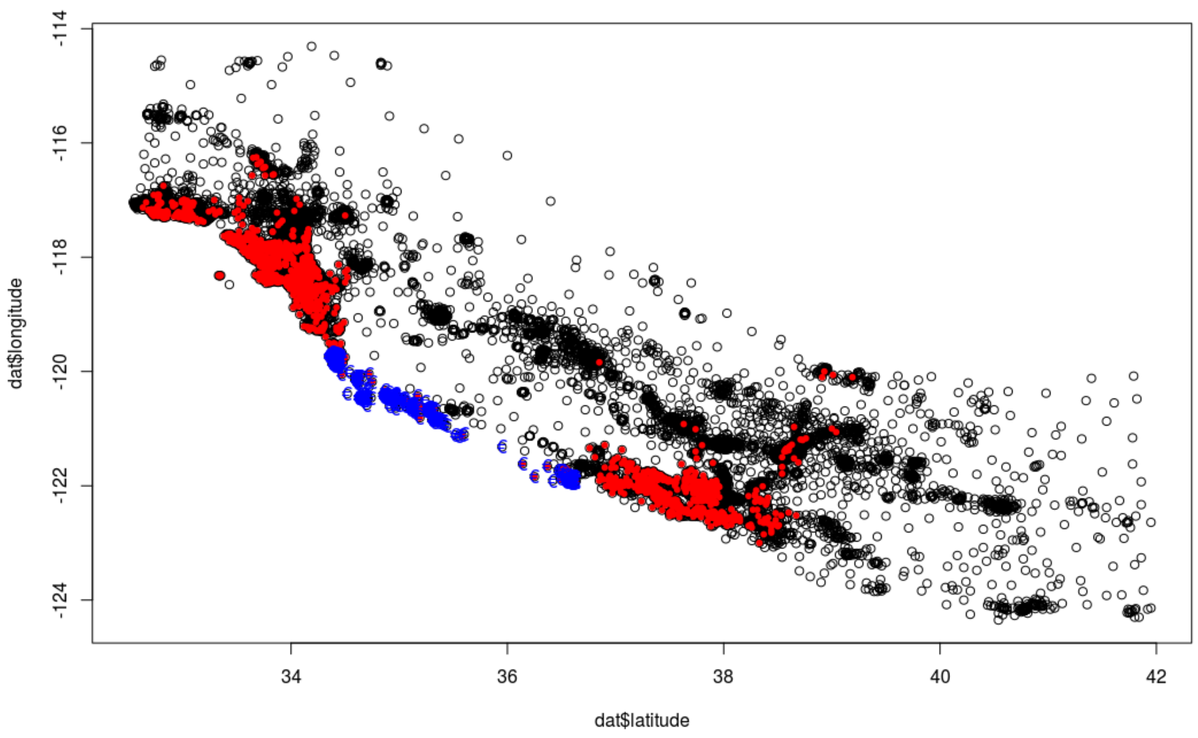


Figure 6.5: Appendix Ref 5

The linear regression model has plotted a blue euro sign for every time it thinks the census tract is above value 3,00,000. It's almost a sharp line that the linear regression defines. Also notice the shape is almost vertical since the latitude variable was not very significant in the regression. The blue and the red dots do not overlap especially in the east and north west. It turns out, the linear regression model isn't really doing a good job. And it has completely ignored everything to the right and left side of the picture. So that's interesting and pretty wrong.

6.1.2 Applying Regression Trees to the problem

I built a regression tree by using the rpart command.I am predicting median_house_value as a function of latitude and longitude, using the dataset.

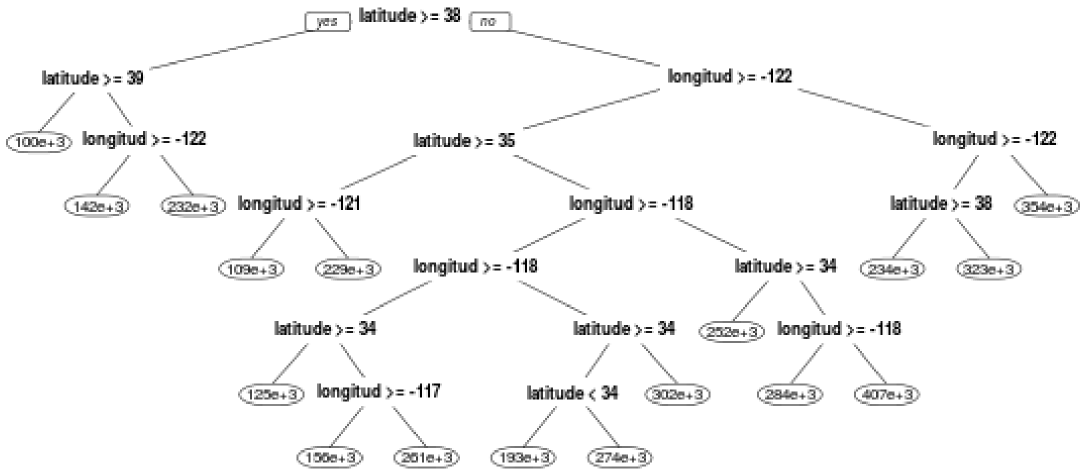


Figure 6.6: Appendix Ref 6

The leaves of the tree are important.

- In a classification tree, the leaves would be the classification that I assign .
- In regression trees, instead predicting the number.That number here is the average of the median house prices in that bucket.

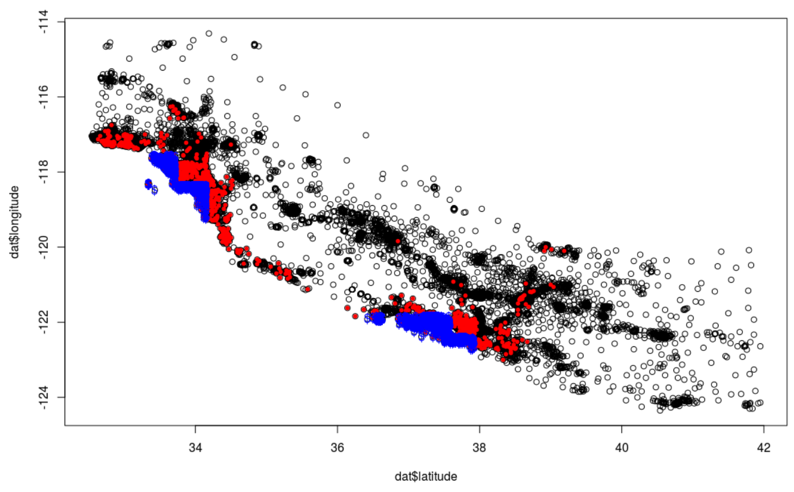


Figure 6.7: Appendix Ref 7

The fitted values are greater than 3,00,000, the colour is blue, The Regression tree has done a much better job and has kind of overlapped the red dots. It has left the low value area out, and has correctly managed to classify some of those points in the bottom right and top right,but still not accurate as it left the houses in the center.Now I further run Regression Tree to better the Model and predictions

6.1.3 Making a Regression Tree Model

- Training and Splitting median_house_value
- Finding and adding important variables to regression- latitude,longitude + median_income + population
- Making Regression Tree

-

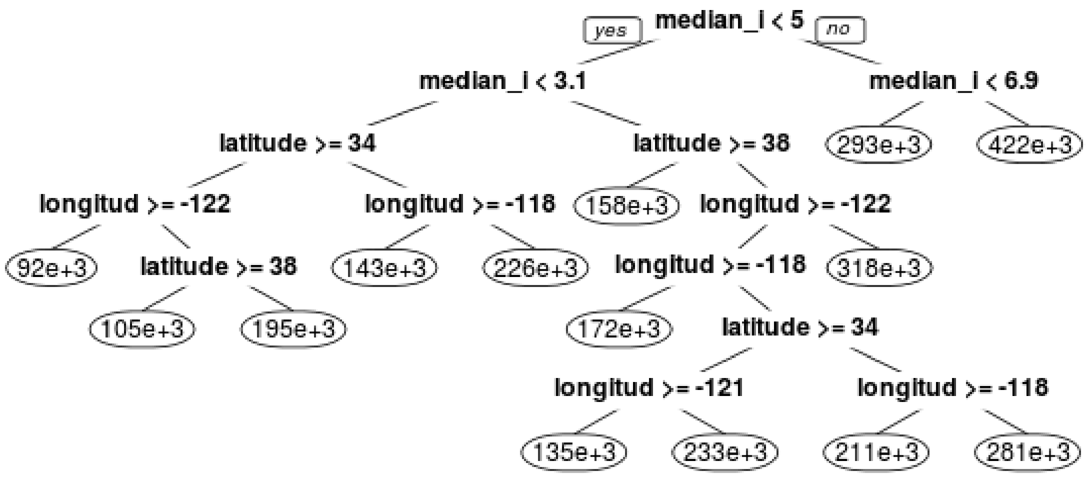


Figure 6.8: Appendix Ref 8

Results: The median_income is the most important variable and split ,and decision starts from there,then algorithm further distinguish between more income values,and then algorithm makes decision based on the locations,longitude and latitude
tree.sse= [1] 3.269545e+13

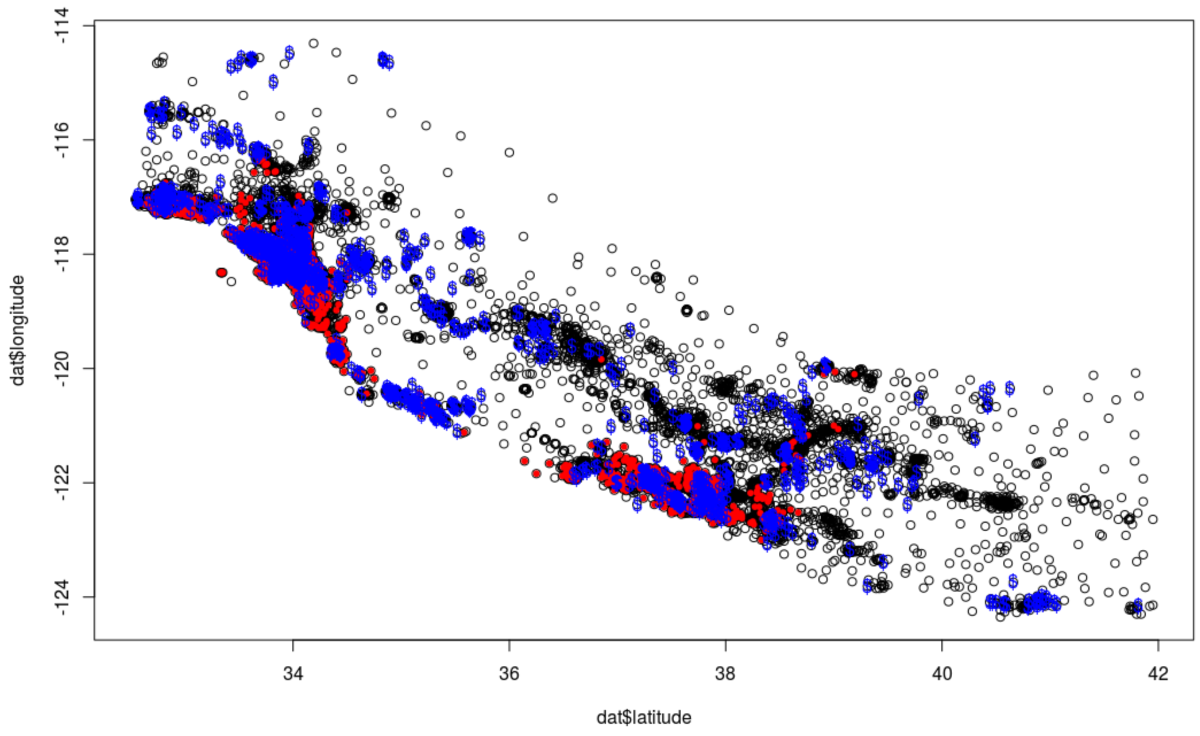


Figure 6.9: Appendix Ref 9

6.1.4 Conclusions

Even though the Decision Trees appears to be working very well in certain conditions, it comes with its own drawbacks.The model has very high chances of "over-fitting". If no limit is set, in the worst case, it will end up putting each observation into a leaf node. Maybe If i further try to use PCA and Cross validation with Regression maybe I can remove the problem of overfitting that I can see with decision tree right now.For that I am going to do futher implementation of PCA and Cross validation on the project of UNSUPERVISED LEARNING with this same data .

Appendix

1

```
#Analayze Data
str(dat)
```

2

```
#Building/laying down Maps/Location
plot(dat$longitude,dat$latitude)
```

3

```
#Marking Different regions,< 1H OCEAN =Purple,NEAR BAY #=Blue,INLAND=Black,ISLAND =Green,NEAR OCEAN=Pink

plot(dat$longitude,dat$latitude)
points(dat$longitude[dat$ocean_proximity=="NEAR␣BAY"], dat$latitude[dat$ocean_proximity=="NEAR␣BAY"], col="blue", pch=19)
points(dat$longitude[dat$ocean_proximity=="ISLAND"], dat$latitude[dat$ocean_proximity=="ISLAND"], col="green", pch=19)
points(dat$longitude[dat$ocean_proximity=="NEAR␣OCEAN"], dat$latitude[dat$ocean_proximity=="NEAR␣OCEAN"], col="pink", pch=19)
points(dat$longitude[dat$ocean_proximity=="INLAND"], dat$latitude[dat$ocean_proximity=="INLAND"], col="black", pch=19)
points(dat$longitude[dat$ocean_proximity=="<1H␣OCEAN"], dat$latitude[dat$ocean_proximity=="<1H␣OCEAN"], col="purple", pch=19)
```

4

```
#Performing Linear regression
\begin{lstlisting}
#Building/laying down Maps/Location
plot(dat$longitude,dat$latitude)
```

5

```
#comparing linear regression results with real data
plot(dat$latitude,dat$longitude)
points(dat$latitude[dat$median_house_value>=300000], dat$longitude[dat$median_house_value>=300000], col="red", pch=20)
latlonlm$fitted.values
points(dat$latitude[latlonlm$fitted.values >= 300000], dat$longitude[latlonlm$fitted.values >= 300000], col="blue", pch="   ")
)
```

6

```
#Making Decision Tree
library(rpart)
library(rpart.plot)
latlontree = rpart(median_house_value ~ latitude + longitude, data=dat)
prp(latlontree)
```

7

```
#Comparing Decision Tree Results with Real Data
plot(dat$latitude,dat$longitude)
points( dat$latitude[dat$median_house_value>=300000],dat$longitude[dat$median_house_value>=300000], col="red", pch=20)
fittedvalues = predict(latlontree)
points(dat$latitude[fittedvalues >= 300000], dat$longitude[fittedvalues >= 300000], col="blue", pch="$")

#Training and Splitting the Data
library(caTools)
set.seed(123)
split = sample.split(dat$median_house_value, SplitRatio = 0.7)
train = subset(dat, split==TRUE)
test = subset(dat, split==FALSE)
```

8

```
# Create a CART model
tree = rpart(median_house_value ~ latitude + longitude + median_income + population , data=train)
prp(tree)

# Finding results for Regression Tree
tree.pred = predict(tree, newdata=test)
tree.sse = sum((tree.pred - test$median_house_value)^2)
tree.sse
```

9

```
# Comparing results of Regression Tree with Real data
```

```
fittedvalues2 <-tree.pred
plot(dat$latitude,dat$longitude)
points( dat$latitude[dat$median_house_value>=300000],dat$longitude[dat$median_house_value>=300000], col="red", pch=20)
fittedvalues = predict(latlontree)
points(dat$latitude[fittedvalues2 >= 300000], dat$longitude[fittedvalues2 >= 300000], col="blue", pch="$")
```

List of Figures