**House Pricing**

**Dataset:** <https://www.kaggle.com/datasets/shree1992/housedata?select=output.csv>

**Response variable:** price

**Predictor variables**: bedrooms, bathrooms, condition, sqft\_living, sqft\_lot, floors, waterfront, view, sqft\_above, sqft\_basement, yr\_built, yr\_renovated

**Structure of the data:**

> house.df <- read.csv("RatanTejaPunati\_housing\_price.csv")

> View(house.df)

> str(house.df)

'data.frame': 4600 obs. of 18 variables:

$ date : chr "2014-05-02 00:00:00" "2014-05-02 00:00:00" "2014-05-02 00:00:00" "2014-05-02 00:00:00" ...

$ price : num 313000 2384000 342000 420000 550000 ...

$ bedrooms : num 3 5 3 3 4 2 2 4 3 4 ...

$ bathrooms : num 1.5 2.5 2 2.25 2.5 1 2 2.5 2.5 2 ...

$ sqft\_living : int 1340 3650 1930 2000 1940 880 1350 2710 2430 1520 ...

$ sqft\_lot : int 7912 9050 11947 8030 10500 6380 2560 35868 88426 6200 ...

$ floors : num 1.5 2 1 1 1 1 1 2 1 1.5 ...

$ waterfront : int 0 0 0 0 0 0 0 0 0 0 ...

$ view : int 0 4 0 0 0 0 0 0 0 0 ...

$ condition : int 3 5 4 4 4 3 3 3 4 3 ...

$ sqft\_above : int 1340 3370 1930 1000 1140 880 1350 2710 1570 1520 ...

$ sqft\_basement: int 0 280 0 1000 800 0 0 0 860 0 ...

$ yr\_built : int 1955 1921 1966 1963 1976 1938 1976 1989 1985 1945 ...

$ yr\_renovated : int 2005 0 0 0 1992 1994 0 0 0 2010 ...

$ street : chr "18810 Densmore Ave N" "709 W Blaine St" "26206-26214 143rd Ave SE" "857 170th Pl NE" ...

$ city : chr "Shoreline" "Seattle" "Kent" "Bellevue" ...

$ statezip : chr "WA 98133" "WA 98119" "WA 98042" "WA 98008" ...

$ country : chr "USA" "USA" "USA" "USA" ...

> house.df<-house.df[-4351,]

> house.df<-house.df[-4347,]

> house.df <- house.df[house.df$price != 0, ]

> house.df <- na.omit(house.df)

> str(house.df) # structure of the data frame

'data.frame': 4549 obs. of 18 variables:

$ date : chr "2014-05-02 00:00:00" "2014-05-02 00:00:00" "2014-05-02 00:00:00" "2014-05-02 00:00:00" ...

$ price : num 313000 2384000 342000 420000 550000 ...

$ bedrooms : num 3 5 3 3 4 2 2 4 3 4 ...

$ bathrooms : num 1.5 2.5 2 2.25 2.5 1 2 2.5 2.5 2 ...

$ sqft\_living : int 1340 3650 1930 2000 1940 880 1350 2710 2430 1520 ...

$ sqft\_lot : int 7912 9050 11947 8030 10500 6380 2560 35868 88426 6200 ...

$ floors : num 1.5 2 1 1 1 1 1 2 1 1.5 ...

$ waterfront : int 0 0 0 0 0 0 0 0 0 0 ...

$ view : int 0 4 0 0 0 0 0 0 0 0 ...

$ condition : int 3 5 4 4 4 3 3 3 4 3 ...

$ sqft\_above : int 1340 3370 1930 1000 1140 880 1350 2710 1570 1520 ...

$ sqft\_basement: int 0 280 0 1000 800 0 0 0 860 0 ...

$ yr\_built : int 1955 1921 1966 1963 1976 1938 1976 1989 1985 1945 ...

$ yr\_renovated : int 2005 0 0 0 1992 1994 0 0 0 2010 ...

$ street : chr "18810 Densmore Ave N" "709 W Blaine St" "26206-26214 143rd Ave SE" "857 170th Pl NE" ...

$ city : chr "Shoreline" "Seattle" "Kent" "Bellevue" ...

$ statezip : chr "WA 98133" "WA 98119" "WA 98042" "WA 98008" ...

$ country : chr "USA" "USA" "USA" "USA" ...

This data converts a CSV file to a data frame, removes specified rows, rows with a price of 0, and rows with missing values (NA), and then displays the updated data frame and its structure. The result shown above shows the structure of the data frame, which includes 4549 observations and 18 variables. Each variable has an associated data type and value.

**Summary Statistics:**

> describe(house.df)

vars n mean sd median trimmed mad min max

date\* 1 4549 37.16 19.70 39.00 37.41 25.20 1 70.0

price 2 4549 549470.38 368230.65 465000.00 492447.74 230562.83 7800 7062500.0

bedrooms 3 4549 3.39 0.90 3.00 3.36 1.48 0 9.0

bathrooms 4 4549 2.15 0.78 2.25 2.12 0.74 0 8.0

sqft\_living 5 4549 2132.57 956.06 1970.00 2033.56 837.67 370 13540.0

sqft\_lot 6 4549 14837.59 35971.80 7680.00 8451.47 4111.25 638 1074218.0

floors 7 4549 1.51 0.54 1.50 1.47 0.74 1 3.5

waterfront 8 4549 0.01 0.08 0.00 0.00 0.00 0 1.0

view 9 4549 0.23 0.77 0.00 0.00 0.00 0 4.0

condition 10 4549 3.45 0.68 3.00 3.33 0.00 1 5.0

sqft\_above 11 4549 1822.42 854.58 1590.00 1715.20 711.65 370 9410.0

sqft\_basement 12 4549 310.15 462.04 0.00 224.02 0.00 0 4820.0

yr\_built 13 4549 1970.79 29.76 1976.00 1973.05 34.10 1900 2014.0

yr\_renovated 14 4549 808.48 979.40 0.00 759.40 0.00 0 2014.0

street\* 15 4549 2240.85 1292.85 2239.00 2241.30 1661.99 1 4474.0

city\* 16 4549 26.71 11.97 33.00 28.07 4.45 1 44.0

statezip\* 17 4549 39.76 20.88 42.00 40.21 26.69 1 77.0

country\* 18 4549 1.00 0.00 1.00 1.00 0.00 1 1.0

range skew kurtosis se

date\* 69.0 -0.12 -1.16 0.29

price 7054700.0 3.99 36.31 5459.61

bedrooms 9.0 0.47 1.29 0.01

bathrooms 8.0 0.59 1.80 0.01

sqft\_living 13170.0 1.72 8.39 14.18

sqft\_lot 1073580.0 11.32 219.75 533.34

floors 2.5 0.55 -0.54 0.01

waterfront 1.0 12.19 146.57 0.00

view 4.0 3.37 10.68 0.01

condition 4.0 0.96 0.21 0.01

sqft\_above 9040.0 1.45 3.74 12.67

sqft\_basement 4820.0 1.65 4.19 6.85

yr\_built 114.0 -0.51 -0.67 0.44

yr\_renovated 2014.0 0.39 -1.85 14.52

street\* 4473.0 0.00 -1.20 19.17

city\* 43.0 -0.77 -0.78 0.18

statezip\* 76.0 -0.15 -1.12 0.31

country\* 0.0 NaN NaN 0.00

|  |
| --- |
| **date**: The date variable has a skewness of -0.12, and its range is 1 to 70. **price**: The houses prices range widely from 7800 to 7062500, and their high (3.99) skewness indicates a right skew. **bedrooms**: There are 0 to 9 bedrooms, with a little positive skew (0.47). **bathrooms**: Positively skewed data, with bathrooms ranging from 0 to 8. **sqft\_living**: Positively skewed square footage of living area with a wide range (1.72–1.72). **sqft\_lot**: The lot's square footage, positively skewed and with a broad range (11.32). **floors**: Positively skewed (0.55) number of floors, ranging from 1 to 3.5. **waterfront**: A substantially positively skewed binary variable that indicates if the home has a waterfront (12.19). **view**: Positively skewed categorical variable that reflects the viewpoint (3.37).  **condition**: The house's condition is positively skewed (0.96), with a range of 1 to 5. **sqft\_above**: Positively skewed square footage of the house (excluding the basement) (1.45). **sqft\_basement**: Positively skewed square footage of the basement (1.65)... **yr\_built**: The year of construction, significantly negatively skewed (-0.51), ranging from 1900 to 2014. **yr\_renovated**: The year of renovation, positively skewed (0.39) and ranging from 0 to 2014... **street**: A categorical variable with no skewness that represents the street. **city**: A negatively skewed (-0.77) categorical variable that represents the city. **statezip**: A categorical variable with no discernible skewness that represents the state and zip code. **country**: A constant variable that has no volatility and only one value, or one skewness. |

**Converting to Binary :**

> house.df$price <- ifelse(house.df$price > mean(house.df$price), 1, 0)

> str(house.df)

'data.frame': 4549 obs. of 18 variables:

$ date : chr "2014-05-02 00:00:00" "2014-05-02 00:00:00" "2014-05-02 00:00:00" "2014-05-02 00:00:00" ...

$ price : num 0 1 0 0 1 0 0 0 0 1 ...

$ bedrooms : num 3 5 3 3 4 2 2 4 3 4 ...

$ bathrooms : num 1.5 2.5 2 2.25 2.5 1 2 2.5 2.5 2 ...

$ sqft\_living : int 1340 3650 1930 2000 1940 880 1350 2710 2430 1520 ...

$ sqft\_lot : int 7912 9050 11947 8030 10500 6380 2560 35868 88426 6200 ...

$ floors : num 1.5 2 1 1 1 1 1 2 1 1.5 ...

$ waterfront : int 0 0 0 0 0 0 0 0 0 0 ...

$ view : int 0 4 0 0 0 0 0 0 0 0 ...

$ condition : int 3 5 4 4 4 3 3 3 4 3 ...

$ sqft\_above : int 1340 3370 1930 1000 1140 880 1350 2710 1570 1520 ...

$ sqft\_basement: int 0 280 0 1000 800 0 0 0 860 0 ...

$ yr\_built : int 1955 1921 1966 1963 1976 1938 1976 1989 1985 1945 ...

$ yr\_renovated : int 2005 0 0 0 1992 1994 0 0 0 2010 ...

$ street : chr "18810 Densmore Ave N" "709 W Blaine St" "26206-26214 143rd Ave SE" "857 170th Pl NE" ...

$ city : chr "Shoreline" "Seattle" "Kent" "Bellevue" ...

$ statezip : chr "WA 98133" "WA 98119" "WA 98042" "WA 98008" ...

$ country : chr "USA" "USA" "USA" "USA" ...

After a binary conversion of the 'price' variable (1 if the price exceeds the mean price, zero otherwise), the data frame's structure, which includes 18 variables and 4549 observations, remains the same.

**Analysis:**

Dataset is divided into 60% training data and 40% validation data.

> str(train.df)

'data.frame': 2729 obs. of 13 variables:

$ price : Factor w/ 2 levels "0","1": 2 1 1 1 2 2 1 1 1 2 ...

$ bedrooms : num 3 3 1 2 5 4 4 2 2 4 ...

$ bathrooms : num 2.5 2.5 1 1 3.75 1 2.5 1 1.5 1.75 ...

$ sqft\_living : int 2780 2160 690 2550 4130 2550 2160 980 1068 1660 ...

$ sqft\_lot : int 33503 3000 1950 21675 226076 4000 7826 2130 1758 3800 ...

$ floors : num 1.5 1.5 1 1 2 2 1 1 2 1.5 ...

$ waterfront : int 0 0 0 0 0 0 0 0 0 0 ...

$ view : int 1 0 0 1 0 0 0 0 0 0 ...

$ condition : int 4 3 3 4 3 3 4 4 3 3 ...

$ sqft\_above : int 2110 1260 690 1610 3170 2370 1390 860 1068 1660 ...

$ sqft\_basement: int 670 900 0 940 960 180 770 120 0 0 ...

$ yr\_built : int 1969 1909 1928 1958 1985 1905 1976 1918 1990 1926 ...

$ yr\_renovated : int 0 2011 1954 1972 0 2010 1992 1974 2009 2003 ...

> str(valid.df)

'data.frame': 1820 obs. of 13 variables:

$ price : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 1 1 1 ...

$ bedrooms : num 3 3 3 4 2 2 3 3 3 3 ...

$ bathrooms : num 1.5 2 2.25 2.5 1 2 1.75 1.5 1.75 1.5 ...

$ sqft\_living : int 1340 1930 2000 1940 880 1350 1710 1570 1370 1180 ...

$ sqft\_lot : int 7912 11947 8030 10500 6380 2560 7320 6700 5858 10277 ...

$ floors : num 1.5 1 1 1 1 1 1 1 1 1 ...

$ waterfront : int 0 0 0 0 0 0 0 0 0 0 ...

$ view : int 0 0 0 0 0 0 0 0 0 0 ...

$ condition : int 3 4 4 4 3 3 3 4 3 3 ...

$ sqft\_above : int 1340 1930 1000 1140 880 1350 1710 1570 1370 1180 ...

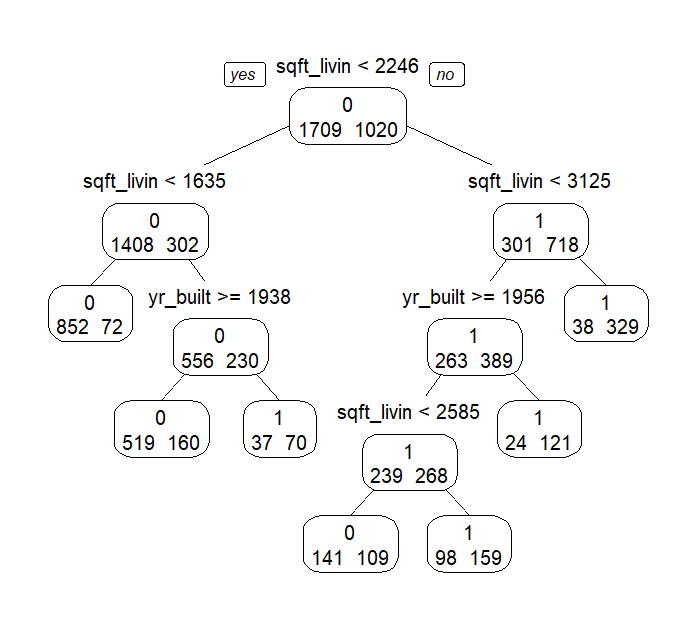
$ sqft\_basement: int 0 0 1000 800 0 0 0 0 0 0 ...

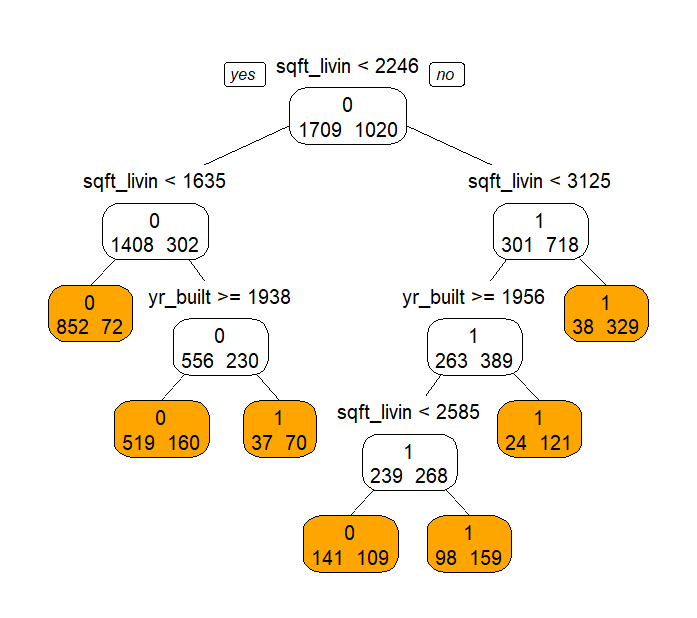
$ yr\_built : int 1955 1966 1963 1976 1938 1976 1948 1956 1987 1983 ...

$ yr\_renovated : int 2005 0 0 1992 1994 0 1994 0 2000 2009 ...

This gives a quick overview of the structure and variables in the 'train.df' and 'valid.df' data frames. The 'train.df' data frame consists of 2729 observations and 15 variables. The 'valid.df' data frame has 1820 observations and 15 variables with the same structure as 'train.df'.

**Single Classification tree**





A single classification tree is fit with all features. The classification tree consists of two variables and six decision nodes.

> tr$variable.importance

sqft\_living sqft\_above bathrooms bedrooms sqft\_basement floors

445.8512468 303.2760317 176.1339732 138.7743134 96.4387240 72.3215347

yr\_built sqft\_lot condition view yr\_renovated

57.4229934 3.9959899 2.6425673 2.5309412 0.1455129

sqft\_living is significantly more important than all other variables in determining price, followed by sqft\_above, bathrooms, and so on.

> ####classification matrix for training data

> pred <- predict(tr, train.df, type = "class")

> c.mat <- table(pred, train.df$price)

> c.mat # row: predicted ; column: actual

pred 0 1

0 1512 341

1 197 679

> sum(diag(c.mat))/sum(c.mat) # this gives accuracy: overall correct classification percentage

[1] 0.8028582

> c.mat[2,2]/sum(c.mat[,2]) # this gives True positive percentage, or sensitivity

[1] 0.6656863

> c.mat[1,1]/sum(c.mat[,1]) # this gives True negatie percentage, or specificity

[1] 0.8847279

> ####classification matrix for validation data

> pred <- predict(tr, valid.df, type = "class")

> c.mat <- table(pred, valid.df$price)

> c.mat # row: predicted ; column: actual

pred 0 1

0 986 230

1 149 455

> sum(diag(c.mat))/sum(c.mat) # this gives accuracy: overall correct classification percentage

[1] 0.7917582

> c.mat[2,2]/sum(c.mat[,2]) # this gives True positive percentage, or sensitivity

[1] 0.6642336

> c.mat[1,1]/sum(c.mat[,1]) # this gives True negatie percentage, or specificity

[1] 0.8687225

**Performance with Training Data:**

The model's overall accuracy using the training data was roughly 80.29%.

True Positive Rate (sensitivity): 66.57%.

True Negative Rate (Specificity): 88.47 percent.

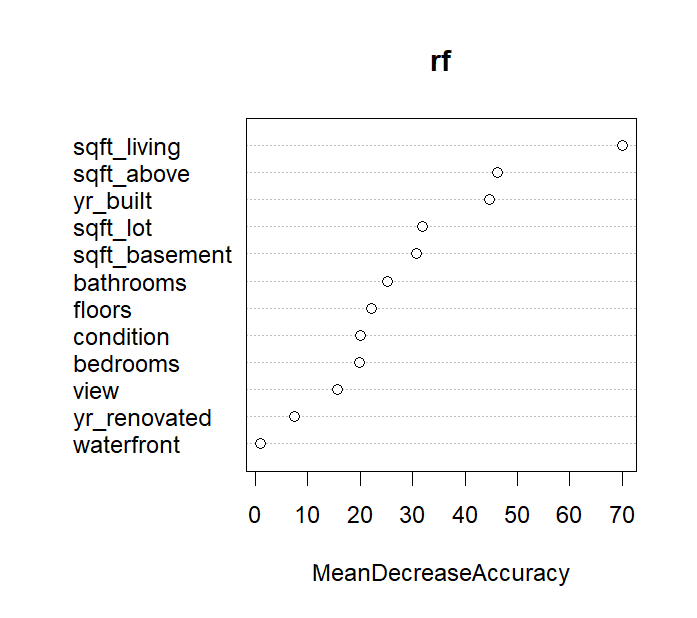
**Performance with Validation Data:**

The model performed well on the validation dataset, with an accuracy of approximately 79.18%.

True Positive Rate (sensitivity): 66.42%.

True Negative Rate (Specificity): 86.87 percent.

**Random Forest:**



The plot shows how each feature includes to the model's overall accuracy. Variables with greater Mean Decrease Accuracy scores are more effective at predicting the outcome. For example, sqft\_living appears to be highly significant, indicating that the size of the living area is an excellent predictor of the model's accuracy.

> ####classification matrix for training data

> rf.pred <- predict(rf, train.df, type = "class")

> c.mat <- table(rf.pred, train.df$price)

> c.mat # row: predicted ; column: actual

rf.pred 0 1

0 1696 66

1 13 954

> sum(diag(c.mat))/sum(c.mat) # this gives accuracy: overall correct classification percentage

[1] 0.9710517

> c.mat[2,2]/sum(c.mat[,2]) # this gives True positive percentage, or sensitivity

[1] 0.9352941

> c.mat[1,1]/sum(c.mat[,1]) # this gives True negatie percentage, or specificity

[1] 0.9923932

> ####classification matrix for validation data

> rf.pred <- predict(rf, valid.df, type = "class")

> c.mat <- table(rf.pred, valid.df$price)

> c.mat # row: predicted ; column: actual

rf.pred 0 1

0 981 208

1 154 477

> sum(diag(c.mat))/sum(c.mat) # this gives accuracy: overall correct classification percentage

[1] 0.8010989

> c.mat[2,2]/sum(c.mat[,2]) # this gives True positive percentage, or sensitivity

[1] 0.6963504

> c.mat[1,1]/sum(c.mat[,1]) # this gives True negatie percentage, or specificity

[1] 0.8643172

**Performance with Training Data:**

The model's overall accuracy using the training data was roughly 97.11%.

True Positive Rate (sensitivity): 93.53%.

True negative rate (specificity) is 99.24%.

**Performance using Validation Data:**

The model's performance on the validation dataset indicated an accuracy of approximately 80.11%.

True Positive Rate (sensitivity): 69.64%.

The true negative rate (specificity) is 86.43%.

**Bagging Model:**

> varImp(bag)

Overall

bathrooms 387.096702

bedrooms 259.809609

condition 119.502447

floors 209.158513

sqft\_above 701.738889

sqft\_basement 204.213849

sqft\_living 831.085812

sqft\_lot 478.894928

view 103.308604

waterfront 9.896997

yr\_built 429.349937

yr\_renovated 153.214380

These values represent the significance of each predictor variable in the bagged ensemble model. Higher values indicate higher significance.

It is important to note that sqft\_living, sqft\_above, and bathrooms have the greatest significance ratings, indicating that they play an important role in the model's predictive performance. On the other hand, waterfront appears to be the least important of the characteristics considered.

> ####classification matrix for training data

> bag.pred <- predict(bag, train.df, type = "class")

> c.mat <- table(bag.pred, train.df$price)

> c.mat # row: predicted ; column: actual

bag.pred 0 1

0 1704 6

1 5 1014

> sum(diag(c.mat))/sum(c.mat) # this gives accuracy: overall correct classification percentage

[1] 0.9959692

> c.mat[2,2]/sum(c.mat[,2]) # this gives True positive percentage, or sensitivity

[1] 0.9941176

> c.mat[1,1]/sum(c.mat[,1]) # this gives True negatie percentage, or specificity

[1] 0.9970743

> ####classification matrix for validation data

> bag.pred <- predict(bag, valid.df, type = "class")

> c.mat <- table(bag.pred, valid.df$price)

> c.mat # row: predicted ; column: actual

bag.pred 0 1

0 955 206

1 180 479

> sum(diag(c.mat))/sum(c.mat) # this gives accuracy: overall correct classification percentage

[1] 0.7879121

> c.mat[2,2]/sum(c.mat[,2]) # this gives True positive percentage, or sensitivity

[1] 0.6992701

> c.mat[1,1]/sum(c.mat[,1]) # this gives True negatie percentage, or specificity

[1] 0.8414097

**Performance with Training Data:**

Overall, the model's accuracy on the training data was roughly 99.60%.

True Positive Rate (sensitivity): 99.41%.

The true negative rate (specificity) is 99.71 percent.

**Performance using Validation Data:**

Model performance on the validation dataset was slightly lower, at roughly 78.79%.

True positive rate (sensitivity): 69.93 percent.

The true negative rate (specificity) is 84.14%.

**Comparison of LR, NN, and Tree methods**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Training Accuracy | Training Sensitivity | Training Specificity | Validation Accuracy | Validation Sensitivity | Validation Specificity |
| Logistic Regression | 78.20% | 63.09% | 86.78% | 77.09% | 59.73% | 87.27% |
| Neural Network | 81.57% | 73.63% | 86.31% | 78.85% | 71.68% | 83.17% |
| Decision Tree | 80.29% | 66.57% | 88.47% | 79.18% | 66.42% | 86.87% |
| Random Forest | 97.11% | 93.53% | 99.24% | 80.11% | 69.64% | 86.43% |
| Bagging Model | 99.60% | 99.41% | 99.71% | 78.79% | 69.93% | 84.14% |

**Conclusion:**

Random Forest and Bagging Model have the highest accuracy on training data, while Logistic Regression and Neural Network do relatively well.

**Sensitivity**: Random Forest is the most sensitive to both training and validation data, closely followed by the Bagging Model.

**Specificity**: Logistic Regression and Single Classification Tree have higher specificity than other approaches.

Finally, while Random Forest and Bagging Model exhibit higher accuracy and sensitivity on training data, it is critical to examine the relationship between model complexity and generalization performance. Depending on the application's specific objectives and limitations, any of the models discussed above may be an appropriate alternative for projecting house values. Further fine-tuning and optimization of these models may increase their performance on previously discovered data.