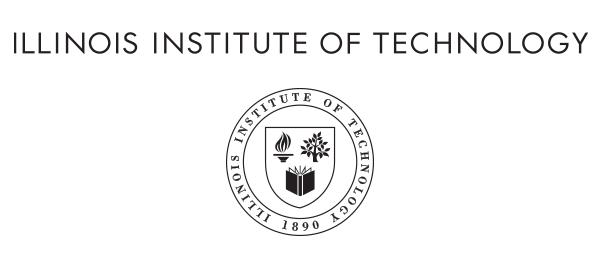
****

**CSP571-Data Preparation and Analysis**

**Professor: Jawahar Panchal**

**California House Price Prediction**

**Atharva Nirali (A20517247)**

**Neha Ramesh Gawali (A20523722)**

**Vaishnavi Mule (A20516627)**

**Vikas Reddy Nomu (A20539316)**

**ABSTRACT**

**1.** [**INTRODUCTION**](#_t4e5692fgcio)

1.1 Purpose of the Project

1.2 Proposed Methodology

1.3 Scope of the Project

**2.** [**DATA PRE-PROCESSING**](#_6sye99o1uir)

2.1 Feature Description of the Dataset

2.2 Data Cleaning

2.2.1 Missing Data

**3.** [**DATA ANALYSIS**](#_axdwkeei5202)

**4.** [**MODEL TRAINING**](#_qobdy5qfwama)

4.1 Feature Engineering

4.1.1 Imputation

4.1.2 Binning

4.1.3 One-hot Encoding

4.1.4 Feature Selection

4.2 Evaluation Metrics

4.2.1 Linear Regression

4.2.2 Random Forest

4.2.3 Decision Tree

4.3 Model Selection

4.4 Test Train Split

**5.** [**MODEL VALIDATION**](#_s51jo16i4426)

5.1 Linear Regression

5.2 Decision Tree

5.3 Testing Results

**6.** [**CONCLUSION**](#_8ca4l9wbe3vy)

**7.** [**FUTURE ENHANCEMENTS**](#_ejhnmheuq33z)

**8.** [**DATA SOURCES**](#_lx8zqwiiurac)

**9.** [**SOURCE CODE**](#_fshdsf1gdsfe)

**10.** [**LITERATURE REVIEW**](#_8dw7xeik771i)

**11.** [**PROJECT REPO LINK**](#_tq6luqya2ddj)

**CALIFORNIA HOUSE PRICE PREDICTION**

**Abstract**

Machine learning model predictions allow businesses to make highly accurate predictions as to the likely outcomes of a question based on previous data. These provide the business with insights that result in tangible business value. For example, if a model predicts a customer is likely to churn, the business can target them with specific communications and outreach that will prevent the loss of that customer. Real estate is the least transparent industry in our ecosystem. Housing prices keep changing day in and day out and sometimes are hyped rather than being based on valuation. Predicting housing prices depend upon many user defined parameters (features) with the real crux of our project. Here we aim to make our evaluations based on every basic parameter that is considered while determining the price such as area, location, number of rooms, bathrooms, bedrooms, etc. We use various regression techniques like linear and multiple variable regression in this pathway. This model should learn from the data and be able to predict the median housing price in any district, given all other metrics.

## INTRODUCTION

**1.1** **Purpose of Project**

The purpose of this system is to determine the price of a house by looking at the various features which are given as input by the user. These features are given to the ML model and based on how these features affect the label it gives out a prediction. This will be done by first searching for an appropriate dataset that suits the needs of the developer as well as the user. Furthermore, after finalizing the dataset, the dataset will go through the process known as data cleaning where all the data which is not needed will be eliminated and the raw data will be turned into a .csv file. Moreover, the data will go through data pre-processing where missing data will be handled and if needed label encoding will be done. Moreover, this will go through data transformation where it will be converted into a NumPy array so that it can finally be sent for training the model. While training various machine learning algorithms will be used to train the model their error rate will be extracted and consequently an algorithm and model will be finalized which can yield accurate predictions.

**1.2 Proposed Methodology**

In the initial phase of our analysis, we performed a comprehensive Exploratory Data Analysis (EDA) on the dataset, delving into summary statistics, the distribution of the target variable (median\_house\_value), and visualizing relationships between variables through scatter plots and correlation matrices. Subsequently, we undertook robust Data Cleaning procedures, addressing missing values either through imputation or removal to ensure a clean dataset. Feature Engineering played a crucial role, allowing us to create new features from existing ones, enhancing the dataset's richness and capturing more meaningful information. The construction of a consistent Data Pipeline facilitated uniform processing of both training and testing datasets, while Stratified Sampling ensured a balanced distribution of target variables in these sets, mitigating bias. Our focus on Training and Model Evaluation involved the selection of models, such as linear regression and random forest, and their assessment using metrics like mean absolute error and cross-validation for robust performance evaluation. The interpretation of model results, considering linear regression coefficients and random forest feature importance, provided insights into the impact of different features on predicted median house values. Through Model Comparison, we evaluated and compared the performance of different models, identifying the most effective one for our regression task. Additionally, we explored the potential benefits of Model Stacking, considering ensemble methods to enhance overall predictive capabilities. This systematic and informed approach guides our decision-making process, ensuring the reliability and effectiveness of the chosen regression model.

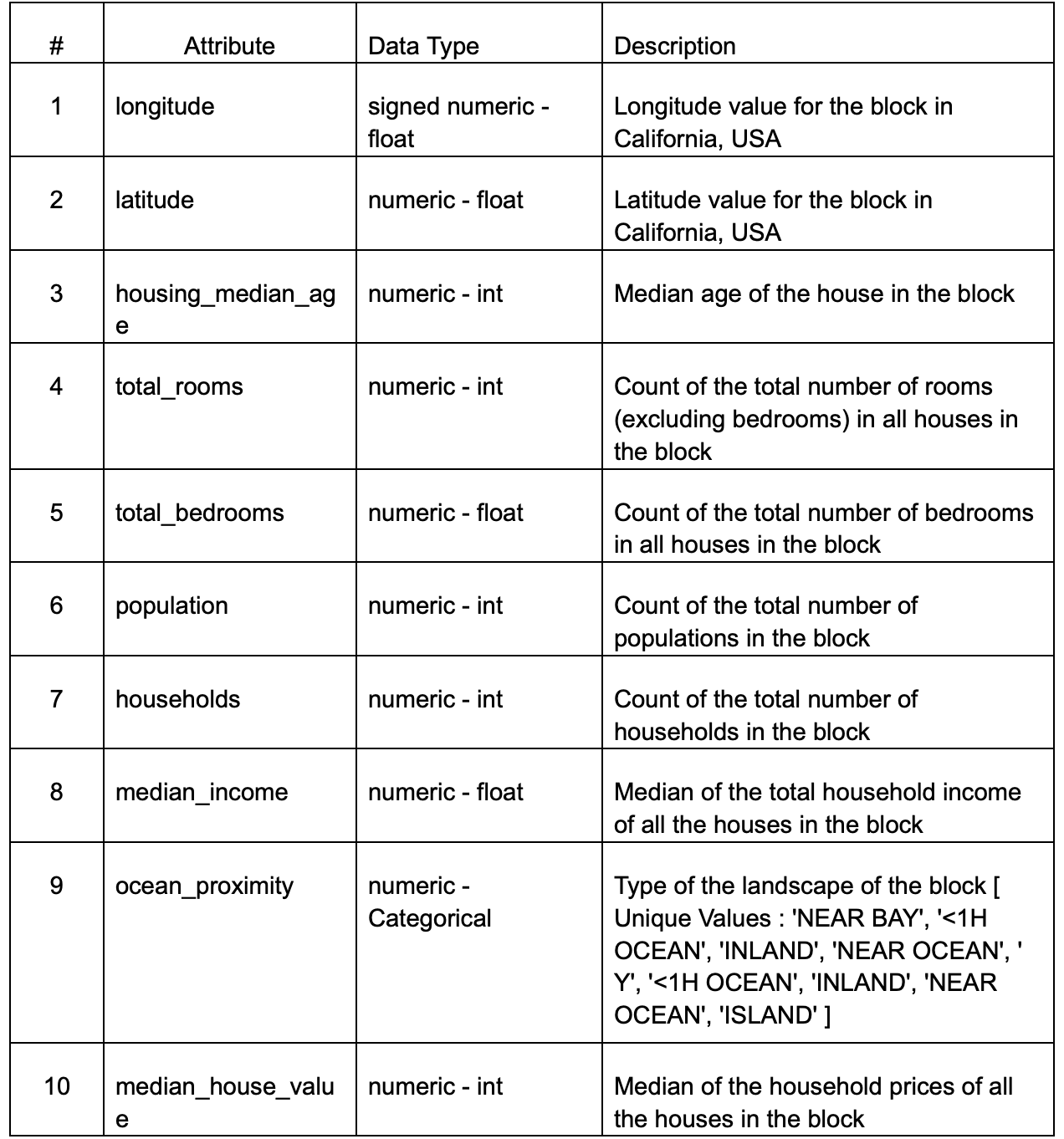
**1.3** **Scope of the Project**

Despite the profound and far-reaching significance brought by this method, its potential for statistical prediction on the pricing of real estate has been in deficiency by its nature. The limit of the traditional hedonic regression model has a significant influence on the procedure of generating the model, making it hard to identify the appropriate variables to set up the eventual

model. Therefore, adopting other methodologies to conduct rigorous and viable research to a different extent is indispensable.

### **Data Pre Processing**

**2.1 Feature description of the dataset**



**2.2 Data Cleaning**

Data cleaning, also known as data cleansing or data scrubbing, is the process of identifying and correcting errors, inconsistencies, and inaccuracies in datasets. It involves cleaning up messy, incomplete, or corrupted data to enhance its quality and reliability. The goal of data cleaning in a data project is to improve the accuracy and integrity of the data, ensuring that it is suitable for analysis and decision-making. This process typically includes handling missing values, removing duplicate records, correcting errors, and standardizing formats, ultimately preparing the data for meaningful insights and effective use in data-driven applications.

### **2.2.1 Missing Data**

There are 207 observations with NA values for “total\_bedrooms”. We can handle this by imputing median value (435) at these places.

# Check if there are any missing values in the numeric columns

if (any(is.na(hist\_data))) {

hist\_data <- hist\_data[, !apply(is.na(hist\_data), 2, any)]

}

### **Data Analysis**

### **Data Processing:**

#### **Loading Libraries and Dataset:**

* Load necessary libraries, including caret, randomForest, rpart, dplyr, and corrplot.
* Read the dataset (housing.csv) into the variable housing.



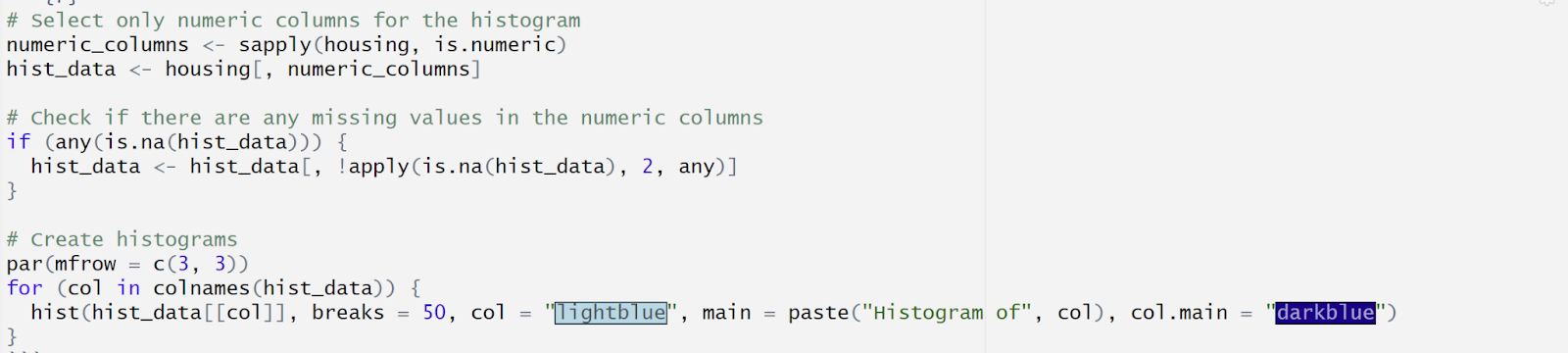
#### **Exploring the Dataset:**

* Display the first few rows of the dataset using head(housing) to get an initial understanding of the data.



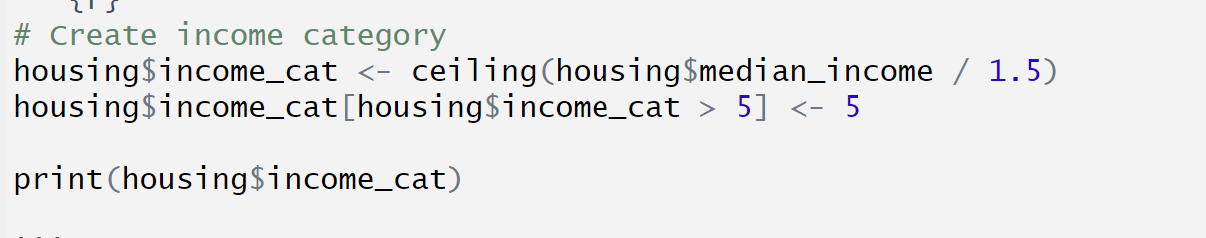
#### **Creating Histograms for Numeric Columns:**

* Select only numeric columns for creating histograms.
* Check for missing values in numeric columns and remove them if present.
* Create histograms for each numeric column.
* Histograms helped us visualize the distribution of numeric variables, identify potential outliers, and assess the data's overall shape.



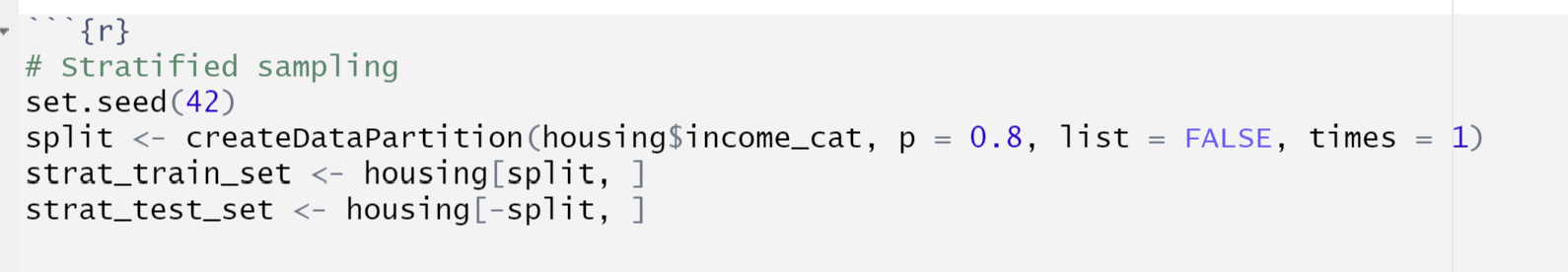
#### **Creating Income Categories:**

* Create an income category variable (income\_cat) based on the ceiling of the ratio of median\_income divided by 1.5.
* Set any income category above 5 to 5.
* Creating income categories can help stratify the data for more meaningful analysis, particularly since income is an important predictor.



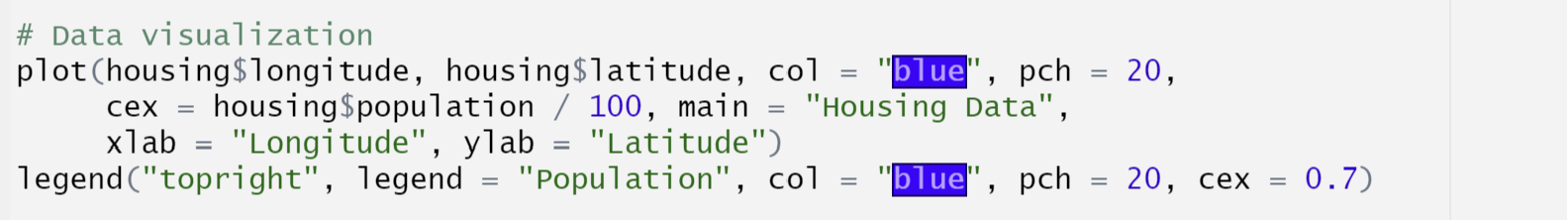
#### **Stratified Sampling:**

* Perform stratified sampling to create training and test sets.
* Stratified sampling ensured that the income categories are well-represented in both training and test sets, reducing the risk of biased model training.



#### **Data Visualization:**

* Visualize geographic data using a scatter plot of longitude and latitude.
* Adjust point size based on population and include a legend.



**Data Analysis**

After we have cleaned the data, we will visualize the data to get some insights into the distribution and skewness of numeric data as well as correlation of the variables with each other.

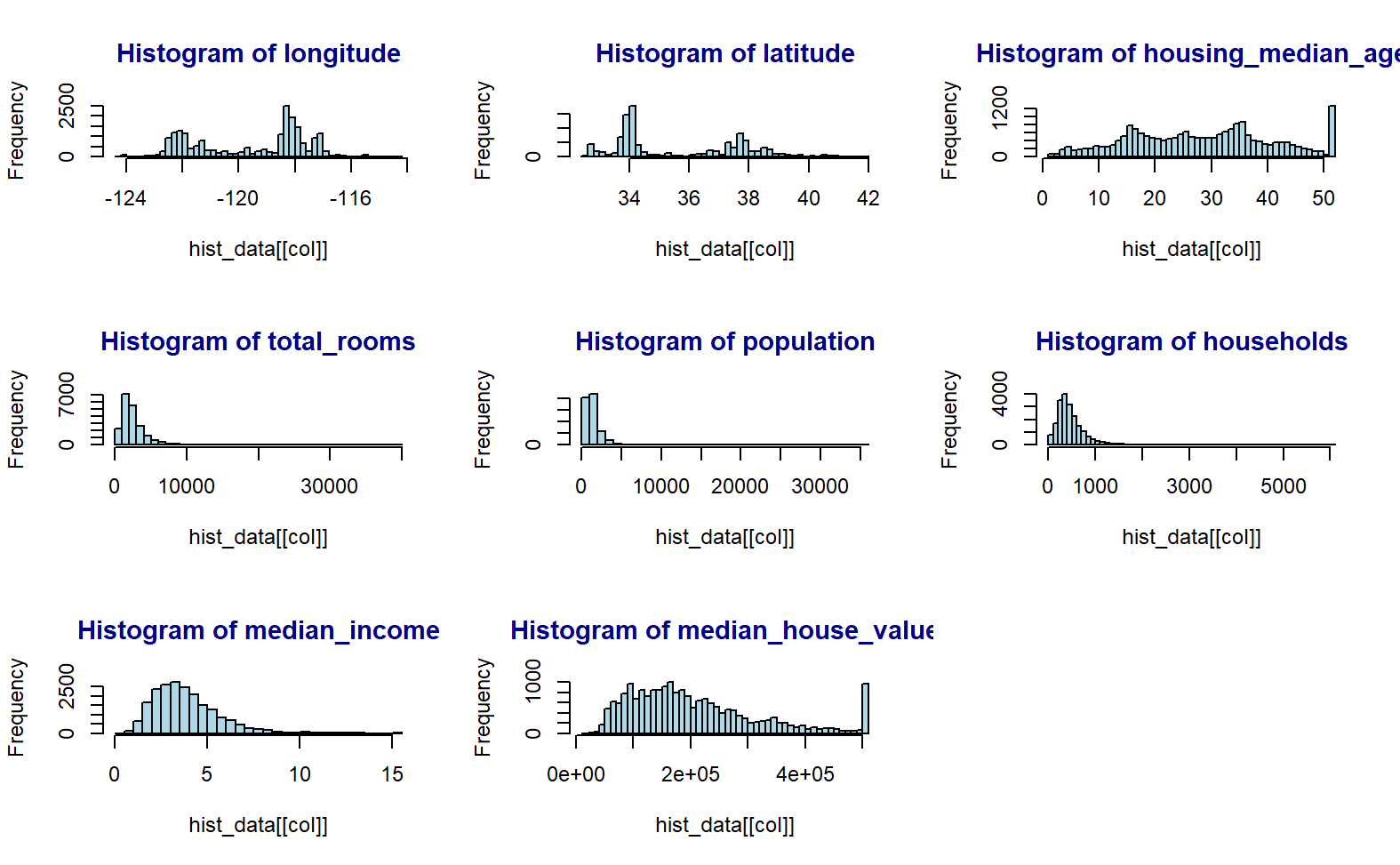
**Histograms for numeric variables:**

par(mfrow = c(3, 3))

for (col in colnames(hist\_data)) {

hist(hist\_data[[col]], breaks = 50, col = "lightblue", main = paste("Histogram of", col), col.main = "darkblue")

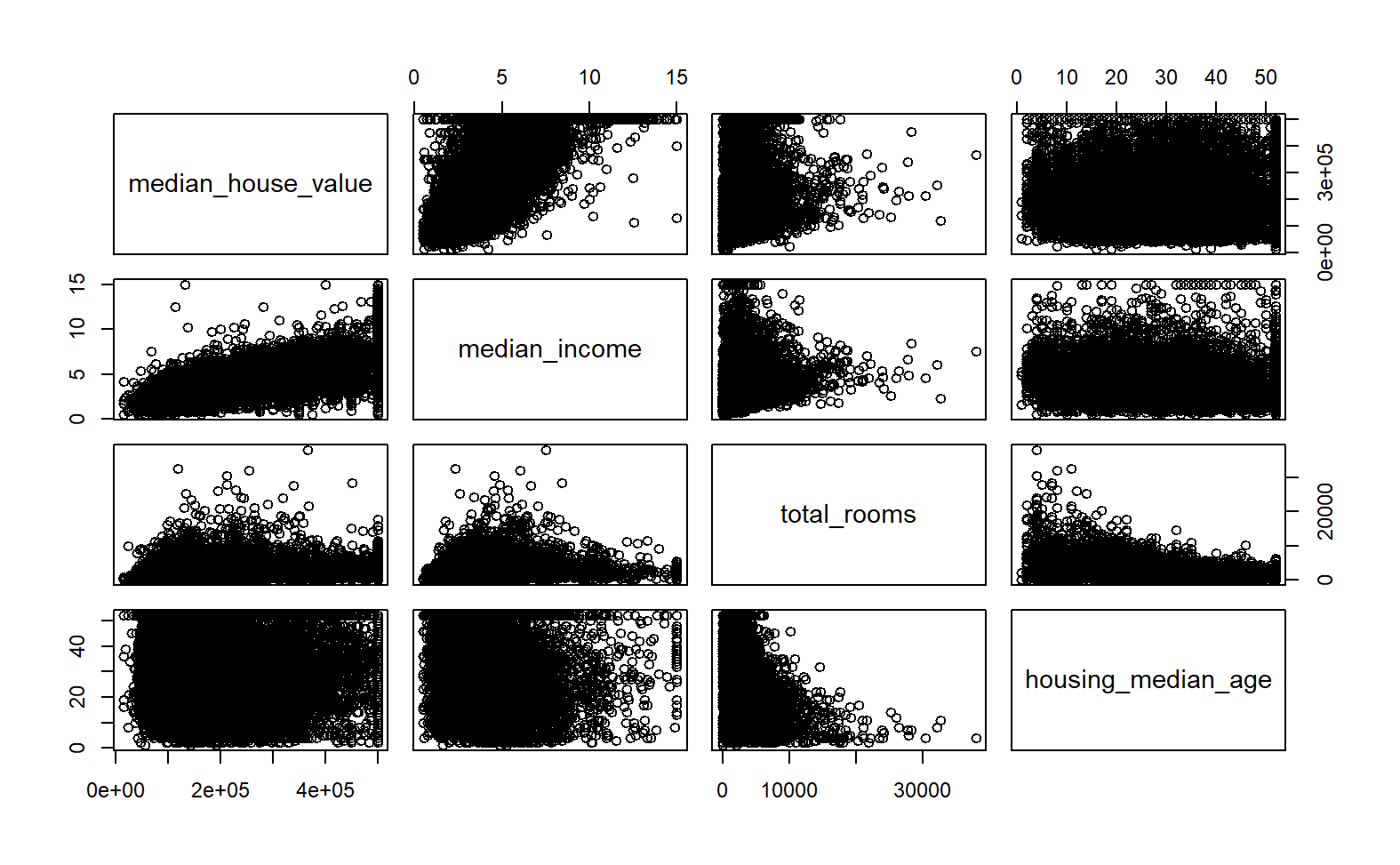
}



**Scatter Matrix and Scatter Plot**

# Scatter matrix

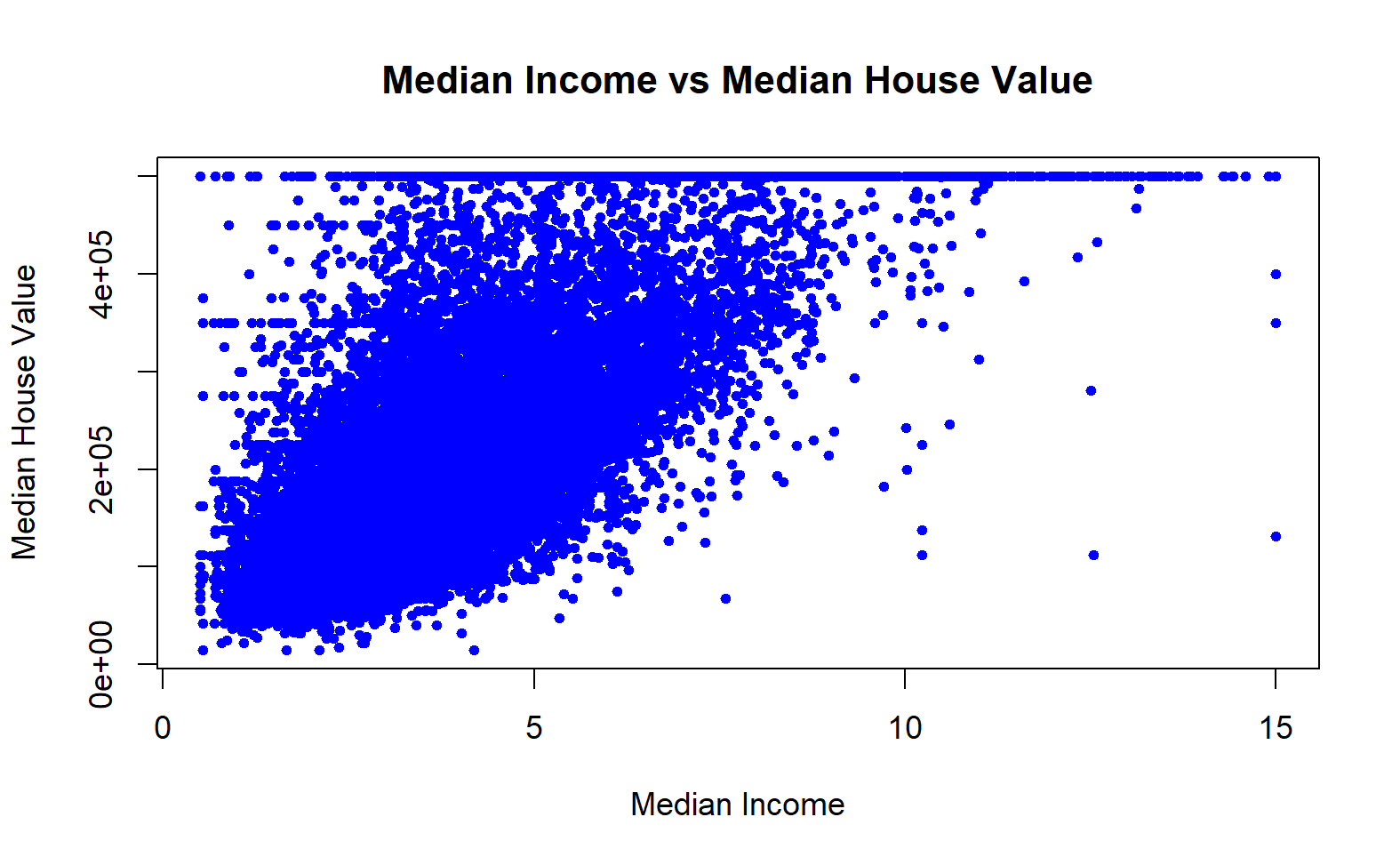
pairs(select(strat\_train\_set, c("median\_house\_value", "median\_income", "total\_rooms", "housing\_median\_age")))



**Scatter Plot**

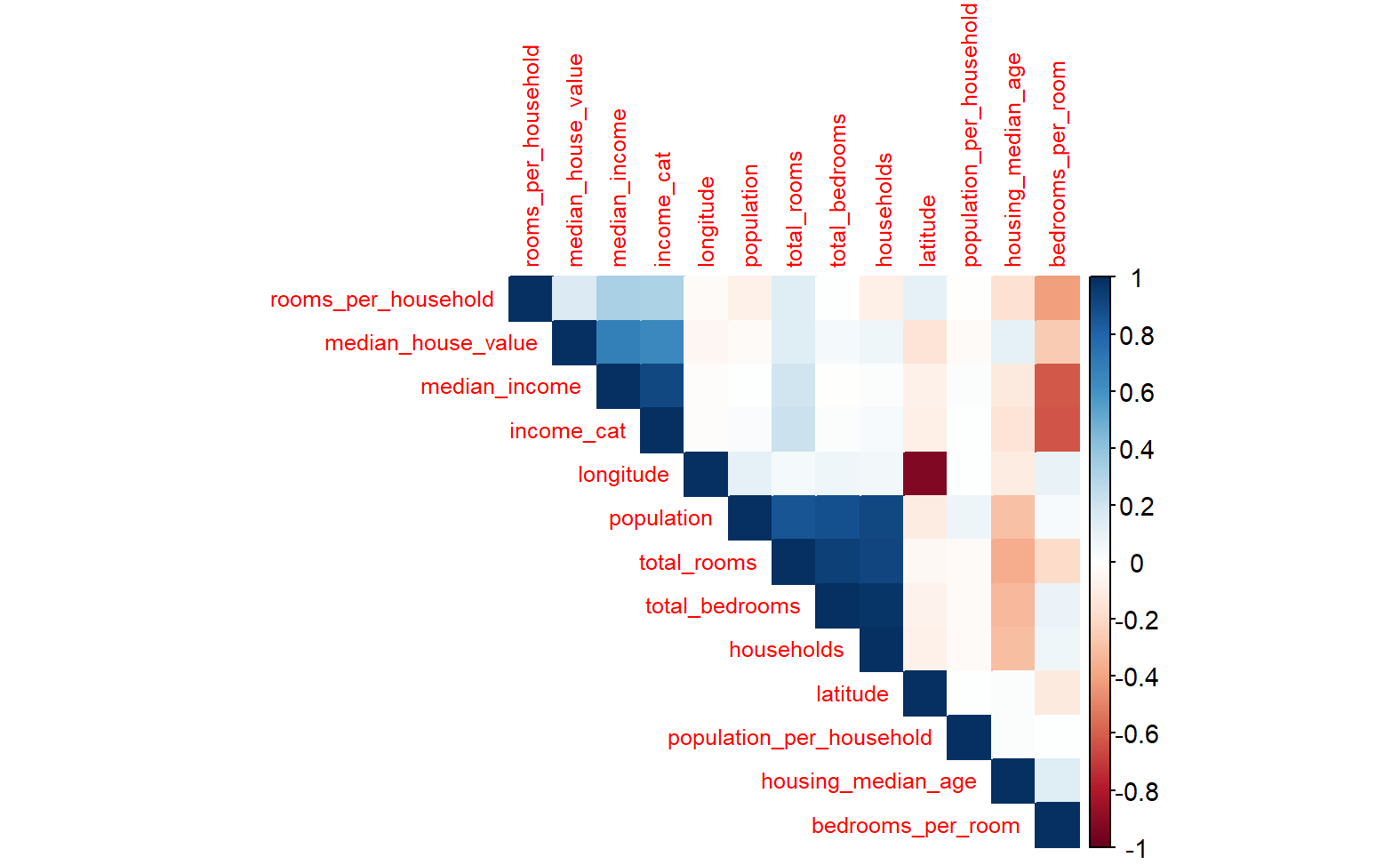
# Scatter plot

plot(housing$median\_income, housing$median\_house\_value, col = "blue", pch = 20, main = "Median Income vs Median House Value", xlab = "Median Income", ylab = "Median House Value", alpha = 0.1)



We further observe the correlation between the numeric variables by checking their correlation coefficients.

| # Correlation matrix  numeric\_columns <- sapply(housing, is.numeric)  numeric\_housing <- housing[, numeric\_columns]  # Compute correlation matrix  corr\_matrix <- round(cor(numeric\_housing), digits = 2)  print(sort(corr\_matrix["median\_house\_value", ], decreasing = TRUE))  plot(corr\_matrix)  corr\_matrix <- cor(numeric\_housing, use = "complete.obs")  corrplot(corr\_matrix, method = "color", type = "upper", order = "hclust", tl.cex = 0.7) |
| --- |



From this correlation matrix, it is observed that there is a high correlation between “households” and “total\_bedrooms”, as well as “households” and “total\_rooms”. This can cause the problem of multicollinearity but since these can be influential in the pricing of homes, we decided to keep the covariates. We can deal with that, if required, using appropriate methods.

### **MODEL TRAINING**

**4.1 Feature engineering**

Feature Engineering is the technique of improving the performance on a dataset by transforming its feature space, and it is the practice of constructing suitable features from given features of the dataset, which leads to improving the performance of the prediction model.

| housing$rooms\_per\_household <- housing$total\_rooms / housing$households  housing$bedrooms\_per\_room <- housing$total\_bedrooms / housing$total\_rooms  housing$population\_per\_household <- housing$population / housing$households  CombinedAttributesAdder <- function(X, add\_bedrooms\_per\_room = TRUE) {  rooms\_per\_household <- X[, "total\_rooms"] / X[, "households"]  population\_per\_household <- X[, "population"] / X[, "households"]    if (add\_bedrooms\_per\_room) {  bedrooms\_per\_room <- X[, "total\_bedrooms"] / X[, "total\_rooms"]  return(cbind(X, rooms\_per\_household, population\_per\_household, bedrooms\_per\_room))  } else {  return(cbind(X, rooms\_per\_household, population\_per\_household))  }  }  DataFrameSelector <- function(X, attribute\_names) {  return(X[, attribute\_names, drop = FALSE])  } |
| --- |

Here we are creating three new features or columns using existing columns and the new columns are:

rooms\_per\_household , population\_per\_household and bedrooms\_per\_room

We get **rooms\_per\_household** by dividing total\_rooms with households.

We get **population\_per\_household** by dividing population with households.

We get **bedrooms\_per\_room** by dividing total\_bedrooms with total\_rooms.  
  
We now add these columns to the existing dataset which will help us in improving the performance of predicting model.

**4.1.1 Imputation**

Missing value imputation is one of the biggest challenges encountered by the data scientist. In addition, most machine learning algorithms are not powerful enough to handle missing data. Missing data can lead to ambiguity, misleading conclusions, and results .

| if (any(**is.na**(hist\_data))) {  hist\_data <- hist\_data[, !apply(is.na(hist\_data), 2, any)]  } |
| --- |

This part of code ensures that it only keeps those columns for which there are no missing values.

| corr\_matrix <- cor(numeric\_housing, use = "**complete.obs**") |
| --- |

"**complete.obs**" argument is used to handle missing values by excluding any pair of observations that have missing values in any of the variables or features.

**4.1.2** **Binning**

Binning is a technique proposed to reduce the impact of statistical noise, to prevent overfitting, reduce overall complexity and make the model more robust. An interval with all observed values is split into smaller sub-intervals, bins, or groups. Also, binning can be considered as a form of discretisation, which is a technique to cut a continuous value range into a finite number of sub-ranges, where a categorical value is associated with each of them.

| housing$income\_cat <- ceiling(housing$median\_income / 1.5)  housing$income\_cat[housing$income\_cat > 5] <- 5 |
| --- |

Here in the first line of code we are dividing the median\_income by 1.5 and take the ceiling, effectively creating discrete bins for income values.

In the second line, we are performing an operation to **cap or limit the values of a variable**, and it's done here to ensure that all values in the income\_cat column are at most 5.If the value in the income\_cat column is greater than 5, we are setting it to 5.

**4.1.3 One-hot Encoding**

One-hot encoding is a technique that is used to convert categorical features to a suitable format to be used as an input in Machine Learning algorithms. It transforms a single variable with 𝑛 observations and 𝑑 distinct values to 𝑑 binary variables, where each observation indicating the presence as 1 or absence as 0.

| encoded\_cat <- dummyVars(~ ., data = cat\_vars)  cat\_vars\_prepared <- predict(encoded\_cat, newdata = cat\_vars) |
| --- |

The **dummyVars** function is used to create dummy variables for the categorical variable "**ocean\_proximity**." The resulting dummy variables are then combined with the numeric variables to create the **housing\_prepared** dataset, which is used in further analysis and modeling.

**4.1.4 Feature Selection**

Feature Selection is an important technique that is used to handle high-dimensional input data and overfitting caused by a curse of dimensionality by selecting a relevant feature subset based on mutual information criterion. Moreover, feature selection has many advantages, such as improving the prediction performance by reducing dimensionality in the dataset. It speeds up the learning process and leads to a better understanding of the considered problem

| tree\_model <- rpart(median\_house\_value ~ ., data = housing\_prepared) |
| --- |

In this step, a decision tree model (tree\_model) is trained using the features in housing\_prepared. The decision tree inherently selects features based on their importance in predicting the target variable (median\_house\_value).

**4.2 Evaluation Metrics**

Several evaluation metrics measure the performance of machine learning algorithms such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R-Squared, and Mean Absolute Error (MAE). However, in this study, the performance of the algorithms is measured by using RMSE and R-Squared.

Root Mean Square Error (RMSE) is used as an evaluation metric in machine learning to

measure the performance of the model. However, RMSE is similar to the Mean Square Error (MAE). Where all errors in MAE have the same weight, but RMSE penalizes the variance, which means it gives more weight to the errors that have large absolute values than those that have small absolute values.

**4.2.1 Evaluation Metrics of Linear Regression:**

Call:

lm(formula = median\_house\_value ~ ., data = housing\_prepared)

Residuals:

Min 1Q Median 3Q Max

-516333 -44007 -10368 29957 718964

Coefficients: (1 not defined because of singularities)

Estimate Std. Error t value Pr(>|t|)

(Intercept) 226011.9 1787.8 126.416 < 2e-16 \*\*\*

longitude -51822.0 2431.8 -21.310 < 2e-16 \*\*\*

latitude -52483.5 2551.3 -20.571 < 2e-16 \*\*\*

housing\_median\_age 13728.8 658.7 20.841 < 2e-16 \*\*\*

total\_rooms -15075.8 2029.1 -7.430 1.15e-13 \*\*\*

total\_bedrooms 33624.0 2892.7 11.624 < 2e-16 \*\*\*

population -40592.3 1395.0 -29.097 < 2e-16 \*\*\*

households 26726.9 2887.2 9.257 < 2e-16 \*\*\*

median\_income 63892.5 1347.2 47.427 < 2e-16 \*\*\*

income\_cat 12150.1 1362.9 8.915 < 2e-16 \*\*\*

`ocean\_proximity<1H OCEAN` -7115.7 1875.0 -3.795 0.000148 \*\*\*

ocean\_proximityINLAND -46507.5 2674.6 -17.389 < 2e-16 \*\*\*

ocean\_proximityISLAND 147528.4 34340.0 4.296 1.75e-05 \*\*\*

`ocean\_proximityNEAR BAY` -10294.0 2593.1 -3.970 7.23e-05 \*\*\*

`ocean\_proximityNEAR OCEAN` NA NA NA NA

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 68560 on 14437 degrees of freedom

Multiple R-squared: 0.6486, Adjusted R-squared: 0.6483

F-statistic: 2050 on 13 and 14437 DF, p-value: < 2.2e-16

**Evaluation Metrics of Random Forest:**

Call:

randomForest(formula = median\_house\_value ~ ., data = housing\_prepared, ntree = 100)

Type of random forest: regression

Number of trees: 100

No. of variables tried at each split: 4

Mean of squared residuals: 2655370925

% Var explained: 80.13

**Evaluation Metrics of Decision Tree:**

n= 16514

Call:

rpart(formula = median\_house\_value ~ ., data = housing\_prepared)

n= 16514

CP nsplit rel error xerror xstd

1 0.30705076 0 1.0000000 1.0001495 0.011804247

2 0.13073484 1 0.6929492 0.6967018 0.009255122

3 0.05656165 2 0.5622144 0.5647159 0.007832996

4 0.04169229 3 0.5056528 0.5097150 0.007751995

5 0.01582424 4 0.4639605 0.4683897 0.007371186

6 0.01361468 5 0.4481362 0.4528117 0.007231778

7 0.01337994 6 0.4345215 0.4422296 0.007171882

8 0.01000000 7 0.4211416 0.4231555 0.007011311

Variable importance

median\_income ocean\_proximityINLAND latitude ocean\_proximity.1H.OCEAN

59 20 7 6

longitude housing\_median\_age total\_rooms population

4 2 1 1

Node number 1: 16514 observations, complexity param=0.3070508

mean=207166.1, MSE=1.339791e+10

left son=2 (13092 obs) right son=3 (3422 obs)

Primary splits:

median\_income < 0.6345238 to the left, improve=0.30705080, (0 missing)

ocean\_proximityINLAND < 0.5 to the right, improve=0.23795160, (0 missing)

latitude < 1.078987 to the right, improve=0.06611940, (0 missing)

ocean\_proximity.1H.OCEAN < 0.5 to the left, improve=0.06444780, (0 missing)

longitude < -1.14316 to the right, improve=0.04114705, (0 missing)

Surrogate splits:

total\_rooms < 5.329399 to the left, agree=0.794, adj=0.006, (0 split)

total\_bedrooms < 11.08626 to the left, agree=0.793, adj=0.000, (0 split)

population < 9.536096 to the left, agree=0.793, adj=0.000, (0 split)

households < 11.74446 to the left, agree=0.793, adj=0.000, (0 split)

Node number 2: 13092 observations, complexity param=0.1307348

mean=174374.6, MSE=8.529763e+09

left son=4 (4712 obs) right son=5 (8380 obs)

Primary splits:

ocean\_proximityINLAND < 0.5 to the right, improve=0.25902270, (0 missing)

median\_income < -0.4194221 to the left, improve=0.16493370, (0 missing)

ocean\_proximity.1H.OCEAN < 0.5 to the left, improve=0.07387965, (0 missing)

latitude < 1.074305 to the right, improve=0.06174550, (0 missing)

longitude < -1.14316 to the right, improve=0.03837117, (0 missing)

Surrogate splits:

ocean\_proximity.1H.OCEAN < 0.5 to the left, agree=0.765, adj=0.347, (0 split)

latitude < 1.088351 to the right, agree=0.729, adj=0.247, (0 split)

housing\_median\_age < -1.038968 to the left, agree=0.666, adj=0.072, (0 split)

longitude < 1.325904 to the right, agree=0.661, adj=0.059, (0 split)

population < -0.9570791 to the left, agree=0.648, adj=0.022, (0 split)

Node number 3: 3422 observations, complexity param=0.05656165

mean=332620.7, MSE=1.216995e+10

left son=6 (2456 obs) right son=7 (966 obs)

Primary splits:

median\_income < 1.64399 to the left, improve=0.30049890, (0 missing)

ocean\_proximityINLAND < 0.5 to the right, improve=0.12577810, (0 missing)

housing\_median\_age < -0.08950014 to the left, improve=0.08359322, (0 missing)

latitude < 1.093033 to the right, improve=0.05638960, (0 missing)

longitude < 0.7633957 to the right, improve=0.04368953, (0 missing)

Surrogate splits:

population < -1.204901 to the right, agree=0.721, adj=0.011, (0 split)

households < -1.194643 to the right, agree=0.721, adj=0.011, (0 split)

total\_rooms < -1.133595 to the right, agree=0.720, adj=0.009, (0 split)

total\_bedrooms < -1.210915 to the right, agree=0.720, adj=0.009, (0 split)

Node number 4: 4712 observations, complexity param=0.01337994

mean=111690.6, MSE=2.737939e+09

left son=8 (2702 obs) right son=9 (2010 obs)

Primary splits:

median\_income < -0.4418594 to the left, improve=0.22946410, (0 missing)

latitude < -0.3771018 to the right, improve=0.04093470, (0 missing)

total\_rooms < -0.1006974 to the left, improve=0.03617024, (0 missing)

longitude < 0.3925382 to the left, improve=0.03022733, (0 missing)

housing\_median\_age < -0.8807235 to the right, improve=0.02425602, (0 missing)

Surrogate splits:

housing\_median\_age < -1.118091 to the right, agree=0.629, adj=0.130, (0 split)

total\_rooms < 0.02373412 to the left, agree=0.622, adj=0.113, (0 split)

population < 0.392417 to the left, agree=0.594, adj=0.048, (0 split)

households < 0.8572581 to the left, agree=0.592, adj=0.043, (0 split)

total\_bedrooms < 1.037149 to the left, agree=0.589, adj=0.037, (0 split)

Node number 5: 8380 observations, complexity param=0.04169229

mean=209621.3, MSE=8.334724e+09

left son=10 (3598 obs) right son=11 (4782 obs)

Primary splits:

median\_income < -0.4032525 to the left, improve=0.13207180, (0 missing)

longitude < 0.6289909 to the right, improve=0.07999800, (0 missing)

latitude < -0.7657044 to the left, improve=0.03873547, (0 missing)

housing\_median\_age < 1.809436 to the left, improve=0.03541905, (0 missing)

total\_rooms < -0.09116149 to the left, improve=0.02774434, (0 missing)

Surrogate splits:

total\_rooms < -0.7042481 to the left, agree=0.604, adj=0.078, (0 split)

latitude < -1.336903 to the left, agree=0.590, adj=0.045, (0 split)

longitude < -1.680779 to the left, agree=0.581, adj=0.025, (0 split)

households < -0.8781542 to the left, agree=0.576, adj=0.013, (0 split)

population < -0.9553058 to the left, agree=0.575, adj=0.010, (0 split)

Node number 6: 2456 observations, complexity param=0.01361468

mean=294694.4, MSE=9.020376e+09

left son=12 (410 obs) right son=13 (2046 obs)

Primary splits:

ocean\_proximityINLAND < 0.5 to the right, improve=0.13597030, (0 missing)

housing\_median\_age < 0.6226009 to the left, improve=0.10436790, (0 missing)

median\_income < 1.00811 to the left, improve=0.08448400, (0 missing)

latitude < 1.088351 to the right, improve=0.05317584, (0 missing)

longitude < -1.262631 to the right, improve=0.04836764, (0 missing)

Surrogate splits:

latitude < 1.266266 to the right, agree=0.864, adj=0.188, (0 split)

longitude < 1.470265 to the right, agree=0.838, adj=0.027, (0 split)

housing\_median\_age < -2.067559 to the left, agree=0.837, adj=0.022, (0 split)

Node number 7: 966 observations

mean=429046.1, MSE=7.222639e+09

Node number 8: 2702 observations

mean=90072.17, MSE=1.451299e+09

Node number 9: 2010 observations

mean=140751.8, MSE=2.994729e+09

Node number 10: 3598 observations

mean=171371.9, MSE=6.158443e+09

Node number 11: 4782 observations, complexity param=0.01582424

mean=238400.3, MSE=8.043154e+09

left son=22 (1982 obs) right son=23 (2800 obs)

Primary splits:

longitude < 0.6439248 to the right, improve=0.09102827, (0 missing)

housing\_median\_age < 1.492947 to the left, improve=0.06687641, (0 missing)

latitude < -0.7703863 to the left, improve=0.04961092, (0 missing)

median\_income < 0.1367957 to the left, improve=0.02804605, (0 missing)

ocean\_proximity.1H.OCEAN < 0.5 to the right, improve=0.02378603, (0 missing)

Surrogate splits:

latitude < -0.695475 to the left, agree=0.882, adj=0.715, (0 split)

ocean\_proximity.1H.OCEAN < 0.5 to the right, agree=0.602, adj=0.041, (0 split)

housing\_median\_age < -1.276335 to the left, agree=0.595, adj=0.024, (0 split)

population < -1.201797 to the left, agree=0.587, adj=0.003, (0 split)

median\_income < 0.6308896 to the right, agree=0.587, adj=0.003, (0 split)

Node number 12: 410 observations

mean=216460.5, MSE=6.196189e+09

Node number 13: 2046 observations

mean=310371.8, MSE=8.114035e+09

Node number 22: 1982 observations

mean=206239.4, MSE=5.026366e+09

Node number 23: 2800 observations

mean=261165.7, MSE=8.928193e+09

**4.3 Model Selection**

From the three models we have used we are going to select Random Forest as the performance of the Random Forest is better than of the other models.

**4.4 Test Train Split**

In order to train our models and further test for measuring the accuracy, we first split our dataset into 80% for training and 20% for testing set respectively.

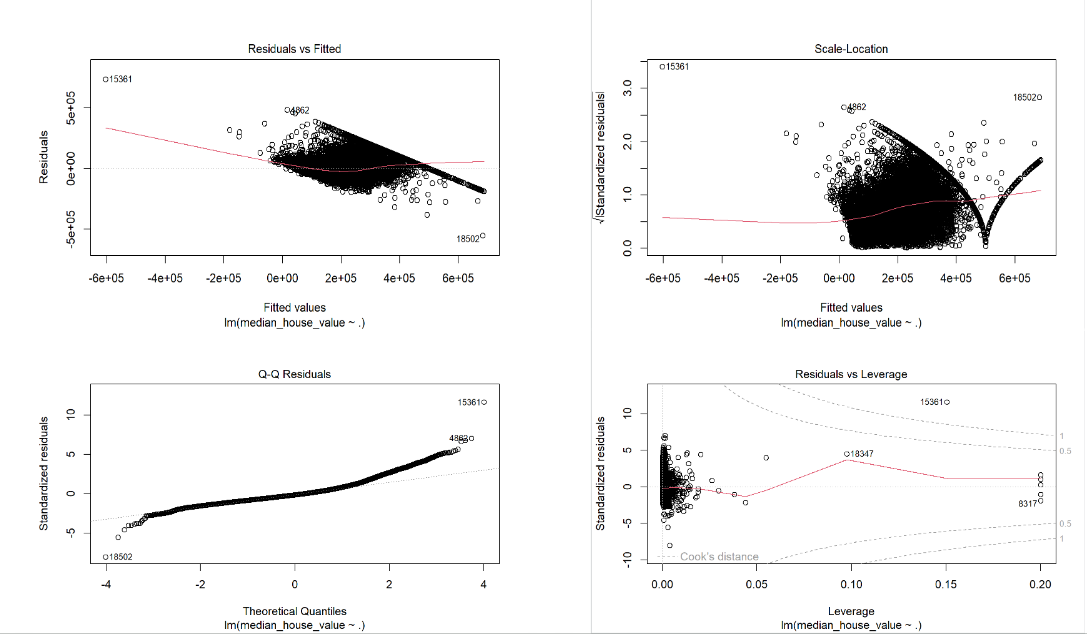
| set.seed(42)  split <- createDataPartition(housing$income\_cat, p = 0.7, list = FALSE, times = 1)  strat\_train\_set <- housing[split, ]  strat\_test\_set <- housing[-split, ] |
| --- |

The argument **p = 0.7** indicates that 70% of the data should be used for training, and the rest for testing.

The **times = 1** argument specifies that the partitioning should be done once.

### **Model Validation**

### **5.1 Linear Regression**



### 

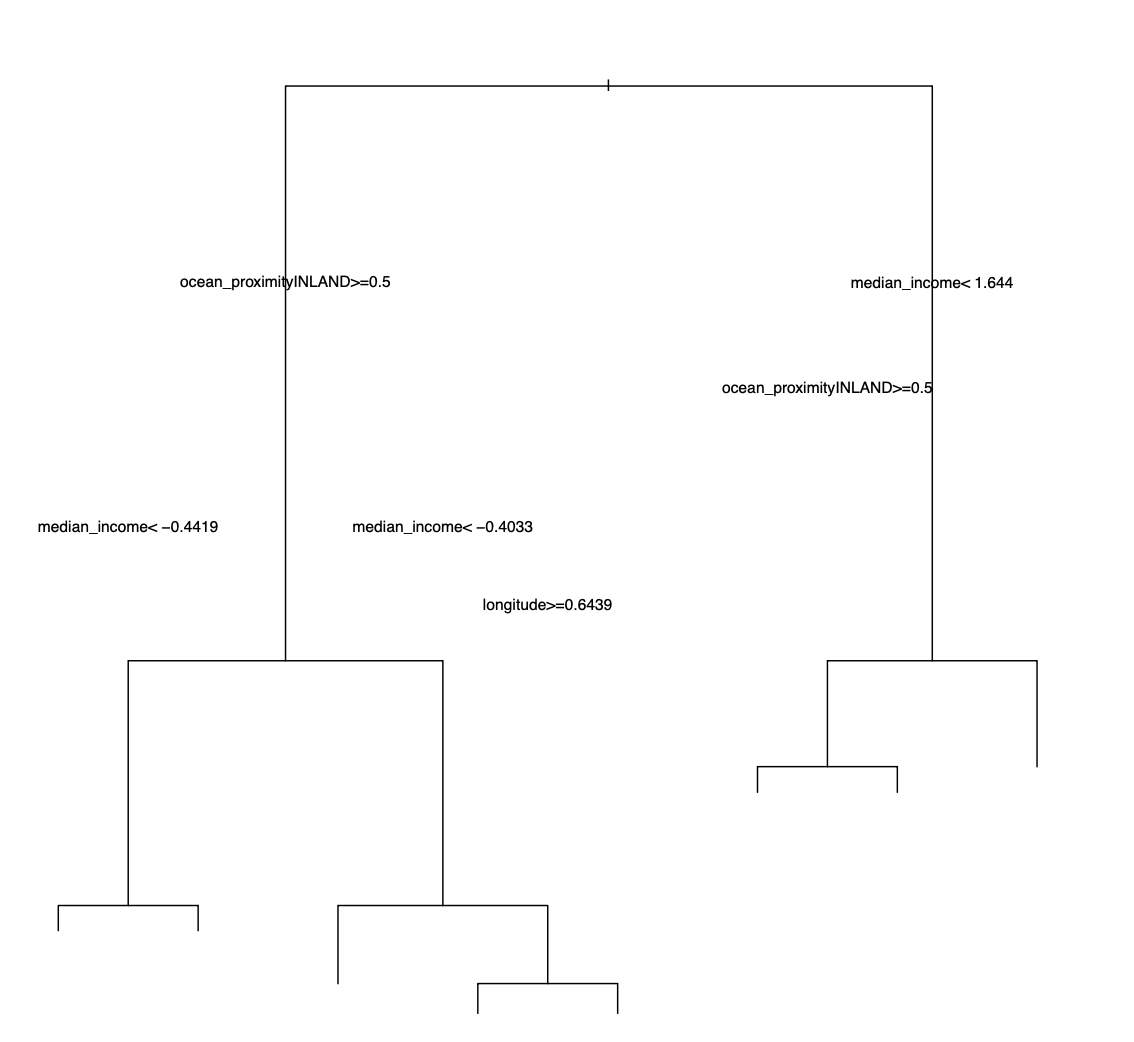
### 

### 

### 

### 

### **5.2 Decision Tree**



### 

### **5.3 Testing Results**

| final\_model <- randomForest(median\_house\_value ~ ., data = housing\_prepared, ntree = 30, max\_features = 8)  X\_test <- subset(strat\_test\_set, select = -c(median\_house\_value))  y\_test <- strat\_test\_set$median\_house\_value  X\_test\_num <- X\_test[, num\_attribs]  X\_test\_cat <- X\_test[, cat\_attribs, drop = FALSE]  X\_test\_num\_prepared <- predict(num\_pipeline, newdata = X\_test\_num)  X\_test\_cat\_prepared <- predict(encoded\_cat, newdata = X\_test\_cat)  # Combine numeric and categorical variables in the test set  X\_test\_prepared <- cbind(X\_test\_num\_prepared, X\_test\_cat\_prepared)  colnames(X\_test\_prepared) <- make.names(colnames(X\_test\_prepared))  final\_predictions <- predict(final\_model, newdata = X\_test\_prepared)  print(final\_predictions)  summary(final\_predictions)  final\_rmse <- sqrt(mean((final\_predictions - y\_test)^2))  print(final\_rmse)  Min. 1st Qu. Median Mean 3rd Qu. Max.  49433 131923 191101 207110 259522 500001  [1] 52207.11 |
| --- |

**Output:**

Min. 1st Qu. Median Mean 3rd Qu. Max.

49433 131923 191101 207110 259522 500001

## [1] 52207.11

**Actual Price vs Predicted Price**

# Compare actual price vs. predicted price

plot\_predictions <- data.frame(Actual = y\_test, Predicted = final\_predictions)

# Scatter plot

plot(plot\_predictions$Actual, plot\_predictions$Predicted,

main = "Actual Price vs. Predicted Price",

xlab = "Actual Price", ylab = "Predicted Price",

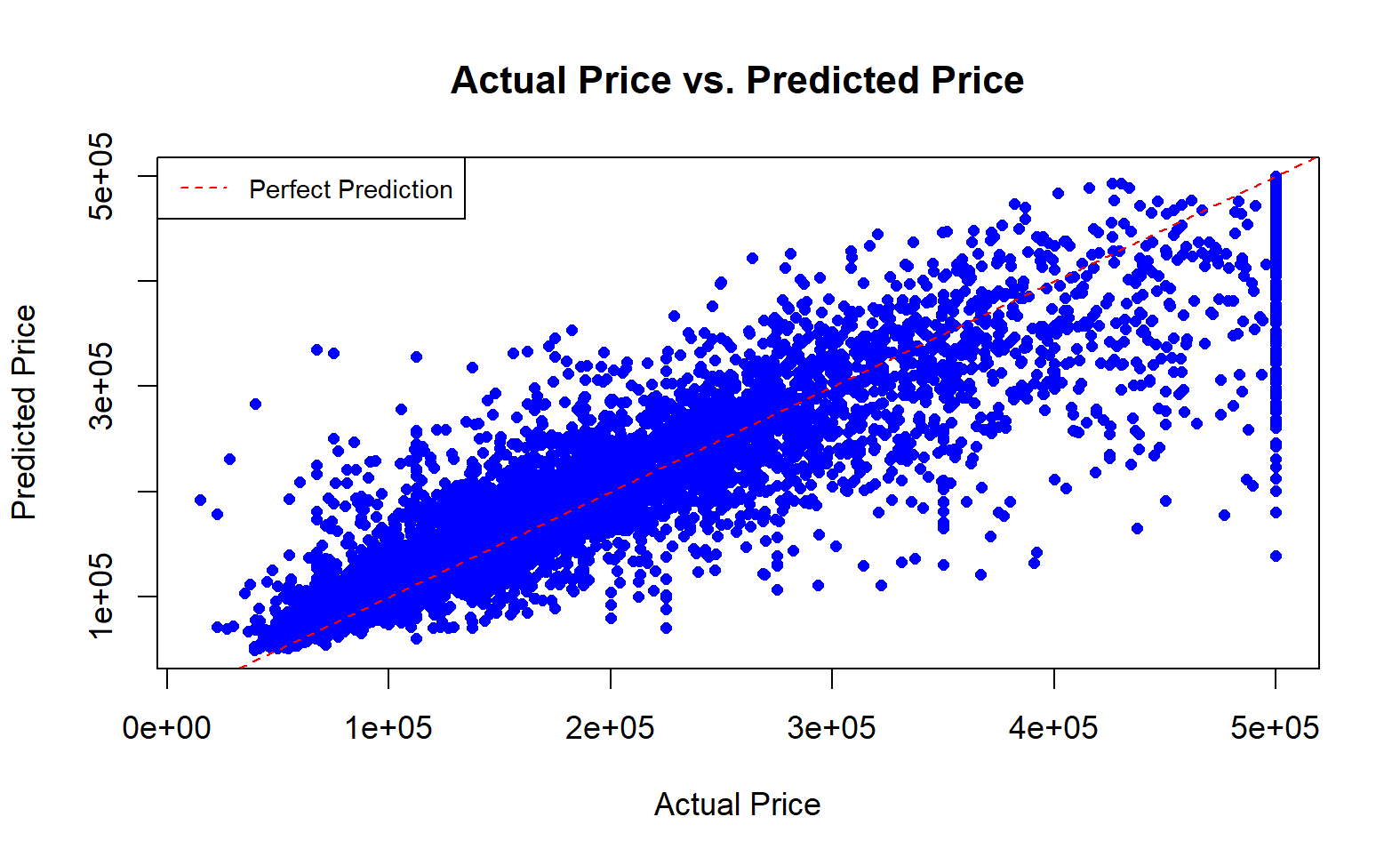
pch = 16, col = "blue", cex = 0.8)

# Add a diagonal line for reference (perfect prediction)

abline(a = 0, b = 1, col = "red", lty = 2)

# Add legend

legend("topleft", legend = "Perfect Prediction", col = "red", lty = 2, cex = 0.8)



### **Conclusion**

In our analysis, we explored three distinct regression models—Linear Regression, Decision Tree, and Random Forest—as tools to predict housing prices based on a variety of features. The Linear Regression model provided valuable insights into the relationships between housing prices and individual features, with coefficients revealing the specific impact of each variable. Notably, latitude, longitude, and median income emerged as the most influential factors. The Decision Tree model, delving into non-linear patterns and feature interactions, identified median income as the pivotal predictor, closely followed by proximity to inland areas. The Random Forest model, an ensemble of decision trees, surpassed individual decision trees by achieving an impressive 80.13% variance explained. Feature importance analysis underscored the significance of median income, proximity to inland areas, and geographic coordinates. Across all models, a consistent trend emerged, highlighting the dominant role of median income in predicting housing prices. Geographic patterns also played a crucial role, with latitude, longitude, and proximity to specific geographic areas, such as inland and near bay, significantly influencing housing values. The superiority of the Random Forest ensemble over the Decision Tree model underscored the effectiveness of ensemble methods in enhancing predictive accuracy. Overall, our exploration provided nuanced insights into the diverse factors shaping housing prices, emphasizing the importance of both individual predictors and ensemble modeling techniques.

### **7.** **FUTURE ENHANCEMENTS**

There are some significant changes which can be brought to the existing model. Currently we can just predict houses using only certain parameters and no other action can be done. There is a possible opening for an opportunity where the model can recommend the house that a customer can buy within their budget range based on their previous searches. The model is now currently limited to predicting houses using certain parameters but we can add more parameters according to customers choice.Using different models or algorithms to predict more accurately.

### **8. DATA SOURCES**

Dataset:

The US Census Bureau has released Census Data for California, which has 20640

records. The sample dataset contains 10 distinct metrics for each Californian block

group, such as population, median income, and median housing price. The median

house value attribute of the dataset will be predicted utilizing the various features as

independent variables.

<https://www.kaggle.com/datasets/camnugent/california-housing-prices?resou>

[rce=download&select=housing.csv%29](https://www.kaggle.com/datasets/camnugent/california-housing-prices?resou)

Reference Data:

Median house prices California districts from the 1990 census

<https://www.kaggle.com/datasets/fedesoriano/california-housing-prices-dataextra->

[Features](https://www.kaggle.com/datasets/fedesoriano/california-housing-prices-dataextra-)

The Boston house-price data of Harrison, D. and Rubinfeld, D.L.

<https://www.kaggle.com/datasets/fedesoriano/the-boston-houseprice-data>

Real Estate listings (900k+) in the US broken by State and zip code

<https://www.kaggle.com/datasets/ahmedshahriarsakib/usa-real-estate-dataset>

Property Prices in United States - Cost of Living

<https://www.kaggle.com/datasets/themrityunjaypathak/property-prices-in-unit>

[ed-states](https://www.kaggle.com/datasets/themrityunjaypathak/property-prices-in-unit)

### **9.** **SOURCE CODE**

Data Manipulation and Exploration:

* ‘caret’ : A comprehensive package for classification and regression training.
* ‘randomForest’ : For creating and analyzing random forests.

Data Loading:

* read.csv: A base R function for reading comma-separated values from a file.

Data Visualization:

* base R plotting functions: Used for creating histograms, scatter plots, and visualizations.

Data Preprocessing:

* ‘createDataPartition’ : From the ‘caret’ package, used for creating data partitions.
* ‘preProcess’ : From the ‘caret' package, used for preprocessing numeric variables.
* ‘dummyVars’ : From the ‘caret’ package, used for dummy encoding categorical variables.

Modeling:

* ‘Lm’ : Base R function for linear regression.
* ‘Rpart’ : For building decision trees.
* ‘randomForest’ : For creating random forest models.

Model Evaluation:

* ‘train’ : From the ‘caret’ package, used for model training and grid search.
* ‘trainControl’ : From the ‘caret’ package, used for controlling the training process.

Final Model and Evaluation on Test Set:

* ‘randomForest’ : Used to train the final model.
* Various operations for preparing the test set and evaluating the final model on it.

Other Functions:

* Customfunctions(‘CombinedAttributesAdder’,‘DataFrameSelector’ , ‘MyLabelBinarizer’ ) for specific data transformations.

Dependency Management:

* The code is dependent on external data, presumably in a file named "housing.csv."

### **10.** **LITERATURE REVIEW**

1. <https://medium.com/@ageitgey/machine-learning-is-fun-80ea3ec3c471>
2. [https://www.sas.com/en\_us/insights/analytics/machinelearning.html#machine-learning-importance](https://medium.com/@ageitgey/machine-learning-is-fun-80ea3ec3c471)
3. <http://www.wired.co.uk/article/machine-learning-ai-explained>
4. <https://deeplearning4j.org/ai-machinelearning-deeplearning>
5. David E. Rapach , Jack K. Strauss “ Forecasting real housing price growth in the Eighth District states”
6. Vasilios Plakandaras+ and Theophilos , Rangan Gupta\*, Periklis Gogas “Forecasting the U.S. Real House Price Index”
7. Gupta and Das (2010) Forecasting the US Real House Price Index: Structural and Non-Structural Models with and without Fundamentals
8. Rangan Gupta “Forecasting US real house price returns over 1831–2013: evidence from copula models
9. Shahasane, A., Gosavi, M., Bhagat, A., Mishra, N., & Nerurkar, A. (2023, April 4).House Price Prediction Using Machine Learning. www.irjet.net; IRJET.<https://www.irjet.net/archives/V10/i4/IRJET-V10I4194.pdf.> The paper is about predicting real estate prices using machine learning algorithms and techniques. It considers various factors that affect house prices, such as area, location, population, size, number of bedrooms and bathrooms, parking space, elevator, etc.The paper uses a dataset from a reputed website to perform data analysis and apply linear regression and sklearn models to increase the accuracy. It also covers data cleaning, outlier removal, feature engineering, dimensionality reduction, gridsearchcv for hyperparameter tuning, k-fold cross-validation, etc.The paper aims to build a house price prediction system with a user-friendly front-end that will help users choose their desired destination and get an idea about the price rates.
10. Eltanani, S. (2022). Combining Machine Learning models to predict House Prices.[www.solent.ac.uk](http://www.solent.ac.uk);SOLENT. <https://www.solent.ac.uk/documents/degree-shows/isaac-ake-project-scaids.pdf.> This paper states the critical importance of accurate house price valuation in real estate decision-making. It emphasizes the role of predictive models and factors influencing property price changes. Numerical property characteristics and spatial data are discussed, highlighting their impact on price predictions. The paper also emphasizes the development of a software tool to enhance house price prediction algorithms. It suggests that the housing market could benefit from improved mechanisms for projecting house values, with regression methods and machine learning techniques being potential options. The study uses the California House Price Prediction data and employs various regression models, concluding that the Random Forest Regression Model is the most suitable for predicting housing prices.

### **11 Project Repository**

[Link](https://drive.google.com/drive/folders/1rp0pG7ELJBivdTgy73QWKAq6pEX4zM4d?usp=drive_link)

URL: : <https://drive.google.com/drive/folders/1rp0pG7ELJBivdTgy73QWKAq6pEX4zM4d?usp=drive_link>