[School Logo]

[School Name]

[Project title]

[Couse name and code]

[Student Name]

[Student Registration Number]

Contents

1 Introduction 3
1.1 Project background 3
1.2 Project Goals and deliverables 3
1.3 Tools and technologies applied 3
2 Purpose Statement 3
3 Methodology 4
3.1 About the dataset 4
3.2 Exploratory Data Analysis 5
3.2.1 A quick look at the data structure 5
3.2.2 Explore and visualize the data to gain insights 7
3.3 Data preparation with Weka 12
3.3.1 Feature combination 12
3.3.2 Data cleaning 13
3.3.3 Encoding categorical variables 14
3.3.4 Feature Scaling and Transformation 14
3.3.5 Splitting the data 14
3.4 Algorithm selection and Data mining 14
3.4.1 Decision tree J48 in Weka Classification algorithm 14
3.4.2 Random Forest Classification algorithm 18
3.4.3 Linear regression 20

1 Introduction

1.1 Project background

In today's world, customers are faced with multiple choices for every decision. Assuming a person is looking for a movie to watch without any specific idea of what they want. There is a wide range of possibilities for how their search might pan out. They might waste a lot of time browsing the internet and trawling through various sites hoping to strike gold. They might look for recommendations from other people.

Just as a person might benefit from a recommendation system to quickly find a movie that suits their tastes, a real estate agency can utilize data-driven approaches to streamline the decision-making process for home buyers and investment. By leveraging the California housing prices dataset from Statlib repository. This project aims to provide a real estate agency with personalized prediction systems, that aims to predict the median housing price of a districts give other data about the district

1.2 Project Goals and deliverables

The business objective is to accurately predict a district's median housing price. This median housing prices is essential in determining whether it is worth investing in a given area. Getting this right, it is critical as it directly affects revenue.

Given that most of the real estate agency current solution; to estimate the housing prices is done manually by experts: a team gathers up to date information about a district and when they cannot get the median housing price, they estimate it using complex rules. This is costly and time consuming and their estimates are bound not to be so great. Therefore, the business objective is to train a model to predict a district's median housing price, given other data about that district.

1.3 Tools and technologies applied

The execution of this project has been performed using some main tools: Python, Weka and Jupyter notebook. The models have been executed on Weka. The exploratory data analysis and data visualization has been executed on Jupyter notebook.

Several packages have been used to perform the initial EDA. This included a combination of Python libraries like matplotlib and seaborn for data visualization. Packages like Numpy and Pandas have been used for data wrangling and manipulation. For the models inbuild models in Weka have been used.

2 Purpose Statement

The purpose of this study is to predict a districts median housing price. Looking at the business objective and dataset we can categorize this task as a supervised learning task. The assumption made is that the districts demographic characteristics and houses this population live inform the median housing prices in that specific district.

Given that this is an estimation task. The performance measure for regression problems is typically root mean square error (RMSE). It gives an idea of how much error the system typically makes in its predictions with a higher weight given to large errors

3 Methodology

This report is composed o several different components that explore various aspects of the California housing dataset. This section contains information about the dataset, the exact techniques employed in EDA, the data mining techniques used to classify, numeric prediction. It also discusses the success applied of the data mining techniques.

3.1 About the dataset

The dataset for this project has been acquired from a repository on GitHub from the following link https://github.com/Ashuku001/sales-dataset/tree/3bbe5894ebea69a09445f9a6bfe8ee93af0c5aed/datasets. The data includes metrics such as longitude, latitude, housing median age, total rooms, total bedrooms, population, households, median income, median house value, ocean proximity for each block group in California. A brief description about the dataset is as shown below

Column Name	Description
longitude	The longitudinal coordinate of the property.
latitude	The latitudinal coordinate of the property.
housing_median_age	The median age of houses in the area (in years).
total_rooms	The total number of rooms in all houses within the block.
total_bedrooms	The total number of bedrooms in all houses within the block.
population	The total population within the block.
households	The total number of households within the block.
median_income	The median income of households within the block
median_house_value	The median value of houses within the block (in dollars).
ocean_proximity	The proximity of the block to the ocean (e.g., "Near Ocean," "Inland").

Table 1 Attribute descriptions

There are 10 attributes in total from this dataset.

Column	Non-Null Count	percentage missing	Info
longitude	20640	0	float64
latitude	20640	0	float64
housing_median_age	20640	0	float64
total_rooms	20640	0	float64
total_bedrooms	20640	1.0029	float64
population	20640	0	float64
households	20640	0	float64
median_income	20640	0	float64
median_house_value	20640	0	float64
ocean_proximity	20640	0	string

3.2 Exploratory Data Analysis

It is important to have an in-depth understanding of the dataset used in this project analysis to have an idea of the models that would give the best most accurate results, and promising data transformation that would improve models' performance. This thorough examination is necessary to understand the underlying structure of the dataset and to draw insights about the validity of our analysis and how well the analysis responds to the business objective.

The study begins with a brief analysis of the available dataset to get a sense of the main characteristics or attributes that are relevant to the project's end goal. Considering the numerous attributes an exploratory data analysis is necessary to study the features of the attributes and their relevance to the study of making predictions and draw customer insights. As part of the EDA, visualizations have been included that will help us understand the various attributes that we can use to improve results.

Several packages have been used to create visualizations that provide information about product and customer insights inform of heatmaps, correlation matrix, histograms, scatterplots and etcetera. Even though Weka has data visualization it is limited therefore the packages include; matplotlib, seaborn, Numpy and Pandas on Jupyter notebook. The visualizations are accompanied by brief descriptions that will discuss the findings and scope of potential modelling and transformation that can will be performed in the next stages of the analysis.

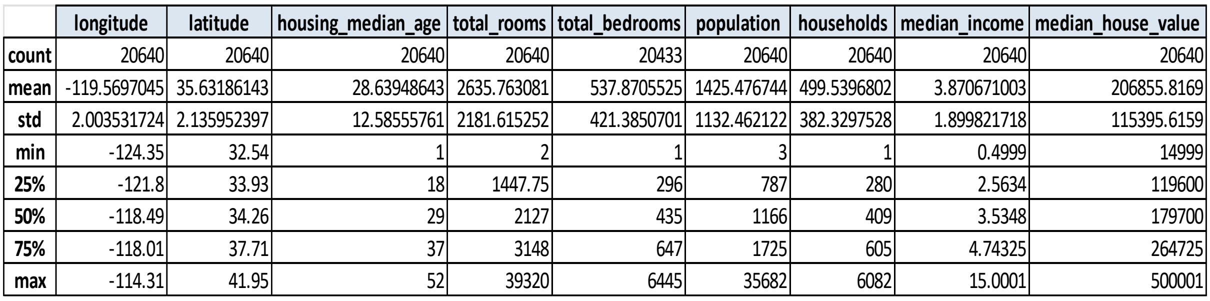
3.2.1 A quick look at the data structure

For this study it was necessary to get a sense of various aspects of the datasets before they are used to create complex models. The python packages manipulate, compares and visualizes different aspects of the datasets. The sales dataset has the following information relevant to this study.

Column	Non-Null Count	percentage missing	Info
longitude	20640	0	float64
latitude	20640	0	float64
housing_median_age	20640	0	float64
total_rooms	20640	0	float64
total_bedrooms	20640	1.0029	float64
population	20640	0	float64
households	20640	0	float64
median_income	20640	0	float64
median_house_value	20640	0	float64
ocean_proximity	20640	0	string

Table 2 Size, Missing data and data types

All attributes are numerical except for ocean_proximity which is type text. Looking at the summary of the numerical attributes.



	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
count	20640	20640	20640	20640	20433	20640	20640	20640	20640
mean	-119.5697045	35.63186143	28.63948643	2635.763081	537.8705525	1425.476744	499.5396802	3.870671003	206855.8169
std	2.003531724	2.135952397	12.58555761	2181.615252	421.3850701	1132.462122	382.3297528	1.899821718	115395.6159
min	-124.35	32.54	1	2	1	3	1	0.4999	14999
25%	-121.8	33.93	18	1447.75	296	787	280	2.5634	119600
50%	-118.49	34.26	29	2127	435	1166	409	3.5348	179700
75%	-118.01	37.71	37	3148	647	1725	605	4.74325	264725
max	-114.31	41.95	52	39320	6445	35682	6082	15.0001	500001

Table 3 Summary of the numerical attributes

The count gives the total non-null values for each column. Std gives the standard deviation which measure how dispersed the values are. Th 25%, 50% and 75% rows show the corresponding percentiles. The percentile indicates the value below which a give percentage of observations in a group of observation fall. For example, with a percentile of 25% of the districts have a housing_median_age lower than 18, while 50% are lower than 29 and 75% are lower than 37.

To visualize the data distribution, we plot the numerical attributes using a histogram that shows the number of instances on the vertical axis that have a given range on the horizontal axis.

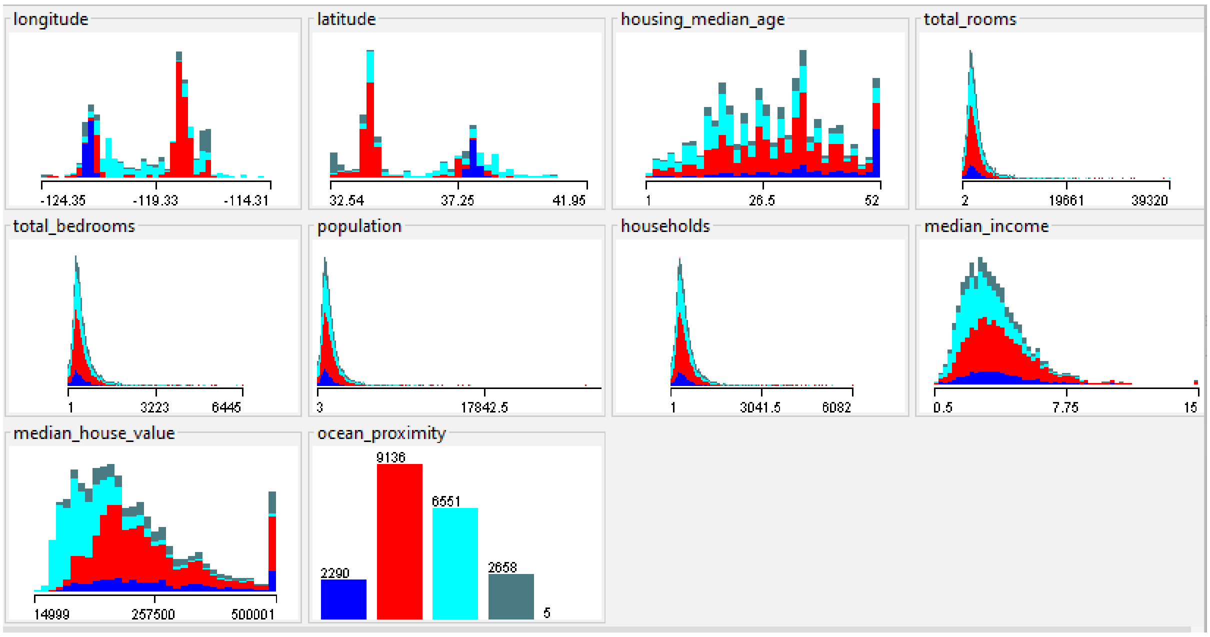
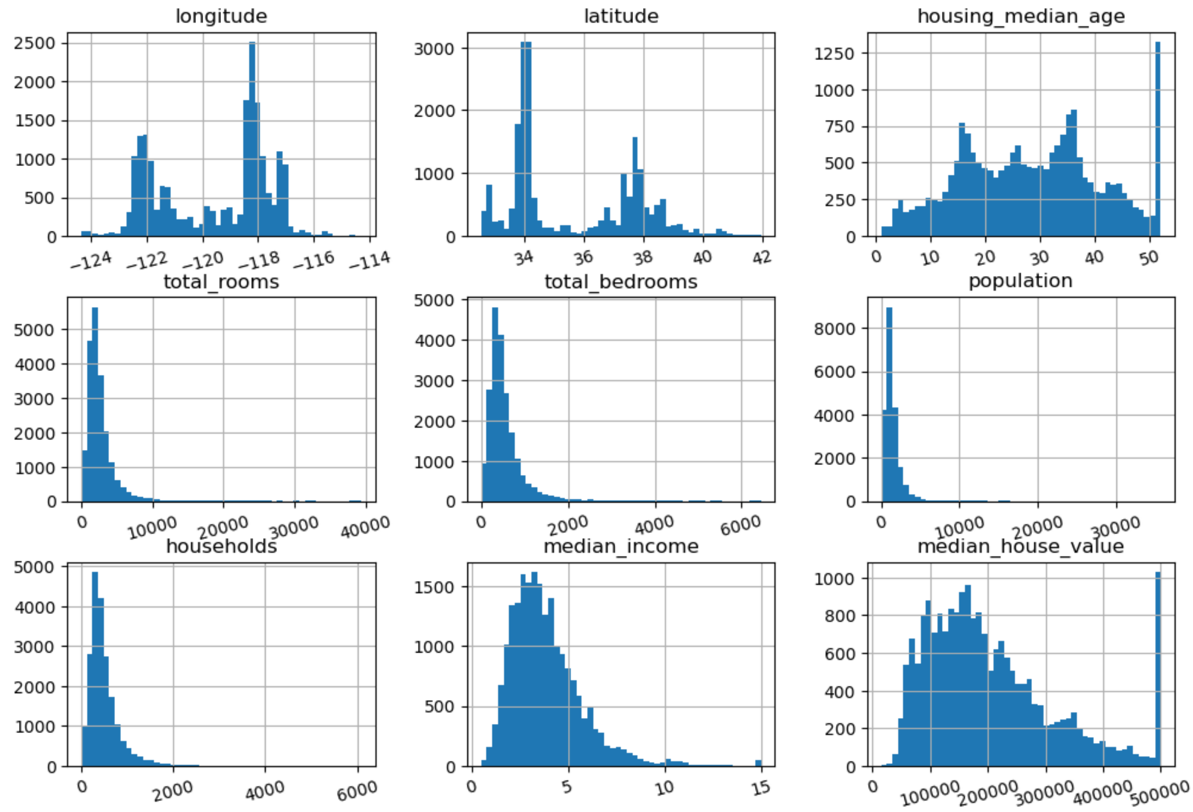


Table 4 A histogram for numerical attributes and categorical attributes

Looking at this histogram we notice the following:

1. Some histograms are skewed right. They extend much farther to the right of the median than to the left. This can make it harder for some machine learning algorithms to detect patterns. This calls for data transformations the attributes to have more symmetrical and bell-shaped distributions.

Plotting the same histograms in Jupyter notebook.



From the plots the following ideas about the dataset can be drawn.

1. The dataset is still skewed to the right therefore there is need for data transformation.
2. The attributes have very different scales we will have to scale the dataset.
3. Looking at the median_income horizontal axis it seams that the data was scaled.
4. For the median_house_value it seams the data was capped at 500,000 also the housing_median_age was capped at 50.

3.2.2 Explore and visualize the data to gain insights

After taking a quick glance at the data to get a general understanding of the kind of data we are manipulating. We go a little deeper. The goal is to study the relationship between the attributes against our target attribute median house price.

Before we perform any further EDA, we need to ensure that the data is representative of the entire population before splitting. This is known as stratified sampling where we divide the population into homogeneous subgroups called strata and the right number of instances are sampled from each stratum to guarantee that the test set is representative of the overall population. We base our stratified sampling on the median_income. We assume that median_income is a very important attribute to predict median housing prices. Since the median_income is a numerical attribute, we will have to create an income category attribute. A plot of the median income is as shown below.

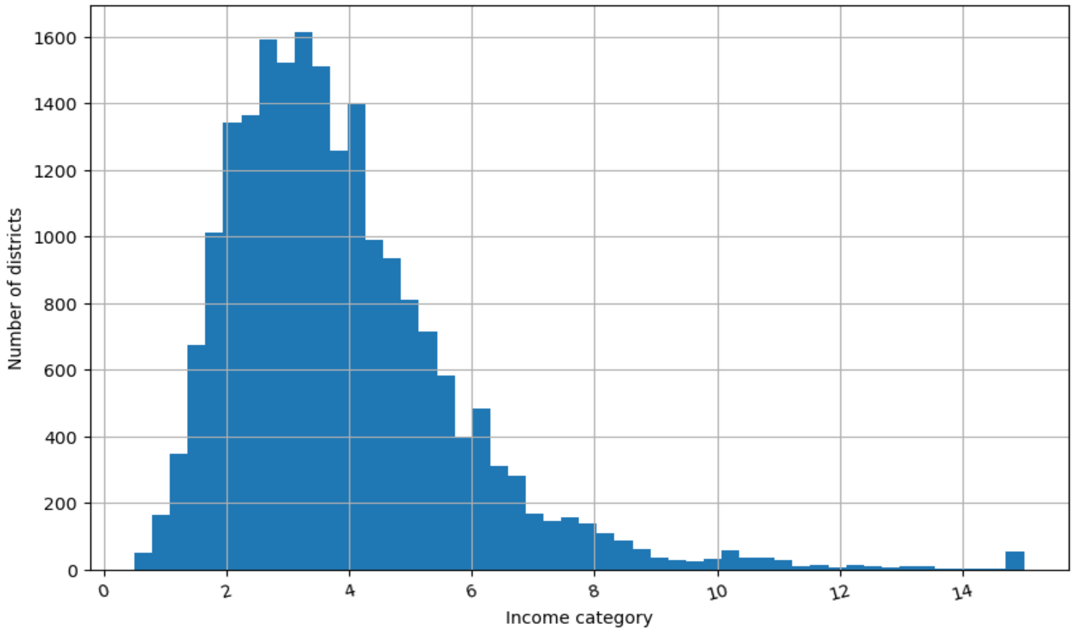


Table 5 A plot of median_income values

Looking at the median income histogram closely more median income values are clustered around 1.5 to 6 while some go beyond the value 6. It is important to have a sufficient number of instances in the dataset from this stratum 6 to 15 otherwise the estimates from the resulting model will be biased. This means not to have too many strata and each stratum large enough

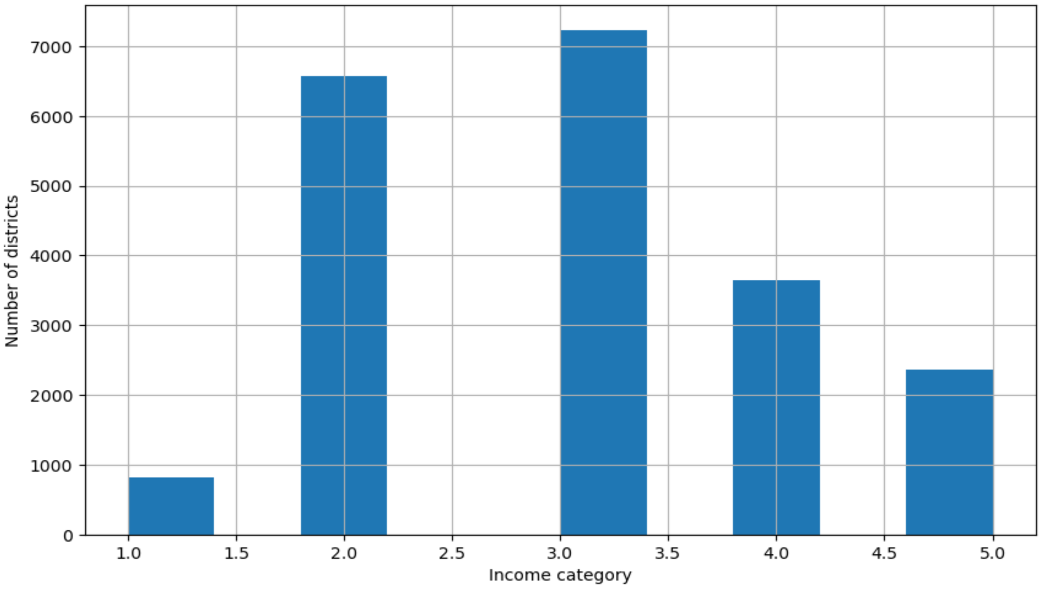


Table 6 New income category

3.2.2.1 Visualizing geographical data

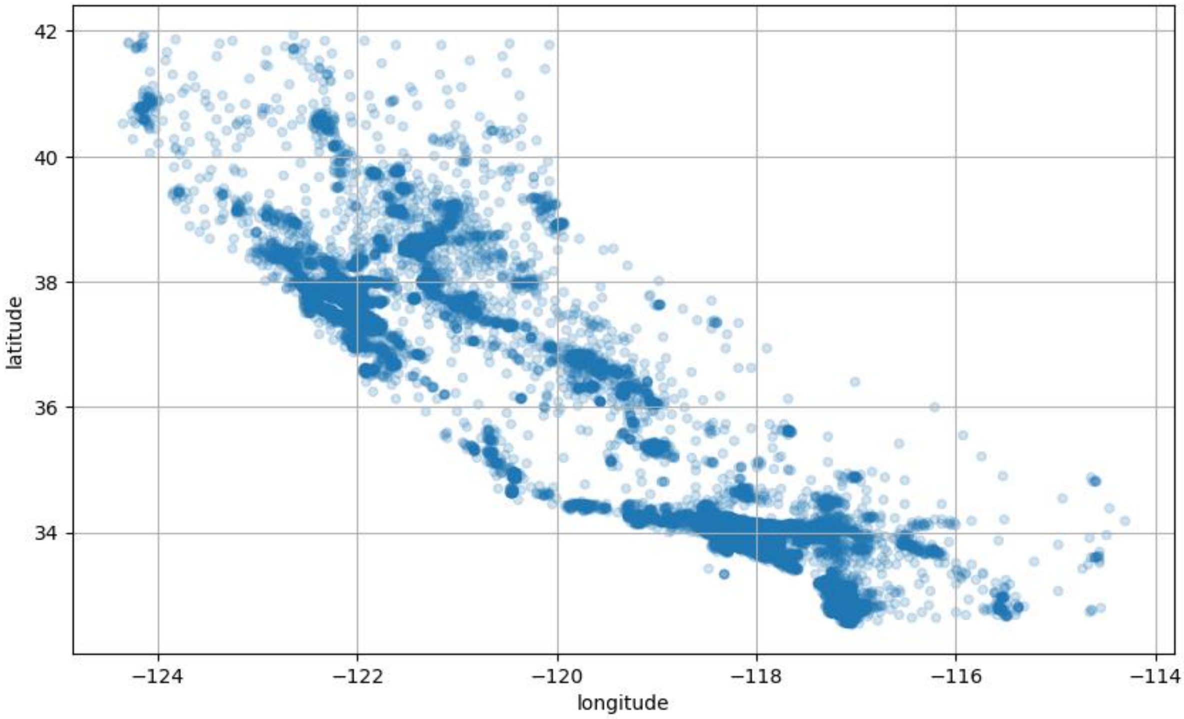


Table 7 Geographical scatter plot of the data

From the scatter plot we can see high density areas with a darker shade.

We look at distribution of housing prices. Using the following scatter flow. The radius of each circle represents the population and colour represents the price.

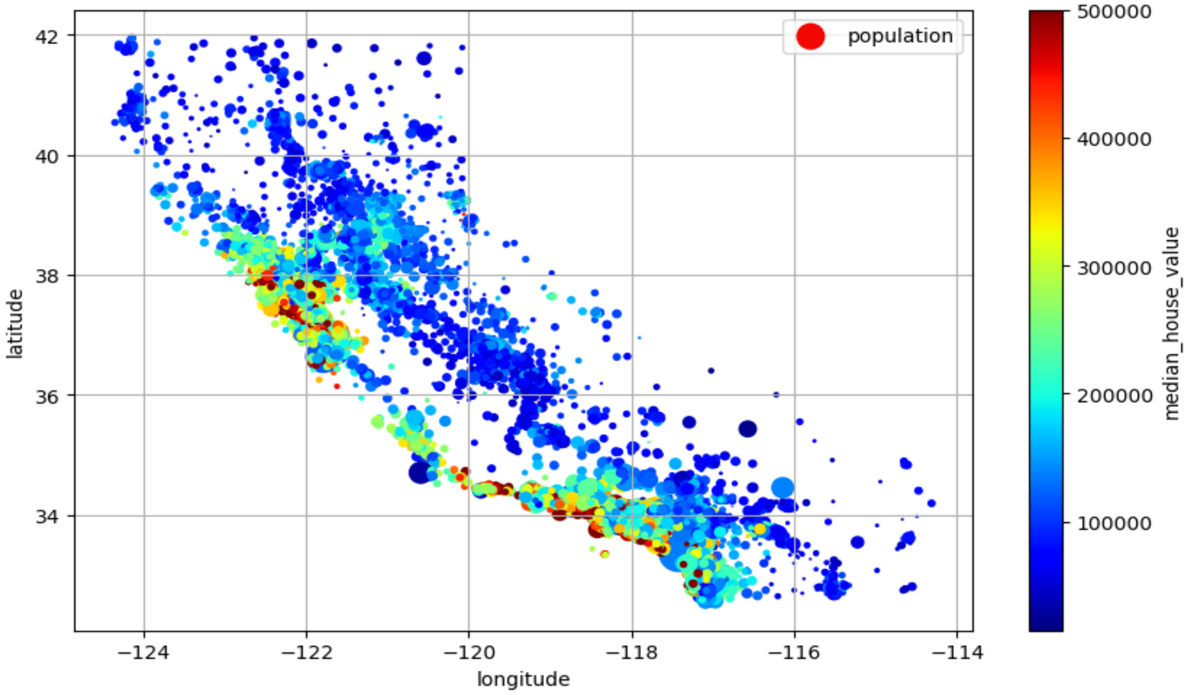


Table 8 Median housing price distribution

The image shows that the housing prices are related to the location e.g., close to the ocean and to the population density. As we know this is a California dataset it the ocean borders California to the west and population is denser towards the ocean.

3.2.2.2 Correlations

The correlation coefficient ranges from -1 to 1. We check correlations against quantity of products bought by a specific customer.

Column	Corr Coeffient
median_house_value	1
median_income	0.688075
total_rooms	0.134153
housing_median_age	0.105623
households	0.065843
total_bedrooms	0.049686
population	-0.02465
longitude	-0.045967
latitude	-0.14416

Table 9 Correlation coefficient using Pearson's r

Visualizing correlation using a heatmap

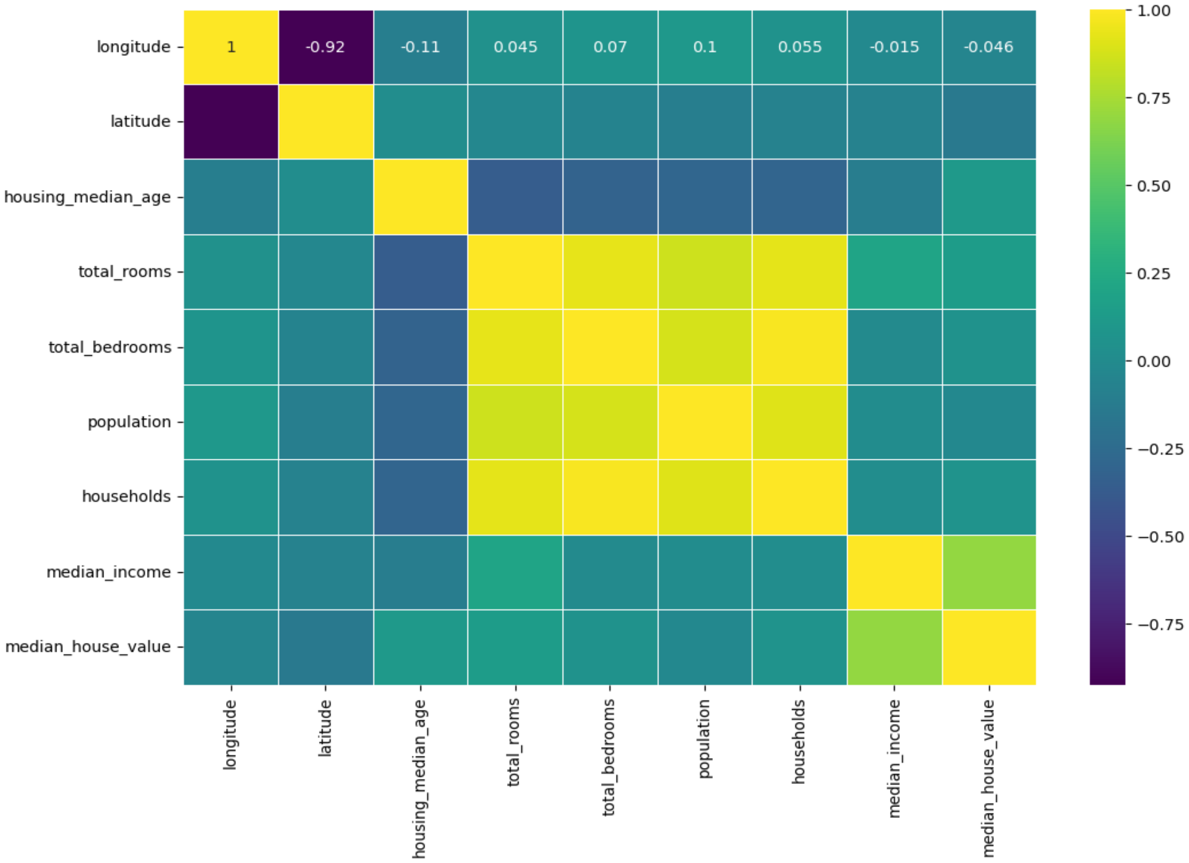


Table 10 Correlation heatmap

Checking correlation using Panda's scatter_matrix () function. In this case we plot scatter plot on attributes with top 5 coefficient against median_housing_value

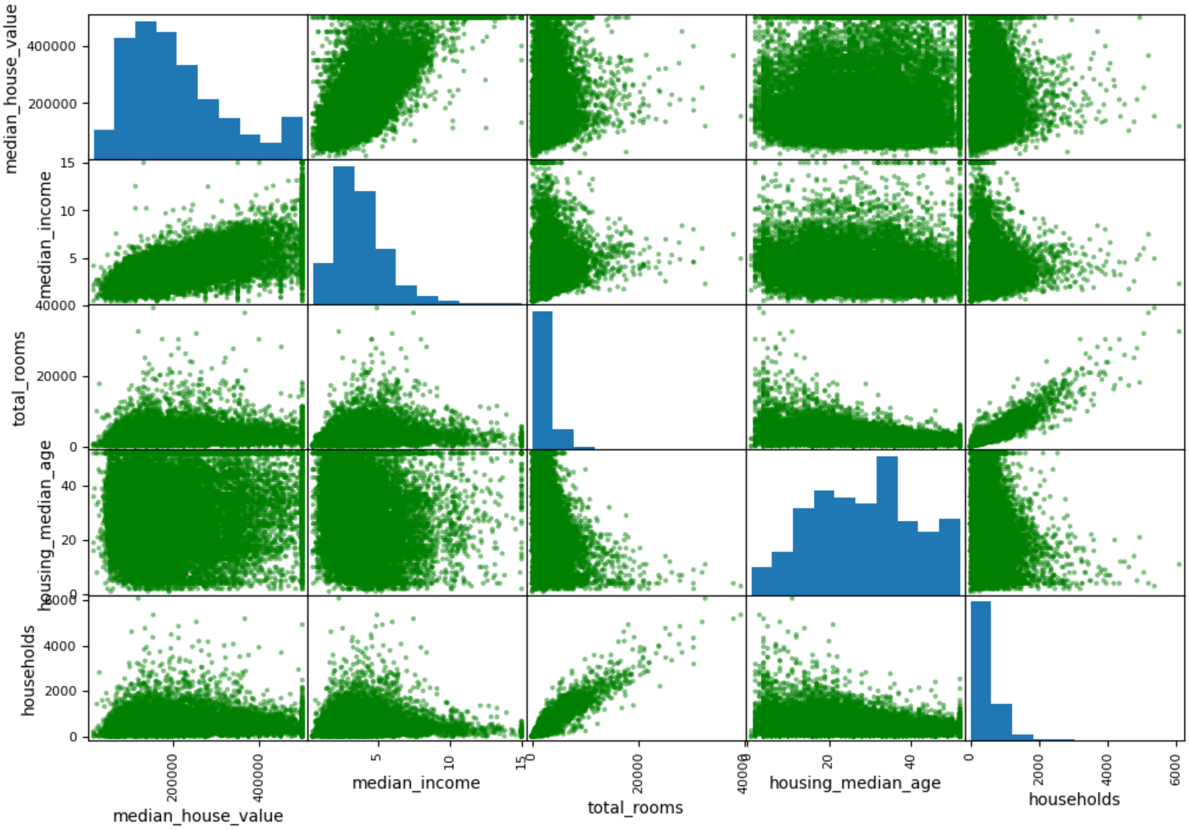


Table 11 Scatter plot showing correlation between attributes

The main diagonal is full of strain lines since Pandas plotted each variable against itself. i.e., correlation of 1. After all the correlation coefficient, scatter plot and heat map. It seams the most promising attribute to predict the median house value is the median income. This confirms our initial assumption during stratified sampling. Zooming in.

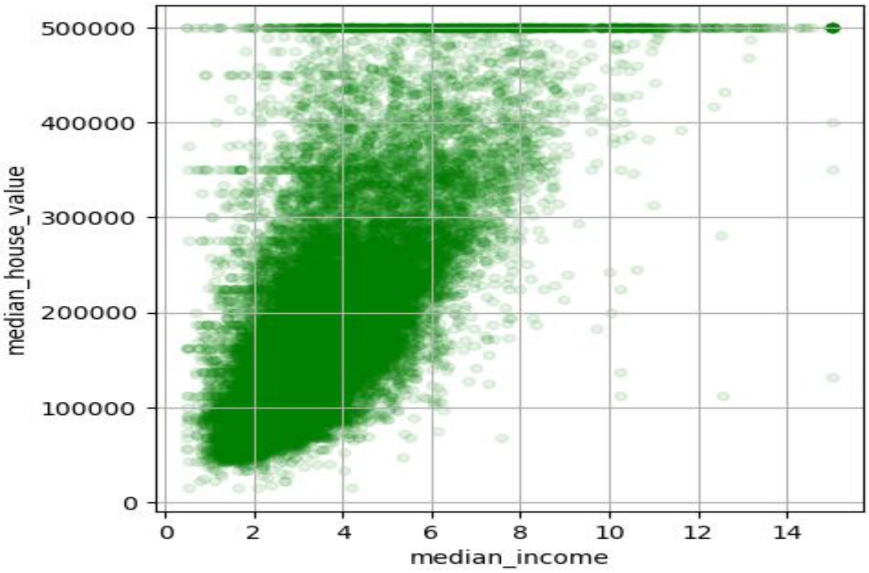


Table 12 Scatter plot of median house value vs median income

The plot reveals a few things;

1. The correlation is quite strong, as we can clearly see the upward trend and the points are not too dispersed.
2. The price cap we noticed earlier is visible as a horizontal line at $500,000
3. There is also a faint horizontal line around $450,000 and another at around $350,000. There might be need to remove the districts corresponding to this to prevent the algorithms from learning to reproduce these data. So, we keep this in mind.

From the above EDA it has given as a general idea of how the dataset looks like. We have identified;

1. A few data anomalies that we might need to clean up before feeding the data to a machine learning algorithm of choice.
2. We also found interesting correlations between attributes, in particular the target attribute median_house_value.
3. We also noticed that some attributes have a skewed right distribution which calls for transformation to ensure uniform distribution of features.

3.3 Data preparation with Weka

This is a crucial step before performing any data mining tasks. This will involve cleaning, transformation, attribute combination and splitting the data to ensure that it is in the best possible state for analysis. Here is a brief description of the steps taken to prepare the data.

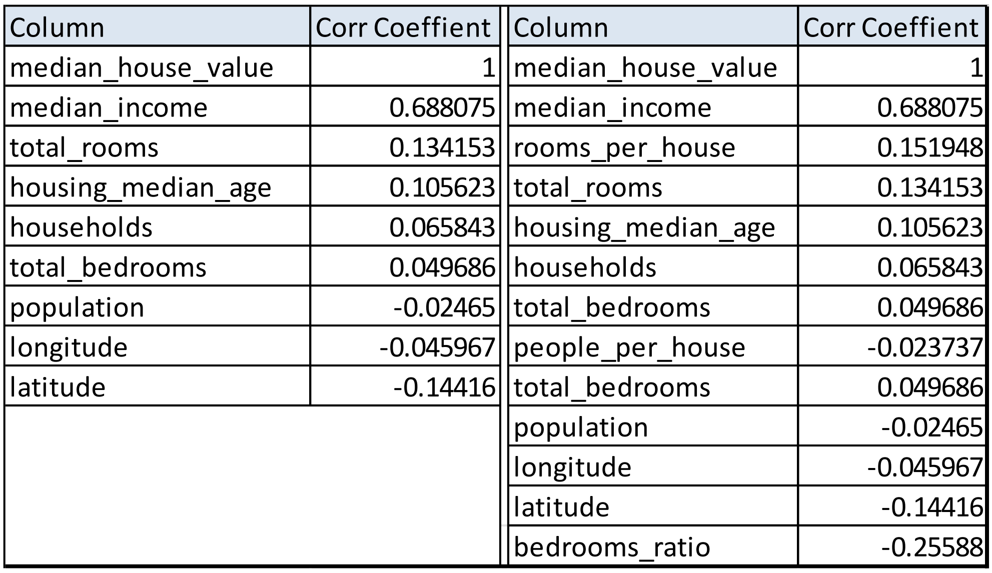
3.3.1 Feature combination

The total number of rooms in a district is not very useful if we don't know how many households there are. For example;

1. The total number of rooms in a district is not very useful if we don’t know how many households there are. Let’s try to see if knowing the number of rooms per household would have some effect on predicting the target variable.
2. Similarly, the total number of bedrooms by itself is not useful. We compare it to the number of rooms.
3. Population also seems like an interesting attribute combination to look at.

We create 3 new attributes $rooms\_per\_house = total\_rooms\ /\ households$, $bedrooms\_ratio = total\_bedrooms\ /\ total\_rooms$ and $people\_per\_house = population\ /\ households.$

We compare correlation to see what has changed.



Column	Corr Coeffient	Column	Corr Coeffient
median_house_value	1	median_house_value	1
median_income	0.688075	median_income	0.688075
total_rooms	0.134153	rooms_per_house	0.151948
housing_median_age	0.105623	total_rooms	0.134153
households	0.065843	housing_median_age	0.105623
total_bedrooms	0.049686	households	0.065843
population	-0.02465	total_bedrooms	0.049686
longitude	-0.045967	people_per_house	-0.023737
latitude	-0.14416	total_bedrooms	0.049686
		population	-0.02465
		longitude	-0.045967
		latitude	-0.14416
		bedrooms_ratio	-0.25588

Table 13 Comparison of coefficients after attribute combination

Not bad the new rooms_per_house seems to be more correlated compared to total_rooms. Looking at the bedrooms_ratio attribute is much more correlated with the median house value than the total number of rooms or bedrooms. However, houses with a lower bedroom to room ratio tend to be more expensive as expressed by the -0.26-correlation coefficient. After feature combination we see that the number of rooms per household is more informative than just total number of rooms in a district this is obvious the larger the house the more expensive they are.

For the rest of data processing will be done via Weka filters.

3.3.2 Data cleaning

Earlier we noticed that total bedroom has some missing values. The missing value is about 1 percent. We have 3 options;

1. Get rid of the corresponding districts
2. Get rid of the whole attribute

3. Set the missing value to some value e.g., zero, the mean, the median. A process called imputation

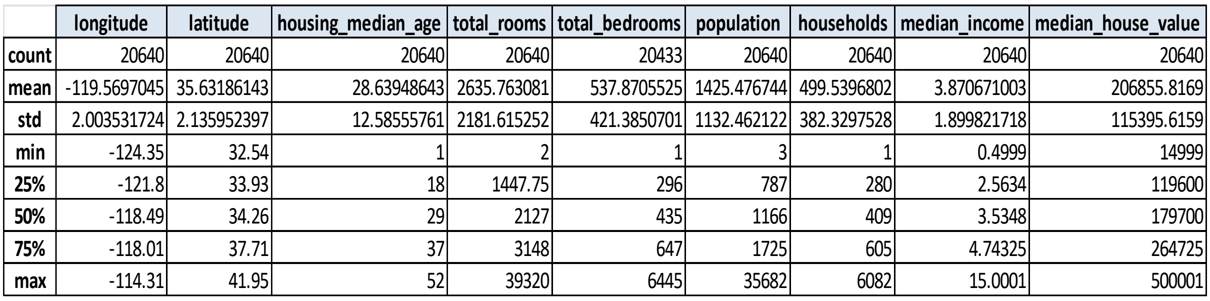
We go with option 3 as it is less destructive from Weka, we Select *filters → unsupervised → attribute → ReplaceMissingValues.* Weka uses the mean by default, to replace the missing values in the numerical attributes.

3.3.3 Encoding categorical variables

To encode categorical variables, we used one hot encoding. This was done through the $NominalToBinary$ filter which is found at $Filters > Unsupervised > Attribute > NominalToBinary$. This converted the categorical variables into binary attributes $Gender, Income, Customer\ Segment$ one-hot encoding.

3.3.4 Feature Scaling and Transformation

One important transformation we have to apply is feature scaling this is cause machine learning algorithms don't perform well when we input numerical attributes having different scales.



	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
count	20640	20640	20640	20640	20433	20640	20640	20640	20640
mean	-119.5697045	35.63186143	28.63948643	2635.763081	537.8705525	1425.476744	499.5396802	3.870671003	206855.8169
std	2.003531724	2.135952397	12.58555761	2181.615252	421.3850701	1132.462122	382.3297528	1.899821718	115395.6159
min	-124.35	32.54	1	2	1	3	1	0.4999	14999
25%	-121.8	33.93	18	1447.75	296	787	280	2.5634	119600
50%	-118.49	34.26	29	2127	435	1166	409	3.5348	179700
75%	-118.01	37.71	37	3148	647	1725	605	4.74325	264725
max	-114.31	41.95	52	39320	6445	35682	6082	15.0001	500001

For instance, as seen from above dataset summary. The total number of rooms ranges from 2 to 39320 while the median income ranges from 0.5 to 15. Without any scaling most models will be biased towards ignoring the median income and focusing more on the number of rooms.

We use standardization to scale the numerical attributes. Standardization does not restrict value to a specific range and it is much less affected by outliers. We Navigate to $filters -> unsupervised -> attribute -> Standardize$.

3.3.5 Splitting the data

The data was split into training and testing subsets using a 70%/30% percentage split, additionally used 10-fold cross-validation to evaluate model performance

3.4 Algorithm selection and Data mining

3.4.1 Decision tree J48 in Weka Classification algorithm

Classification is a supervised machine learning technique in which the quantity or output variable is categorical for example spam/ham, churn/not churned. In the case for our dataset, we have 1 categorical attribute which is ocean proximity.

For easier analysis of the results, we perform feature selection and remove some attributes from the datasets; this are population, longitude total_bedrooms, people_per_house and households

Because our business objective it to predict median housing price. This target value is numerical. To perform classification, we will have to convert it into categorical attribute. We create a new attribute called house_price_cat which will be categorical divided among cheap = 0-80,000, , medium = 80,001-250,000 and expensive 250,00 >. Below is a plot of number of districts against median house value.

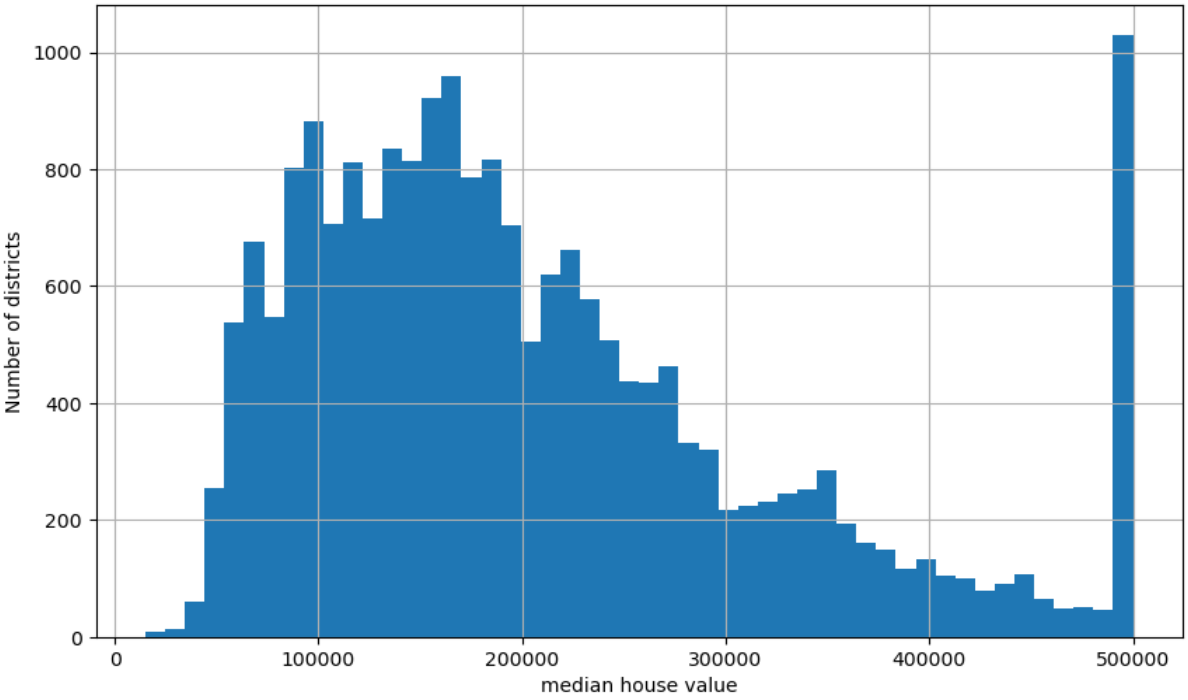


Table 14 Plot of number of districts vs median house value with a median house value cap at 500,000

Remember, we had an issue with the capping at 500,000 as seen from the above bar plot we filter out the districts associated with this data to prevent our model from learning to reproduce this data anomalies.

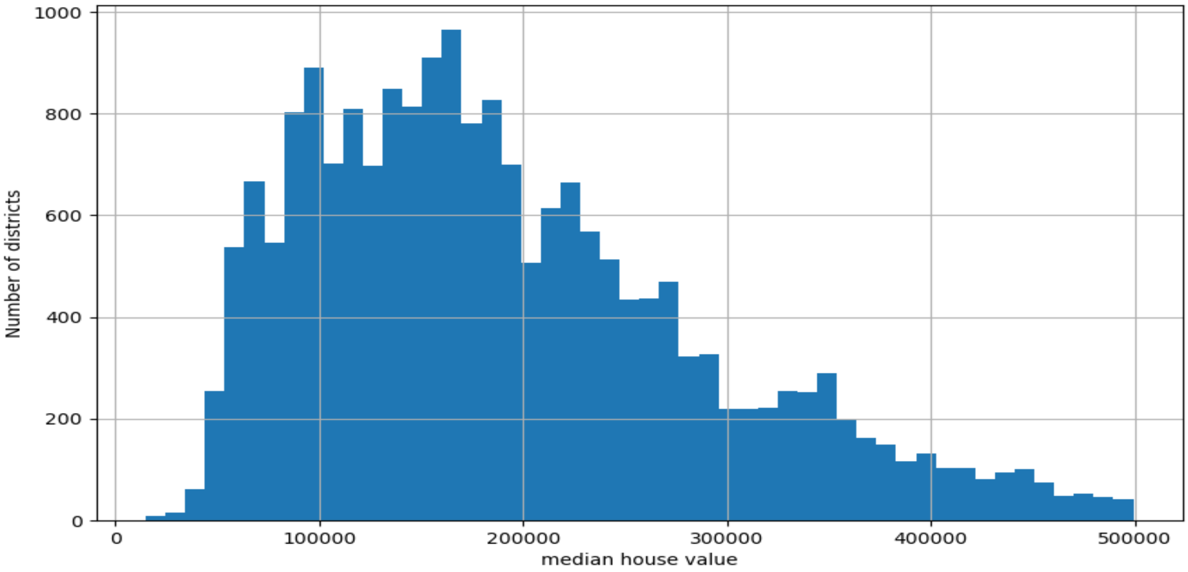


Table 15 Plot of number of districts vs median house value after filtering districts associated with 500,000 cap

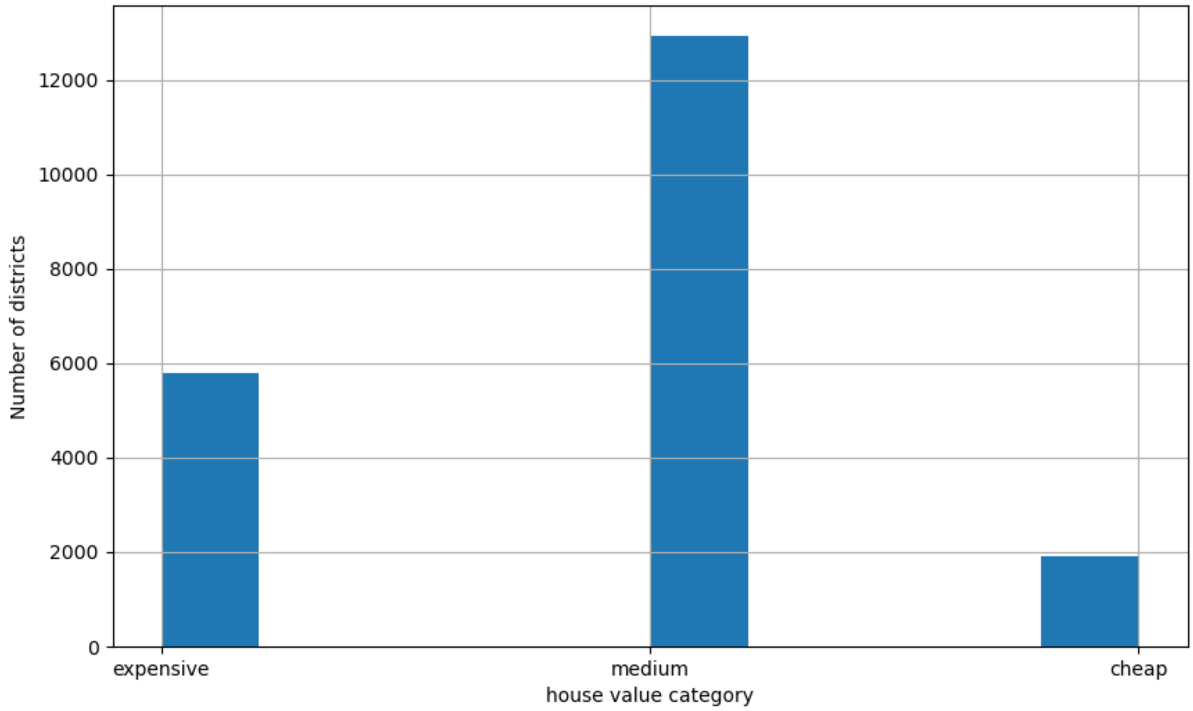


Table 16 new house value category attribute

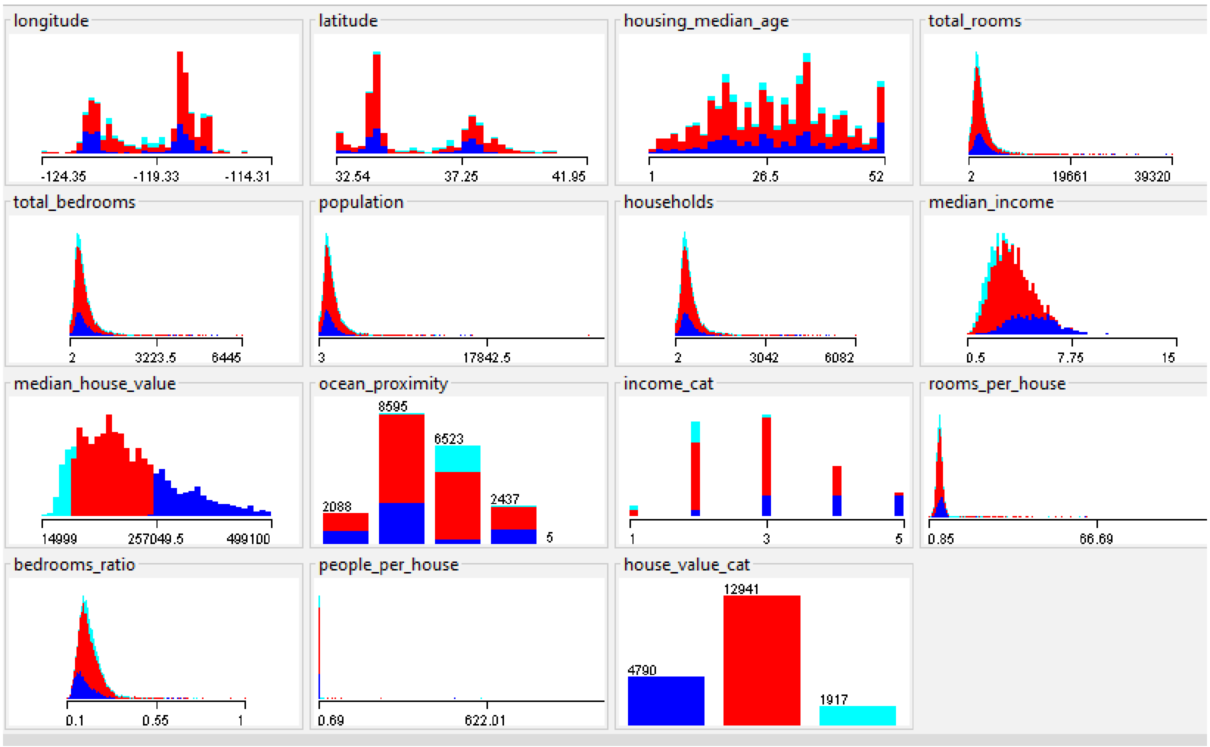


Table 17 Attributes to train for classification

For classification task we used Decision tree algorithm J48. Which was applied to a dataset to categorize house values into three classes cheap, medium and expensive. The model is build using a 10-fold cross validation approach which divides the dataset into 10 parts.

Model configuration:

- **Algorithm:** J48 (pruned tree)
- **Confidence Factor:** 0.25
- **Minimum Number of Instances per Leaf:** 2
- **Test Mode:** 10-fold cross-validation

Tree Structure Overview

The decision tree is structured based on various attributes, with a focus on predicting housing prices categorized into three classes: "expensive," "medium," and "cheap." Key factors influencing the classification include income_cat, ocean_proximity, median_income, housing_median_age, latitude, and total_rooms.

The model distinguishes housing prices by iteratively splitting on attributes like proximity to the ocean, income levels, and housing characteristics.

For example, houses classified as "expensive" typically have higher median incomes, closer proximity to the ocean (e.g., "<1H OCEAN," "NEAR OCEAN," "NEAR BAY"), and more rooms.

Model Summary:

- **Number of Leaves:** 442
- **Size of the Tree:** 874

The tree is relatively complex with 874 leaves.

Performance evaluation

Model performance metrics

Metric	Value
Correctly Classified Instances	4701 (79.75%)
Incorrectly Classified Instances	1193 (20.24%)
Kappa Statistic	0.5744
Mean Absolute Error (MAE)	0.1753
Root Mean Squared Error (RMSE)	0.3236
Relative Absolute Error (RAE)	52.57%
Root Relative Squared Error (RRSE)	79.50%
Total Number of Instances	5,894

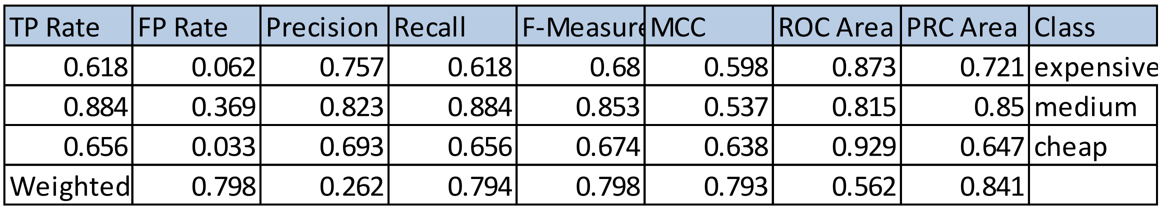
Figure 1Performance evaluation decision tree

Confusion matrix

Class	a (expensive)	b (medium)	c (cheap)
a (expensive)	868	536	1
b (medium)	277	3441	173
c (cheap)	2	204	392

Table 18 Confusion matrix decision tree

Detailed Accuracy by Class



TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.618	0.062	0.757	0.618	0.68	0.598	0.873	0.721	expensive
0.884	0.369	0.823	0.884	0.853	0.537	0.815	0.85	medium
0.656	0.033	0.693	0.656	0.674	0.638	0.929	0.647	cheap
Weighted	0.798	0.262	0.794	0.798	0.793	0.562	0.841	

Model Insights

- The model performs best in predicting the "medium" class but struggles with distinguishing between "expensive" and "medium" in some cases, as seen by the higher number of misclassifications between these two classes.
- The complexity of the tree with 442 leaves suggests a detailed decision-making process, but the relatively moderate kappa statistic (0.5744) indicates there's room for improvement, perhaps by simplifying the tree or using additional data preprocessing.

Conclusion:

This decision tree model provides a decent predictive capability for housing prices with an accuracy of about 80%. However, its performance varies across classes, with better precision and recall for "medium" housing prices compared to "expensive" and "cheap." Future improvements could involve tuning the model to reduce overfitting, which might improve classification rates for the "expensive" and "cheap" categories

3.4.2 Random Forest Classification algorithm

The ransom forest model was developed to classify housing prices into here categories expensive, medium and cheap.

The Random Forest algorithm was configured with 100 iterations and utilized Random Tree as the base learner. The model-building process took 8.47 seconds, and the model was evaluated on the test split.

Model configuration:

- **Algorithm:** Random forest
- **Confidence Factor:** 0.25
- **Minimum Number of Instances per Leaf:** 2
- **Test Mode:** 10-fold cross-validation

Tree Structure Overview

The decision tree is structured based on various attributes, with a focus on predicting housing prices categorized into three classes: "expensive," "medium," and "cheap." Key factors influencing the classification include income_cat, ocean_proximity, median_income, housing_median_age, latitude, and total_rooms.

The model distinguishes housing prices by iteratively splitting on attributes like proximity to the ocean, income levels, and housing characteristics.

For example, houses classified as "expensive" typically have higher median incomes, closer proximity to the ocean (e.g., "<1H OCEAN," "NEAR OCEAN," "NEAR BAY"), and more rooms.

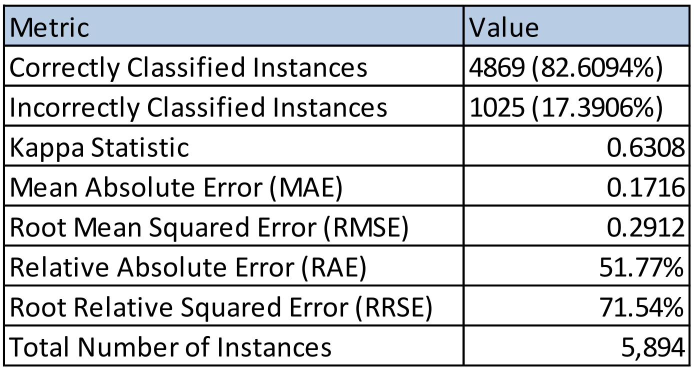
Model Summary:

- **Number of Leaves:** 442
- **Size of the Tree:** 874

The tree is relatively complex with 874 leaves.

Performance evaluation

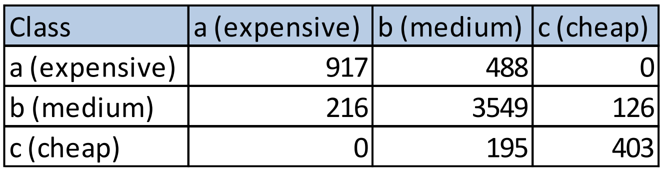
Model performance metrics



Metric	Value
Correctly Classified Instances	4869 (82.6094%)
Incorrectly Classified Instances	1025 (17.3906%)
Kappa Statistic	0.6308
Mean Absolute Error (MAE)	0.1716
Root Mean Squared Error (RMSE)	0.2912
Relative Absolute Error (RAE)	51.77%
Root Relative Squared Error (RRSE)	71.54%
Total Number of Instances	5,894

Performance evaluation random forest

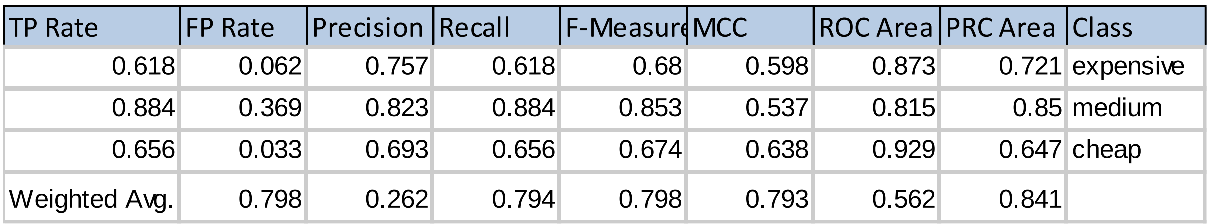
Confusion matrix



Class	a (expensive)	b (medium)	c (cheap)
a (expensive)	917	488	0
b (medium)	216	3549	126
c (cheap)	0	195	403

Table 19 Confusion matrix random forest

Detailed Accuracy by Class



TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.618	0.062	0.757	0.618	0.68	0.598	0.873	0.721	expensive
0.884	0.369	0.823	0.884	0.853	0.537	0.815	0.85	medium
0.656	0.033	0.693	0.656	0.674	0.638	0.929	0.647	cheap
Weighted Avg.	0.798	0.262	0.794	0.798	0.793	0.562	0.841	

Model Insights

- **Intuitive Sense**: The model’s performance appears to make intuitive sense. Variables such as median income, proximity to the ocean, and room ratios are logically linked to housing prices, contributing to the model's ability to classify homes into the "expensive," "medium," and "cheap" categories.
- **Generalization**: The model generalizes well across the different price categories, as evidenced by its overall accuracy and kappa statistic. However, the model does show

some difficulty in distinguishing between "expensive" and "medium" homes, as indicated by the confusion matrix.

- **Cautious Interpretation:** While the results suggest that features like median income and ocean proximity may influence housing prices just like we predicted during exploratory data analysis.

Conclusion

The Random Forest model achieved an accuracy of 82.61% in classifying housing prices into "expensive," "medium," and "cheap" categories, demonstrating a strong overall performance. The model showed high effectiveness in classifying "medium" homes, though it encountered challenges in distinguishing between "expensive" and "medium" categories. The results suggest that attributes such as median income, proximity to the ocean, and room ratios may influence housing prices.

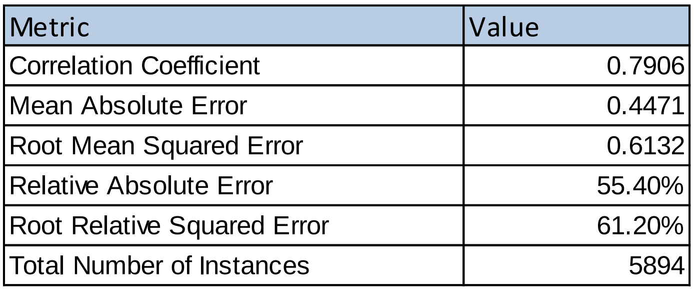
3.4.3 Linear regression

The equation derived by the model is

```
median_house_value =

    -0.5098 * longitude +

    -0.5    * latitude +

     0.1223 * housing_median_age +

     0.0259 * total_rooms +

     0.0327 * total_bedrooms +

    -0.3803 * population +

     0.3737 * households +

     0.6273 * median_income +

     0.3518 * ocean_proximity=<1H OCEAN,NEAR OCEAN,NEAR BAY,ISLAND +

     0.0427 * ocean_proximity=NEAR OCEAN,NEAR BAY,ISLAND +

    -0.1151 * ocean_proximity=NEAR BAY,ISLAND +

     1.7471 * ocean_proximity=ISLAND +

     0.0362 * income_cat +

     0.0566 * rooms_per_house +

     0.1539 * bedrooms_ratio +

    -0.233
```

Performance evaluation

Model performance metrics



Metric	Value
Correlation Coefficient	0.7906
Mean Absolute Error	0.4471
Root Mean Squared Error	0.6132
Relative Absolute Error	55.40%
Root Relative Squared Error	61.20%
Total Number of Instances	5894

Performance evaluation linear regression

From the results, the correlation coefficient of 0.7906 indicates there is a strong relationship between median_house_value predicted and actual values i.e., 79%, thus showing the model itself as effective in indicating the relationship of independent variables to target variable.

Model Insights

- **Intutive sense**: Model coefficients make sense intuitively. For example, for median_income, we have the highest positive coefficient (+0.6273), which makes intuitive sense since greater income levels tend to mean higher housing values. Also, longitude and latitude coefficients are negative, meaning houses that are positioned far from central regions or desirable locations will be less valued.
  Moreover, ocean_proximity also behaves as would be expected whereby properties near the ocean or on an island have generally high values. This is common in California
- **Generalization Ability**: The model appears to generalize well due to significant correlation coefficient. Nevertheless, errors (MAE and RMSE) indicate that though it captures general trends but it can be improved especially by accounting for outliers or other nonlinear relationships with data.
  The Relative Absolute Error (55.398%) and Root Relative Squared Error (61.2042%) indicate that the model has some limitations in predicting the