Text

Description automatically generated

First imported all the required libraries for our project.

Table

Description automatically generated

Above screenshot displayed the first five rows of the dataset. Each rows represent one district. So, the values are average values of houses in one district.

There is total 8 features: MedInc - it is the median income in the group

HouseAge – median house age of the district

AveRooms- Average number of rooms

aveBedrms – Average number of bedrooms

Population – populations of the district

AveOccup – Average number of members in the household

Latitude and Longitude – latitude and longitude of the district

Table

Description automatically generated

This is the basic description of the dataset. Here 25%,50% and 75% represent the values at 25 percentile, 50 percentile and 75 percentiles. We can see the difference between 75% and max is huge so we can say that there are some extreme values.

Graphical user interface, text, application, email

Description automatically generated

This is the target attribute of the dataset. Which is median house value of the district. Here we have to predict the numeric value, so this is a regression problem. This dataset has labels, so this is a classification problem.

Graphical user interface, text, application, email

Description automatically generated

Above we checked for null values using isnull () function. In our dataset there is no null values.

Chart, treemap chart

Description automatically generated

To get the insight of the dataset first we implemented the heatmap. Heatmap helps visualize the correlation between attributes.

From the heatmap we can see that median income and median house value are highly corelated. Average bedrooms and median house value are the least corelated.

Latitude and longitude directly less corelated to median house value but we implemented scatterplot with latitude and longitude with house value. So, we can know the prices with the location of the house.

Graphical user interface, chart, scatter chart

Description automatically generated

Above scatterplot has been implemented with latitude as Y axis and longitude as X axis and the datapoints are house values. we can see that all datapoints represent the map of California. From this scatterplot we can see that high value houses are mostly in the cities like San Francisco, Los Angeles, San Diego, San Jose.

For one more analysis we implemented the pair plot which will show the relation between each attribute pairwise.

Table

Description automatically generated with medium confidence

In above pair plot first dropped the attribute latitude and longitude because they separately do not affect the target attribute. From the above pair plot, we can see that as the avgrooms and avgbedrooms increases price of the house also increases. Also, aveOccup is the least corelated with the other attributes.

Scatter chart

Description automatically generated with medium confidence

Here we split the data to train and test model. We dropped the MedHouseVal column and assign all the other attributes to “x” and assign the MedHouseVal to “y” because that is our target column. To split the data, we use train\_test\_split (). The size of dataset for test we keep 0.2 which means 80% of the data will be used to train the model and 20% data will be used to test the data. And we took random\_state = 35 so every time we run the code, we get the same datasets.

Graphical user interface, text, application, email

Description automatically generated

To train the model for linearRegression, first we created the model using LinearRegression(). We use X\_train and Y\_train to train the model. To test the model, we used X\_valid and assign the predicted output to LR\_Pred. Also, we calculated the MAE, MSE and accuracy score. The result of linear regression is in the screenshot above.

Graphical user interface, text, application

Description automatically generated

To train the model for Random Forest, first we created the model. We use X\_train and Y\_train to train the model. To test the model, we used X\_valid and assign the predicted output to RF\_Pred. Also, we calculated the MAE, MSE and accuracy score. The result of linear regression is in the screenshot above. From all three model random forest has the maximum accuracy and minimum error rate.

Graphical user interface, text, application, email

Description automatically generated

To train the model for Decision Tree, first we created the model. We use X\_train and Y\_train to train the model. To test the model, we used X\_valid and assign the predicted output to DT\_Pred. Also, we calculated the MAE, MSE and accuracy score. The result of linear regression is in the screenshot above.

Graphical user interface

Description automatically generated with medium confidence

In above screenshot we have printed the result from all three models with the actual value. LR\_Predicted is result from linear Regression, RF\_Predicted is the result from random forest and DT\_Predicted is the result of Decision tree model.