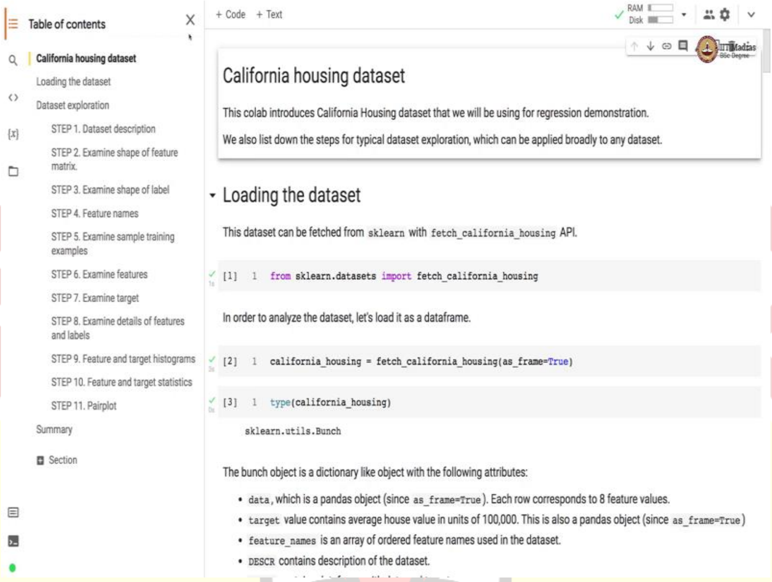
INDIAN INSTITUTE OF TECHNOLOGY MADRAS • सिद्धिर्भवति कर्मजा •

IIT Madras

ONLINE DEGREE

Machine Learning Program
Indian Institute of Technology, Madras
California Housing Dataset

(Refer Slide Time: 0:10)



California housing dataset
Loading the dataset
Dataset exploration
STEP 1. Dataset description
STEP 2. Examine shape of feature matrix.
STEP 3. Examine shape of label
STEP 4. Feature names
STEP 5. Examine sample training examples
STEP 6. Examine features
STEP 7. Examine target
STEP 8. Examine details of features and labels
STEP 9. Feature and target histograms
STEP 10. Feature and target statistics
STEP 11. Pairplot
Summary
Section

California housing dataset

This colab introduces California Housing dataset that we will be using for regression demonstration.

We also list down the steps for typical dataset exploration, which can be applied broadly to any dataset.

Loading the dataset

This dataset can be fetched from `sklearn` with `fetch_california_housing` API.

```
[1]   from sklearn.datasets import fetch_california_housing
```

In order to analyze the dataset, let's load it as a dataframe.

```
[2]   california_housing = fetch_california_housing(as_frame=True)
```

```
[3]   type(california_housing)

    sklearn.utils.Bunch
```

The bunch object is a dictionary like object with the following attributes:

- `data`, which is a pandas object (since `as_frame=True`). Each row corresponds to 8 feature values.
- `target` value contains average house value in units of 100,000. This is also a pandas object (since `as_frame=True`)
- `feature_names` is an array of ordered feature names used in the dataset.
- `DESCR` contains description of the dataset.

Namaste. Welcome to the next video of Machine Learning Practice Course. In this video, we will introduce you to California housing dataset. We are going to use California housing dataset for demonstrating linear regression implementation in skearn. While exploring this dataset, you will also understand the typical steps in dataset exploration that can be broadly applied to any other dataset.

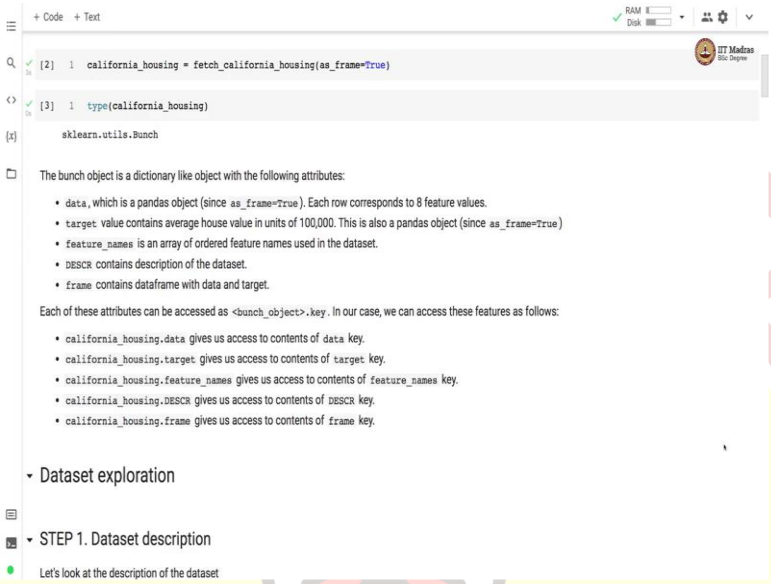
Let us begin. First, we load the dataset. So, let us look at the steps. So, we load the dataset will see the description of the dataset will examine the shapes of feature and label matrix. Then we will look at the names of the feature. We will examine the sample of training examples, we will examine the features, we will examine the target.

We will also examine details of features and labels. Then we will see what, how are the features and labels are distributed through histogram plots. And we look at feature and target statistics. And finally, we will plot a pair plot between the features. So, you are going to explore this dataset in these 11 different steps. So, let us begin by loading the dataset.

So, California housing dataset can be fetched from sklearn with fetch_California _housing API. We can find this API in sklearn.datasets module. In order to analyze the dataset, let us load it as a dataframe. In order to load this dataset as a dataframe, we set as_frame parameter to true in the fetch California housing API.

So, when you execute this cell, we basically get California housing data frame that contains the dataset that is stretched using this particular API. If we look at the type of this particular object, it is the bunch type right and we have seen that bunch is the type that is returned by load and fetch commands.

(Refer Slide Time: 2:53)



```
[2] california_housing = fetch_california_housing(as_frame=True)
```

```
[3] type(california_housing)
```

```
sklearn.utils.Bunch
```

The bunch object is a dictionary like object with the following attributes:

- `data`, which is a pandas object (since `as_frame=True`). Each row corresponds to 8 feature values.
- `target` value contains average house value in units of 100,000. This is also a pandas object (since `as_frame=True`)
- `feature_names` is an array of ordered feature names used in the dataset.
- `DESCR` contains description of the dataset.
- `frame` contains dataframe with data and target.

Each of these attributes can be accessed as `<bunch_object>.key`. In our case, we can access these features as follows:

- `california_housing.data` gives us access to contents of `data` key.
- `california_housing.target` gives us access to contents of `target` key.
- `california_housing.feature_names` gives us access to contents of `feature_names` key.
- `california_housing.DESCR` gives us access to contents of `DESCR` key.
- `california_housing.frame` gives us access to contents of `frame` key.

Dataset exploration

STEP 1. Dataset description

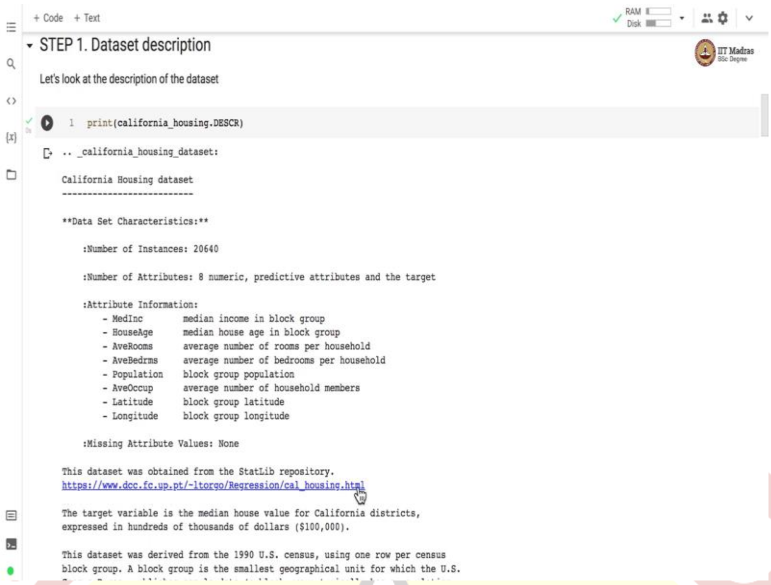
Let's look at the description of the dataset

A bunch object is a dictionary like object with the following attributes. It has got data, which is a pandas object, since as_frame = true, and each row corresponds to 8 feature values. Then there is a target, which contains the average house value in unit of 100,000 is also a pandas object, since as_frame = true.

Then your feature names, which is an array of ordered feature names used in the dataset. We have DESCR that contains a description of the dataset and frame that contains data frame with data and target together. So here, we just have the features in this particular data frame, which is data. And then in the target data frame, we only have labels rather, this is a series in target series, we have only labels, and the frame contains both the features and labels.

So, each of these attributes can be accessed as bunch object.key and these are the keys. So, in our case, we can access these features as follows. So, if you want to access data, we use California_housing.data because California_housing is a bunch object. And we access data through, by basically calling California_housing_data that gives us access to the content of the data key or the data frame. Then we can in the similar manner access the target as California _ housing.target. We can get feature names, description and train in the similar manner.

(Refer Slide Time: 4:55)



STEP 1. Dataset description

Let's look at the description of the dataset

```
print(california_housing.DESCR)
```

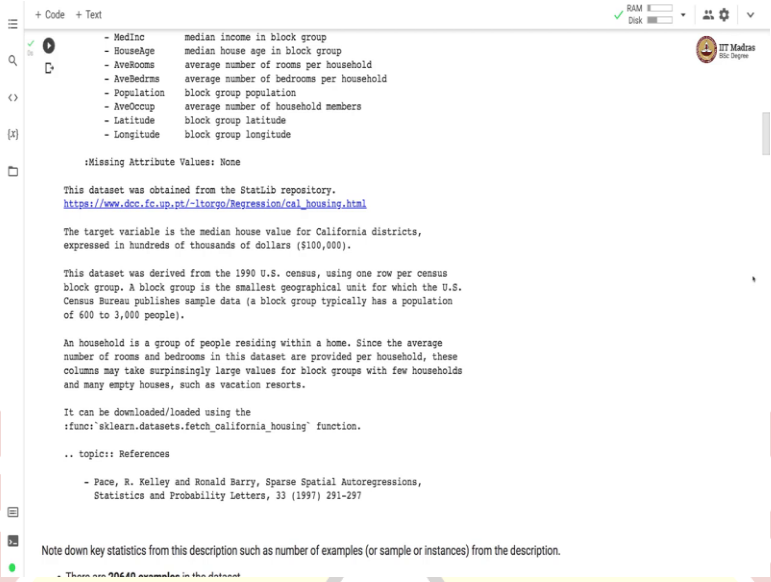
```
.. _california_housing_dataset:

California Housing dataset
--------------------------

**Data Set Characteristics:**

    :Number of Instances: 20640

    :Number of Attributes: 8 numeric, predictive attributes and the target

    :Attribute Information:
        - MedInc        median income in block group
        - HouseAge      median house age in block group
        - AveRooms      average number of rooms per household
        - AveBedrms     average number of bedrooms per household
        - Population    block group population
        - AveOccup      average number of household members
        - Latitude      block group latitude
        - Longitude     block group longitude

    :Missing Attribute Values: None

This dataset was obtained from the StatLib repository.
https://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html

The target variable is the median house value for California districts,
expressed in hundreds of thousands of dollars ($100,000).

This dataset was derived from the 1990 U.S. census, using one row per census
block group. A block group is the smallest geographical unit for which the U.S.
```

Now, that we have loaded the dataset, let us let us explore it. Let us first look at a Dataset Description. We can print a dataset description and you know how we can access the description; it is California _ housing.DESCR , that prints out the description of this dataset.

And you can see that in the description it is written that total number of instances is 20,640 number of attributes are 8, they are numeric and predictive attribute. These are these are 8 predictive attributes and then there is a target and then there is a section on attribute information where the names of that atributes are listed along with short description.

So, there is MedINC which is median income in block group. Then there is HouseAge which is median house age in block group, then there is average rooms, average bedrooms which are average number of rooms per household and average number of bedrooms per household. And there is a population which is block group population, average occupancy of the household members, latitude and longitude of the block. There are no missing values as specified in the documentation. And this dataset was obtained from StatLib repository, which is hosted at this particular URL.

(Refer Slide Time: 6:17)



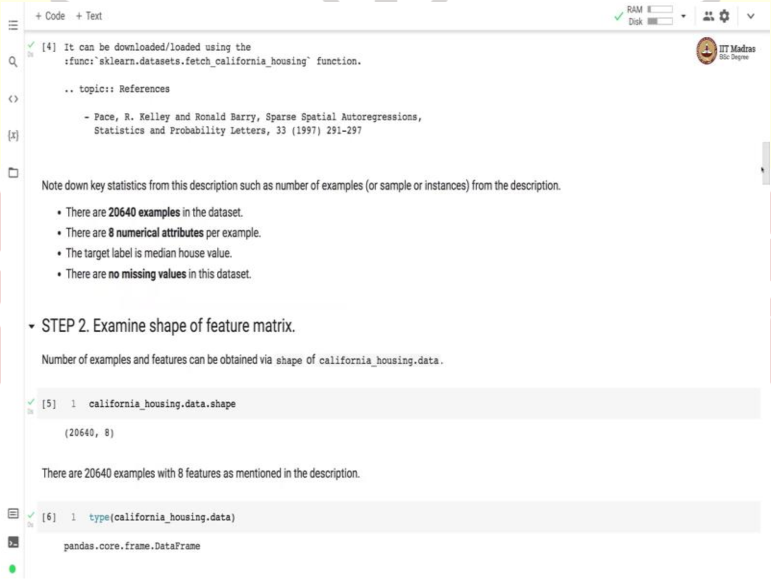
```
        - MedInc        median income in block group
        - HouseAge      median house age in block group
        - AveRooms      average number of rooms per household
        - AveBedrms     average number of bedrooms per household
        - Population    block group population
        - AveOccup      average number of household members
        - Latitude      block group latitude
        - Longitude     block group longitude

    :Missing Attribute Values: None

This dataset was obtained from the StatLib repository.
https://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html

The target variable is the median house value for California districts,
expressed in hundreds of thousands of dollars ($100,000).

This dataset was derived from the 1990 U.S. census, using one row per census
block group. A block group is the smallest geographical unit for which the U.S.
Census Bureau publishes sample data (a block group typically has a population
of 600 to 3,000 people).

An household is a group of people residing within a home. Since the average
number of rooms and bedrooms in this dataset are provided per household, these
columns may take surpinsingly large values for block groups with few households
and many empty houses, such as vacation resorts.

It can be downloaded/loaded using the
:func:`sklearn.datasets.fetch_california_housing` function.

.. topic:: References

    - Pace, R. Kelley and Ronald Barry, Sparse Spatial Autoregressions,
      Statistics and Probability Letters, 33 (1997) 291-297
```

Note down key statistics from this description such as number of examples (or sample or instances) from the description.

- There are **20640 examples** in the dataset

The target variable is expressed in 100s of 1000s of dollars. So, if the value of the house is 200,000 dollars, the target variable will have value 2. If the value is 300,000 dollars, the target variable will contain value 3, and so on. There is a brief description about how this dataset was derived and also given an explanation about household. So, household is a group of people residing within a home.

(Refer Slide Time: 6:51)



```
[4] It can be downloaded/loaded using the
    :func:`sklearn.datasets.fetch_california_housing` function.

    .. topic:: References

        - Pace, R. Kelley and Ronald Barry, Sparse Spatial Autoregressions,
          Statistics and Probability Letters, 33 (1997) 291-297
```

Note down key statistics from this description such as number of examples (or sample or instances) from the description.

- There are **20640 examples** in the dataset.
- There are **8 numerical attributes** per example.
- The target label is median house value.
- There are **no missing values** in this dataset.

STEP 2. Examine shape of feature matrix.

Number of examples and features can be obtained via `shape` of `california_housing.data`.

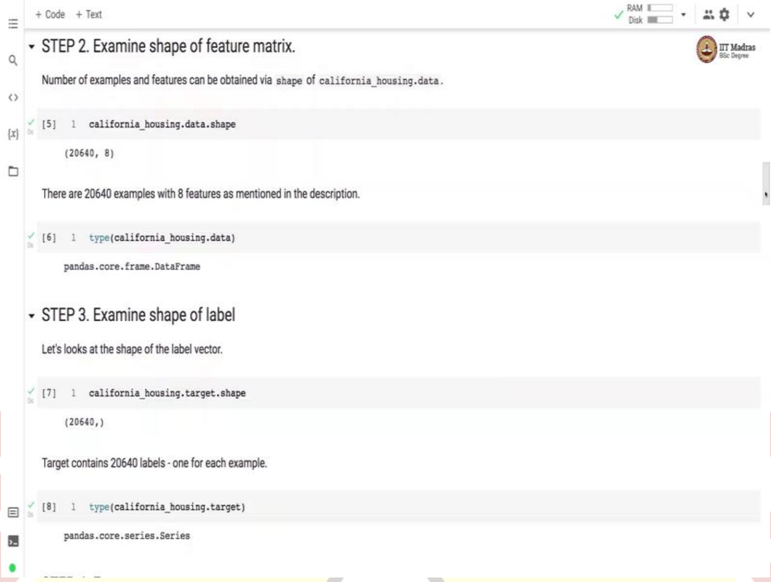
```
[5] california_housing.data.shape

(20640, 8)
```

There are 20640 examples with 8 features as mentioned in the description.

```
[6] type(california_housing.data)

pandas.core.frame.DataFrame
```

So, from this description, we will note down key statistics such as number of examples from the description. So, there are 20,640 examples in the dataset. There are 8 numerical attributes per example, the target value, the target label is median house value, and there are no missing values in this dataset. Let us examine the feature the shape of feature matrix.

(Refer Slide Time: 7:15)



STEP 2. Examine shape of feature matrix.

Number of examples and features can be obtained via `shape` of `california_housing.data`.

```
[5] california_housing.data.shape
```

```
(20640, 8)
```

There are 20640 examples with 8 features as mentioned in the description.

```
[6] type(california_housing.data)
```

```
pandas.core.frame.DataFrame
```

STEP 3. Examine shape of label

Let's looks at the shape of the label vector.

```
[7] california_housing.target.shape
```

```
(20640,)
```

Target contains 20640 labels - one for each example.

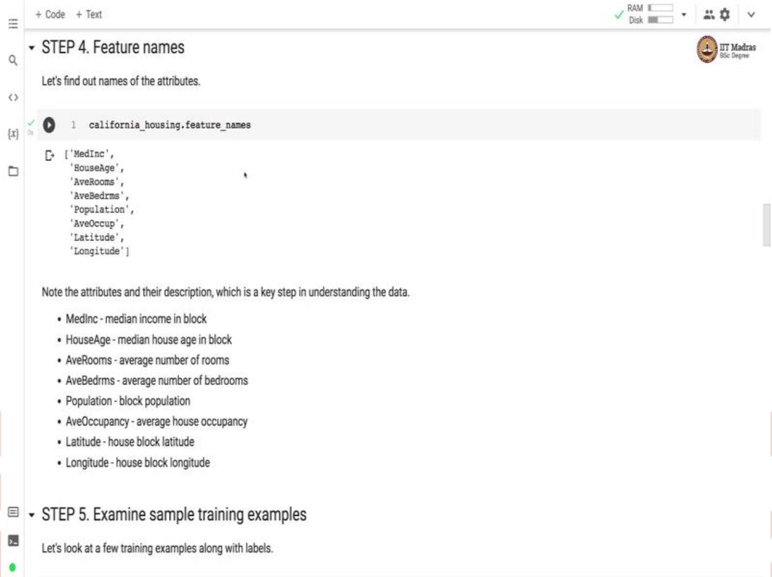
```
[8] type(california_housing.target)
```

```
pandas.core.series.Series
```

So, we can examine, we can obtain the shape by simply calling California _ housing.data, this gives us data frame, and we call the shape on the data from object, we get the shape as (20640,8) so that means there are 20,640 samples are examples with 8 features. Let us look at a type of California_housing.data object and you can see that it is a data frame.

Now, let us examine the shape of label. And when we get the shape of label with California _ housing.target that contains the label series, we call the shape on the label series. And we see that there are 20,640 labels one for each example, and the type of target is a series. So, you should compare the shape of the target with the first dimension of shape of the data. So, in this case, it is 20640 in both the cases and that should be the case. So, this is a good sanity check, to check whether the data and targets are loaded correctly.

(Refer Slide Time: 8:45)



STEP 4. Feature names

Let's find out names of the attributes.

```
california_housing.feature_names
```

```
['MedInc',
 'HouseAge',
 'AveRooms',
 'AveBedrms',
 'Population',
 'AveOccup',
 'Latitude',
 'Longitude']
```

Note the attributes and their description, which is a key step in understanding the data.

- MedInc - median income in block
- HouseAge - median house age in block
- AveRooms - average number of rooms
- AveBedrms - average number of bedrooms
- Population - block population
- AveOccupancy - average house occupancy
- Latitude - house block latitude
- Longitude - house block longitude

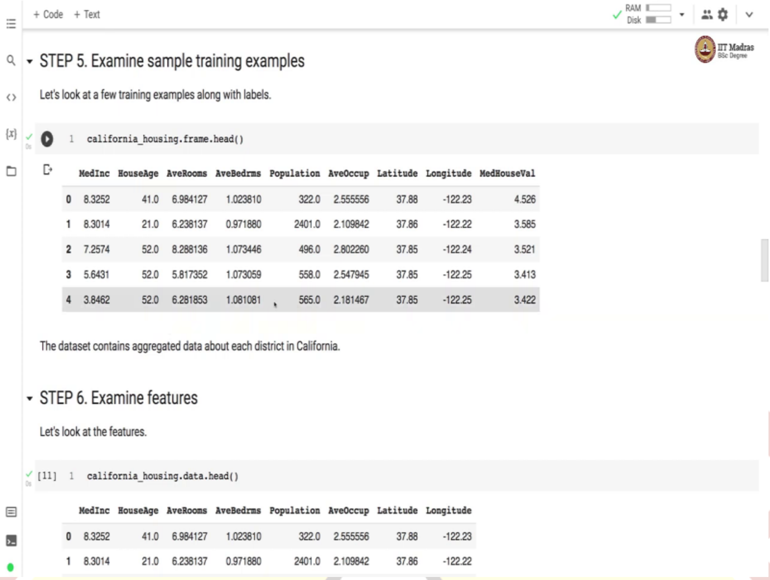
STEP 5. Examine sample training examples

Let's look at a few training examples along with labels.

Let us look at the names of the features or attributes. And we can obtain them by calling California _ housing.feature _ names. We get an ordered, we get an array that contains the feature names in order in which they appear in the data. So, we have MedINC, HouseAge, average rooms, average bedrooms, population, average occupancy, latitude and longitude.

So, you should note down attributes under description. And this is a key step in understanding the data. You should first know what each attribute corresponds to and how that attribute is calculated. So, here we know that there are some med income is median income in the block. House age is the median house age, average room is average number of rooms and so on.

(Refer Slide Time: 9:50)



STEP 5. Examine sample training examples

Let's look at a few training examples along with labels.

```
california_housing.frame.head()
```

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	MedHouseVal
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23	4.526
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22	3.585
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24	3.521
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25	3.413
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25	3.422

The dataset contains aggregated data about each district in California.

STEP 6. Examine features

Let's look at the features.

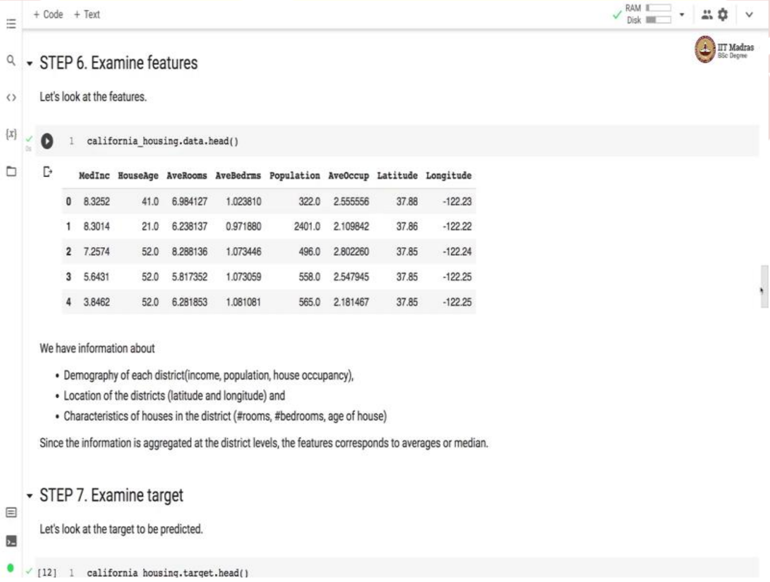
```
[11] california_housing.data.head()
```

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22

Let us examine sample training data. So here, what we will do is, since you want to examine features and labels together we will use frame. So, California _ housing.frame contains labels, as well as features. So, we call the head function on this data frame to obtain first 5 examples. And here, you can see, all 8 features listed along with the 9 column, which is the median house value.

And remember, this median house value is expressed in terms of 100s of 1000s of dollars. So, here the price is 4.526, that means the price of the house will be 452,600 dollars. So, this dataset contains aggregated data about each district in California and this aggregation is done via mean and median.

(Refer Slide Time: 11:03)



STEP 6. Examine features

Let's look at the features.

```
california_housing.data.head()
```

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25

We have information about

- Demography of each district(income, population, house occupancy),
- Location of the districts (latitude and longitude) and
- Characteristics of houses in the district (#rooms, #bedrooms, age of house)

Since the information is aggregated at the district levels, the features corresponds to averages or median.

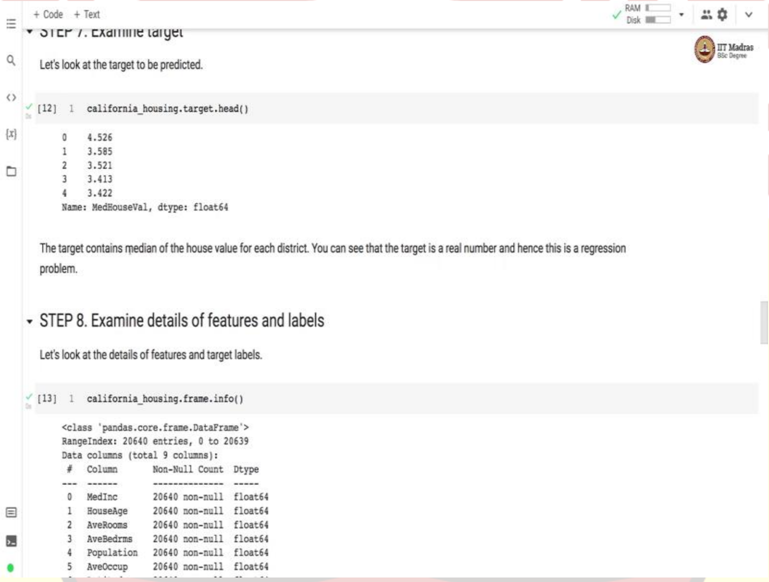
STEP 7. Examine target

Let's look at the target to be predicted.

```
[12] california_housing.target.head()
```

So, let us examine the features separately. So, instead of examining the features and labels together, in this case, we are just examining the features and that so we are using California _ housing.data frame. So, now, we have information about demography of each district, that is we have income, population and house occupancy, you also locations of the district through latitude and longitude, and we have characteristics of houses in the district in terms of rooms, bedrooms, and age of the house. Since the information is aggregated at the district level, the feature corresponds to either averages or medians.

(Refer Slide Time: 11:58)



STEP 7. Examine target

Let's look at the target to be predicted.

```
[12] california_housing.target.head()
```

```
0    4.526
1    3.585
2    3.521
3    3.413
4    3.422
Name: MedHouseVal, dtype: float64
```

The target contains median of the house value for each district. You can see that the target is a real number and hence this is a regression problem.

STEP 8. Examine details of features and labels

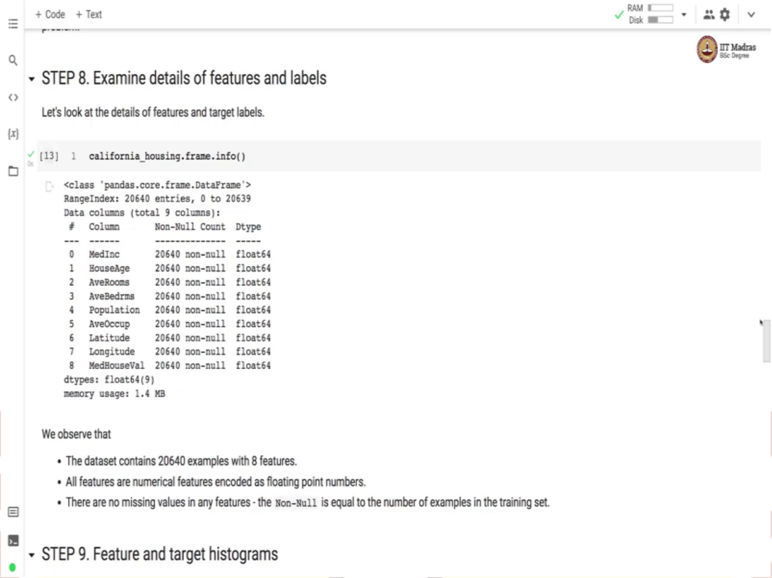
Let's look at the details of features and target labels.

```
[13] california_housing.frame.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   MedInc      20640 non-null  float64
 1   HouseAge    20640 non-null  float64
 2   AveRooms    20640 non-null  float64
 3   AveBedrms   20640 non-null  float64
 4   Population  20640 non-null  float64
 5   AveOccup    20640 non-null  float64
```

You also examine the target by looking at the target series and calling head on it, so that we look at first five values in the target. And here the target contains median of house value for each district. And we can see that the target is a real number. And hence, this dataset is can be used for linear regression problem.

(Refer Slide Time: 12:31)



STEP 8. Examine details of features and labels

Let's look at the details of features and target labels.

```
[13]  california_housing.frame.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   MedInc       20640 non-null  float64
 1   HouseAge     20640 non-null  float64
 2   AveRooms     20640 non-null  float64
 3   AveBedrms    20640 non-null  float64
 4   Population   20640 non-null  float64
 5   AveOccup     20640 non-null  float64
 6   Latitude     20640 non-null  float64
 7   Longitude    20640 non-null  float64
 8   MedHouseVal  20640 non-null  float64
dtypes: float64(9)
memory usage: 1.4 MB
```

We observe that

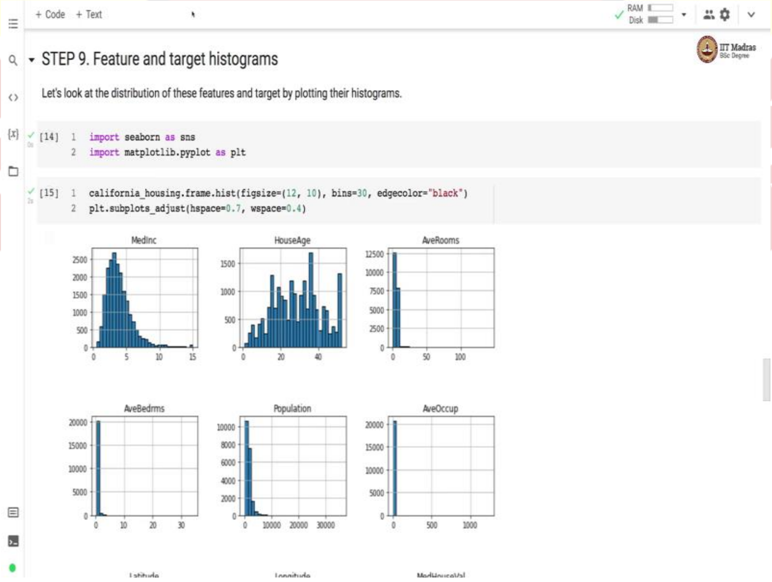
- The dataset contains 20640 examples with 8 features.
- All features are numerical features encoded as floating point numbers.
- There are no missing values in any features - the `Non-Null` is equal to the number of examples in the training set.

STEP 9. Feature and target histograms

Let us examine details of features and labels by calling info on the frame. So, when we call info we get the following information. So, there are 20,640 entries and there are total 9 columns in which there are 8 features and 1 target column. So, you can see that, there are 20,640, non-nulls in each of the column and the type of each of the columns is float 64.

So, we observed the dataset contains 20,640 examples with 8 features, all features are numerical features encoded as floating-point numbers, and there are no missing values in any feature. The non-nulls, are equal to number of examples in the training set.

(Refer Slide Time: 13:30)

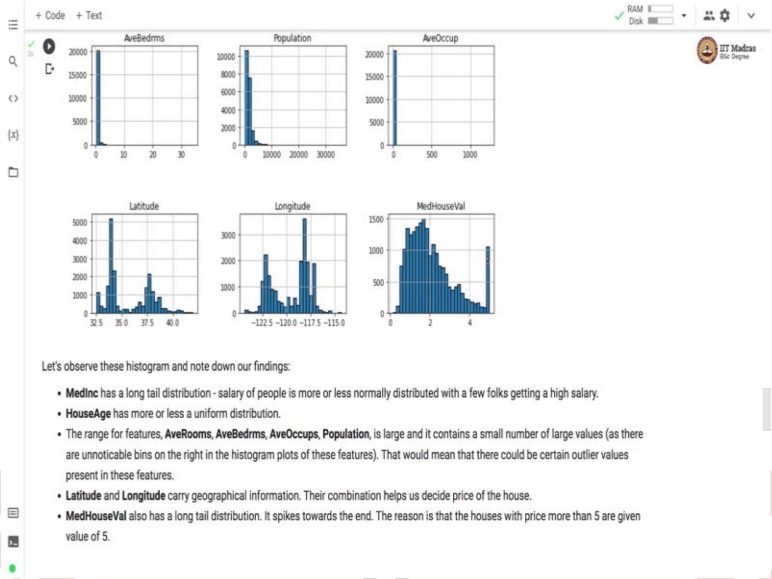


STEP 9. Feature and target histograms

Let's look at the distribution of these features and target by plotting their histograms.

```
[14]  import seaborn as sns
      import matplotlib.pyplot as plt
```

```
[15]  california_housing.frame.hist(figsize=(12, 10), bins=30, edgecolor="black")
      plt.subplots_adjust(hspace=0.7, wspace=0.4)
```



Let's observe these histogram and note down our findings:

- **MedInc** has a long tail distribution - salary of people is more or less normally distributed with a few folks getting a high salary.
- **HouseAge** has more or less a uniform distribution.
- The range for features, **AveRooms**, **AveBedrms**, **AveOccups**, **Population**, is large and it contains a small number of large values (as there are unnoticable bins on the right in the histogram plots of these features). That would mean that there could be certain outlier values present in these features.
- **Latitude** and **Longitude** carry geographical information. Their combination helps us decide price of the house.
- **MedHouseVal** also has a long tail distribution. It spikes towards the end. The reason is that the houses with price more than 5 are given value of 5.

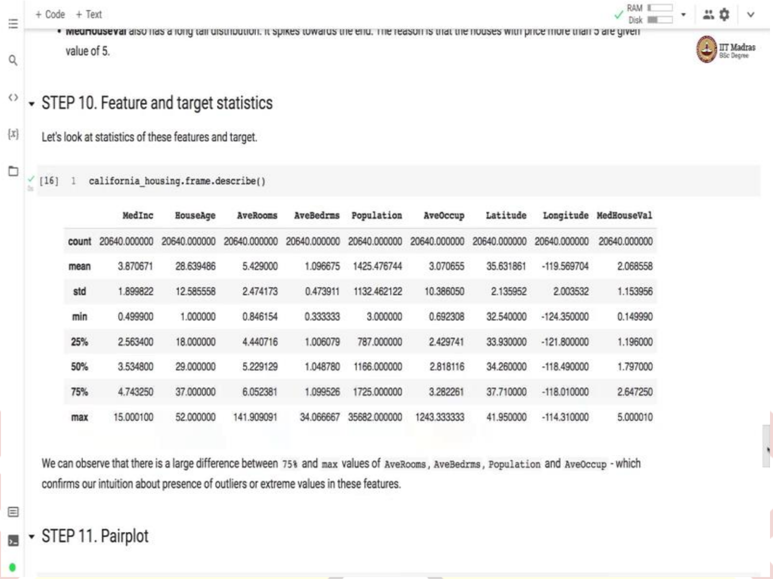
Let us look at a distribution of these features and target, by plotting the histograms. So, you can plot the histogram based on the frame, based on the content of the frame data frame. So here, we have plotted histograms for all 9 columns, which also include a target column, which is median house value.

So, we have to observe these histograms and write down our findings, which is one important step in the data exploration. So, what do we see from these from this figure, we can see that this median INC has a long tail distribution. So, the salary of people is more or less normally distributed, you can see a normal distribution type of things, but there is a long tail over here and that denotes that some people are getting higher salaries.

So, HouseAge is more or less a uniform distribution with a certain spikes in houses. And features like average room, average bedroom, population, average occupancy, the range is quite large and it contains a small number of large values as there are unnoticeable beams in these regions in each of these features. And this may well indicate that there could be certain outlier values present in these features.

Latitude and Longitude carry geographic information, and then combination helps us decide the price of the house. So, the target is also a long tail distribution, it spikes towards the end. And the reason for the spike is that houses with price more than 5 are given value of 5 so, they are essentially truncated at 500,000 US dollars. So, that is why this particular peak.

(Refer Slide Time: 15:54)



- **MedHouseVal** also has a long tail distribution. It spikes towards the end. The reason is that the houses with price more than 5 are given value of 5.

STEP 10. Feature and target statistics

Let's look at statistics of these features and target.

```
[16]  california_housing.frame.describe()
```

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	MedHouseVal
count	20640.000000	20640.000000	20640.000000	20640.000000	20640.000000	20640.000000	20640.000000	20640.000000	20640.000000
mean	3.870671	28.639486	5.429000	1.096675	1425.476744	3.070655	35.631861	-119.569704	2.068558
std	1.899822	12.585558	2.474173	0.473911	1132.462122	10.386050	2.135952	2.003532	1.153956
min	0.499900	1.000000	0.846154	0.333333	3.000000	0.692308	32.540000	-124.350000	0.149990
25%	2.563400	18.000000	4.440716	1.006079	787.000000	2.429741	33.930000	-121.800000	1.196000
50%	3.534800	29.000000	5.229129	1.048780	1166.000000	2.818116	34.260000	-118.490000	1.797000
75%	4.743250	37.000000	6.052381	1.099526	1725.000000	3.282261	37.710000	-118.010000	2.647250
max	15.000100	52.000000	141.909091	34.066667	35682.000000	1243.333333	41.950000	-114.310000	5.000010

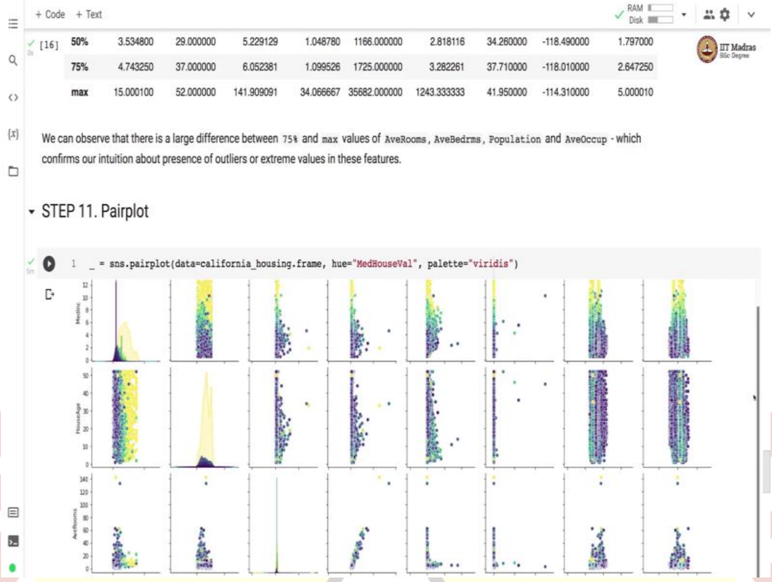
We can observe that there is a large difference between `75%` and `max` values of `AveRooms`, `AveBedrms`, `Population` and `AveOccup` - which confirms our intuition about presence of outliers or extreme values in these features.

STEP 11. Pairplot

Let us look at the statistics of features and target. And we can obtained the statistics by calling describe on the data frame containing both features and target. So, here, we get the count, the mean of each of the feature, what is the standard deviation and what are the values, what are the minimum and maximum values and what are the values at different quartiles, 25 percent, 50 percent and 75 percent.

You can see that in average rooms, average bedrooms, average occupancy and population there is significant difference, between seventy fifth percentile and maximum. And that is what confirms our intuition about presence of outliers or extreme values in these features based on examination of the histograms. You can see that in each of the feature the difference is quite large, between these two values.

(Refer Slide Time: 17:01)



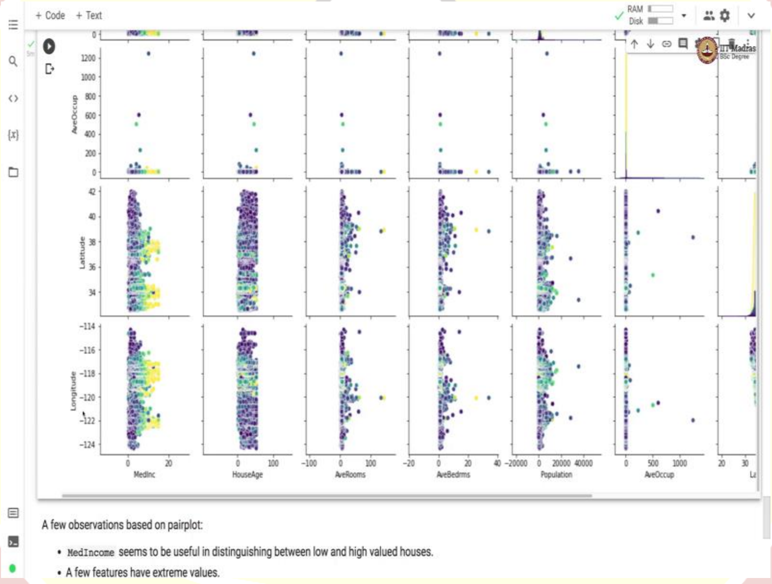
[16]

50%	3.534800	29.000000	5.229129	1.048780	1166.000000	2.818116	34.260000	-118.490000	1.797000
75%	4.743250	37.000000	6.052381	1.099526	1725.000000	3.282261	37.710000	-118.010000	2.647250
max	15.000100	52.000000	141.909091	34.066667	35682.000000	1243.333333	41.950000	-114.310000	5.000010

We can observe that there is a large difference between `75%` and `max` values of `AveRooms`, `AveBedrms`, `Population` and `AveOccup` - which confirms our intuition about presence of outliers or extreme values in these features.

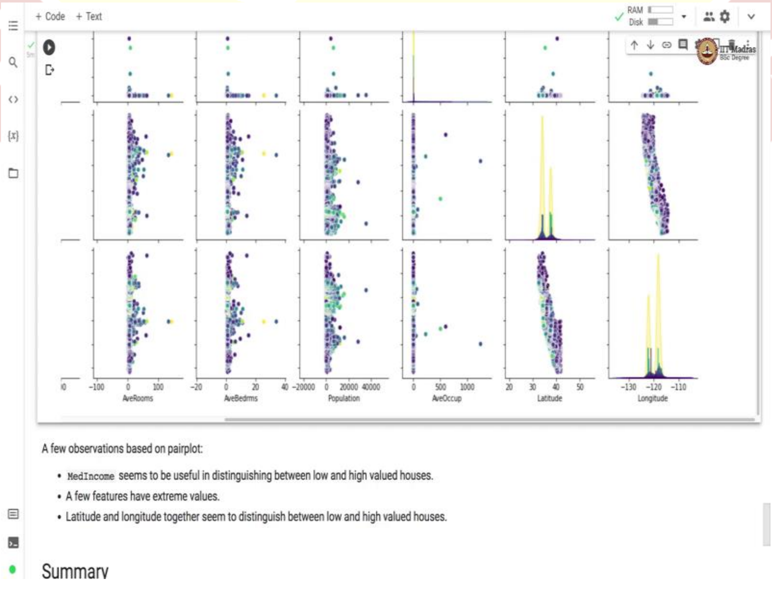
STEP 11. Pairplot

```
_ = sns.pairplot(data=california_housing.frame, hue="MedHouseVal", palette="viridis")
```



A few observations based on pairplot:

- `MedIncome` seems to be useful in distinguishing between low and high valued houses.
- A few features have extreme values.



A few observations based on pairplot:

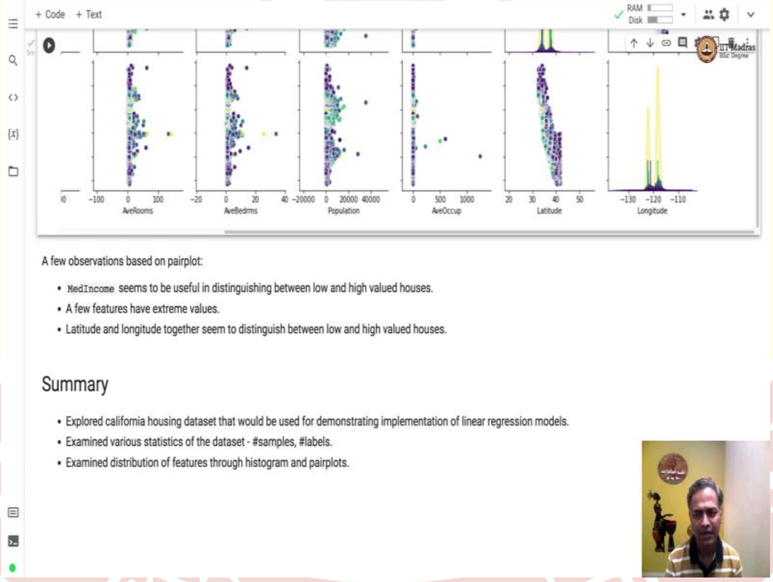
- `MedIncome` seems to be useful in distinguishing between low and high valued houses.
- A few features have extreme values.
- Latitude and longitude together seem to distinguish between low and high valued houses.

Summary

Finally, let us plot a pairplot which is plot between the features and what we have done is we have coloured this particular pairplot with the target which is median house value. So, pairplot contains a lot of data, it has got all the features on the x axis and all the features on the y axis and the way you have to read this is, so when you see this particular plot which has MedINC on x axis that is medium income on x axis and longitude on y axis.

So, here you can see that and the colors here are by the value of the colors are based on the median house value. So, there are 5 different colors that you will see and they are listed over here. So, you can see that median income is kind of indicative of the house price. In the same way, we can also see that the latitude longitude combined also gives us an indication of prices of the house. And you can see that there are certain features that have got extreme values something like average occupancy or population average bedroom they have some extreme values. And we have seen these multiple times through histogram plots as well as through these statistics.

(Refer Slide Time: 19:02)



A few observations based on pairplot:

- `MedIncome` seems to be useful in distinguishing between low and high valued houses.
- A few features have extreme values.
- Latitude and longitude together seem to distinguish between low and high valued houses.

Summary

- Explored california housing dataset that would be used for demonstrating implementation of linear regression models.
- Examined various statistics of the dataset - #samples, #labels.
- Examined distribution of features through histogram and pairplots.

So, that is it from this exploration. We explored California housing dataset that will be useful for linear regression demonstration, we examine various features, various statistics of the dataset, looked at number of samples number of labels, we examine distributions of features to histogram and pairplot.

So, hope you now have an idea about how to examine any given dataset. So, there are certain steps that we took in examining this dataset, which can be broadly applied to any other dataset. So, in the next video, we will use this dataset and build our first linear regression model with sklearn API's. Till then, thank you and Namaste.