Kmit

SONET

Data Science

(Level-I)

Machine Learning

Data Preprocessing

Steps for Machine Learning

Look at the big picture.

Get the data.

Files/web/pandas

Discover and visualize the data to gain insights.

Prepare the data for Machine Learning algorithms.

Select a model and train it.
Train the model

Fine-tune your model.

Present your solution.

Launch, monitor, and maintain your system

Look at the big picture.

Selection of Domain

Data Set: California census data

Metrics for each block group:

Population, median income, median housing price, …

Problem:

Build a model to learn from this data and be able to predict the **median housing price** in any district, given median_income

As a Data Scientist , follow the steps of machine learning by starting with:

Frame the Problem

Select the Performance Measure

Check the Assumptions

Look at the big picture.

Frame the Problem

Supervised/Unsupervised/reinforcement/classification/regressions/?

Batch Learning / Online Learning?

Supervised Learning – Labelled data is available

Regression – Asked to predict a value

Single regression problem – use median income to predict median_house_price

Batch learning is fine – small data, no rapid changes, no continues flow

Look at the big picture.

Select Performance Measure

There are various performance measures for each model.

We selected the regression model for our problem

Root Mean Square Error(RMSE)

$$\mathrm{RMSE}(\mathbf{X}, h) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(h(\mathbf{x}^{(i)}) - y^{(i)}\right)^2}$$

Mean Absolute Error (MAE)

$$\mathrm{MAE}(\mathbf{X}, h) = \frac{1}{m}\sum_{i=1}^{m}\left| h(\mathbf{x}^{(i)}) - y^{(i)} \right|$$

m – Number of observations

h – Hypothesis function

h(x(i)) – Predicted Value for ith instance

y(i) – Actual Value for ith instance

RMSE is more sensitive to outliers than the MAE.

Look at the big picture.

Check Assumptions

Make the assumptions that what they can expect in different way.

There is the chance of expectation of result as “cheap”, “reasonable”, “ Costly”

If so, there will be the wastage of working hours and efforts

Discuss with management and team and get clarity on this assumption

Lets continue with regression for now….

Get Data

Downloading

Loading

Creating Training and test data

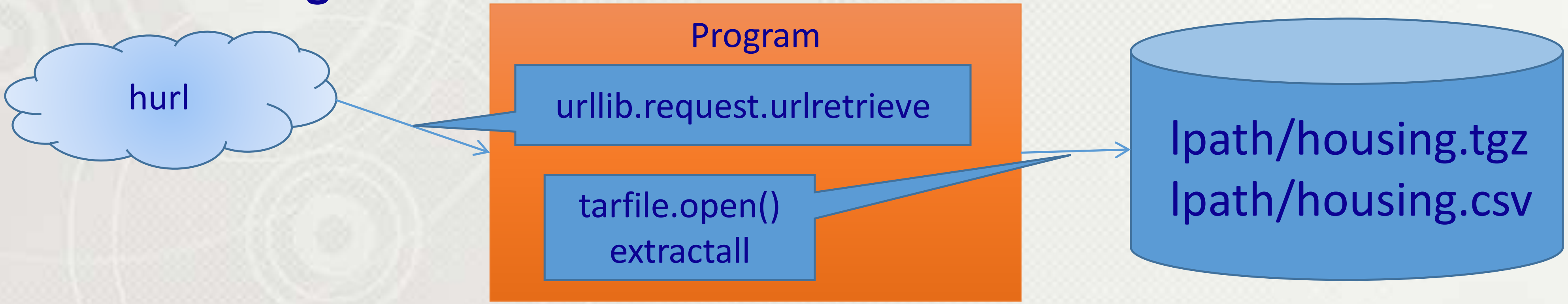
Get Data

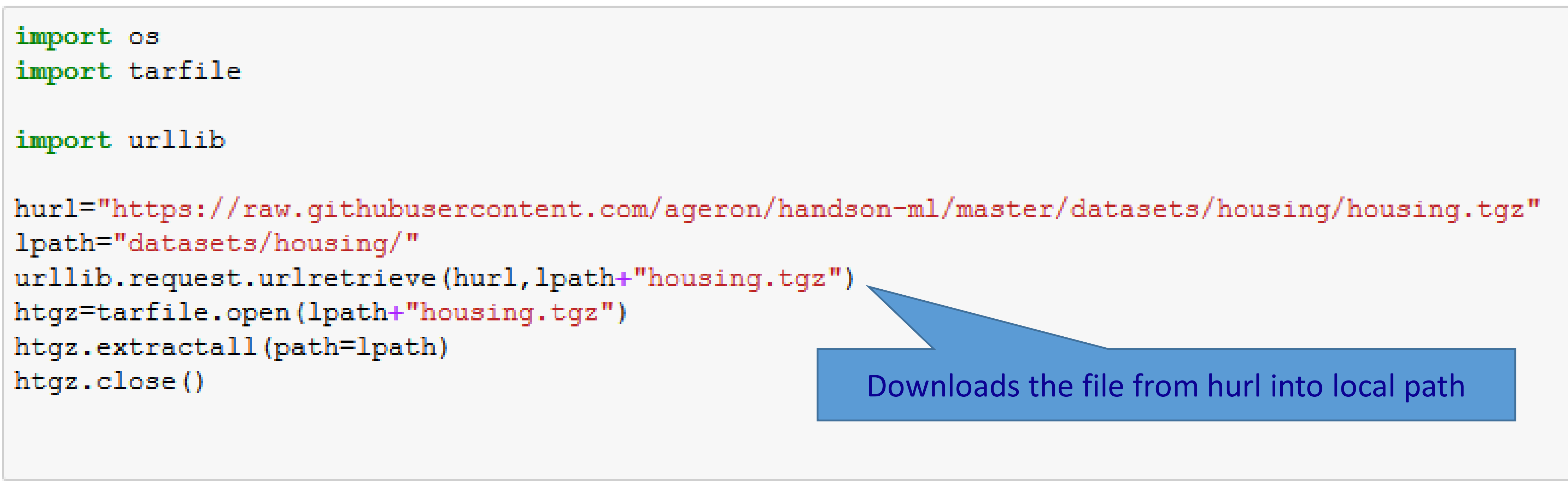
Downloading

- Data may be available in various forms/ sources
  - Web
  - Files
  - Databases
- Lets take CSV from given url.

Get Data

Downloading



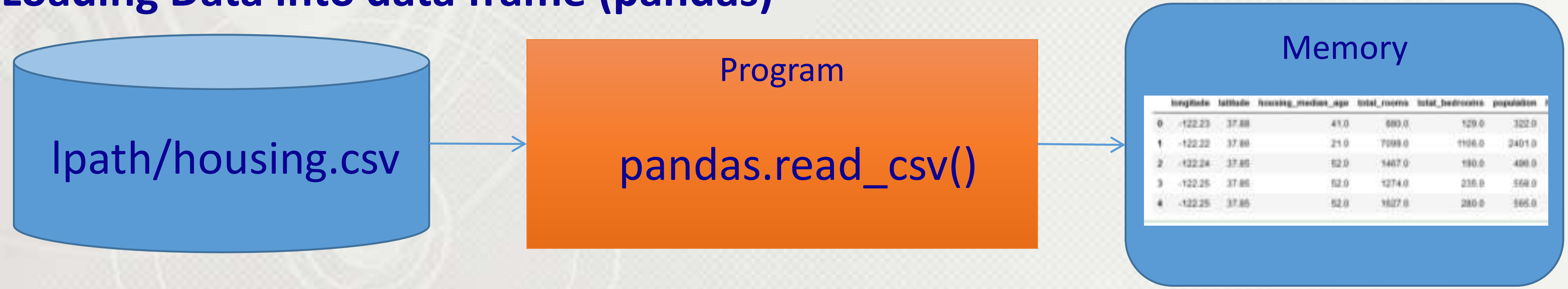


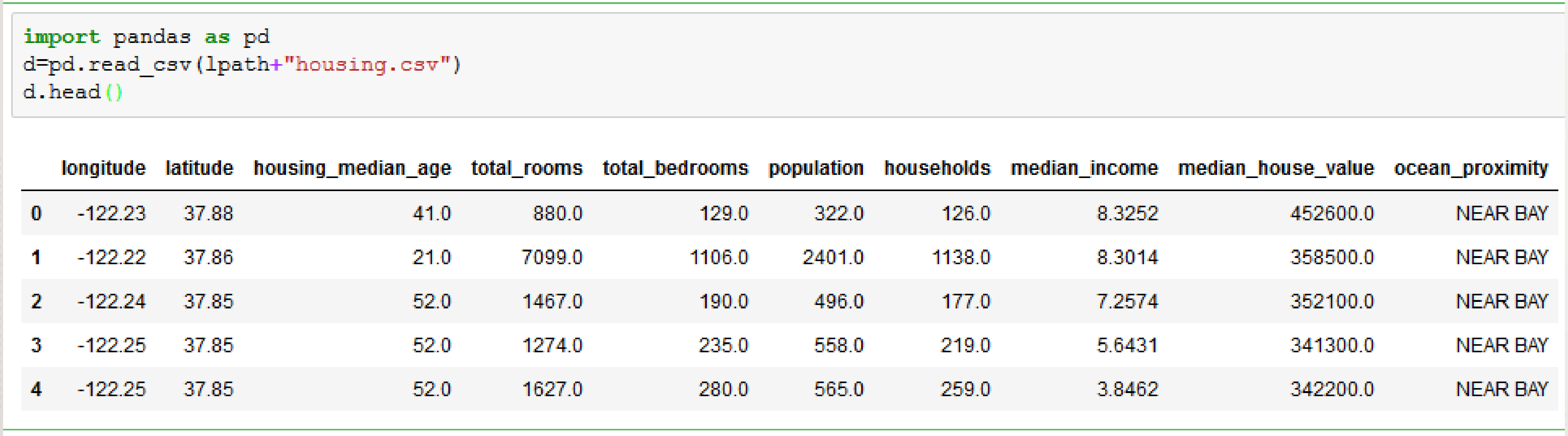
```
import os
import tarfile

import urllib

hurl="https://raw.githubusercontent.com/ageron/handson-ml/master/datasets/housing/housing.tgz"
lpath="datasets/housing/"
urllib.request.urlretrieve(hurl,lpath+"housing.tgz")
htgz=tarfile.open(lpath+"housing.tgz")
htgz.extractall(path=lpath)
htgz.close()
```

Downloads the file from hurl into local path

Get Data

Loading Data into data frame (pandas)





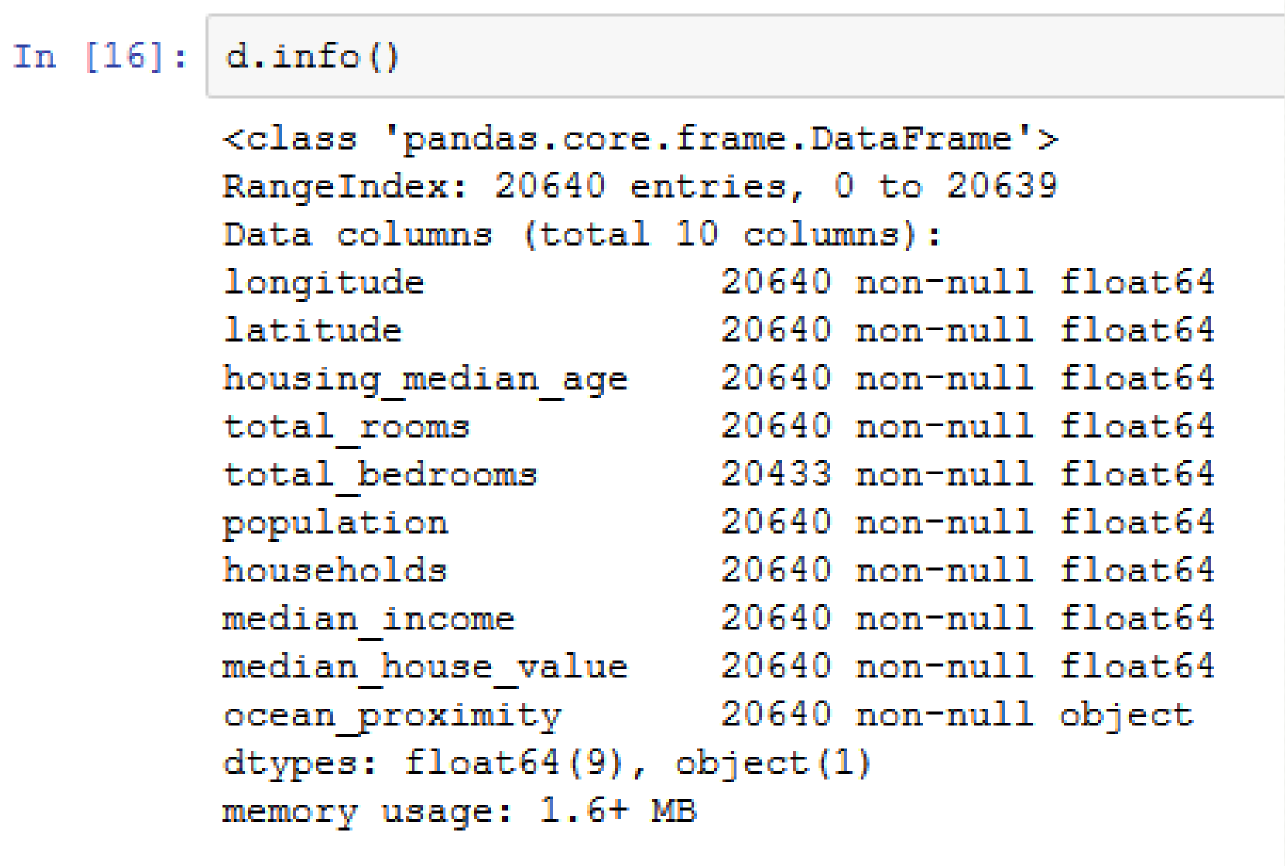
```
import pandas as pd
d=pd.read_csv(lpath+"housing.csv")
d.head()
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	452600.0	NEAR BAY
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	358500.0	NEAR BAY
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	352100.0	NEAR BAY
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	341300.0	NEAR BAY
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	342200.0	NEAR BAY

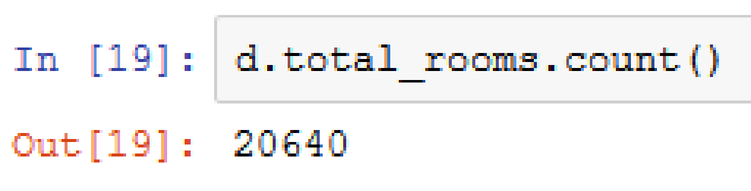
Get Data

Loading Data into data frame (pandas)

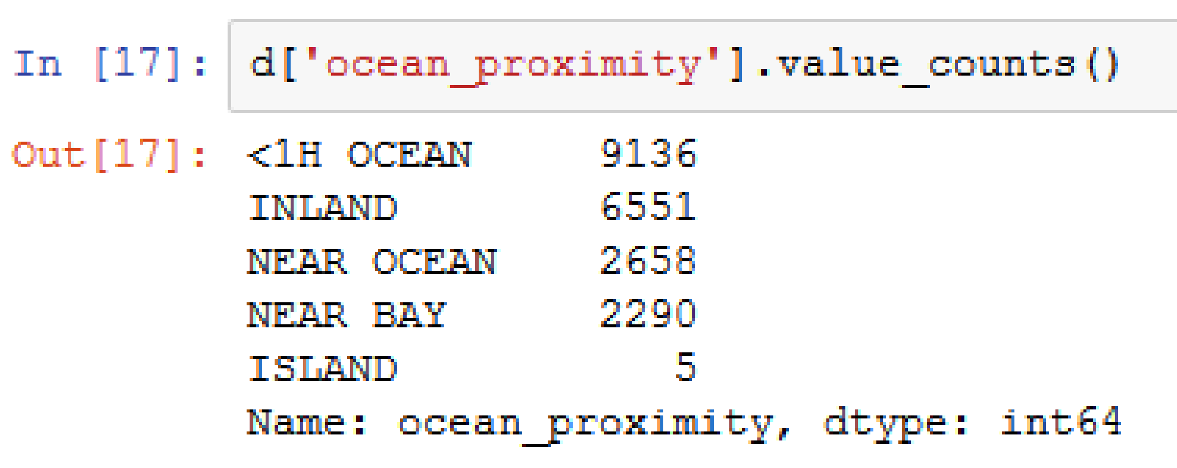
Let’s see the various statistics of loaded dataframe



```
In [16]: d.info()

         <class 'pandas.core.frame.DataFrame'>
         RangeIndex: 20640 entries, 0 to 20639
         Data columns (total 10 columns):
         longitude             20640 non-null float64
         latitude              20640 non-null float64
         housing_median_age    20640 non-null float64
         total_rooms           20640 non-null float64
         total_bedrooms        20433 non-null float64
         population            20640 non-null float64
         households            20640 non-null float64
         median_income         20640 non-null float64
         median_house_value    20640 non-null float64
         ocean_proximity       20640 non-null object
         dtypes: float64(9), object(1)
         memory usage: 1.6+ MB
```



```
In [19]: d.total_rooms.count()

Out[19]: 20640
```

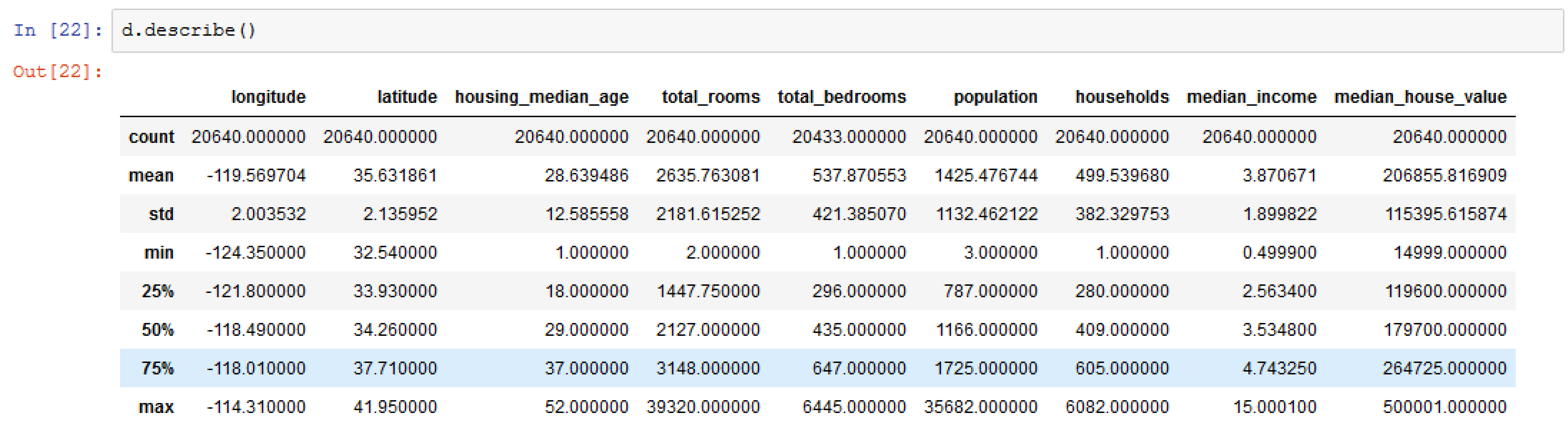


```
In [17]: d['ocean_proximity'].value_counts()

Out[17]: <1H OCEAN     9136
         INLAND        6551
         NEAR OCEAN    2658
         NEAR BAY      2290
         ISLAND           5
         Name: ocean_proximity, dtype: int64
```

Get Data

Loading Data into data frame (pandas)

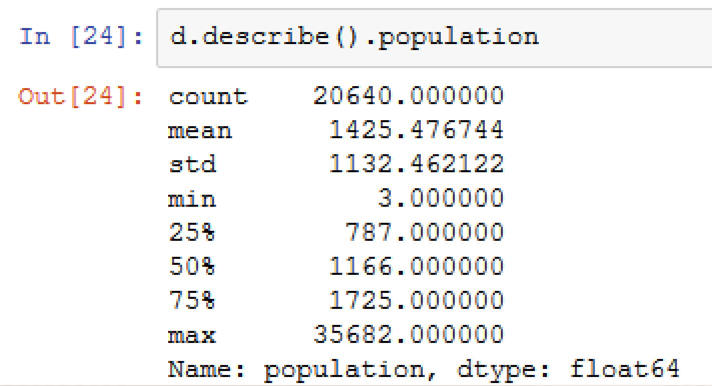
Let’s see the various statistics of loaded dataframe



```
In [22]: d.describe()
```

Out[22]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
count	20640.000000	20640.000000	20640.000000	20640.000000	20433.000000	20640.000000	20640.000000	20640.000000	20640.000000
mean	-119.569704	35.631861	28.639486	2635.763081	537.870553	1425.476744	499.539680	3.870671	206855.816909
std	2.003532	2.135952	12.585558	2181.615252	421.385070	1132.462122	382.329753	1.899822	115395.615874
min	-124.350000	32.540000	1.000000	2.000000	1.000000	3.000000	1.000000	0.499900	14999.000000
25%	-121.800000	33.930000	18.000000	1447.750000	296.000000	787.000000	280.000000	2.563400	119600.000000
50%	-118.490000	34.260000	29.000000	2127.000000	435.000000	1166.000000	409.000000	3.534800	179700.000000
75%	-118.010000	37.710000	37.000000	3148.000000	647.000000	1725.000000	605.000000	4.743250	264725.000000
max	-114.310000	41.950000	52.000000	39320.000000	6445.000000	35682.000000	6082.000000	15.000100	500001.000000

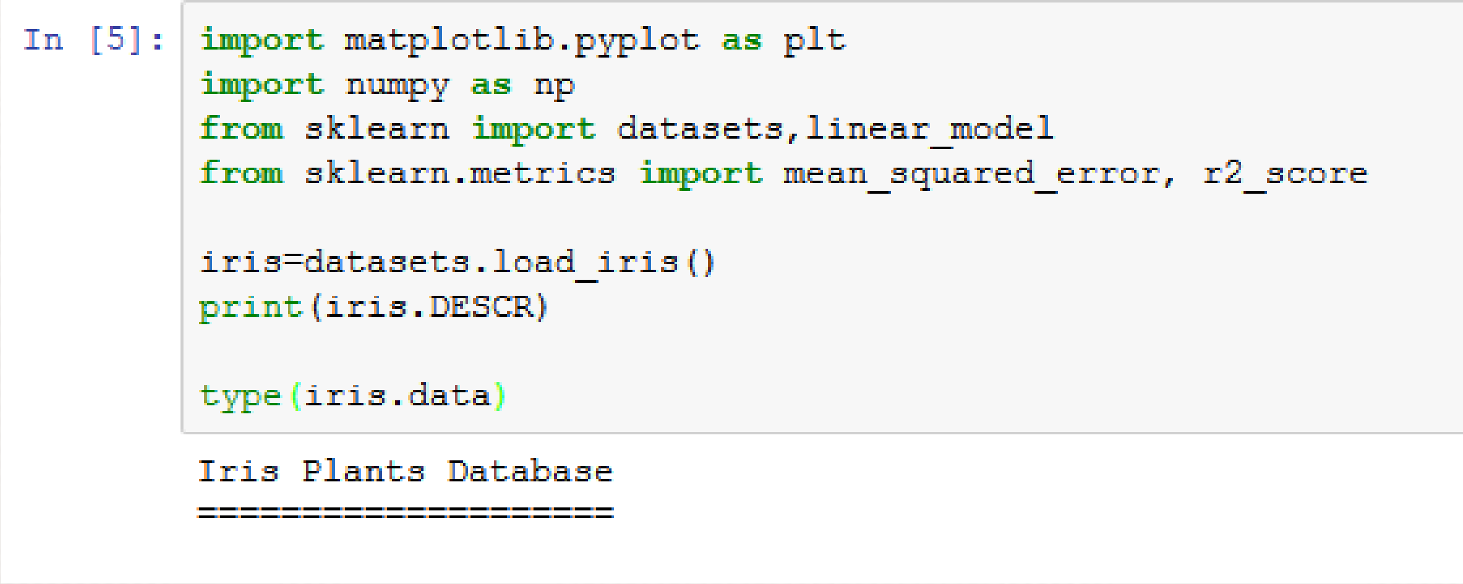


```
In [24]: d.describe().population

Out[24]: count    20640.000000
         mean      1425.476744
         std       1132.462122
         min          3.000000
         25%        787.000000
         50%       1166.000000
         75%       1725.000000
         max      35682.000000
         Name: population, dtype: float64
```

Get Data

Sample datasets in sklearn

load_boston([return_X_y])	Load and return the boston house-prices dataset (regression).
load_iris([return_X_y])	Load and return the iris dataset (classification).
load_diabetes([return_X_y])	Load and return the diabetes dataset (regression).
load_digits([n_class, return_X_y])	Load and return the digits dataset (classification).
load_linnerud([return_X_y])	Load and return the linnerud dataset (multivariate regression).
load_wine([return_X_y])	Load and return the wine dataset (classification).
load_breast_cancer([return_X_y])	Load and return the breast cancer wisconsin dataset (classification).

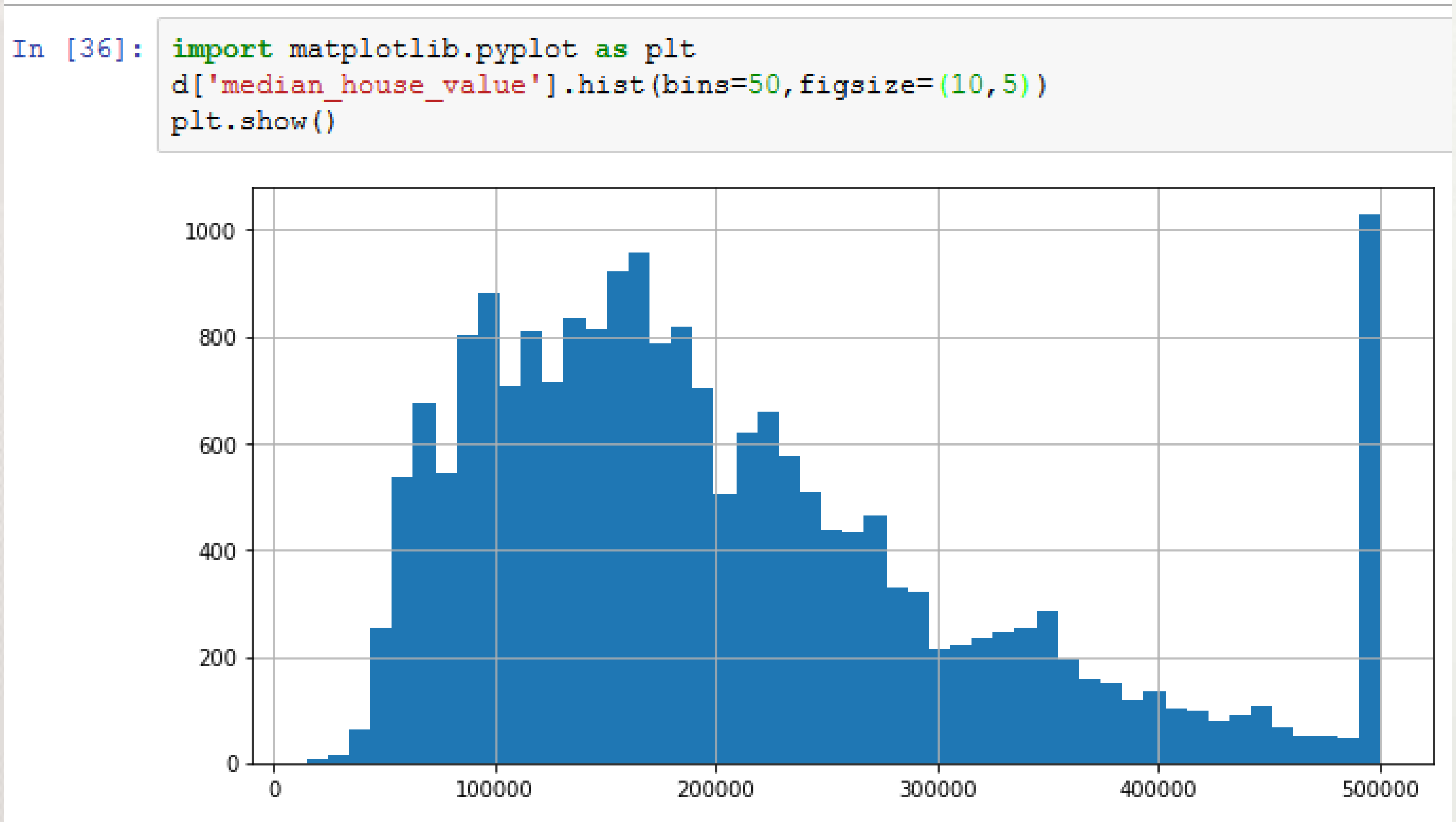


```
In [5]: import matplotlib.pyplot as plt
        import numpy as np
        from sklearn import datasets,linear_model
        from sklearn.metrics import mean_squared_error, r2_score

        iris=datasets.load_iris()
        print(iris.DESCR)

        type(iris.data)
```

```
Iris Plants Database
====================
```

Visualize data

matplotlib is a library to plot graphs and visualize data



```
In [36]: import matplotlib.pyplot as plt
         d['median_house_value'].hist(bins=50,figsize=(10,5))
         plt.show()
```

Conclusion

Covered Initial two steps of Data Science Project

Look at Big Picture

Getting Data

Next:

Pre-Processing

THANK YOU