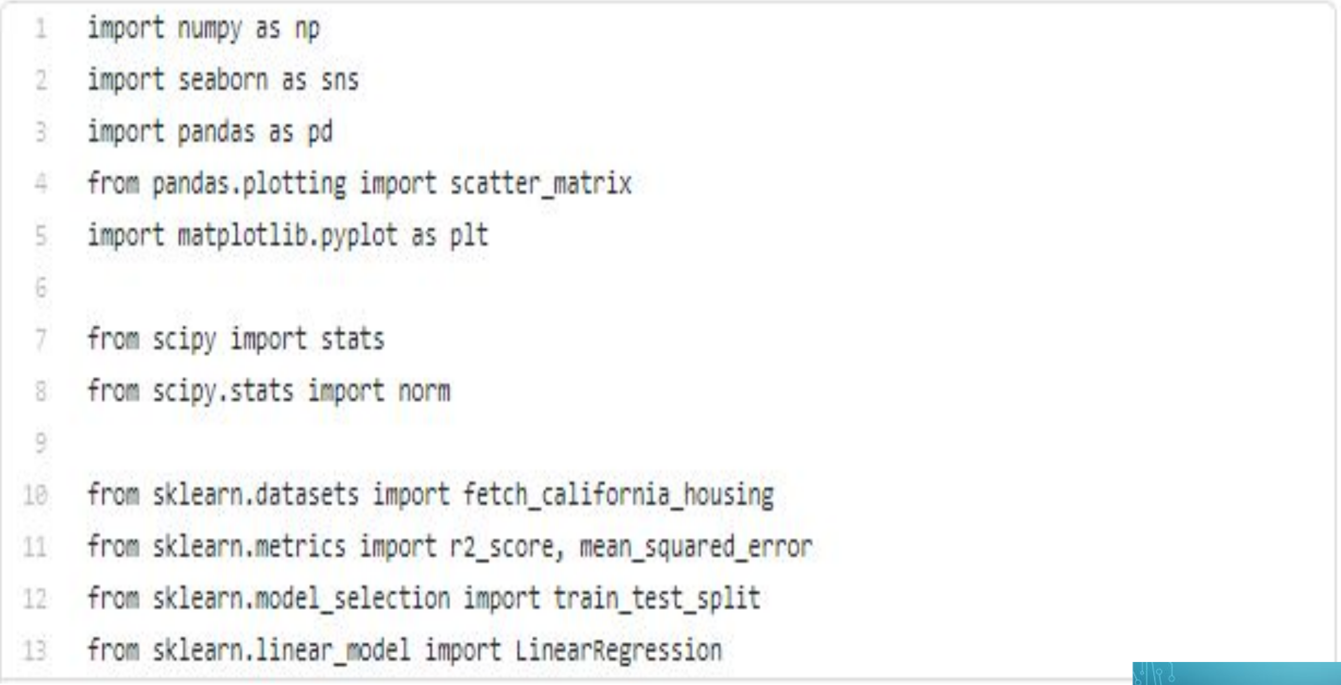
Autocorrelation

- A correlation analysis provides information on the strength and direction of the linear relationship between two variables.
- Autocorrelation refers to the degree of correlation between the values of the same variables across different observations in the data.
- For example, one might expect the air temperature on the 1st day of the month to be more similar to the temperature on the 2nd day compared to the 31st day.
- If the temperature values that occurred closer together in time are, in fact, more similar than the temperature values that occurred farther apart in time, the data would be autocorrelated.

Implementing Autocorrelation

- To showcase some of the concepts previously introduce, we implemented a linear regression model onto the California housing dataset. Here is the code along with a brief explanation for each block.
- First, we import the required libraries.



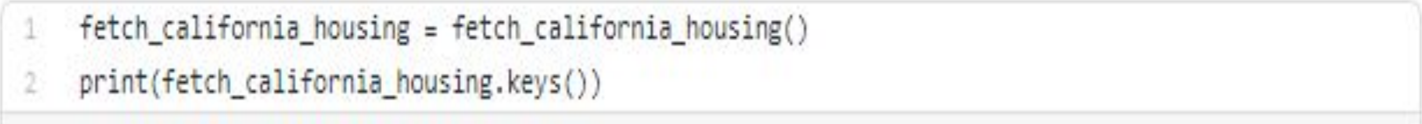
```
import numpy as np
import seaborn as sns
import pandas as pd
from pandas.plotting import scatter_matrix
import matplotlib.pyplot as plt

from scipy import stats
from scipy.stats import norm

from sklearn.datasets import fetch_california_housing
from sklearn.metrics import r2_score, mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

California Housing Data Set- Details

Column title	Description	Range*	Datatype
longitude	A measure of how far west a house is; a higher value is farther west	•Longitude values range from -180 to +180 •Data set min: -124.3 •Data set max: -114.3	float64
latitude	A measure of how far north a house is; a higher value is farther north	•Latitude values range from -90 to +90 •Data set min: 32.5 •Data set max: 42.5	float64
housingMedianAge	Median age of a house within a block; a lower number is a newer building	•Data set min: 1.0 •Data set max: 52.0	float64
totalRooms	Total number of rooms within a block	•Data set min: 2.0 •Data set max: 37937.0	float64
totalBedrooms	Total number of bedrooms within a block	•Data set min: 1.0 •Data set max: 6445.0	float64
population	Total number of people residing within a block	•Data set min: 3.0 •Data set max: 35682.0	float64
households	Total number of households, a group of people residing within a home unit, for a block	•Data set min: 1.0 •Data set max: 6082.0	float64
medianIncome	Median income for households within a block of houses (measured in tens of thousands of US Dollars)	•Data set min: 0.5 •Data set max: 15.0	float64
medianHouseValue	Median house value for households within a block (measured in US Dollars)	•Data set min: 14999.0 •Data set max: 500001.0	float64

Implementing Autocorrelation…

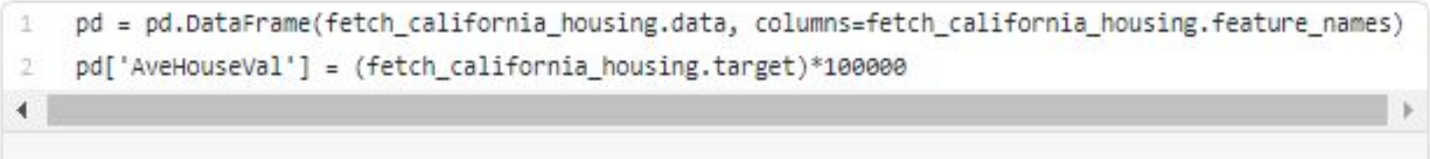
- Next, we load the housing data from the scikit-learn library :



```
fetch_california_housing = fetch_california_housing()
print(fetch_california_housing.keys())
```

- To know more about the features, we print (california_housing_dataset.DESCR)

- These are eight independent variables based on which we can predict the value of the house. The prices of the house are indicated by the variable AveHouseVal , which defines our dependent variable.
- We now load the data into a pandas dataframe using pd.DataFrame



```
pd = pd.DataFrame(fetch_california_housing.data, columns=fetch_california_housing.feature_names)
pd['AveHouseVal'] = (fetch_california_housing.target)*100000
```

Implementing Autocorrelation…

- **Data preprocessing**

  After loading the data, it’s a good practice to see if there are any missing values in the data. We count the number of missing values (none for this dataset) for each feature using isnull() .

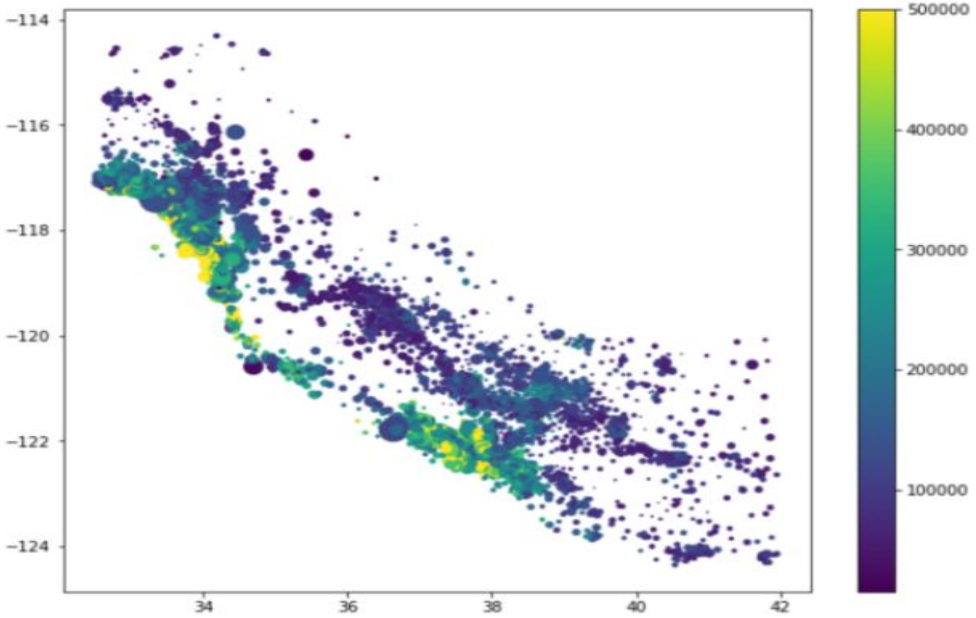
  ```
  pd.isnull().sum()
  ```

- **Exploratory Data Analysis**

  Let’s first plot the distribution of the target variable AveHouseVal depending on Latitude and Longitude . The image is supposed to plot the state of California, USA. As observed, houses located close to the sea are more expensive than the rest.

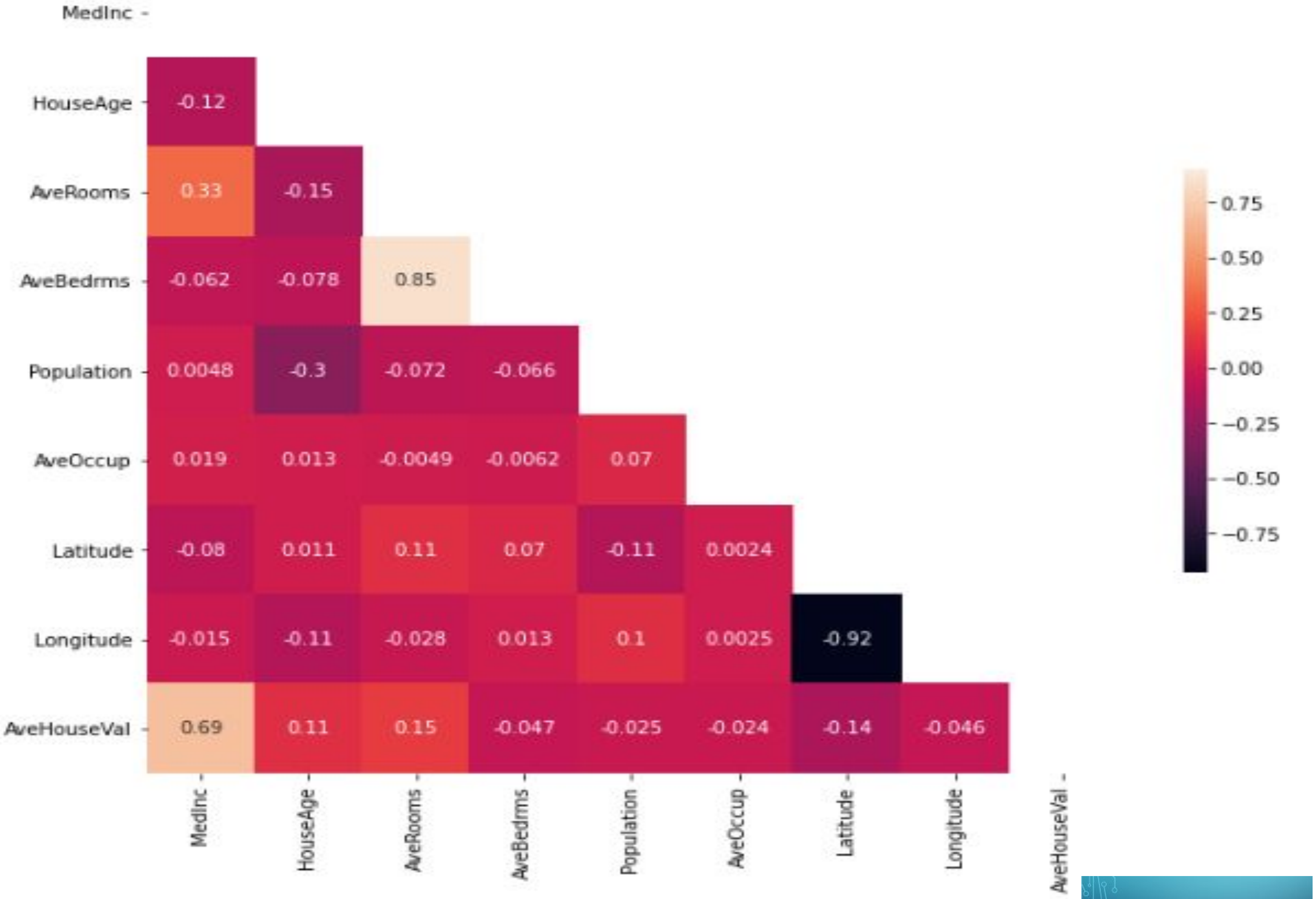
  ```
  plt.figure(figsize=(10,8))
  plt.scatter(pd['Latitude'], pd['Longitude'],
  c=pd['AveHouseVal'], s=pd['Population']/100)
  plt.colorbar()
  ```

Correlation matrix



Next, we create a correlation matrix that measures the linear relationships between the variables.

Next, we create a correlation matrix that measures the linear relationships between the variables in Boston-Housing dataset.



	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	AveHouseVal
MedInc									
HouseAge	-0.12								
AveRooms	0.33	-0.15							
AveBedrms	-0.062	-0.078	0.85						
Population	0.0048	-0.3	-0.072	-0.066					
AveOccup	0.019	0.013	-0.0049	-0.0062	0.07				
Latitude	-0.08	0.011	0.11	0.07	-0.11	0.0024			
Longitude	-0.015	-0.11	-0.028	0.013	0.1	0.0025	-0.92		
AveHouseVal	0.69	0.11	0.15	-0.047	-0.025	-0.024	-0.14	-0.046	

Observations of Autocorrelation:

- To fit a linear regression model, we select those features that have a high correlation with our dependent variable AveHouseVal.
  - By looking at the correlation matrix we can see that MediaInc has a strong positive correlation with AverageHouseVal (0.69).
  - The other two variables with highest correlation are HouseAve and AveRooms .

An important point when selecting features for a linear regression model is to check for multicollinearity.

- For example, the features Latitude and Longitude have 0.92 correlation, so we should not include both of them simultaneously in our regression model.
- Since the correlation between the variables MediaInc , HouseAve and AveRooms is not high, we consider those three variables for our regression model.

Implementing Autocorrelation…

Training and testing the model

We use scikit-learn’s LinearRegression to train our model on both the training and test sets.

The final step is to evaluate the performance of the algorithm. This step is particularly important to compare how well different algorithms perform on a particular dataset.

What is Multicollinearity?

- *"Multicollinearity occurs when two or more independent variables are highly correlated with one another in a regression model."*

- Let's take a simple example from our everyday life to explain this. Som loves watching television while munching on chips. The more television he watches, the more chips he eats and the happier he gets!
- Now, if we could quantify happiness and measure Som's happiness while he's busy doing his favorite activity, which do you think would have a greater impact on his happiness? Having chips or watching television?
- That's difficult to determine because the moment we try to measure Som's happiness from eating chips, he starts watching television. And the moment we try to measure his happiness from watching television, he starts eating chips.
- Eating chips and watching television are highly correlated in the case of Som and we cannot individually determine the impact of the individual activities on his happiness. This is the multicollinearity problem!

1. Explain Multi linearity with example ?

2. How do we print Mean Absolute Error,
   Mean Squared Error, Root Mean Squared Error ?

3. Define Autocorrelation ?

4. Show the implementation of Autocorrelation with a program?