

```
In [39]: import numpy as np
import pandas as pd
from sklearn.datasets import fetch_california_housing
from sklearn.model_selection import train_test_split
housing=fetch_california_housing()
housing

Out[39]: {'data': array([[ 8.3252, 41.0, 6.98412698, ..., 2.55555556,
    [ 37.88, -122.23 ],
    [ 8.3014, 21.0, 6.23813708, ..., 2.10984183,
    [ 37.86, -122.22 ],
    [ 7.2574, 52.0, 8.28813559, ..., 2.80225989,
    [ 37.85, -122.24 ],
    ...,
    [ 1.7, 17.0, 5.20554273, ..., 2.3256351,
    [ 39.43, -121.22 ],
    [ 1.8672, 18.0, 5.32951289, ..., 2.12320917,
    [ 39.43, -121.32 ],
    [ 2.3886, 16.0, 5.25471698, ..., 2.61698113,
    [ 39.37, -121.24 ]]),
'target': array([4.526, 3.585, 3.521, ..., 0.923, 0.847, 0.894]),
'frame': None,
'target_names': ['MedHouseVal'],
'feature_names': ['MedInc',
'HouseAge',
'AveRooms',
'AveBedrms',
'Population',
'AveOccup',
'Latitude',
'Longitude'],
'DESCR': '\n\nCalifornia Housing dataset\n-----\n\n**Data Set Characteristics:**\n\n :
Number of Instances: 20640\n\n :Number of Attributes: 8 numeric, predictive attributes and the target\n\n :Attribute Information:\n
- MedInc median income in block group\n - HouseAge median house age in block group\n - AveRooms average number
of rooms per household\n - AveBedrms average number of bedrooms per household\n - Population block group population\n
- AveOccup average number of household members\n - Latitude block group latitude\n - Longitude block group longitu
de\n\n :Missing Attribute Values: None\n\nThis dataset was obtained from the StatLib repository.\nhttps://www.dcc.fc.up.pt/~ltorgo/Regressi
on/cal_housing.html\n\nThe target variable is the median house value for California districts,\nexpressed in hundreds of thousands of dollars
($100,000).\n\nThis dataset was derived from the 1990 U.S. census, using one row per census\nblock group. A block group is the smallest geogra
phical unit for which the U.S.\nCensus Bureau publishes sample data (a block group typically has a population\nof 600 to 3,000 people).\n\nAn
household is a group of people residing within a home. Since the average\nnumber of rooms and bedrooms in this dataset are provided per househ
old, these\ncolumns may take surprisingly large values for block groups with few households\nand many empty houses, such as vacation resort
s.\n\nIt can be downloaded/loaded using the\nfunc:`sklearn.datasets.fetch_california_housing` function.\n\n.. topic:: References\n\n - Pac
e, R. Kelley and Ronald Barry, Sparse Spatial Autoregressions,\n Statistics and Probability Letters, 33 (1997) 291-297\n'}
```

```
In [10]: housing_df=pd.DataFrame(housing['data'],columns=housing["feature_names"])
```

```
In [11]: housing_df
```

Out[11]:

	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
...
20635	1.5603	25.0	5.045455	1.133333	845.0	2.560606	39.48	-121.09
20636	2.5568	18.0	6.114035	1.315789	356.0	3.122807	39.49	-121.21
20637	1.7000	17.0	5.205543	1.120092	1007.0	2.325635	39.43	-121.22
20638	1.8672	18.0	5.329513	1.171920	741.0	2.123209	39.43	-121.32
20639	2.3886	16.0	5.254717	1.162264	1387.0	2.616981	39.37	-121.24

20640 rows × 8 columns

```
In [13]: housing_df["MedHouseVal"] = housing["target"]
```

Exploratory data analysis on the dataset

```
In [14]: housing_df.head()
```

Out[14]:

	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

```
In [18]: housing_df.info() #info about the type of data in the dataset
```

```
<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
```

```
In [19]: housing_df.describe() #description of dataset
```

Out[19]:

	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

```
In [34]: housing_df["target"] = housing["target"]
```

```
In [35]: housing_df
```

Out[35]:

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	MedHouseVal	target
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23	4.526	4.526
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22	3.585	3.585
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24	3.521	3.521
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25	3.413	3.413
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25	3.422	3.422
...
20635	1.5603	25.0	5.045455	1.133333	845.0	2.560606	39.48	-121.09	0.781	0.781
20636	2.5568	18.0	6.114035	1.315789	356.0	3.122807	39.49	-121.21	0.771	0.771
20637	1.7000	17.0	5.205543	1.120092	1007.0	2.325635	39.43	-121.22	0.923	0.923
20638	1.8672	18.0	5.329513	1.171920	741.0	2.123209	39.43	-121.32	0.847	0.847
20639	2.3886	16.0	5.254717	1.162264	1387.0	2.616981	39.37	-121.24	0.894	0.894

20640 rows × 10 columns

```
In [36]: housing_df=housing_df.drop("MedHouseVal", axis=1)
```

```
In [37]: housing_df
```

Out[37]:

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	target
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
...
20635	1.5603	25.0	5.045455	1.133333	845.0	2.560606	39.48	-121.09	0.781
20636	2.5568	18.0	6.114035	1.315789	356.0	3.122807	39.49	-121.21	0.771
20637	1.7000	17.0	5.205543	1.120092	1007.0	2.325635	39.43	-121.22	0.923
20638	1.8672	18.0	5.329513	1.171920	741.0	2.123209	39.43	-121.32	0.847
20639	2.3886	16.0	5.254717	1.162264	1387.0	2.616981	39.37	-121.24	0.894

20640 rows × 9 columns

Applying Regression Algorithm

```
In [47]: #import algorithm
from sklearn.linear_model import Ridge
np.random.seed(42)

#create data
x=housing_df.drop("target",axis=1)
y=housing_df["target"] #Median house price in $100,1000s

#Split data into train and test set
x_train,x_test,y_train,y_test= train_test_split(x,y, test_size=0.7)
```

```
In [48]: #Instantiate and fit the model
model=Ridge()
model.fit(x_train,y_train)
```

```
Out[48]: Ridge()
```

```
In [49]: #Checking score on test set
model.score(x_test,y_test)
```

```
Out[49]: 0.6031342051497797
```

```
In [ ]:
```