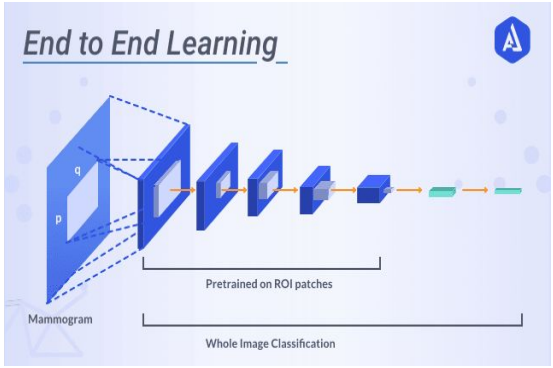
End to End Project

CS550 - End to End Project

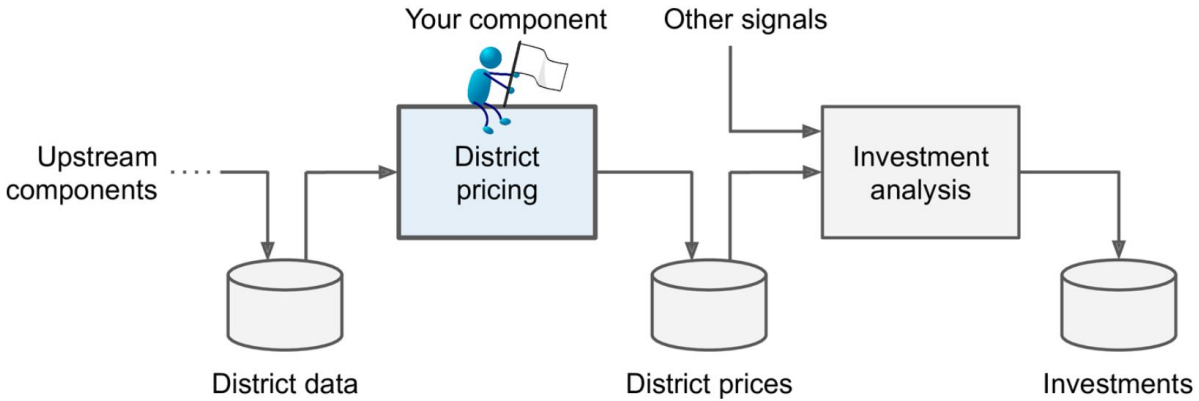


Submitted by: Divya Pandey(19665)
Instructor: Dr. Henry Chang

Table of Content

Introduction

Predicting median housing prices in **Silicon Valley** can help determine if a district is worth investing in. However, other factors should also be considered, and human expertise is crucial in real estate investing.



Introduction

★ End-to-end machine learning projects refer to the process of developing a machine learning model from data preparation to deployment in a production environment.

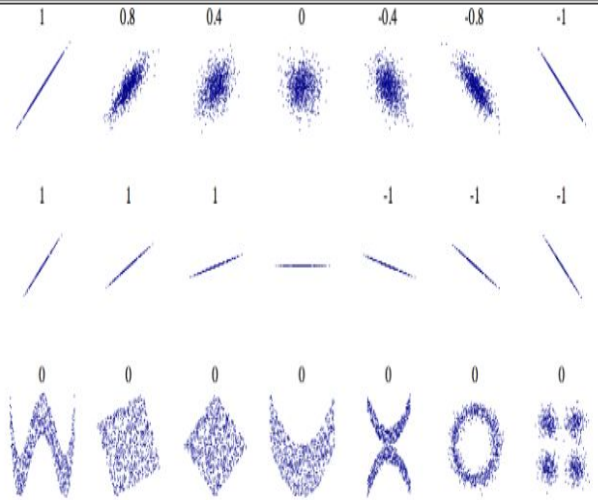
★ It involves several steps, including data collection, preprocessing, feature engineering, model selection, training, and evaluation. The final step is to deploy the model in a production environment where it can be used to make predictions on new data.

Design

- ★ Get the Data
- ★ Create Test Set
- ★ Discover and visualize the data to gain insights
- ★ Prepare the data for Machine Learning algorithms
- ★ Select and train a model
- ★ Fine-tune your model
- ★ Randomized Search
- ★ Analyze the Best Models and Their Errors
- ★ Evaluate Your System on the Test Set

Design

Correlation vs. Linear Regression

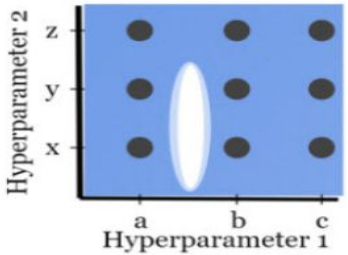


With dominant Hyperparameters (e.g., Hyperparameter 1 is more important) - use Random Search

Grid Search

Pseudocode

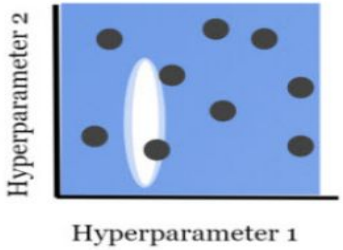
```
Hyperparameter_One = [a, b, c]
Hyperparameter_Two = [x, y, z]
```



Random Search

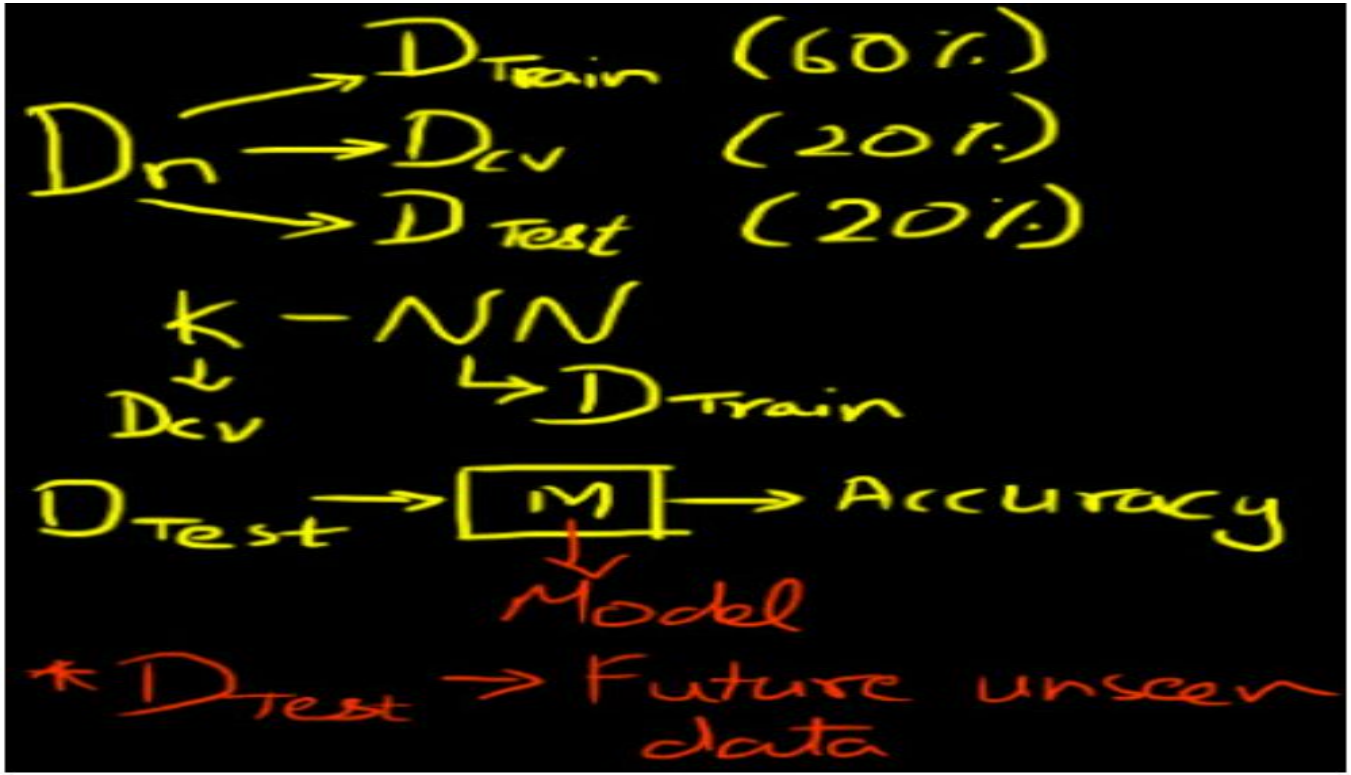
Pseudocode

```
Hyperparameter_One = random.num(range)
Hyperparameter_Two = random.num(range)
```



- Conditions that Random Search is better

Design


D_{Train} (60%)
D_n
D_{cv} (20%)
D_{Test} (20%)
k - NN
D_{cv}
D_{Train}
D_{Test}
M
Accuracy
Model
* D_{Test} → Future unseen data

Implementation

- ★ Go to Colab
- ★ Execute ete.ipynb

Test

- **Download Data**
  - https://raw.githubusercontent.com/ageron/handson-ml2/master/
  - **Load Data by Pandas**
    - pandas.read_csv("housing.csv")

	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

Test

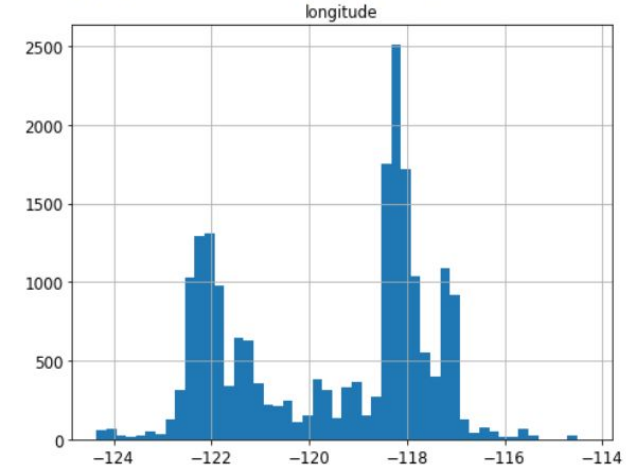
```
housing.describe()
```

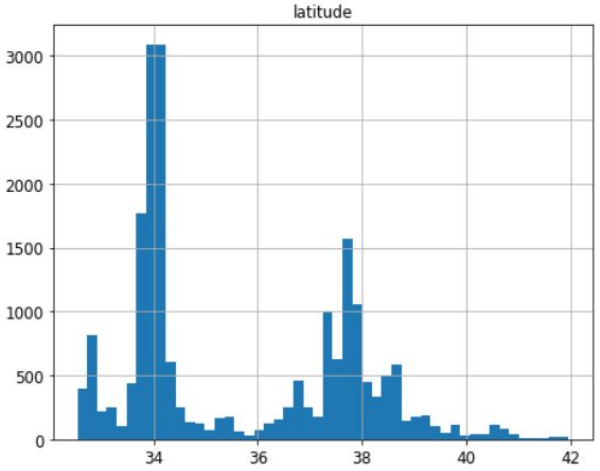
	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
count	20640.000000	20640.000000	20640.000000	20640.000000	20433.000000	20640.000000	20640.000000	20640.000000	20640.000000
mean	-119.569704	35.631861	28.639486	2635.763081	537.870553	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

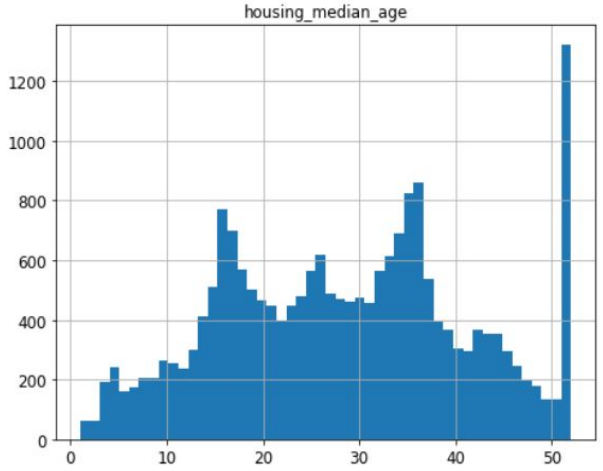
Executing (5s) Cell > save_fig() > savefig() > savefig() > print_figure() > print_png()

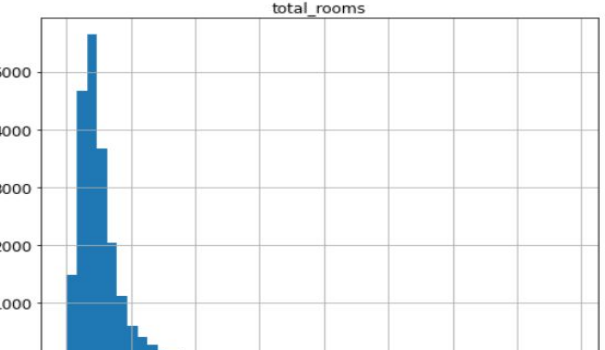
Test

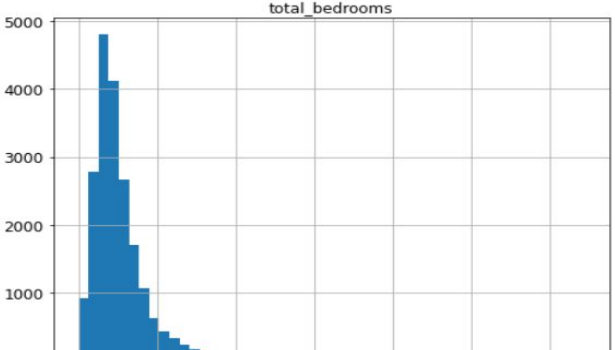
```
Saving figure attribute_histogram_plots
```

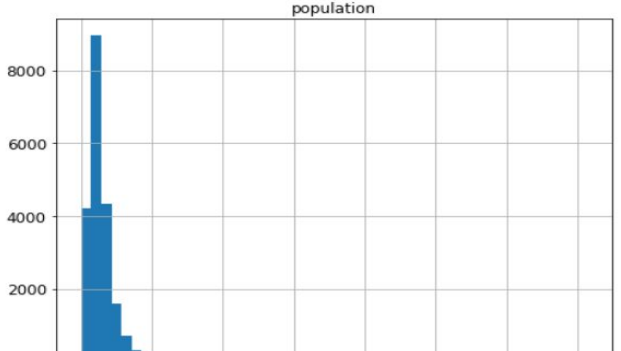




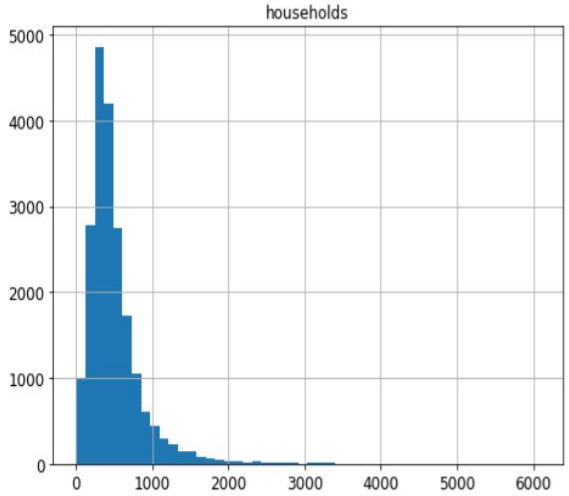


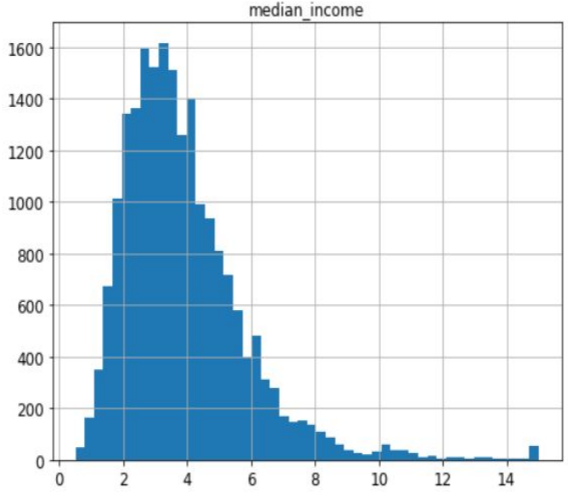


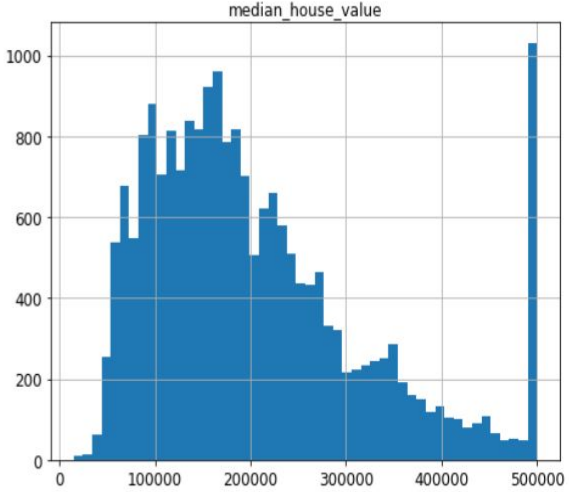




Test


households
5000
4000
3000
2000
1000
0
0
1000
2000
3000
4000
5000
6000

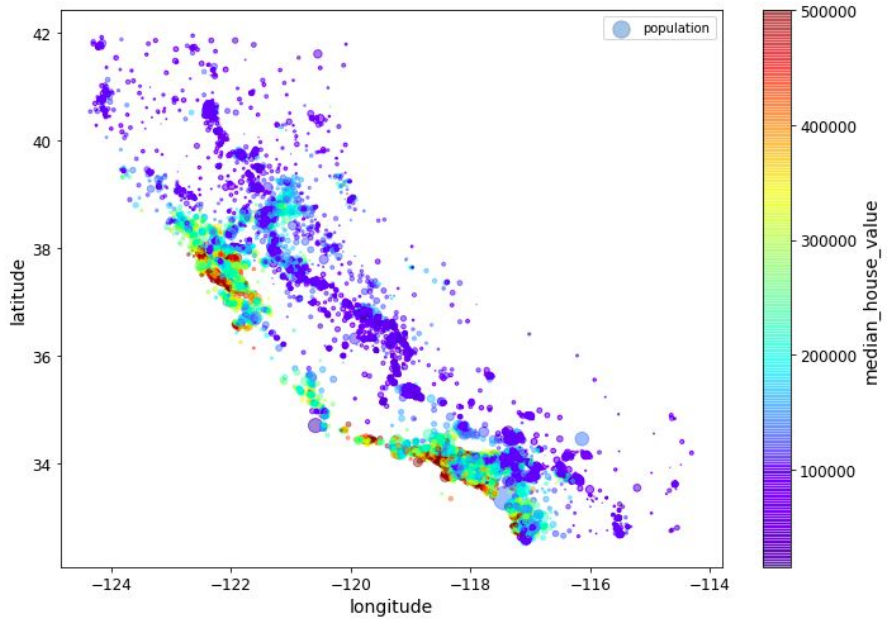

median_income
1600
1400
1200
1000
800
600
400
200
0
0
2
4
6
8
10
12
14


median_house_value
1000
800
600
400
200
0
0
100000
200000
300000
400000
500000

Test

- Visualize Geographical Data

```
housing.plot(kind="scatter", x="longitude", y="latitude", alpha=0.1)
```



Test

- **Looking for Correlation**

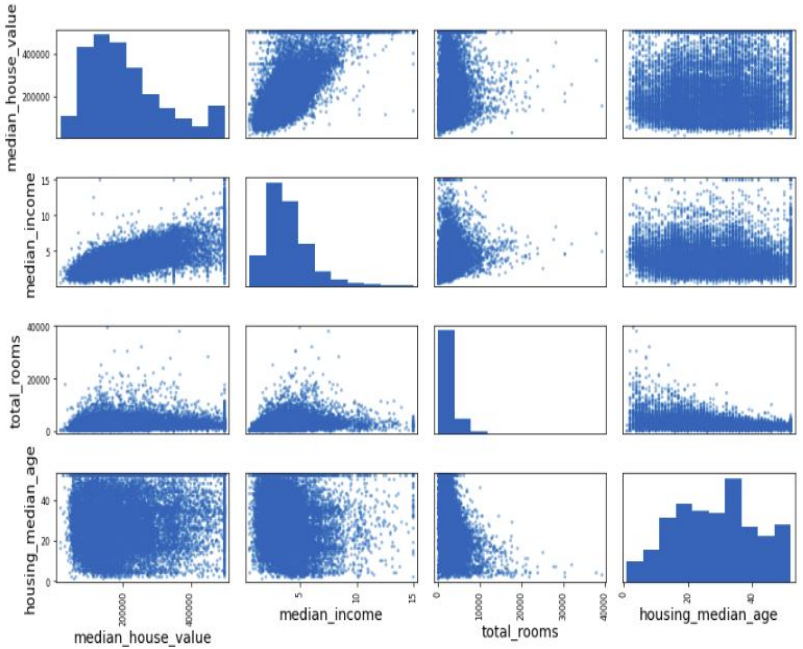
```
corr_matrix = housing.corr()
```

0s

```
corr_matrix["median_house_value"].sort_values(ascending=False)
```

0s

```
median_house_value    1.000000
median_income         0.687151
total_rooms           0.135140
housing_median_age    0.114146
households            0.064590
total_bedrooms        0.047781
population           -0.026882
longitude            -0.047466
latitude             -0.142673
Name: median_house_value, dtype: float64
```

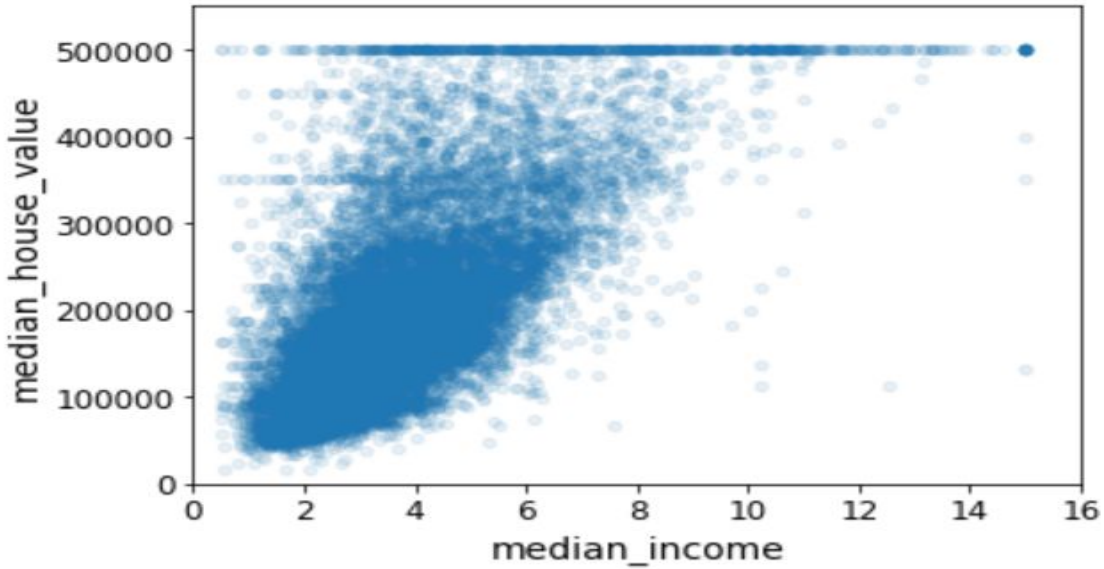


Test

0s

```
housing.plot(kind="scatter", x="median_income", y="median_house_value",
             alpha=0.1)
plt.axis([0, 16, 0, 550000])
save_fig("income_vs_house_value_scatterplot")
```

```
Saving figure income_vs_house_value_scatterplot
```



Enhancement Ideas

- We can use Data Augmentation to generate new data by applying transformations to existing data to help improve the performance of the model.

Conclusion

- In conclusion, our end-to-end machine learning project was successful. We began by exploring and preparing the data, selecting the best machine learning models, evaluating their performance using RMSE, and optimizing the hyperparameters through Grid Search and Random Search.

- Finally, we deployed the model and tested its performance in real-world scenarios. This project highlights the significance of thorough data preparation, model selection, hyperparameter tuning, and performance evaluation in creating a high-performing machine learning system.

GitHubLink

https://github.com/divyapandey03/Machine-Learning/tree/main/End%20to%20End%20Project

References

- *Creating the workspace - jupyter notebooks - grokking data science. (n.d.). Retrieved February 19, 2023, from https://www.educative.io/courses/grokking-data-science/q2KYKwAknDR*
- Grid Search vs Randomized Search