HOUSE PRICE PREDICTION

Date: 08/05/2023
COURSE : IA 651
Shradha Reddy Pulla

INTRODUCTION

- Accurately estimating the value of real estate is an important problem for many stakeholders.
- It is common knowledge that factors such as the size, number of rooms and location affect the price, there are many other things at play.

Overview

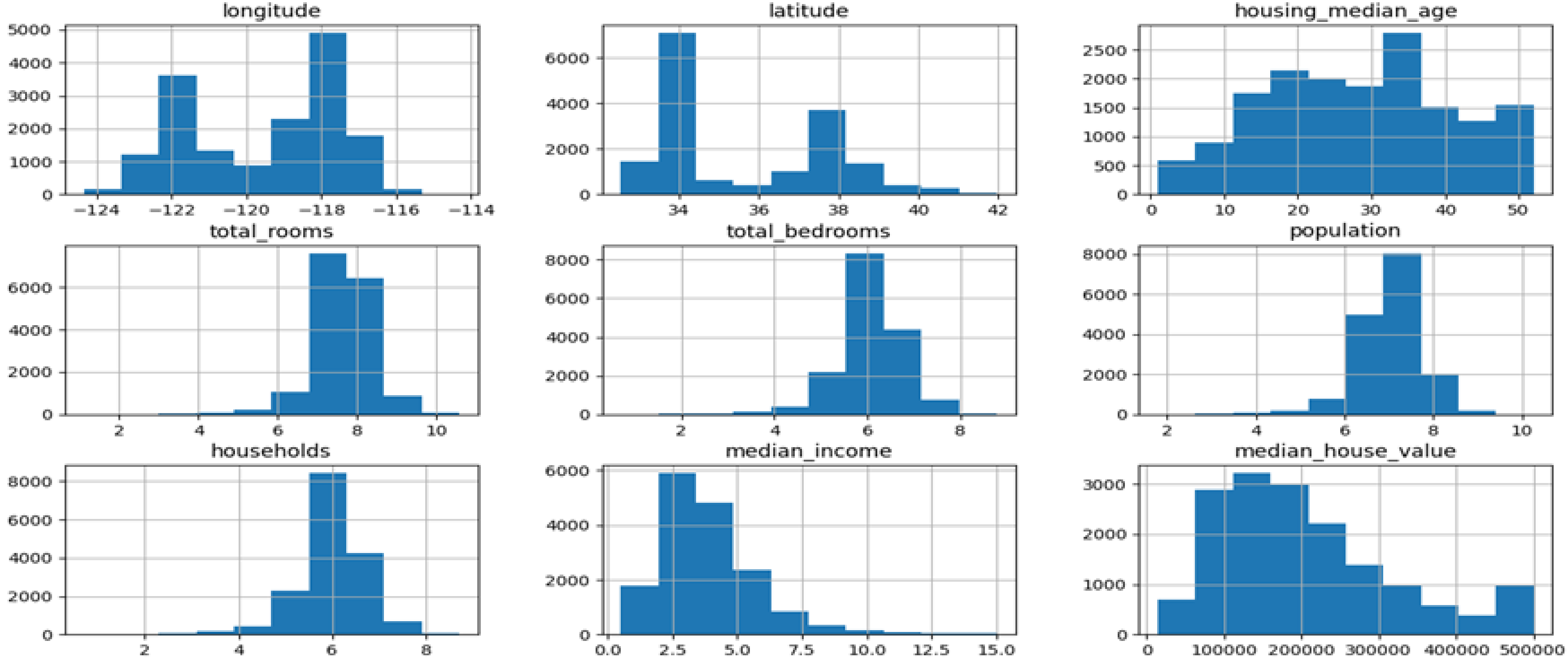
- ❑ Data Exploration
- ❑ Data visualization
- ❑ Data cleaning/pre-processing
- ❑ Feature engineering
- ❑ Model building
- ❑ Model evaluation

GETTING FAMILIAR WITH THE DATA

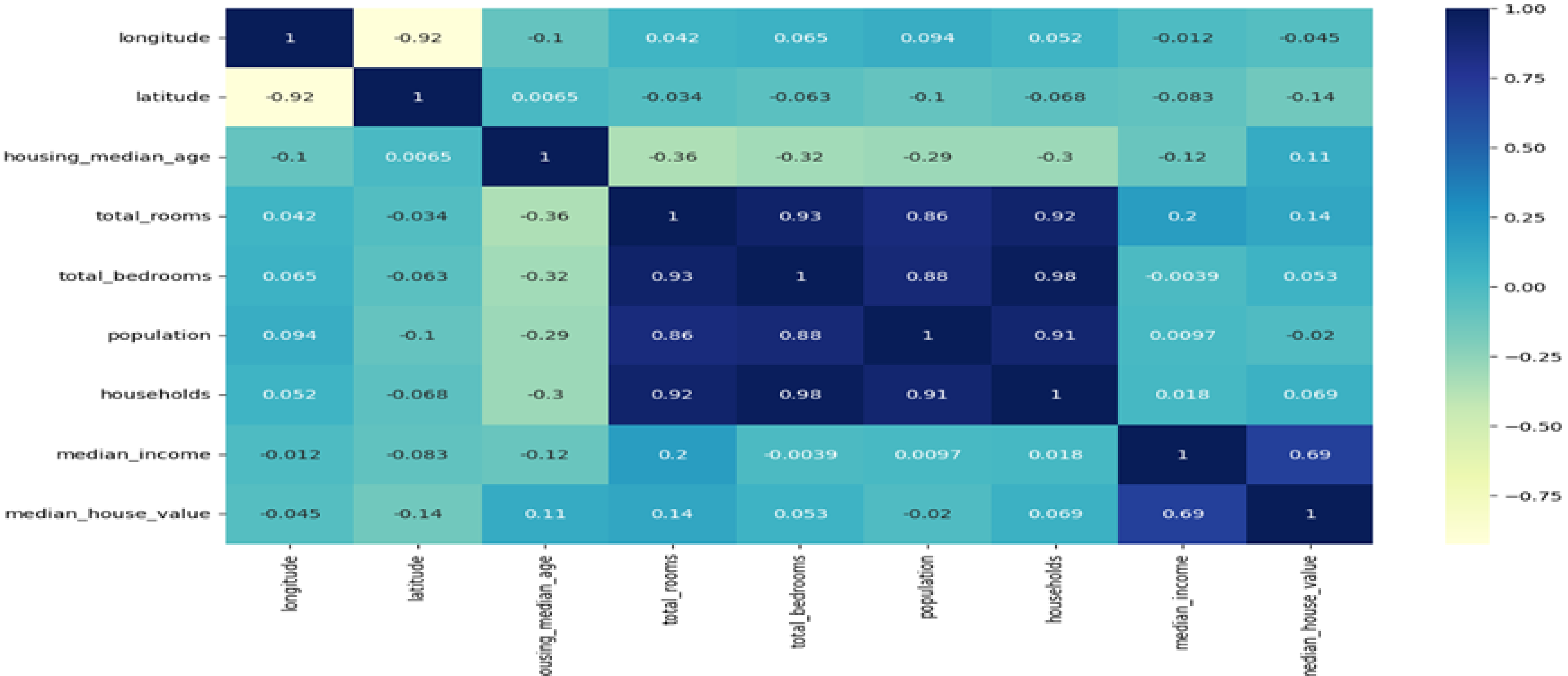
- The data is taken from the kaggle with 20,640 rows and 10 columns.
- Feature standardization was performed on some numeric data variables.
- Attributes like latitude and longitude were used during exploratory analysis not used in further model building.
- The dataset was split into train-validate-test samples using train-test-split from sklearn module.

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income($10,000)	median_house_value	ocean_proximity
0	-122.23	37.86	41	880	129	322	126	8.3252	452600	NEAR BAY
1	-122.22	37.86	21	7099	1106	2401	1138	8.3014	358500	NEAR BAY
2	-122.24	37.85	52	1467	190	496	177	7.2574	352100	NEAR BAY
3	-122.25	37.85	52	1274	235	558	219	5.6431	341300	NEAR BAY
4	-122.25	37.85	52	1627	280	565	259	3.8462	342200	NEAR BAY

Visualization of the dataset

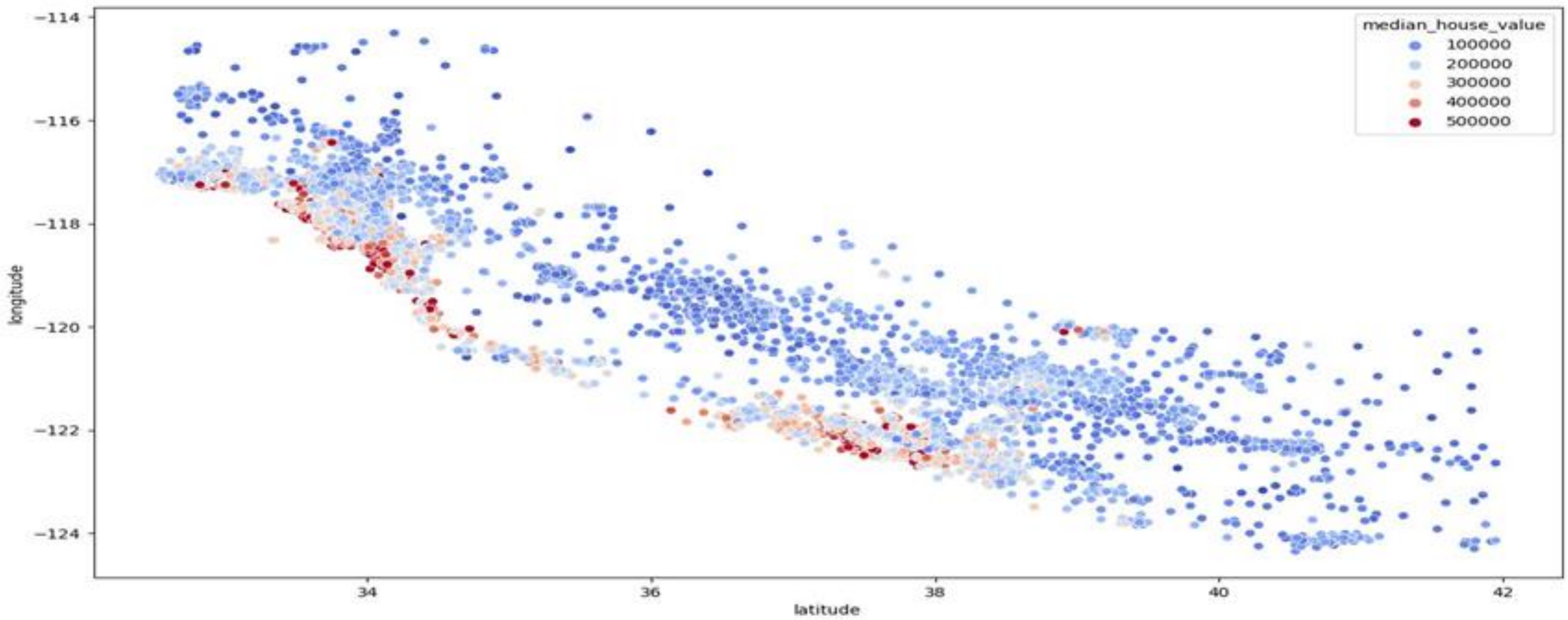


Correlation of predictors and output variable



EARTH MAP OF CALIFORNIA

GEOGRAPHICAL HOUSE PRICE FLUCTUATION.



DATA PRE-PROCESSING

- Removing null values.

#	Column	Non-Null Count	Dtype
0	longitude	20640 non-null	float64
1	latitude	20640 non-null	float64
2	housing_median_age	20640 non-null	float64
3	total_rooms	20640 non-null	float64
4	total_bedrooms	20433 non-null	float64
5	population	20640 non-null	float64
6	households	20640 non-null	float64
7	median_income	20640 non-null	float64
8	median_house_value	20640 non-null	float64
9	ocean_proximity	20640 non-null	object

- ❑ **Standardizing total rooms, total bedrooms, population**
- ❑ **Converting categorical data into numerical data.**

```
<1H OCEAN     7219
INLAND        5195
NEAR OCEAN    2115
NEAR BAY      1815
ISLAND           2
Name: ocean_proximity, dtype: int64
```

Feature engineering

I) feature : Bedroom_ratio = total_bedrooms/Total_rooms

II) feature : Household_rooms = total_rooms/households

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income($10,000)	median_house_value	<1H OCEAN	INLAND	NEAR BAY	NEAR OCEAN	bedroom_ratio	household_room
0	-122.23	37.86	41	880	129	322	126	8.3252	452600	0	0	0	1	0.765363	1.3159
1	-122.22	37.86	21	7099	1106	2401	1138	8.3014	358500	1	0	0	0	0.763839	1.3108
2	-122.24	37.85	52	1467	190	496	177	7.2574	352100	0	1	0	0	0.771696	1.2882
3	-122.25	37.85	52	1274	235	558	219	5.6431	341300	1	0	0	0	0.802899	1.2826
4	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Regression model building

(Linear Regression)

```
from sklearn.linear_model import LinearRegression
from sklearn import metrics
OLS = LinearRegression()
OLS.fit(X_train, y_train)
```

0.0s

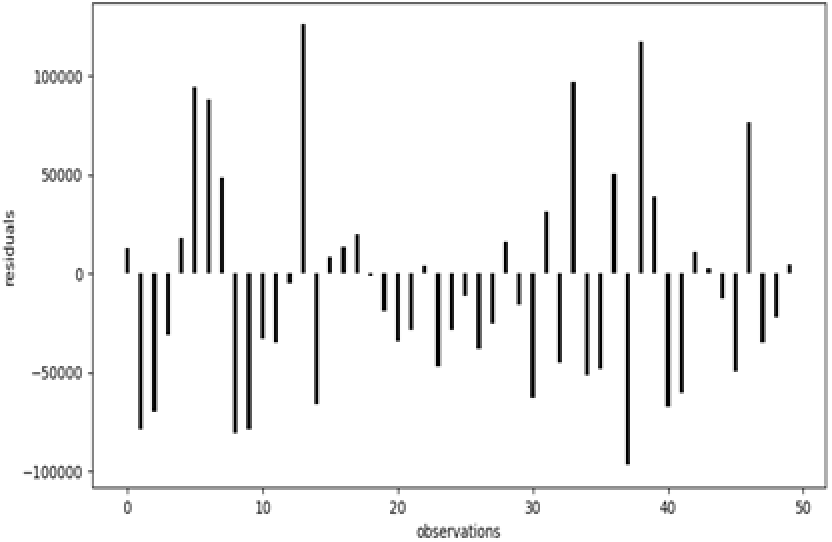
```
▾ LinearRegression
LinearRegression()
```

```
y_pred=OLS.predict(X_test)
print(" The intercept is " +  str(OLS.intercept_))
print(" The coeffiients are " +  str(OLS.coef_))
print(" The R_sqaured value is " +  str(OLS.score(X_test, y_test)))
print("MAE is:", metrics.mean_absolute_error(y_test,y_pred))
print("MSE is:", metrics.mean_squared_error(y_test,y_pred))
```

0.0s

```
 The intercept is -2146719.495872446
 The coeffiients are [-2.72191872e+04 -2.61224400e+04  1.03631750e+03 -6.39771322e+00
  9.97707401e+01 -3.73545857e+01  4.99035484e+01  3.93656164e+04
 -1.49036788e+05 -1.87787533e+05 -1.51883617e+05 -1.45673607e+05]
 The R_sqaured value is 0.6576677709626819
MAE is: 49715.183929114246
MSE is: 4740761051.434472
```

Model accuracy evaluation



	PREDICTIONS	ACTUAL VALUES	error
15175	315646.7385	328200	12553.26153
15424	235926.1562	156900	-79026.1562
16212	157003.3256	87200	-69803.32561
15356	172404.3402	141000	-31404.34025
1899	82931.27063	100800	17868.72937

Drawbacks of OLS

- ❑ Non-linearity
- ❑ Feature Interactions
- ❑ Robustness to outliers and noise
- ❑ Ensemble nature

Regression model building

(Random Forest Regression)

```python
from sklearn.ensemble import RandomForestRegressor
reg = RandomForestRegressor()
reg.fit(X_train, y_train)
```

15.9s

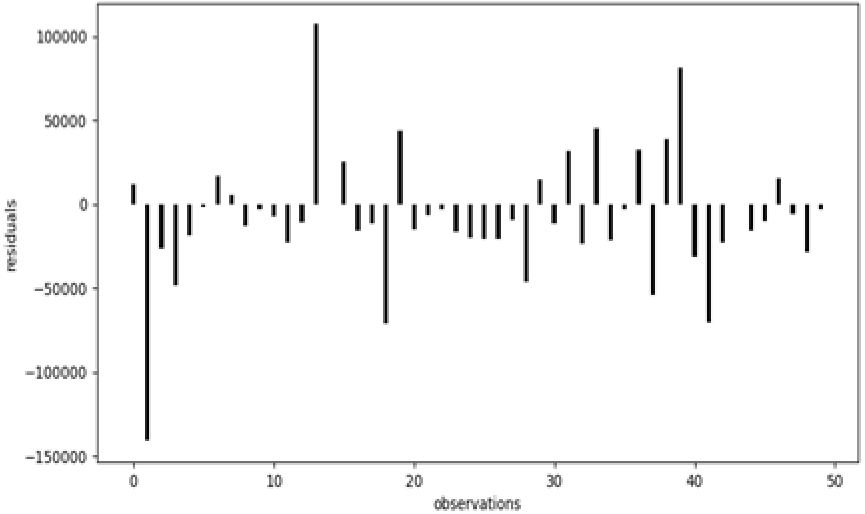
```
▾ RandomForestRegressor
RandomForestRegressor()
```

```python
y_pred_r = reg.predict(X_test)
acc_rf = metrics.r2_score(y_test,y_pred_r)
print("R^2 values is :", acc_rf)
print("MAE is:", metrics.mean_absolute_error(y_test, y_pred_r))
print("MSE is:", metrics.mean_squared_error(y_test, y_pred_r))
print("RMSE is:", np.sqrt(metrics.mean_squared_error(y_test, y_pred_r)))
```

0.3s

```
0.8266384101819658
R^2 values is : 0.8266384101819658
MAE is: 31180.032596036213
MSE is: 2400784393.3809185
RMSE is: 48997.799883065345
```

Model accuracy evaluation



	PREDICTIONS	ACTUAL VALUES	error
15175	306784.03	328200	21415.97
15424	309734.27	156900	-152834.27
16212	111365	87200	-24165
15356	182558	141000	-41558
1899	123728	100800	-22928

THANK YOU!