Analysing Data set of houses in California

Group 6

Batch 3

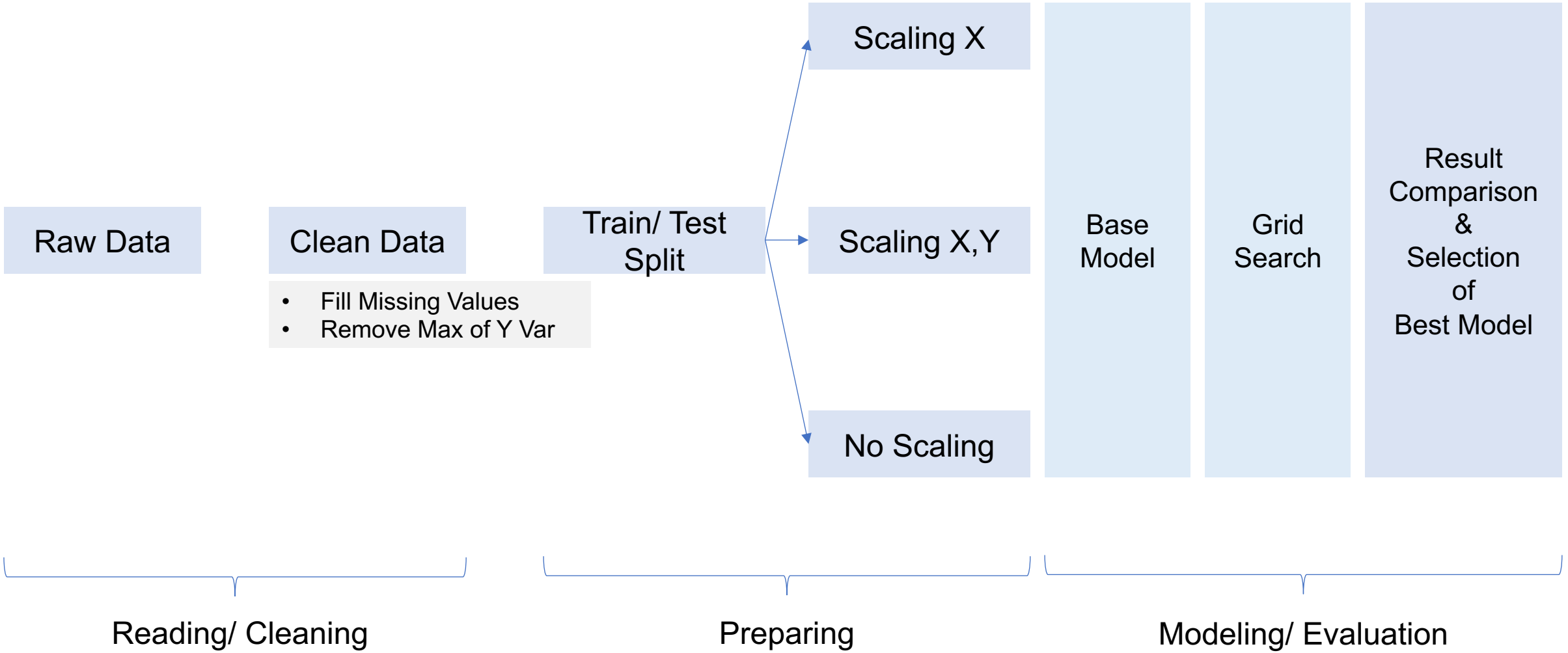
Problem Statement

We have been provided with data of districts in state of California with median house prices; our objective is to predict median house prices basis the data available.

Independent Variables
longitude
latitude
housing_median_age
total_rooms
total_bedrooms
population
households
median_income
ocean_proximity

Dependent Variables
median_house_pricing

Data Pipeline



Reading the data

Column Name	Possible Description	Data Type
longitude	longitude of location	numeric
latitude	latitude of location	numeric
housing_median_age	median age of a houses in a locality	numeric
total_rooms	total rooms in a locality	numeric
total_bedrooms	total bedrooms in a locality	numeric
population	number of people residing in locality	numeric
households	number of group of people residing within a home unit	numeric
median_income	median income for households in (10,000 USD)	numeric
medain_house_value	median house value in a locality (USD)	numeric
ocean_proximity	proximity to ocean location	alphanumeric

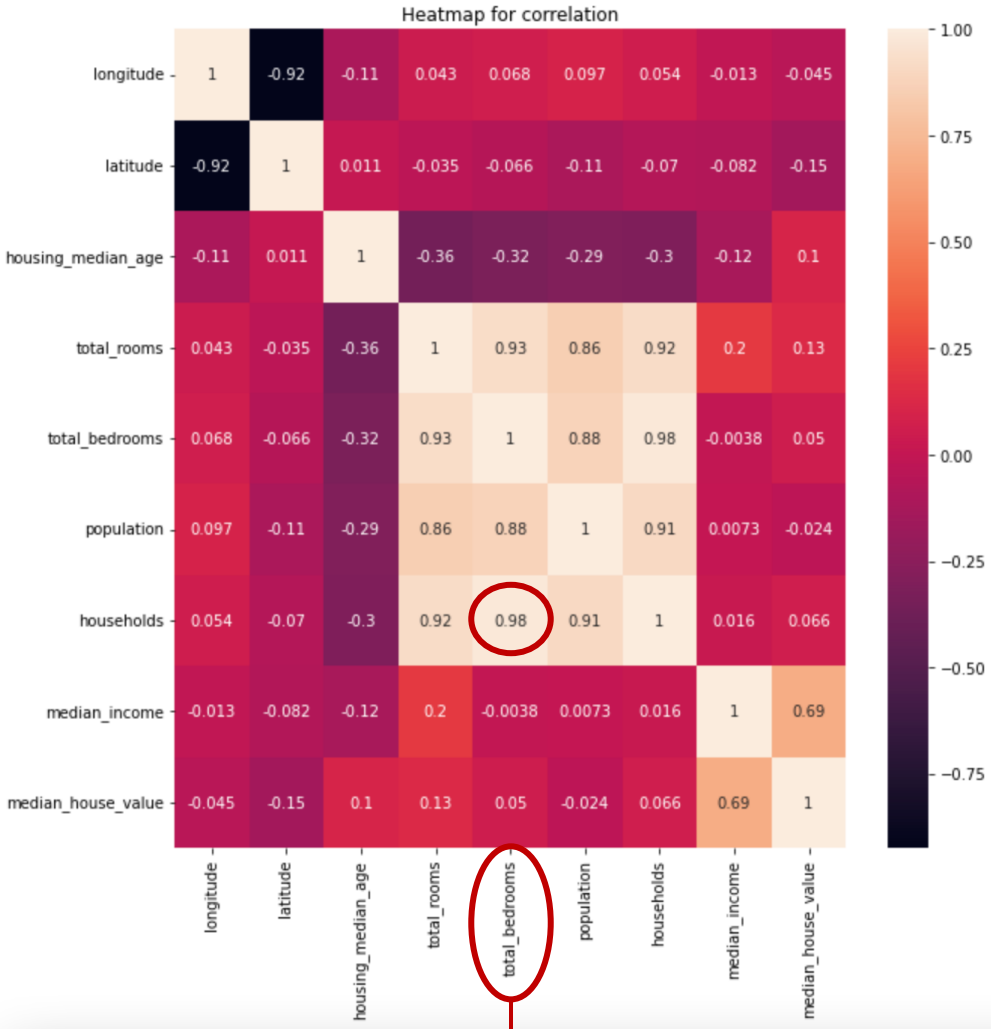
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18565 entries, 0 to 18564
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   longitude           18565 non-null  float64
 1   latitude            18565 non-null  float64
 2   housing_median_age  18565 non-null  int64
 3   total_rooms         18565 non-null  int64
 4   total_bedrooms      18376 non-null  float64
 5   population          18565 non-null  int64
 6   households          18565 non-null  int64
 7   median_income       18565 non-null  float64
 8   median_house_value  18565 non-null  int64
 9   ocean_proximity     18565 non-null  object
dtypes: float64(4), int64(5), object(1)
memory usage: 1.4+ MB
```

Total bedrooms is one column with null values

Method to impute missing value:

- Total bedrooms should be proportional to total_rooms
- Total bedrooms should be proportional to number of house holds

Correlation graph



Highly Correlated Variables are:

1. Households, total_bedrooms, population, total_rooms
2. Latitude and Longitudes
3. Housing Median Age with – Total Rooms, total_bedrooms, Population & Households

1. Population determines #Households which determines #rooms and #Bedrooms
2. Latitude Longitude will be correlated as from same area (California)
3. Median Age => Older areas => Larger houses => Lower population and Lower Number of rooms/ bedrooms etc.

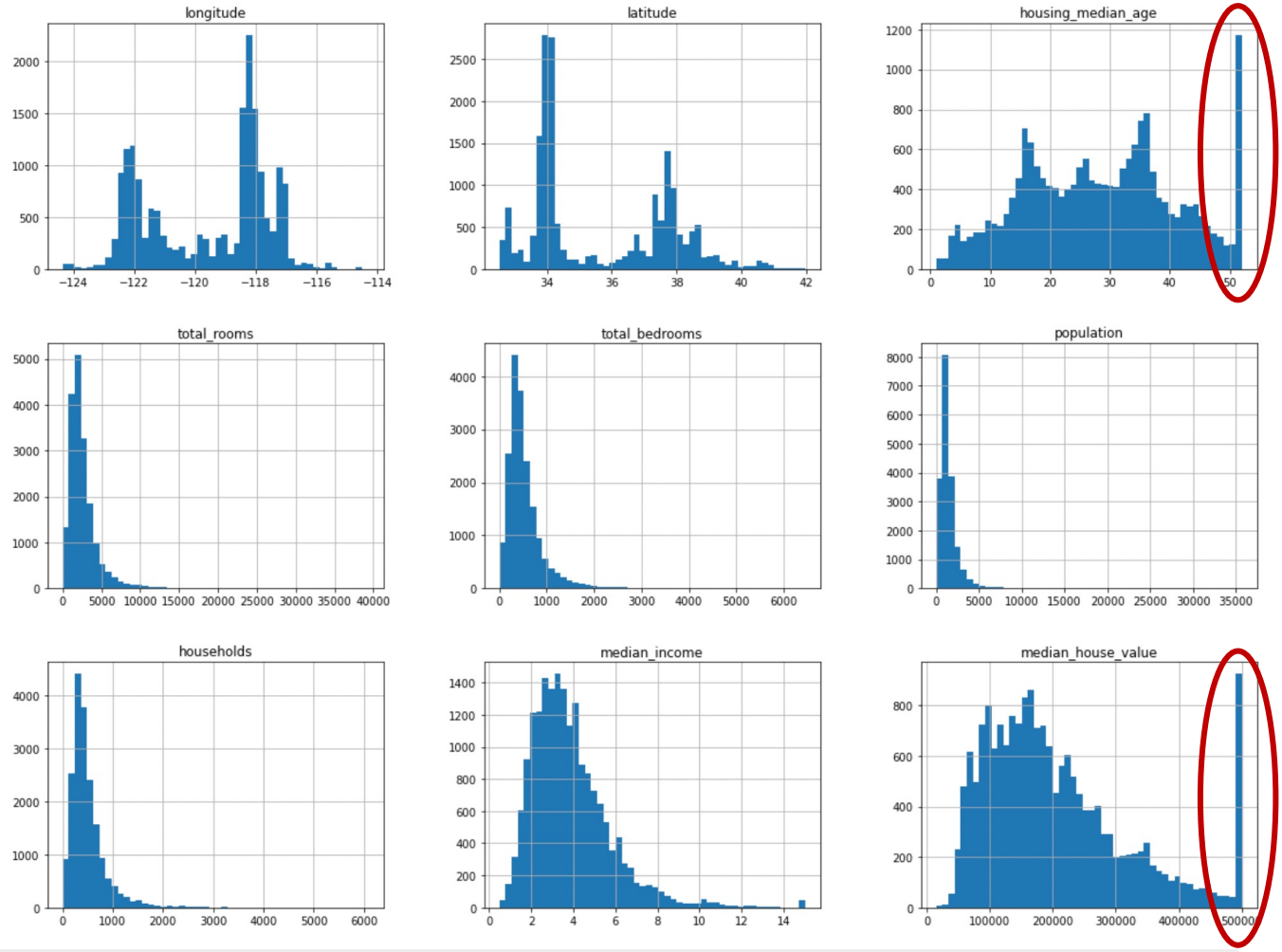
Other reason can be older areas might have lesser amenities leading to lower population

We will use total_households to impute #of bedrooms: (Corr = 0.98)

*Total_bedrooms (missing) = Households (Missing) * (mean)[Total_bedrooms/ Households]*

EDA of the data columns (1/2)

	count	mean	std	min	25%	50%	75%	max
longitude	18565.0	-119.57	2.00	-124.35	-121.80	-118.50	-118.01	-114.31
latitude	18565.0	35.63	2.14	32.54	33.93	34.26	37.71	41.95
housing_median_age	18565.0	28.62	12.56	1.00	18.00	29.00	37.00	52.00
total_rooms	18565.0	2634.03	2195.92	2.00	1442.00	2123.00	3141.00	39320.00
total_bedrooms	18565.0	537.79	424.07	2.00	295.00	434.00	646.00	6445.00
population	18565.0	1426.22	1142.57	3.00	786.00	1166.00	1725.00	35682.00
households	18565.0	499.45	384.55	2.00	279.00	408.00	603.00	6082.00
median_income	18565.0	3.87	1.90	0.50	2.56	3.53	4.74	15.00
median_house_value	18565.0	206617.79	115447.10	14999.00	119300.00	179400.00	264400.00	500001.00



Median_House_Value being continues and having a cap of 500,001 leads to a dirty model and we would ideally like to remove this **4% Loss**

There seems to be a capping on maximum values for Medain_House_Value and Housing_Median_Age;

Housing_Median_Age seems to be a discreet value therefore we can live with the maximum value of 52 (1,129 records)

EDA of data columns (2/2)

Treatment of Categorical Variables

```
<1H OCEAN      7709
INLAND         5895
NEAR OCEAN     2208
NEAR BAY       1880
ISLAND            5
Name: ocean_proximity, dtype: int64
```

```
9   ocean_proximity_INLAND       17697 non-null  uint8
10  ocean_proximity_ISLAND       17697 non-null  uint8
11  ocean_proximity_NEAR BAY     17697 non-null  uint8
12  ocean_proximity_NEAR OCEAN   17697 non-null  uint8
```

Standardization of the Variables

As the scale of variables is different, we need to standardize the variables to have the same mean. We would need standardization for following models:

a. Linear Regression – OLS; Lasso and Ridge
b. Neural Network
c. KNN Regression
d. Support Vector Regression (Requires scaling of the Y Variables as well)

We would not do the standardizations for the following set of models:

a. Decision Trees
b. Random Forest
c. Gradient Boosting

Regression Model (1/2)

Checklist for a Regression Model

- ✓ No missing Values
- ✓ Variables to be standardized
- ✓ Split in Test and Train Set
- ✗ Dependent and Independent variables are normally distributed → We can ignore this for real-life models
- ✗ Non-existence of Multicollinearity in independent variables → **To be treated basis the VIF Factors**

Steps for Linear Regression

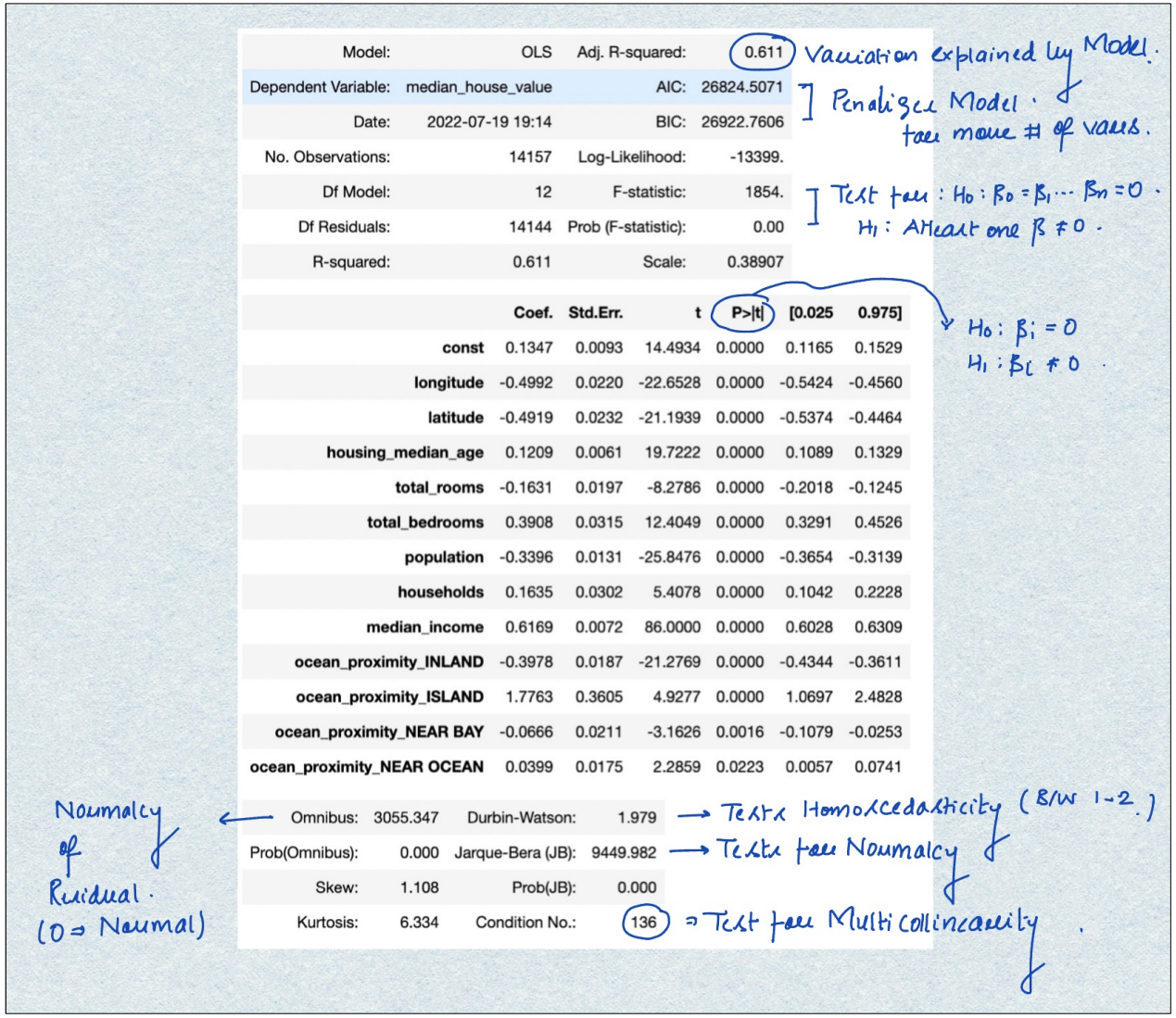
Running the Model

- Split the data into Train Test
- Standardize Test/ Train data separately
- Add constant using *sm.add_constant*
- Run OLS (Ordinary Least Square) model using *sm.OLS*
- Identify any variable having p-value > 0.05 and remove them from the analysis
- Re-run the model

Testing the Model

- Remove Multi-collinearity via VIF Factors
- *Test for normality of residuals (P-P Plot)*
- *Residual plot for homoscedasticity*
- *Measuring R-Squared*
- *Check for the difference between RMSE of Test and Train Data Sets*
- *Check for R2 of Test Dataset*

Regression model (2/2)



Analysis of the Model			
Parameter	**Meaning**	**Model Value**	**Ideal Value**
Adj. R-Square	What % of dependent variable is explained by independent variables	0.611	1.00
AIC/ BIC	Compares efficacy of the model	~26K	As low as possible
Prob (F-statistics)	To check if all coefficients are 0 or not	0	0
P > |t|	To check if individual coefficients are Zero	< 0.05 for all variables	< 0.05 for all variables
Durbin-Watson	To check for Homoscedasticity	1.979	Between 1 and 2
Omnibus	Describes distribution of Residuals (if it is normal or not)	0	~3k
P(Omnibus)	Null hypothesis that errors are normally distributed	0	Close to 1
J-B Test	Testing for errors being normally distributed	9449	Very Low
Condition Number	Tests for multicollinearity	136	Should be < 1000

- Normality assumption can be departed away from in normal cases
- We will test the multi-collinearity via VIF test

Regularized Regression

Lasso Regression

columns	Coef
longitude	-48,488.57
latitude	-47,777.72
housing_median_age	11,740.07
total_rooms	-15,844.79
total_bedrooms	37,960.11
population	-32,988.98
households	15,881.50
median_income	59,915.00
ocean_proximity_INLAND	-38,633.66
ocean_proximity_ISLAND	0.00
ocean_proximity_NEAR BAY	-6,470.37
ocean_proximity_NEAR OCEAN	0.00

Parameter	Value
R-2 on Train Data set	0.5016
R-2 on Test Data set	0.5836

Ridge Regression

columns	Coef
longitude	-48,397.6
latitude	-47,700.7
housing_median_age	11,756.0
total_rooms	-15,738.9
total_bedrooms	37,845.6
population	-32,984.7
households	15,886.8
median_income	59,879.8
ocean_proximity_INLAND	-38,695.3
ocean_proximity_ISLAND	103,497.0
ocean_proximity_NEAR BAY	-6,457.2
ocean_proximity_NEAR OCEAN	3,859.6

Parameter	Value
R-2 on Train Data set	0.6112
R-2 on Test Data set	0.6198

OUTCOME:

1. Ridge Regression performs at par with OLS
2. Performance of Lasso is worse amongst all the models
3. Lasso Removes non-correlated columns (ocean proximity...)

Amongst linear regression OLS/ Ridge is the best model

Reason multi-collinearity is acceptable in the model mainly on account of these variables may have a correlated relationship. However, independently contributing to the model.

Example: BMI = Kg/m2 where both weight and height have a relationship, but predicting BMI with one alone is not accurate

Comparison of Gridsearch Results of Regression

Model_Name	MAE	MSE	RMSE	R2_Train	R2_Test
OLS	45,254	3,820,382,977	61,809	61.13%	61.87%
RIDGE	45,255	3,820,267,465	61,808	61.13%	61.87%
LASSO	45,254	3,820,382,590	61,809	61.13%	61.87%

Params	OLS	Ridge	Lasso
longitude	-48,488.73	-48,484.90	-48,488.57
latitude	-47,777.86	-47,776.28	-47,777.72
housing_median_age	11,740.07	11,742.61	11,740.07
total_rooms	-15,845.03	-15,832.29	-15,844.79
total_bedrooms	37,960.31	37,952.49	37,960.11
population	-32,989.04	-32,989.63	-32,988.98
households	15,881.60	15,877.13	15,881.50
median_income	59,915.06	59,910.48	59,915.00
ocean_proximity_INLAND	-38,633.57	-38,638.71	-38,633.66
ocean_proximity_ISLAND	172,526.72	161,300.08	172,479.55
ocean_proximity_NEAR BAY	-6,470.55	-6,471.84	-6,470.37
ocean_proximity_NEAR OCEAN	3,875.09	3,870.29	3,875.04

OUTCOME:

1. Gridsearch gives parameters for Lasso/ Ridge in such a way that model is very similar to OLS
2. There is no significant difference between 3 models, and anyone can be used

KNN Regression

Running K-NN Model

Parameters for K-NN:
n_neighbors – defining the number of nearest points for regression
metric - what kind of distance metric to choose the nearest points

Grid-search

We will use grid-search to find out the best parameters for the model.

Parameters for the best model are:
1. n_neighbors = 9
2. Metric = euclidean

Parameter	Value
R-2 on Train Data set	0.7513
R-2 on Test Data set	0.6858

Test and Train results both are better

Distance Metrics

Canberra Distance

$$d(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^{n} \frac{|p_i - q_i|}{|p_i| + |q_i|}$$

Euclidian Distance

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2}.$$

Minkowski Distance

$$D(X, Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}.$$

Note: While the train results pre-gridsearch may be better post-gridsearch. However, due to cross validation the test results will be better

Tree-based models

Decision Tree Regression

Parameter	Value
R-2 on Train Data set	0.7631
R-2 on Test Data set	0.6544

Random Forest Regression

Parameter	Value
R-2 on Train Data set	0.9581
R-2 on Test Data set	0.7736

Gradient Boosting

Parameter	Value
R-2 on Train Data set	0.9997
R-2 on Test Data set	0.7985

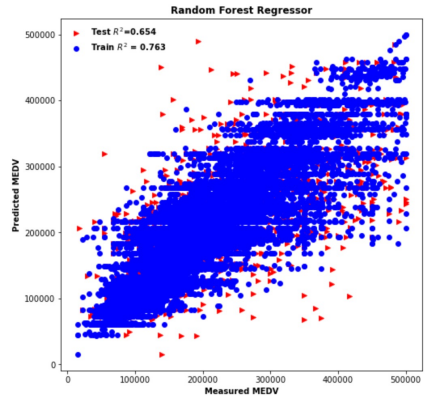
Best Parameters for Grid Search

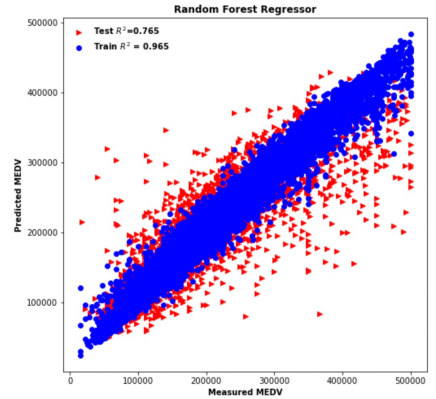
max_depth: 9

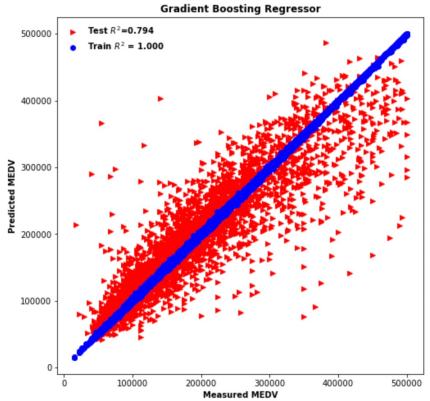
bootstrap : False
max_depth: 30
max_features: ‘sqrt’
n_estimators: 30

n_estimators : 500
max_depth: 10
max_features: ‘log2’
criterion: ‘squared_error’

Best model till now

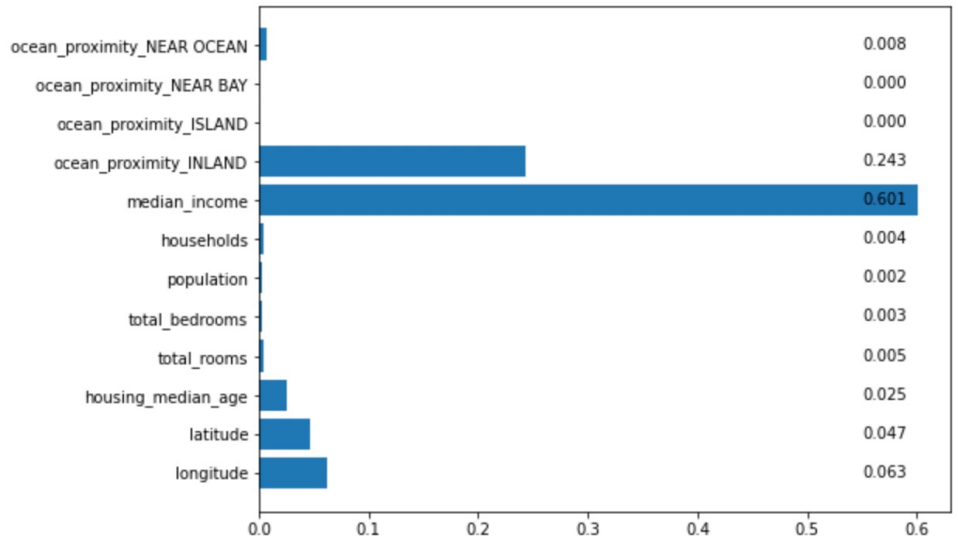




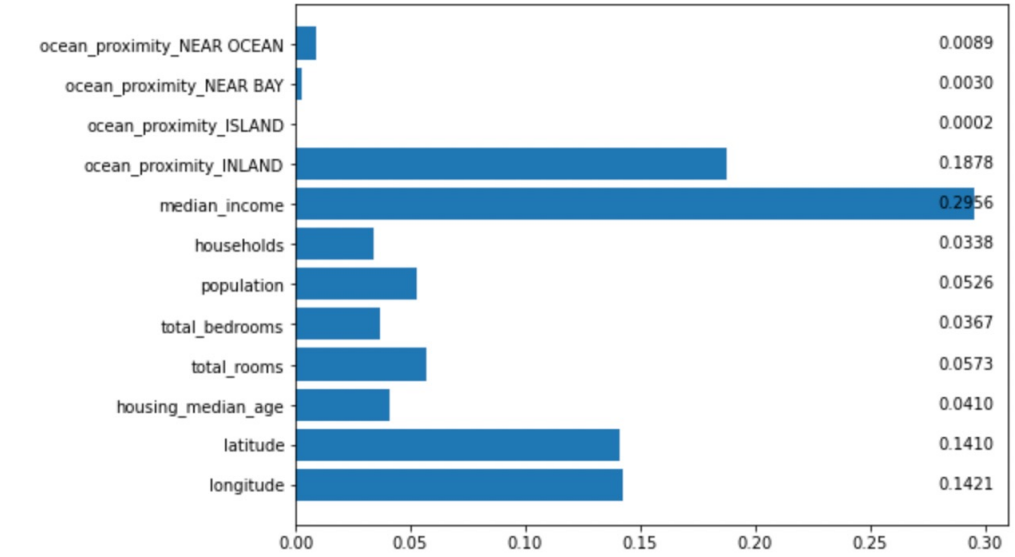


Feature importance – Tree based

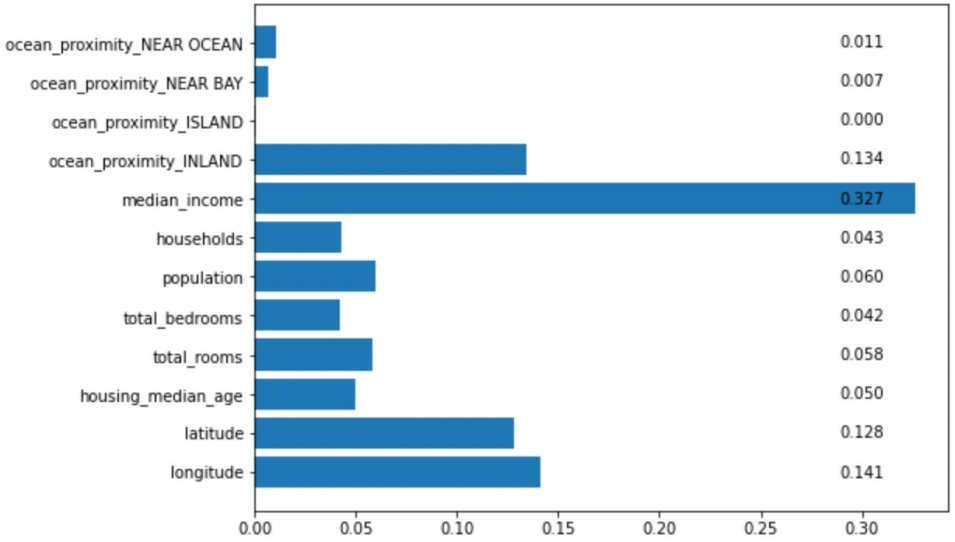
Decision Tree



Gradient Boosting



Random Forest

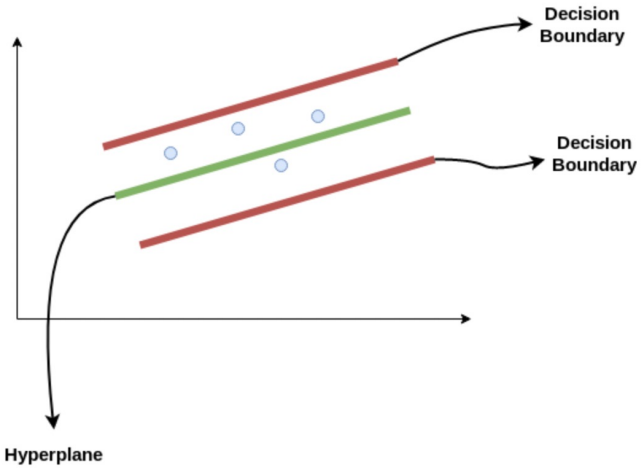


Support Vector Regression (SVR)

Support Vector Machines (SVM) are generally used for classification problems, where a hyperplane in n-dimension is used to classify the data points into classes.

SVM can also be used for regression, where a hyperplane in n-dimension (along with the decision boundaries) contain the maximum number of data points.

We will try to identify the equation of the plane which along with the decision boundary contains the maximum number of points.



Hyperparameter	Values	Purpose
C	numeric	Regularization Parameter. Penalty is L2 Determines the classification boundary
gamma	numeric	The gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'.
kernel	'rbf', 'poly', 'sigmoid', 'linear'	functions to be used for the classification, used to create the different type of decision boundaries

Parameter	Value
R-2 on Train Data set	0.7981
R-2 on Test Data set	0.7479

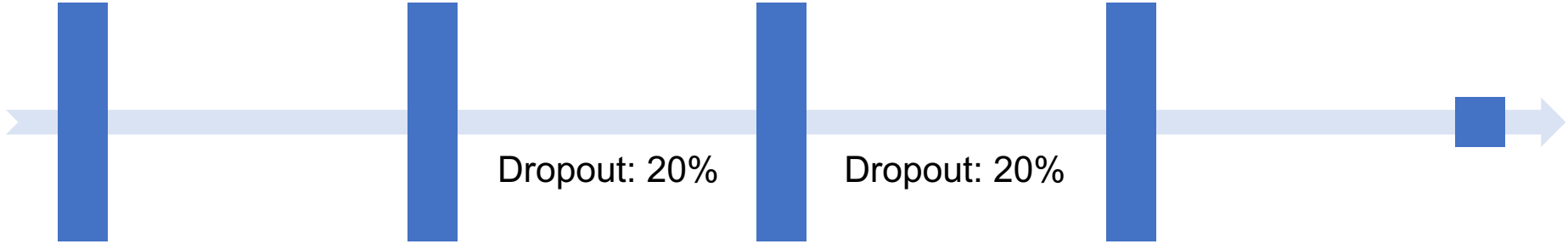
For SVR we would need to scale the Y variables as well as the hyperplane and decision boundaries are determining the distances

Kernel Definition

Kernel Function	Formulas
Linear	$K(x_i, x) = x_i^T x$
Polynomial	$K(x_i,x) = (x_i^T x + 1)^d \quad d=1,2,\ldots\ldots$
Gaussian - Radial Basis Function	$K(x_i,x) = \exp(-\gamma\|x - x_i\|^2)$ $K(x_i,x) = \exp(-\frac{1}{2\sigma^2}\|x - x_i^T\|^2)$

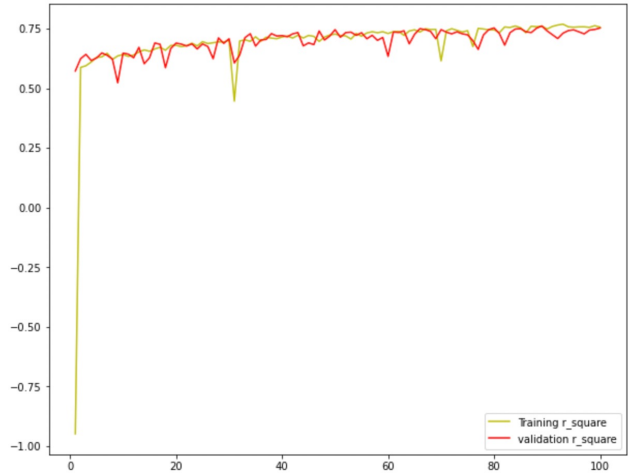
NOTE:
GridSearchCV takes extremely long time to run for SVR therefore use **HalvingGridSearchCV.** It starts with limited resources (number of records) and then iteratively selects best candidates increasing resources

Artificial Neural Networks

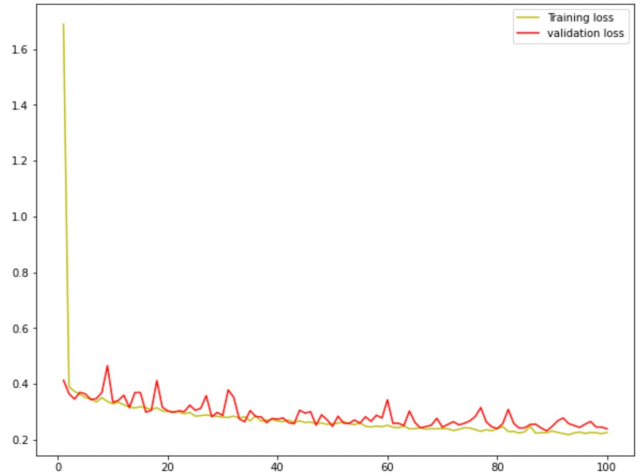


Input Layer	Deep Layer 1	Deep Layer 2	Deep Layer 3	Output Layer
12 Inputs	400 Neuron	320 Neuron	640 Neuron	1 Neuron
	sigmoid	tanh	relu	linear

Grid Search on ANN taking too long to run…



R2 Values for Epochs



Loss Function (MSE)

Parameter	Value
R-2 on Train Data set	0.7885
R-2 on Test Data set	0.7482

Results

Model	R-2 for Train	R-2 for Test	Root Mean Square Error
OLS	0.6113	0.6187	61,809
OLS (Remove Multicollinearity)	0.4930	Not Computed	Not Computed
Lasso	0.6113	0.6187	61,808
Ridge	0.6113	0.6187	61,809
KNN Regression	0.7513	0.6858	56,105
Decision Tree	0.7631	0.6544	58,838
Random Forest	0.9649	0.7649	48,536
Gradient Boosting	**0.9997**	**0.7945**	**45,376**
Support Vector Regression	0.7981	0.7479	50,256
ANN	0.7885	0.7482	49,356

Couldn't run GridSearch

Best Model; Possible overfitting

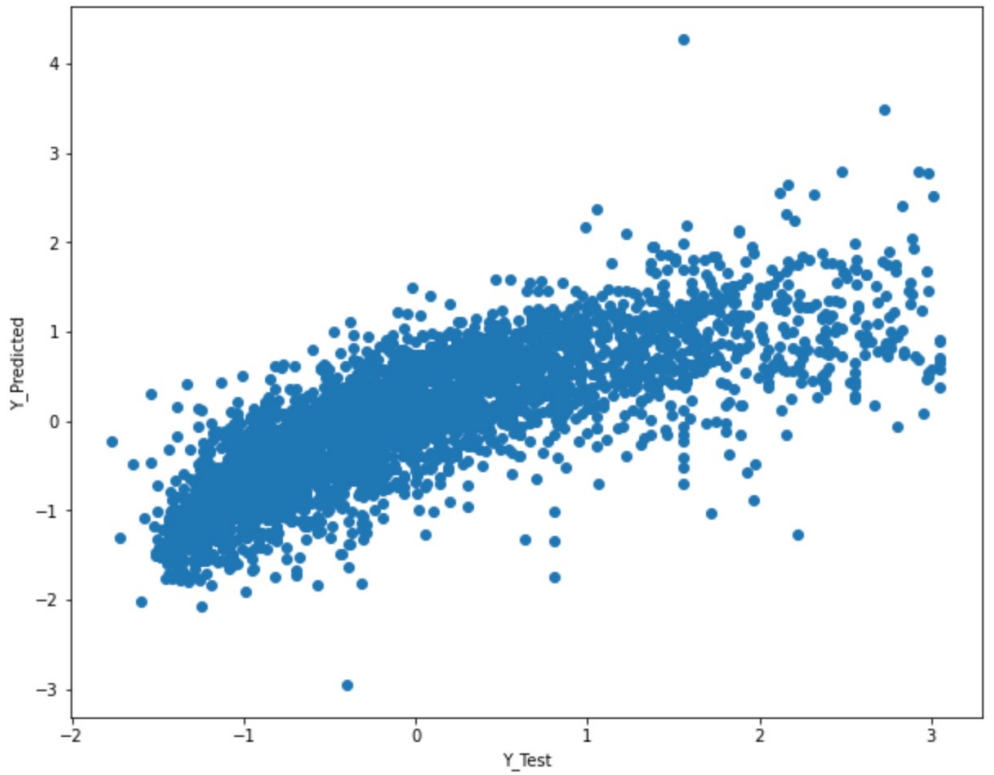
Thank You

Output of the first model

Parameter	Value
R-2 on Test Data set	0.6199
MSE (Train)	60,556
MSE (Test)	61,706

const	0.134725
longitude	-0.499216
latitude	-0.491897
housing_median_age	0.120870
total_rooms	-0.163133
total_bedrooms	0.390821
population	-0.339639
households	0.163509
median_income	0.616856
ocean_proximity_INLAND	-0.397752
ocean_proximity_ISLAND	1.776250
ocean_proximity_NEAR BAY	-0.066618
ocean_proximity_NEAR OCEAN	0.039896

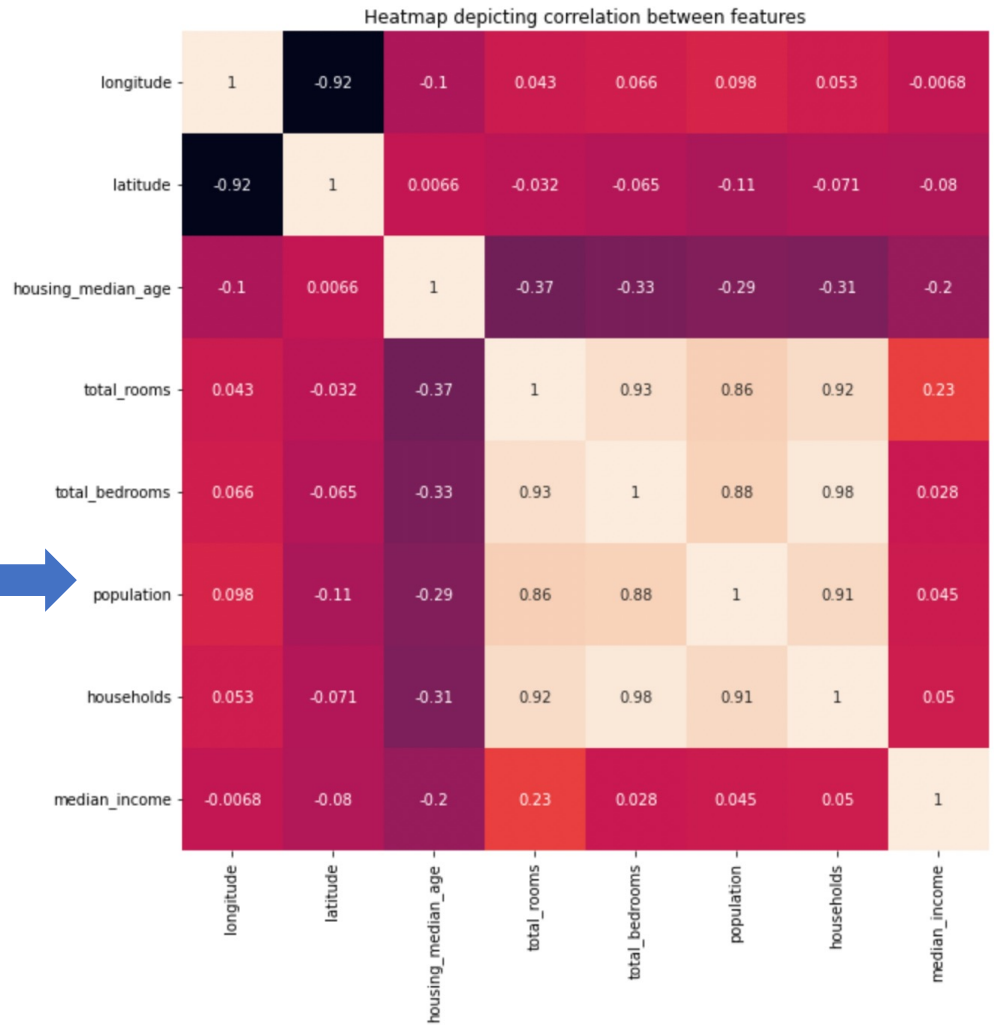
R2 Score 0.6199277978195414



These variables are positively correlated with each other => should have same sign in the model.
This might be pointing towards multi-collinearity which we might need to solve for

An attempt to remove Multi-collinearity (1/2)

column	VIF
ocean_proximity_ISLAND	1.002109
ocean_proximity_NEAR OCEAN	1.311530
ocean_proximity_NEAR BAY	1.640952
ocean_proximity_INLAND	2.816709
housing_median_age	8.364593
median_income	12.036250
population	16.394810
total_rooms	33.608614
households	90.704407
total_bedrooms	95.845383
latitude	796.646021
longitude	834.910141



CORRELATED VARIABLES:

- Total rooms – Total Bedrooms, Population & Household
- Latitude – Longitude

We use household (because of high correlation with others)

We use latitude (can choose either)

Parameters to be used:

ocean_proximity_ISLAND, ocean_proximity_NEAR OCEAN, ocean_proximity_INLAND, housing_median_age, median_income, households, latitude, ocean_proximity_NEAR BAY

Attempt to remove multi-collinearity (2/2)

Model:	OLS	Adj. R-squared:	0.493
Dependent Variable:	median_house_value	AIC:	30561.7635
Date:	2022-07-21 06:56	BIC:	30614.6693
No. Observations:	14157	Log-Likelihood:	-15274.
Df Model:	6	F-statistic:	2297.
Df Residuals:	14150	Prob (F-statistic):	0.00
R-squared:	0.493	Scale:	0.50682

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
const	-0.1017	0.0069	-14.7822	0.0000	-0.1152	-0.0882
ocean_proximity_ISLAND	2.1404	0.4112	5.2051	0.0000	1.3344	2.9464
ocean_proximity_NEAR OCEAN	0.3966	0.0185	21.3804	0.0000	0.3602	0.4330
ocean_proximity_NEAR BAY	0.4887	0.0217	22.5692	0.0000	0.4463	0.5312
housing_median_age	0.1570	0.0064	24.6345	0.0000	0.1445	0.1695
median_income	0.6555	0.0062	105.9503	0.0000	0.6434	0.6676
latitude	-0.1296	0.0065	-19.9200	0.0000	-0.1423	-0.1168

Omnibus:	2385.334	Durbin-Watson:	1.991
Prob(Omnibus):	0.000	Jarque-Bera (JB):	5022.507
Skew:	1.003	Prob(JB):	0.000
Kurtosis:	5.119	Condition No.:	76

While the multi-collinearity related parameters have improved, the model performance has decreased (same happens even if you select some of the parameters).

Which, means the multi-collinearity is not reducing the model performance.

We will be using all the variables.

Outcome of Regression Model

const	0.134725
longitude	-0.499216
latitude	-0.491897
housing_median_age	0.120870
total_rooms	-0.163133
total_bedrooms	0.390821
population	-0.339639
households	0.163509
median_income	0.616856
ocean_proximity_INLAND	-0.397752
ocean_proximity_ISLAND	1.776250
ocean_proximity_NEAR BAY	-0.066618
ocean_proximity_NEAR OCEAN	0.039896

Parameter	Value
R-2 on Test Data set	0.6199
MSE (Train)	60,556
MSE (Test)	61,706