Linear Regression Analysis

Timothy Cox TMC190004, Ryan Sharp RXS180092

Dataset Details

We chose the provided “California Housing Dataset”. This dataset contains 9 predictor variables and one response variable. The dataset contains 20640 rows of data. The goal for this data is to try to predict median house value using our predictor variables. The data is a summary of houses in each California district.

```
housing.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   longitude           20640 non-null  float64
 1   latitude            20640 non-null  float64
 2   housing_median_age  20640 non-null  int64
 3   total_rooms         20640 non-null  int64
 4   total_bedrooms      20433 non-null  float64
 5   population          20640 non-null  int64
 6   households          20640 non-null  int64
 7   median_income       20640 non-null  float64
 8   median_house_value  20640 non-null  int64
 9   ocean_proximity     20640 non-null  object
dtypes: float64(4), int64(5), object(1)
memory usage: 1.6+ MB
```

Predictors:

longitude, latitude, housing_median_age, total_rooms, total_bedrooms, population, households, median_income, ocean_proximity

Response: median_house_value (USD)

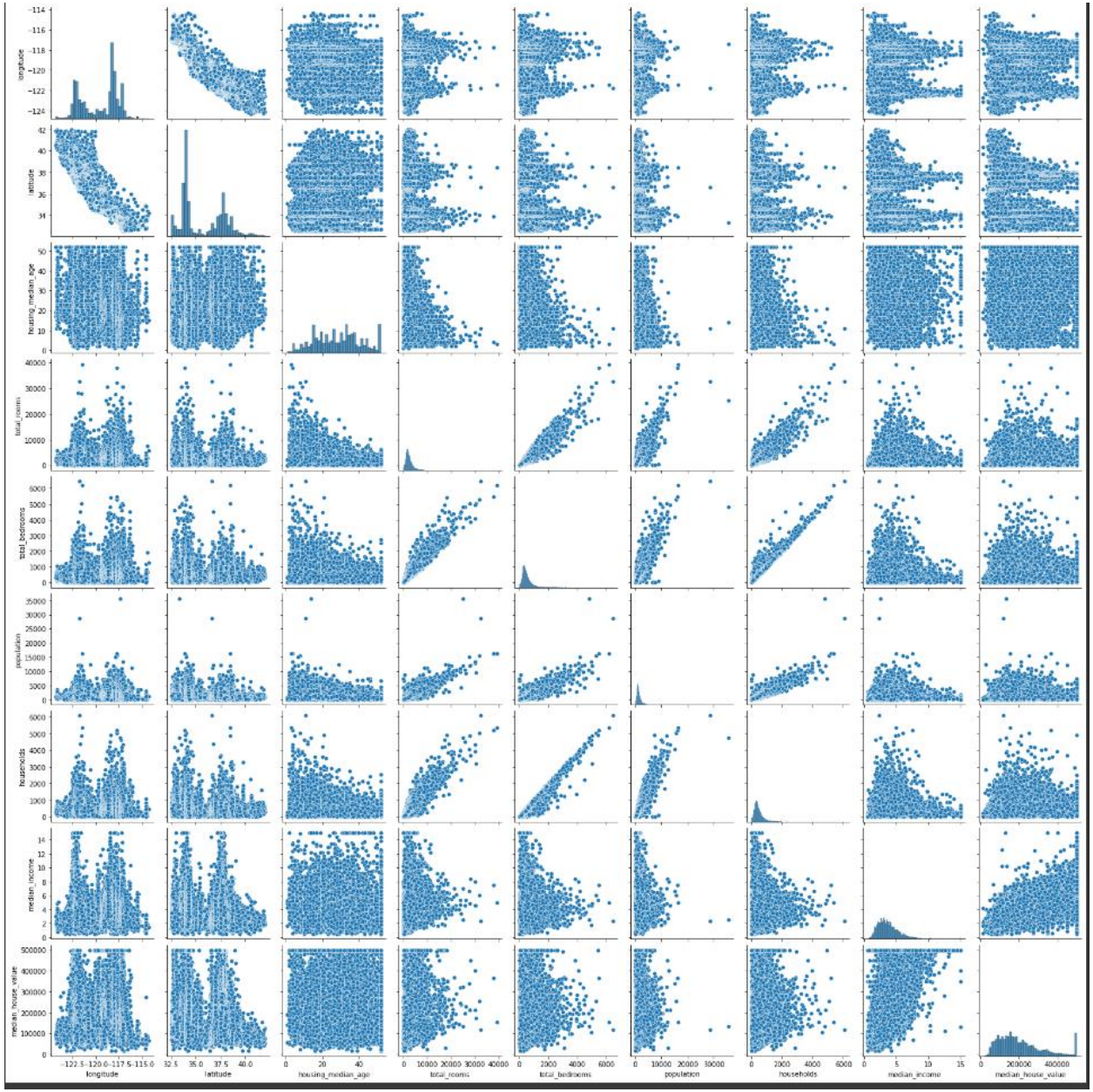
Note: Ocean_proximity is a categorical variable, we must convert it to a group of indicator/dummy variables for linear regression.

Correlation Matrix*

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
longitude	1.00	-0.92	-0.11	0.04	0.07	0.10	0.06	-0.02	-0.05
latitude	-0.92	1.00	0.01	-0.04	-0.07	-0.11	-0.07	-0.08	-0.14
housing_median_age	-0.11	0.01	1.00	-0.36	-0.32	-0.30	-0.30	-0.12	0.11
total_rooms	0.04	-0.04	-0.36	1.00	0.93	0.86	0.92	0.20	0.13
total_bedrooms	0.07	-0.07	-0.32	0.93	1.00	0.88	0.98	-0.01	0.05
population	0.10	-0.11	-0.30	0.86	0.88	1.00	0.91	0.00	-0.02
households	0.06	-0.07	-0.30	0.92	0.98	0.91	1.00	0.01	0.07
median_income	-0.02	-0.08	-0.12	0.20	-0.01	0.00	0.01	1.00	0.69
median_house_value	-0.05	-0.14	0.11	0.13	0.05	-0.02	0.07	0.69	1.00
ocean_proximity_<1H OCEAN	0.32	-0.45	0.05	-0.00	0.02	0.07	0.04	0.17	0.26
ocean_proximity_INLAND	-0.06	0.35	-0.24	0.03	-0.01	-0.02	-0.04	-0.24	-0.48
ocean_proximity_ISLAND	0.01	-0.02	0.02	-0.01	-0.00	-0.01	-0.01	-0.01	0.02
ocean_proximity_NEAR BAY	-0.47	0.36	0.26	-0.02	-0.02	-0.06	-0.01	0.06	0.16
ocean_proximity_NEAR OCEAN	0.05	-0.16	0.02	-0.01	0.00	-0.02	0.00	0.03	0.14

*Note: Some non-interesting variables were cut out from this image to make it a reasonable size to fit into the report. (Full graph in code).

If we look at our last column (our response variable), we can see median_income has the highest correlation with our response. We also can see that several ocean_prox indicator variables have significant correlation with our response.



PairPlot*

We can gather a few interesting insights from this pair plot.

First, looking at the response in the far-right column, the only predictor that looks like it has a linear relationship with our response is median_income (second row from the bottom).

Another noticeable insight from this plot is the center of the plot (Row4 – 7, column 4-7) looks to have a lot of predictors with linear relationships.

These predictors are: Total Rooms, Total bedrooms, population, and total household.

It makes intuitive sense that these predictors could be dependent on each other. It's important to remember that these predictors are based on a summary of houses in a district.

More bedrooms → more rooms → more households → more population.

To deal with these linear relationships between predictors, we will only keep "total rooms" as it has the highest correlation with our response.

Post-EDA Dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 8 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   median_income               20640 non-null  float64
 1   total_rooms                 20640 non-null  int64
 2   ocean_proximity_<1H OCEAN   20640 non-null  uint8
 3   ocean_proximity_INLAND      20640 non-null  uint8
 4   ocean_proximity_ISLAND      20640 non-null  uint8
 5   ocean_proximity_NEAR BAY    20640 non-null  uint8
 6   ocean_proximity_NEAR OCEAN  20640 non-null  uint8
 7   median_house_value          20640 non-null  int64
dtypes: float64(1), int64(2), uint8(5)
memory usage: 584.7 KB
```

After digging deeper into the data, we can reduce our dataset to only variables we believe are important to predicting our response variable. In this case, we have reduced our data to median_income, total_rooms, and ocean_prox indicator variables.

Model Creation/Results

We will use an ordinary least squares model, and a stochastic gradient decent model then compare their performance.

```
OLS train score:  0.5861520324427223
OLS test score:  0.5983851001461998
RMSE:  74519.86116348363
```

The metrics we will look at is training R-squared, test R-squared and root mean squared error.

```
SGD train score:  0.5444082860254069
SGD test score:  0.5561828509099667
RMSE:  78187.89699578121
```

The OLS model provides us with a higher R-squared than the SGD model. We can also notice that for both of our models, test score is higher than train score. This gives evidence that we are not overfitting to the training data. A root mean squared error of 70,000+ in the context of our data is not as bad as it looks. Our response variable is median house value in dollars. This means we are dealing with a response that is expected to be hundreds of thousands of dollars.

Based on these metrics, I would use the OLS model.