# HOUSE PRICE PREDICTION

Using Machine Learning

# Introduction

- We will be building models to predict house prices in California using California Census data.
- It consist of metrics  such as  population, median income, median house price and others for each block group in California which typically consists of population from 600 too 3,000.
- The ultimate goal of the project is to build a prediction engine capable of predicting district’s median housing price.

Project

# Process

| Step 1 | Step 2 | Step 3 | Step 4 | Step 5 | Step 6 |
| --- | --- | --- | --- | --- | --- |
| Get the data. | Discover and visualize the data to gain insights. | Prepare the data for Machine Learning algorithms. | Select a model and train it. | Fine-tune your model. | Present your model |

# Data Prepration

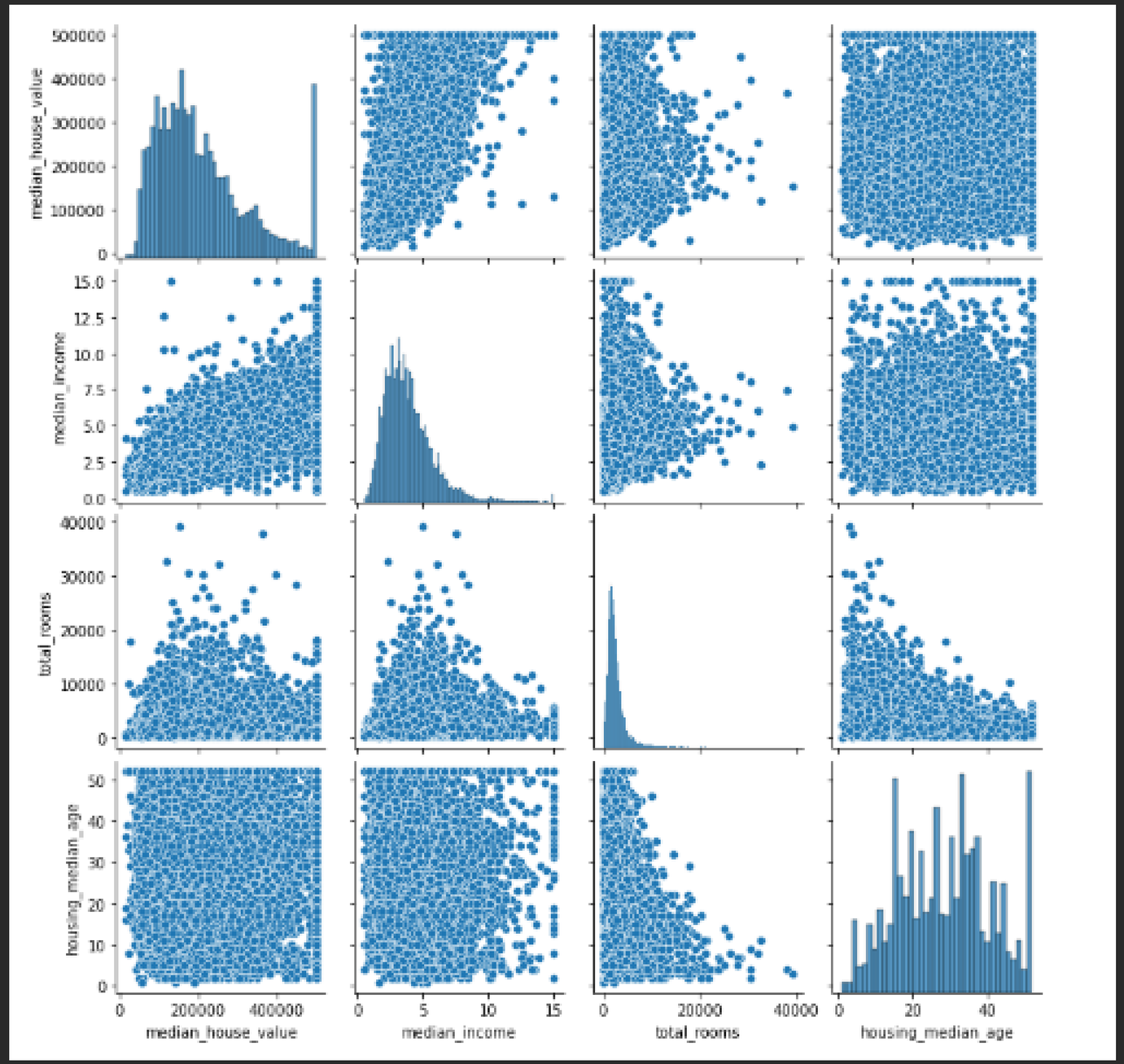
- df.info(),describe(),head() are probably one of the first things we want to inspect having a pandas dataframe; showing feature names, limits/stats and and a few first columns respectively, to get a some initial impression of the data

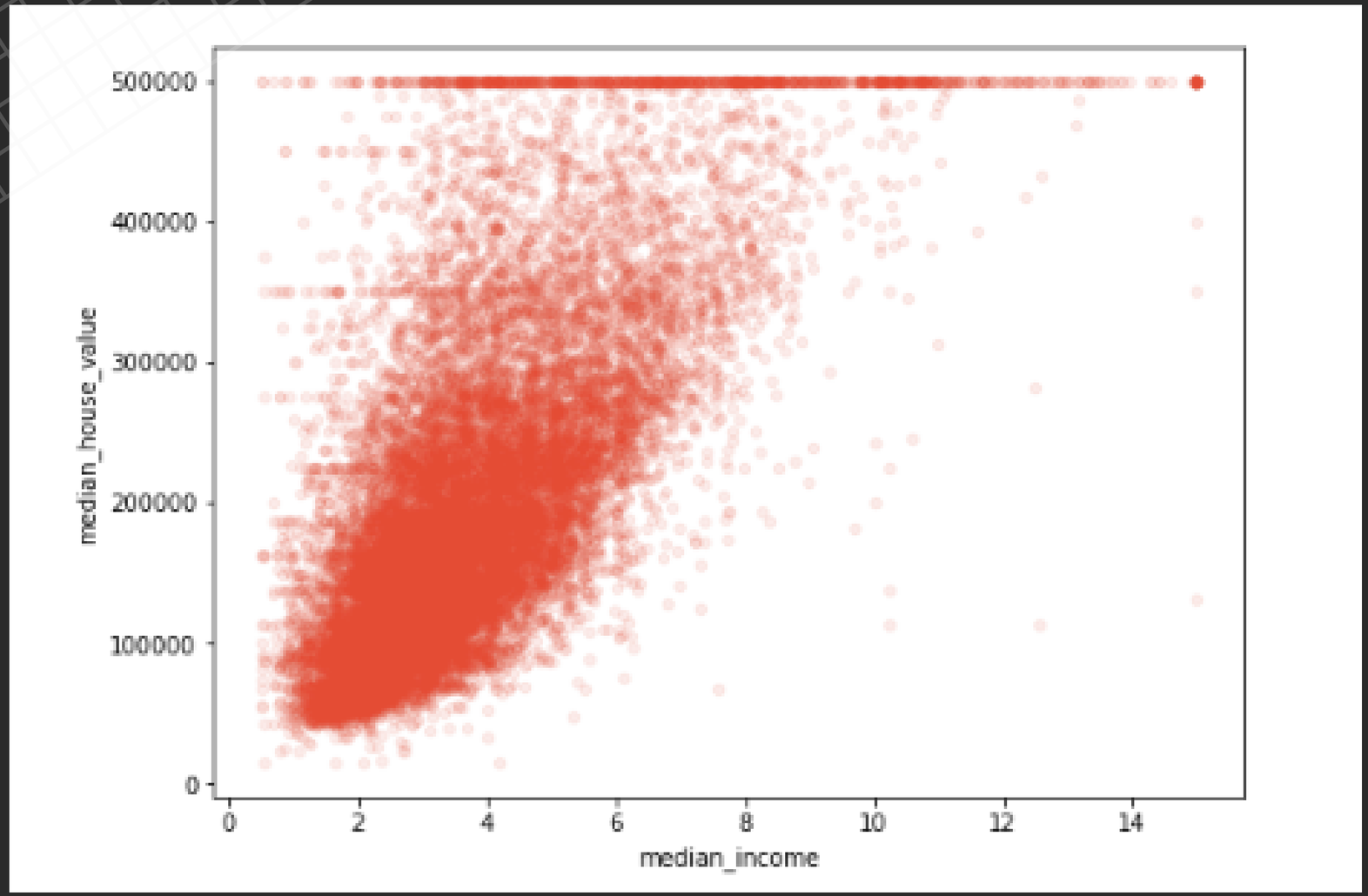
```
data.head()
```

| longitude | latitude | housing_median_age | total_rooms | total_bedrooms | population | households | median_income | median_house_value | ocean_proximity |
|---|---|---|---|---|---|---|---|---|---|
| -122.23 | 37.88 | 41.0 | 880.0 | 129.0 | 322.0 | 126.0 | 8.3252 | 452600.0 | NEAR BAY |
| -122.22 | 37.86 | 21.0 | 7099.0 | 1106.0 | 2401.0 | 1138.0 | 8.3014 | 358500.0 | NEAR BAY |
| -122.24 | 37.85 | 52.0 | 1467.0 | 190.0 | 496.0 | 177.0 | 7.2574 | 352100.0 | NEAR BAY |
| -122.25 | 37.85 | 52.0 | 1274.0 | 235.0 | 558.0 | 219.0 | 5.6431 | 341300.0 | NEAR BAY |
| -122.25 | 37.85 | 52.0 | 1627.0 | 280.0 | 565.0 | 259.0 | 3.8462 | 342200.0 | NEAR BAY |

# DATA ANALYSIS

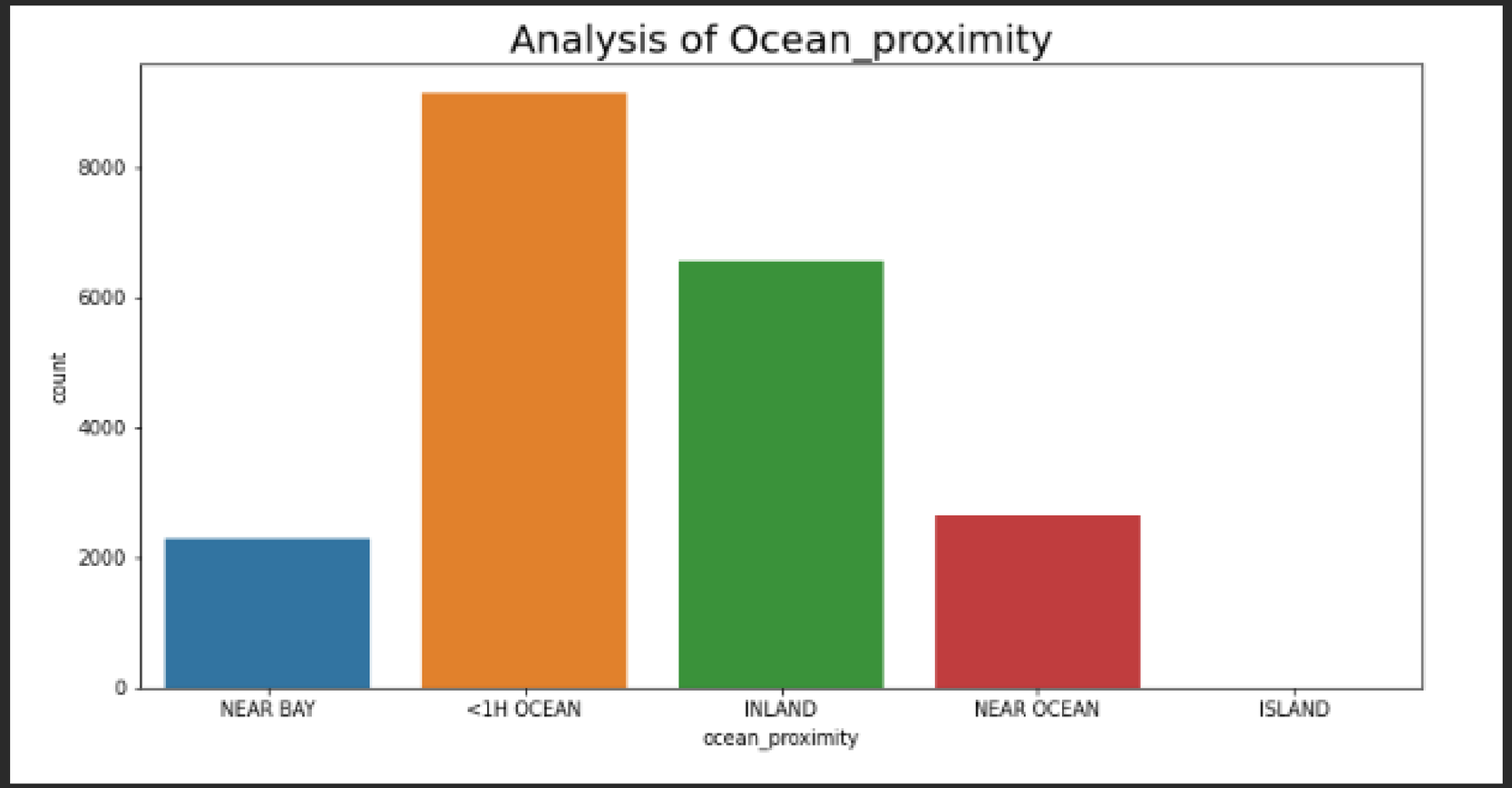
# Scatter Matrix






- This plot reveals a few things. First, the correlation is indeed very strong;
- you can clearly see the upward trend and the points are not too dispersed.

# Ocean Proximity



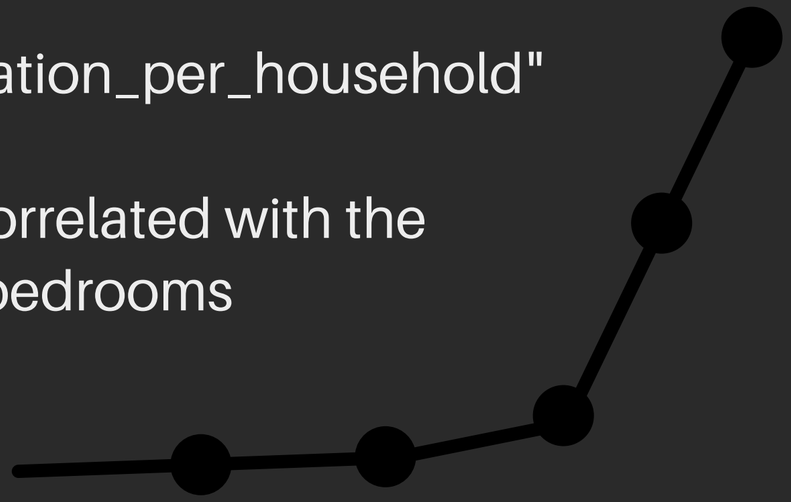
# EDA Conclusions

Conclusions from EDA:

- 1) There are outliers in all features .
- 2) In general, the distribution is close to normal
- 3) High correlation (~0.7) between target variable(median_house_value) and median_income
- 4) High correlation between features, this is multicollinearity
- 5) Features of longitude and latitude are valuable information, use this in feature generation
- 6) Houses (<1H OCEAN) the largest number

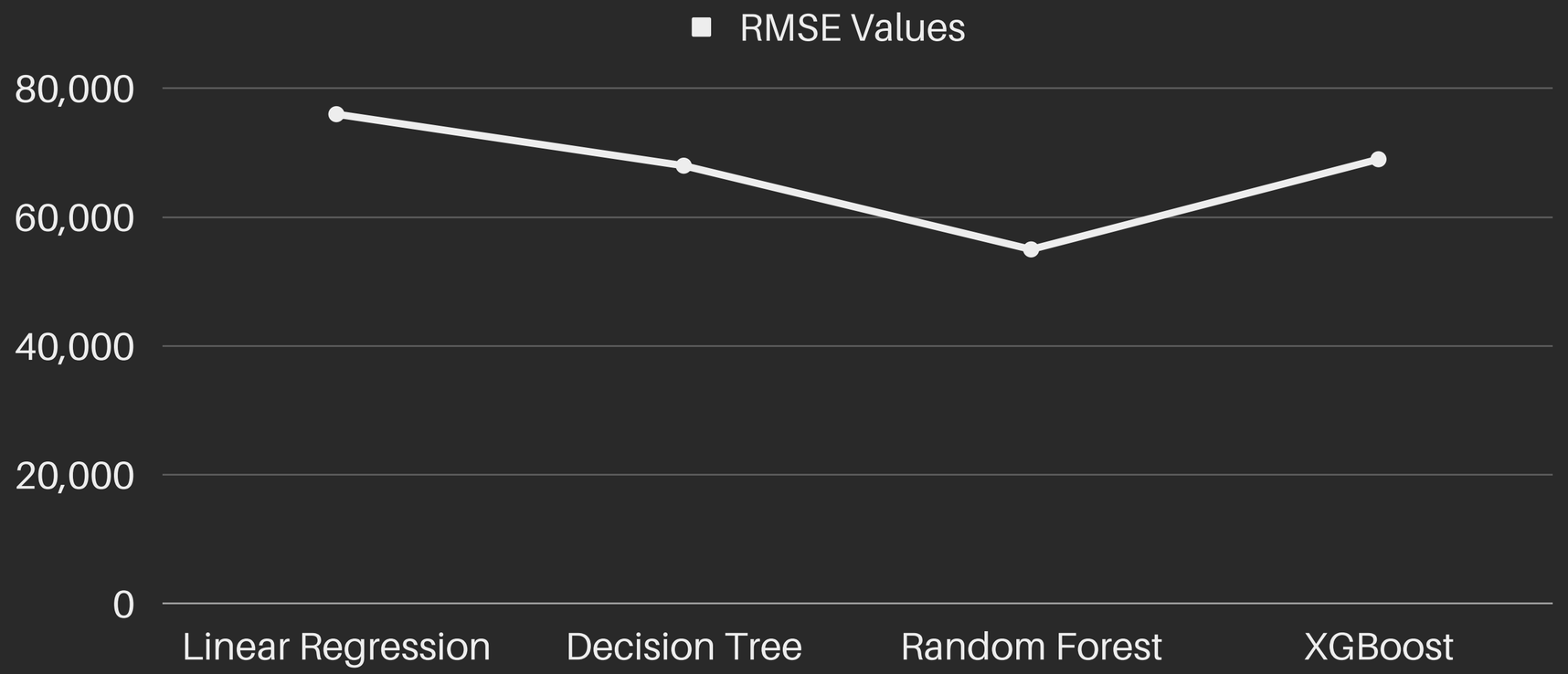
# Feature Engineering

- Creation and modification of the feature matrix data,Feature Engineering is quite important and quite a cyclic process, we want to input a feature matrix that will help teach a model something useful.
- We want to make sure we feed the model data that is most relevant to the prediction of a target variable, perhaps as less overlapping as possible as well.
- Features with very high correlation teach a model similar things, multiple times, maybe consider combing them and dropping the others.
- we will create new feature that is "rooms_per_household","bedrooms_per_room", "population_per_household" as
- The new bedrooms_per_room attribute is much more correlated with the median house value than the total number of rooms or bedrooms



# Model Selection

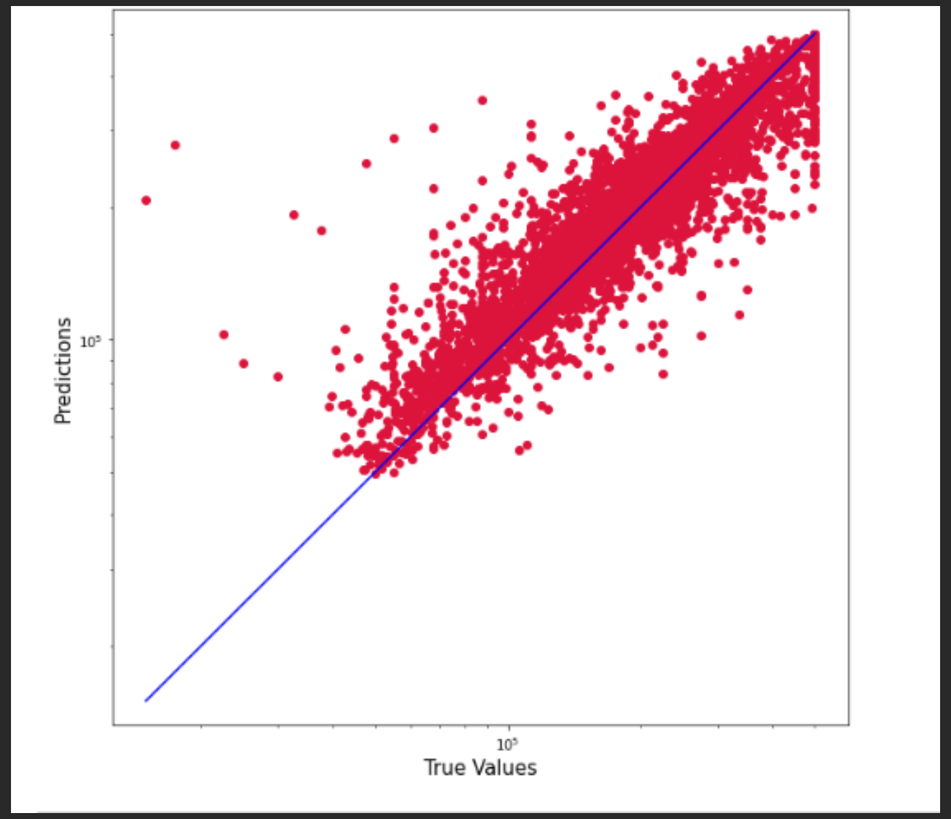
After using all the models we get Random forset as the one of the best model with the error rate of $50,182 even though we see that error rate is pretty high in validation datasets compare to training sets suggesting there might be over fitting issue.



# Model Fine Tuning

After fine tuning the model we get following conclusions:

- It tells us that prediction error  can fluctuate anywhere between $45,685 to $49,691.
- We get r2 score around 81 perceent



# Conclusions and Next Steps

I believe this model could be optimized and tuned more to add accuracy either by adding new features or engineering new features. This model can be used to predict the house prices in any geographic location by just slightly fine tuning the features and parameters.

# Thank you!