Artificial Intelligence (UCS411) LAB PROJECT SUBMISSION

Submitted to: Ms Chandni Submitted by:

Abhinav Lather (102103513)

Yash Rajoria (102103526)

# Problem Statement:

The target variable is the median house value for California districts, expressed in hundreds of thousands of dollars ($100,000).

# Description of the problem:

This dataset was obtained from the StatLib

repository. <https://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html>

This dataset was derived from the 1990 U.S. census, using one row per census block group. A block group is the smallest geographical unit for which the U.S. Census Bureau publishes sample data (a block group typically has a population of 600 to 3,000 people).

A household is a group of people residing within a home. Since the average number of rooms and bedrooms in this dataset are provided per household, these columns may take surprisingly large values for block groups with few households and many empty houses, such as vacation resorts.

We strive to obtain the median house value for the California district, as median prices are a useful tool for understanding the price changes of properties that have transacted in a market. Furthermore, the median is more accurate than the average because it is less affected by a few unusually high or low sale prices.

Hence this helps property dealers and buyers alike in understanding the prices of houses in a particular area, and how they rise and fall, and their relation to the larger framework of property price growth.

# Code and Explanation:



1. **Problem definition** — The given model is one of supervised learning, where we are going to use regression, since we have to find a numerical quantity.
2. **Data** — California Housing Dataset [https://scikit-](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_california_housing.html) [learn.org/stable/modules/generated/sklearn.datasets.fetch\_california\_h](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_california_housing.html) [ousing.html](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_california_housing.html)
3. **Evaluation** — Evaluation using coefficient of determination(r^2),cross validation score and mean absolute error.
4. **Features** — 8 numeric, predictive attributes and the target
5. **Modelling** — We use both the Ridge Regressor and the RandomForestRegressor, with the latter having the highest score.
6. **Experimentation** — What else could we try? Does our deployed model

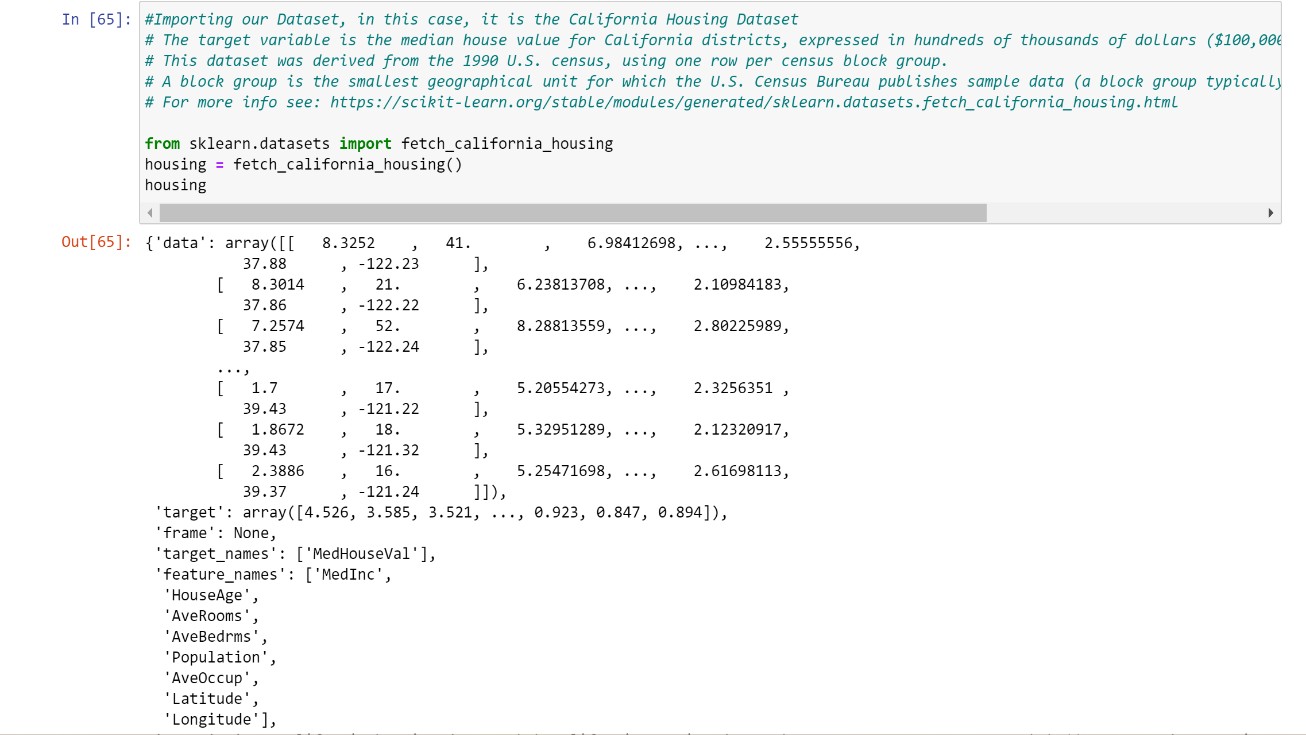
do as we expected? How do the other steps change based on what

we’ve found?

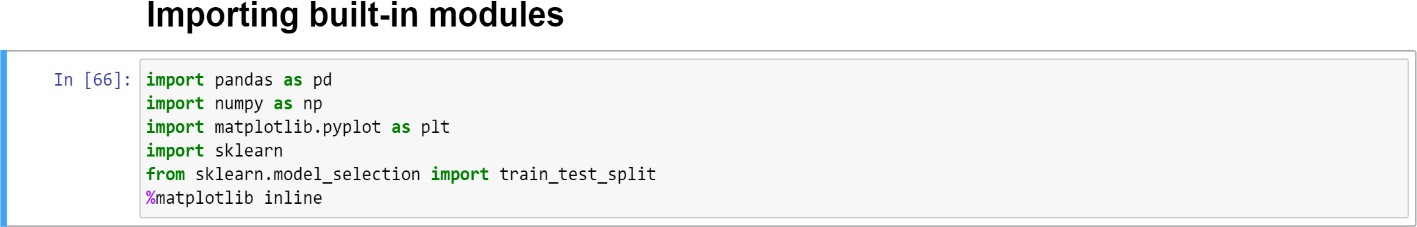
## The above is the framework we rely on.

**The following is how we approached the problem-**

## Firstly, import the dataset from the scikit library



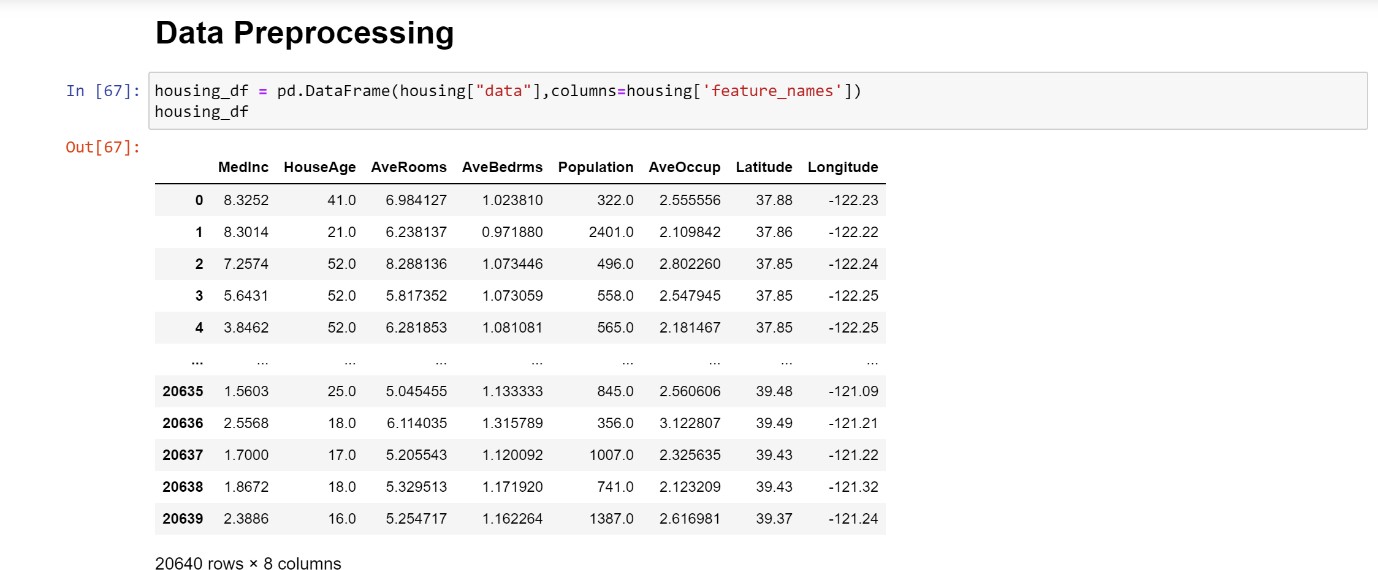
Next, we imported built-in modules –



## A brief description of the modules imported is-

1. **Pandas-**pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.
2. **Numpy-** NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
3. **matplotlib-** Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general- purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.
4. **sklearn-**scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms.

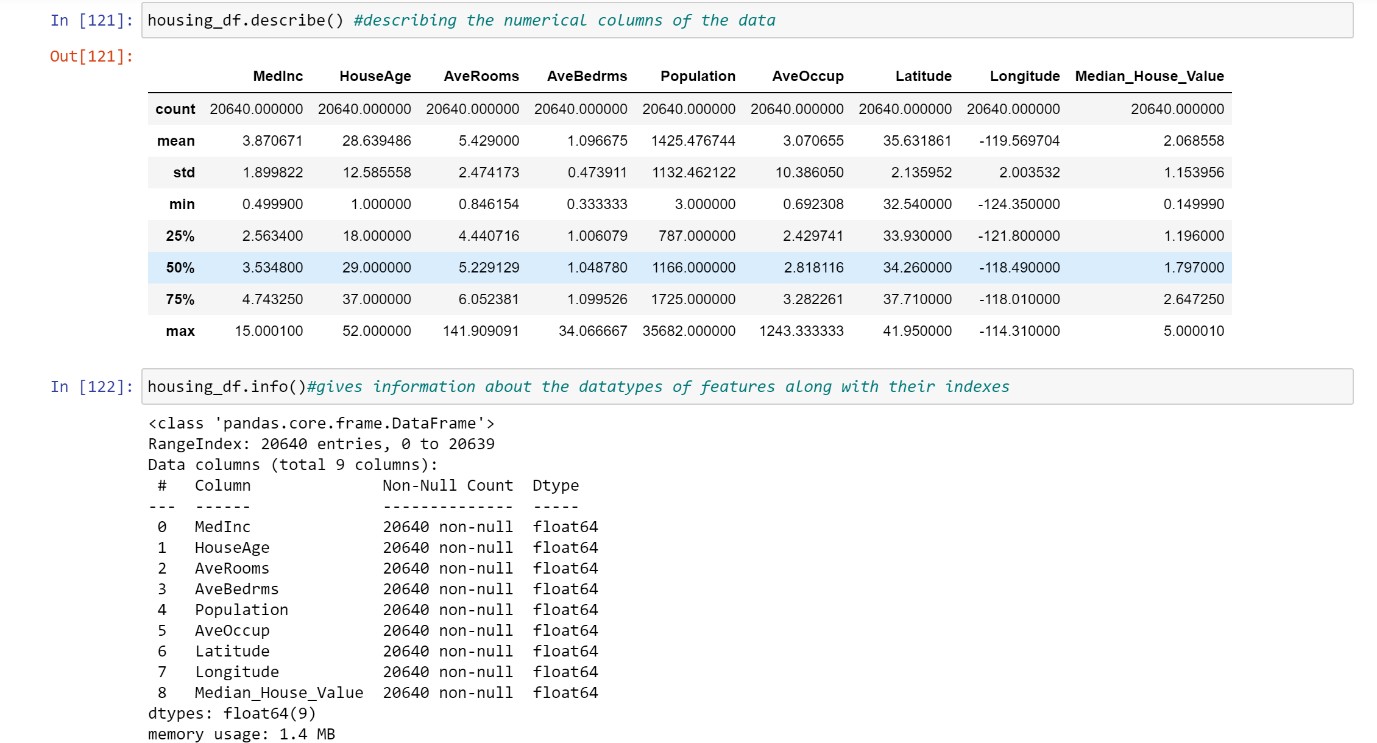
## We then preprocessed the Dataset before building the Machine Learning model.

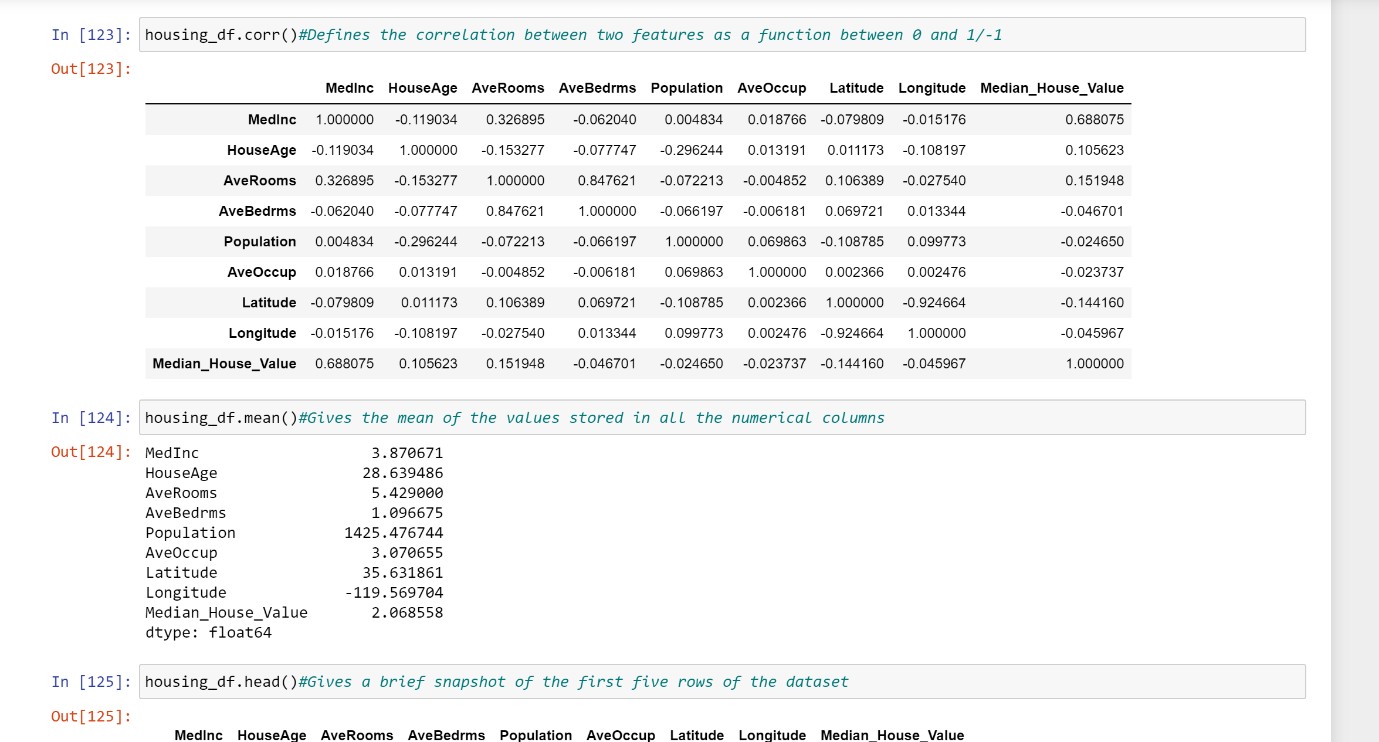


We firstly took out the dataframe with the columns as feature\_names from the larger dataset. We used the features inorder to predict the label, which is the median house value.

We then used the describe function in order to describe the various mathematical factors of the numerical columns, like count,std,max,min mean of the various features which might be useful later.

After that we used the info() function, which shows the datatypes of the various columns along with their indexes and their null criteria. This helped us in the future as we got to know that there is no missing data and that all the columns having floating point numerical values, hence we would not have to convert any column containing non-numerical values into numerical ones before making our model.



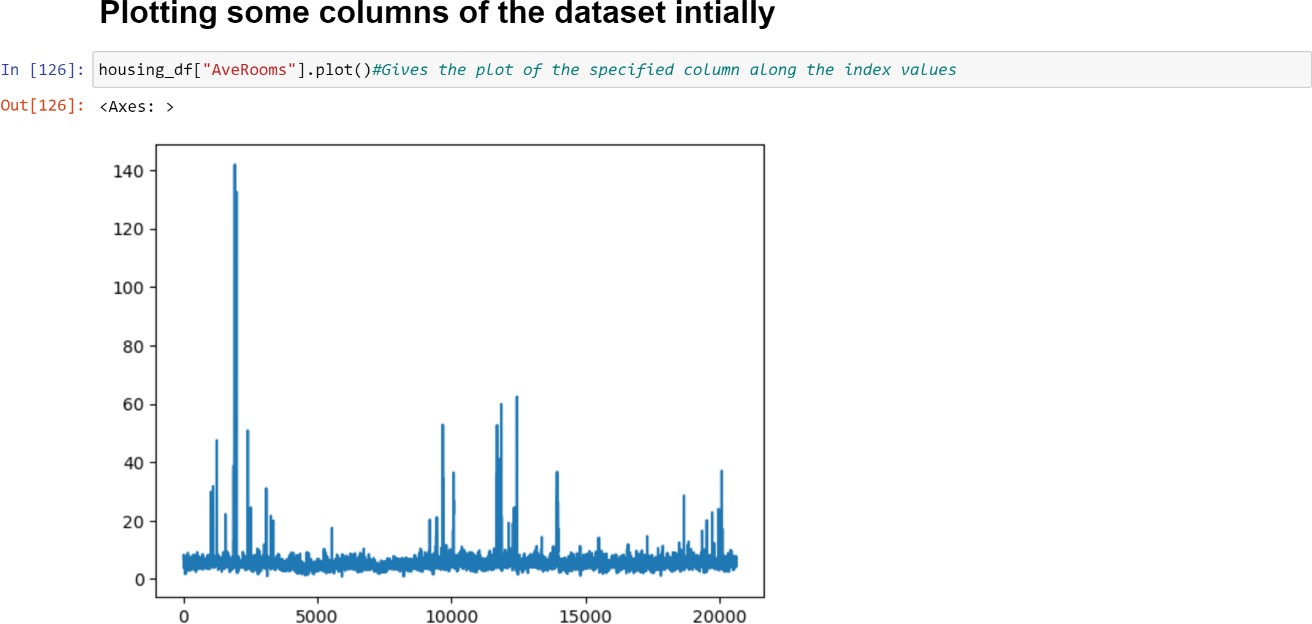


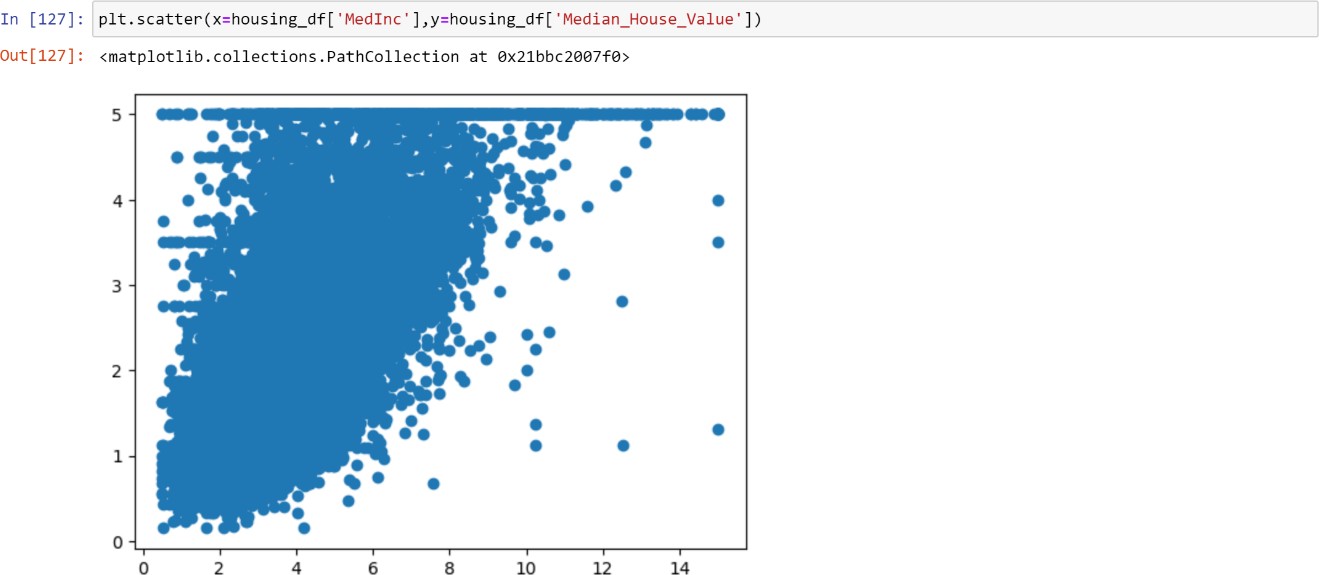
Next we used the corr() function which is used to determine the correlation between the columns. The correlation between median house value and other columns gives us a measure of which columns will be most useful while making our model. A value close to 1 or -1 indicates strong correlation between two columns.

The mean function gives the mean of the columns having numerical datatype.

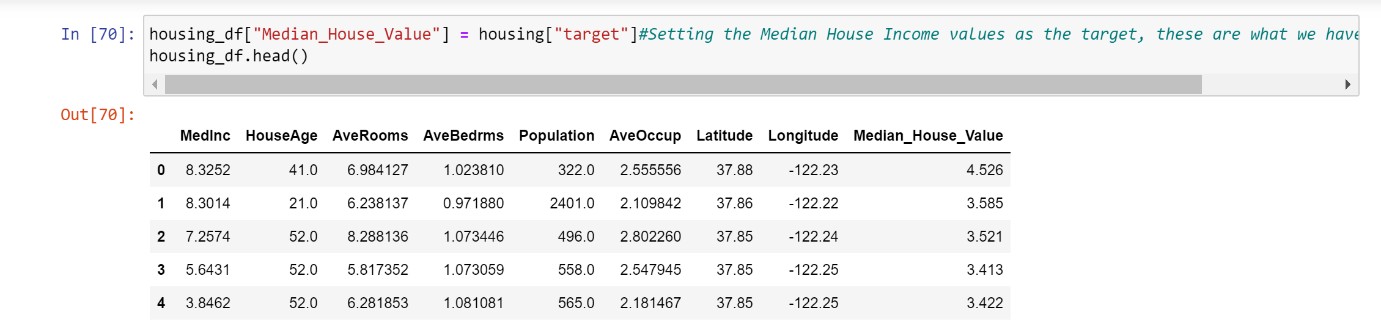
The head and the tail functions simply give us a snapshot of the data we are dealing with.

We then plotted some columns using the matplotlib module, along with their index. We then plotted some columns along the scatter plot, in order to find outliers in the data and check if we could remove some columns/reduce data in order to make our model more efficient.





We then made the column containing the Median House Values equivalent to the target column, since we have to find the Median House Values from the other 8 columns that we have.



We see that the column gets added along the end of our dataset as a target variable.

# Making the Machine Learning Model

**We start off by making the machine learning model using three steps-**

Three main things that we have to do-

1:Split the data into features and labels (usually 'X' and 'Y') 2.Filling (also called imputing) or disregarding missing values

3.Converting non-numerical values to numerical values(also called featuring coding)

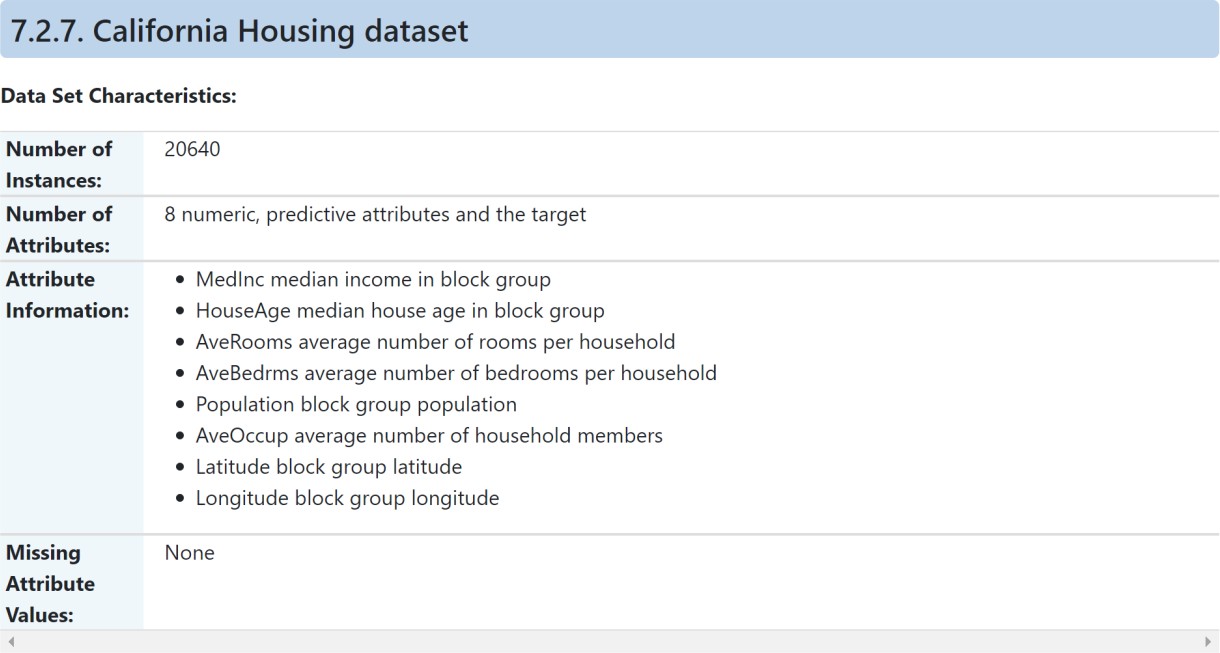
Since there are no missing values in our dataset, and all the columns contain numerical floating type values, so steps 2 and 3 do not apply to our dataset.

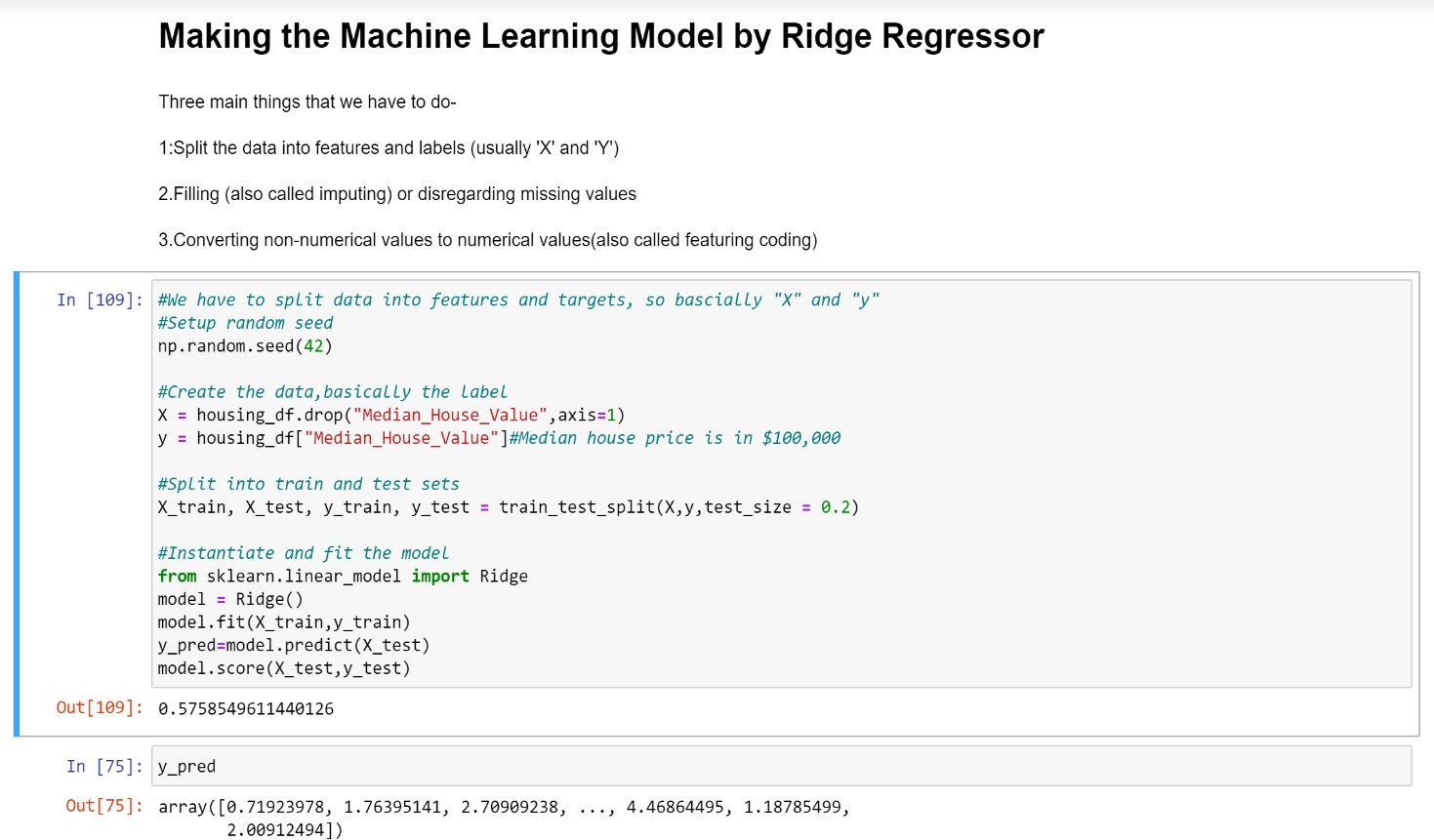
We use 8 numeric and predictive values in order to compute the target. We have initially used the Ridge Regressor in order to calculate the score,

basically how good our predictions formed from the training data by the model

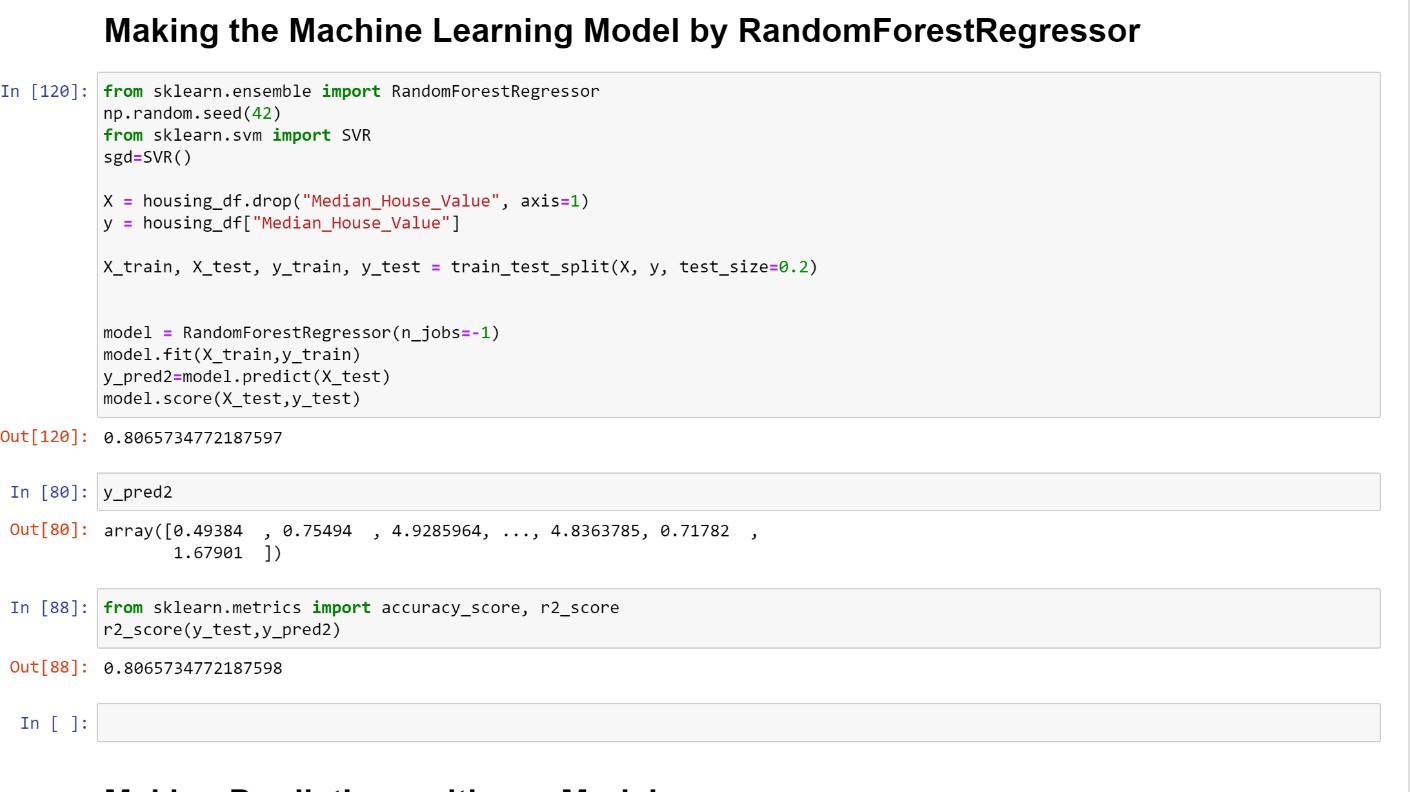
when applied on the testing data are.

It gives a low value of 0.5758549611440126.

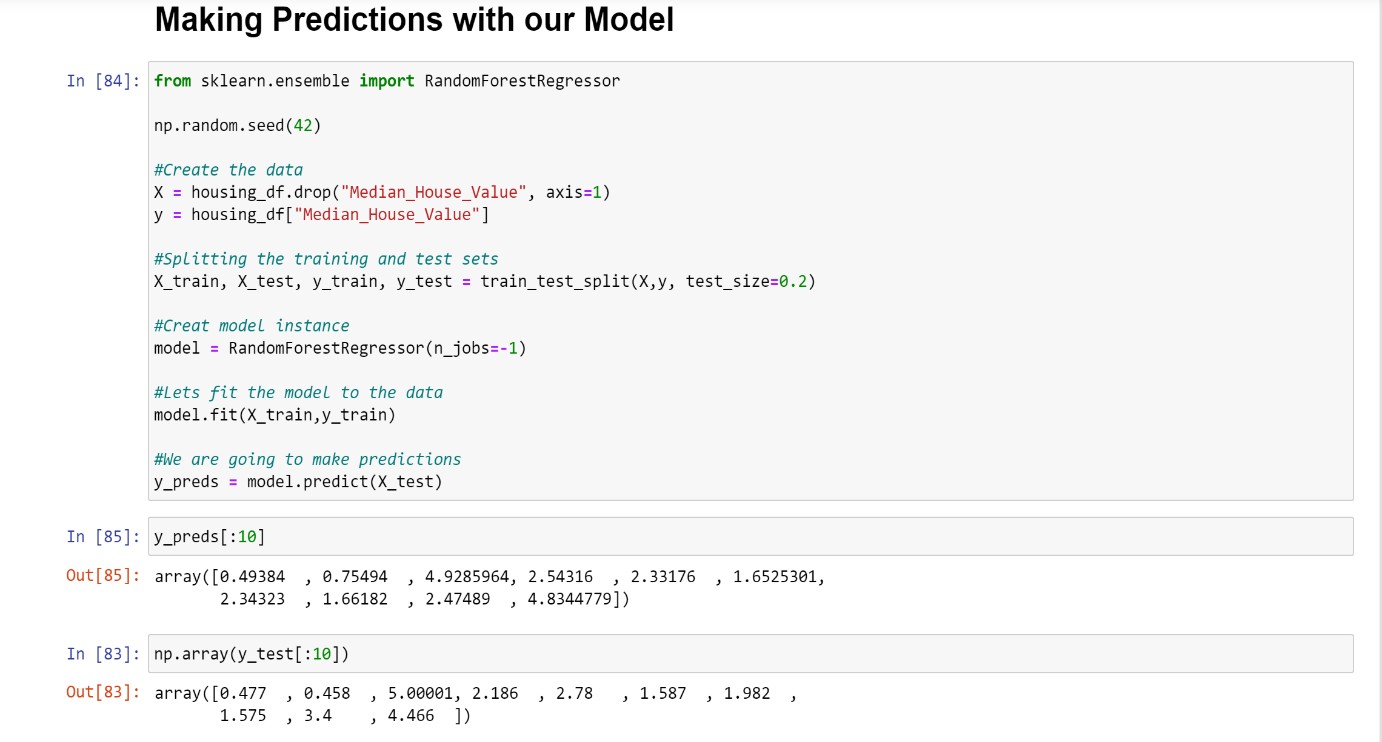


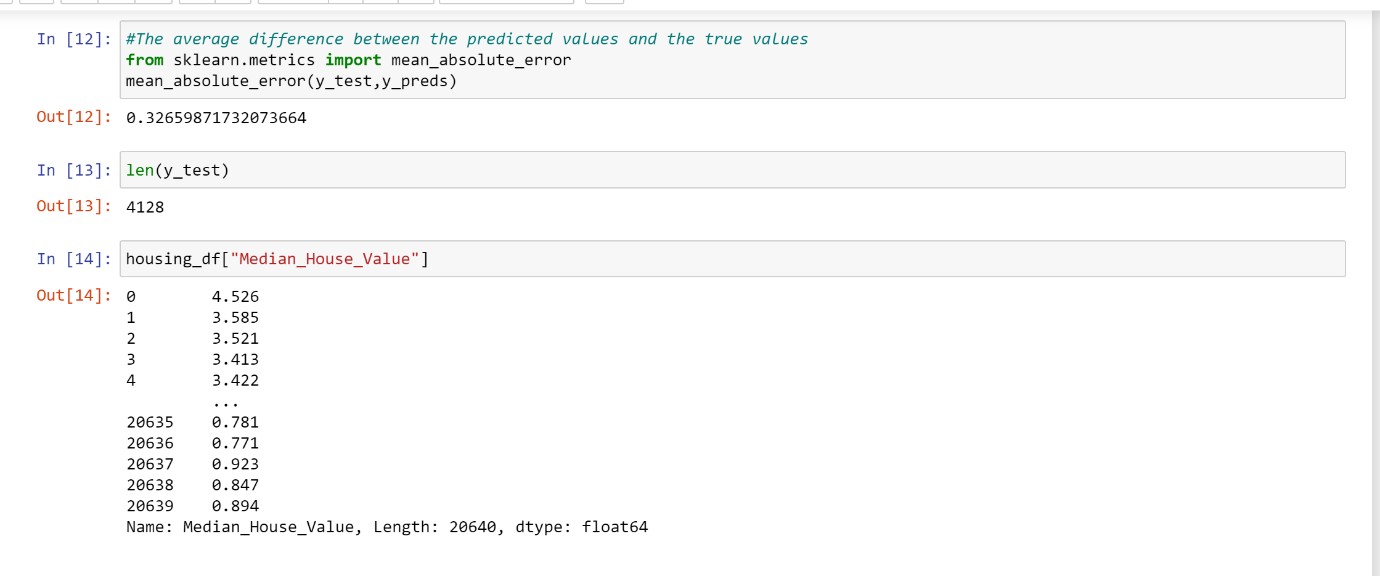


We then made a RandomForestRegressor model which came up with a score of 0.8, which was good, and we tried all other regressors, but they did not give a higher r^2 value, so we chose to stick with it.

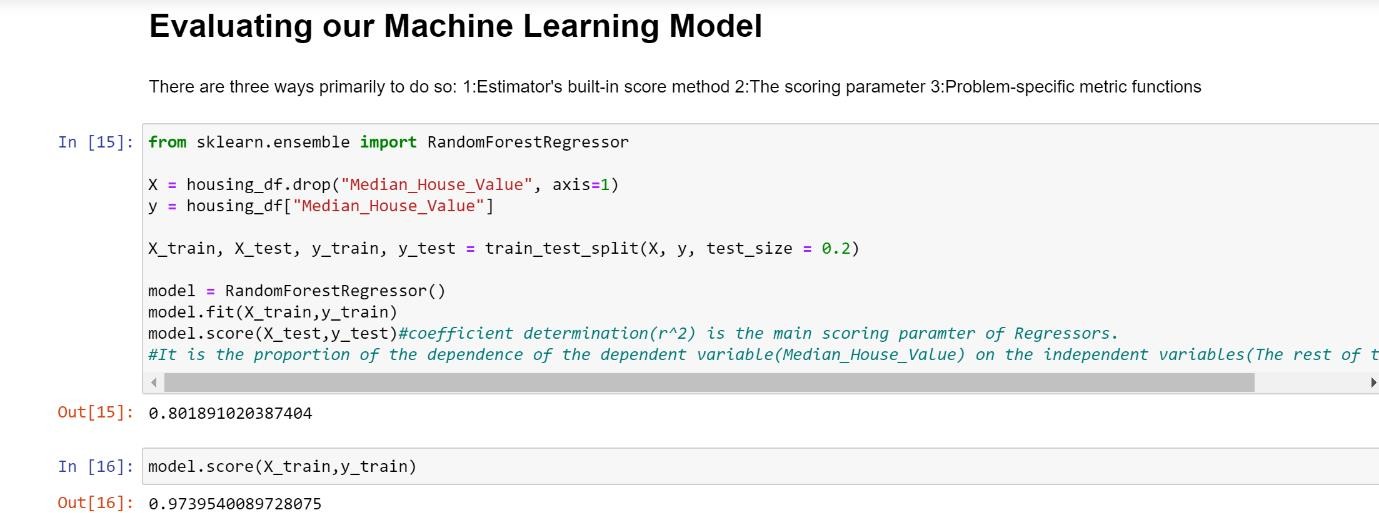


**We then made predictions with our model,** where we tried to find the output of a particular column within the test data, and then displayed the actual numeric values of the test data in an array. We then found that there was a mean absolute error of about 0.3.



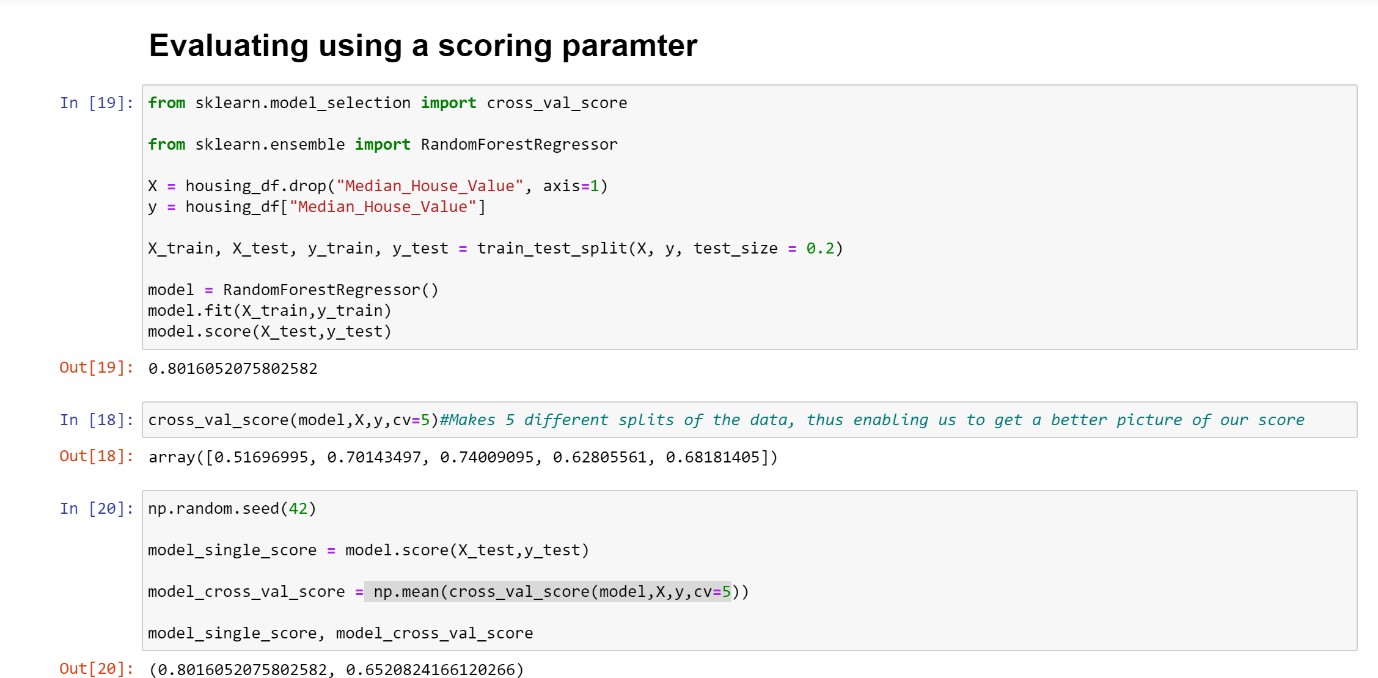


We then evaluated our Machine Learning model along various parameters, mainly the coefficient of determination, cross validation score, and mean absolute error.



The value of the coefficient of determination comes about 0.80, which means that our predictions made by the model are about 80 percent accurate. The conclusion is that we can make an 80 percent accurate guess about the median house values from the 8 features we have used, using our machine learning model.

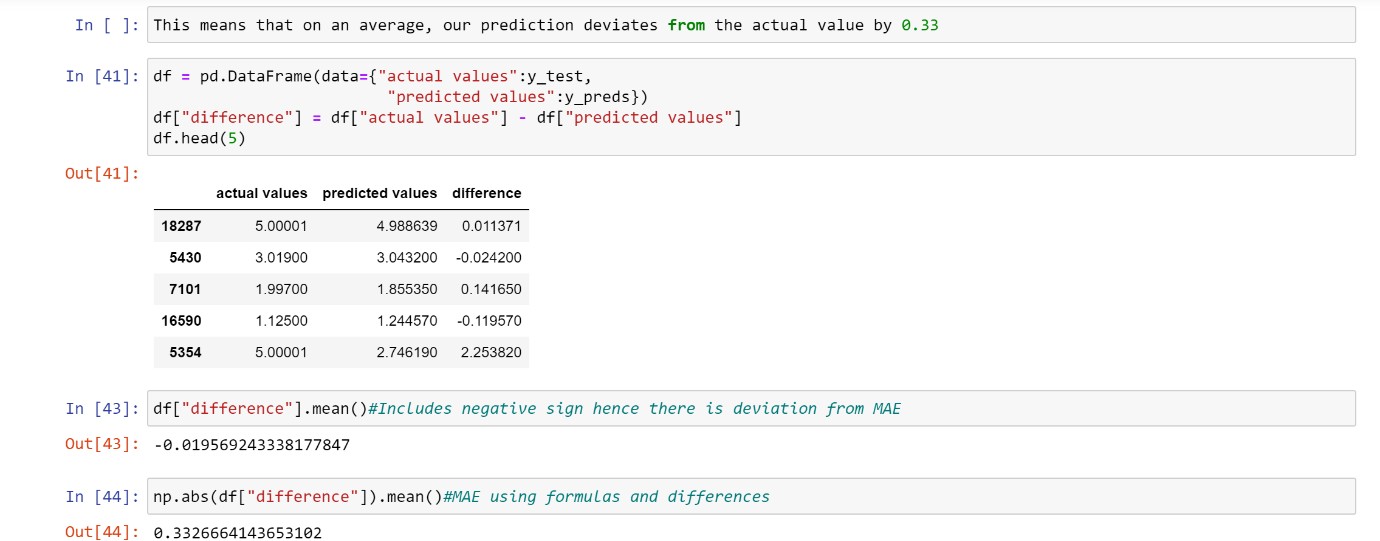
We then used the cross validation score in order to get a more holistic score for our model. It came out to be about 0.65, when the training and testing data were taken for 5 separate times, cv=5.



We then calculated the mean absolute error(MAE), in order to find out how our model’s predicted values were different from the true values.

It was indicative of how wrong our model could be, and it was then calculated that there was a deviation of 0.33 between the predicted and the actual test values on which the predictions were made.





**Conclusion:** We trained about 80% of the data and we tested our predictions on the remaining 20%, we used the RandomForestRegressor to get a score for the predictions made by our model, and we got a decent score of 0.8 after applying all improvement techniques. We evaluated our model based on various parameters like coefficient of determination,cross\_validation score, mean absolute error etc. We successfully predicted about the Median House Values as the label from the other 8 columns we used as features.