|  |  |
| --- | --- |
| Name: Atharv A Swarge | Name: Aditya R Joshi |
| Roll No: A218 | Roll No: A207 |
| SAP ID: 70012000273 | SAP ID: 70012000190 |
| B.Tech IT (3rd Year) | B.Tech IT (3rd Year) |

**House Price Prediction**

**Machine Learning Mini Project**

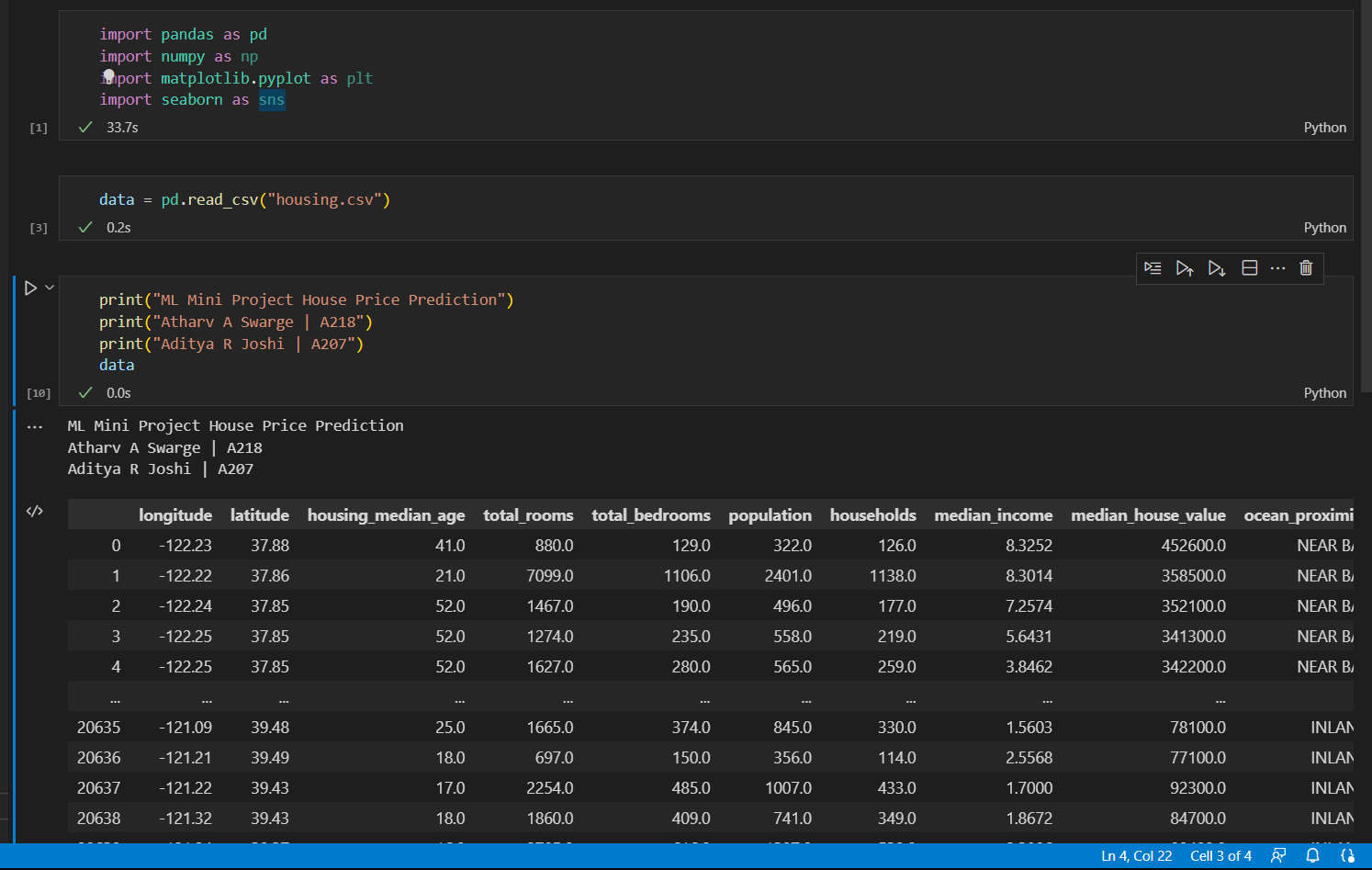
**Under the guidance of**

**Mr. Prashant Udawant**

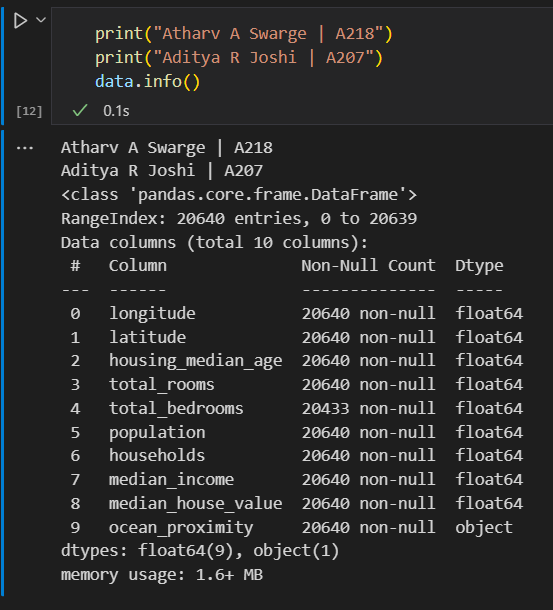
**Assistant Professor**

**Department of Information & Technology**

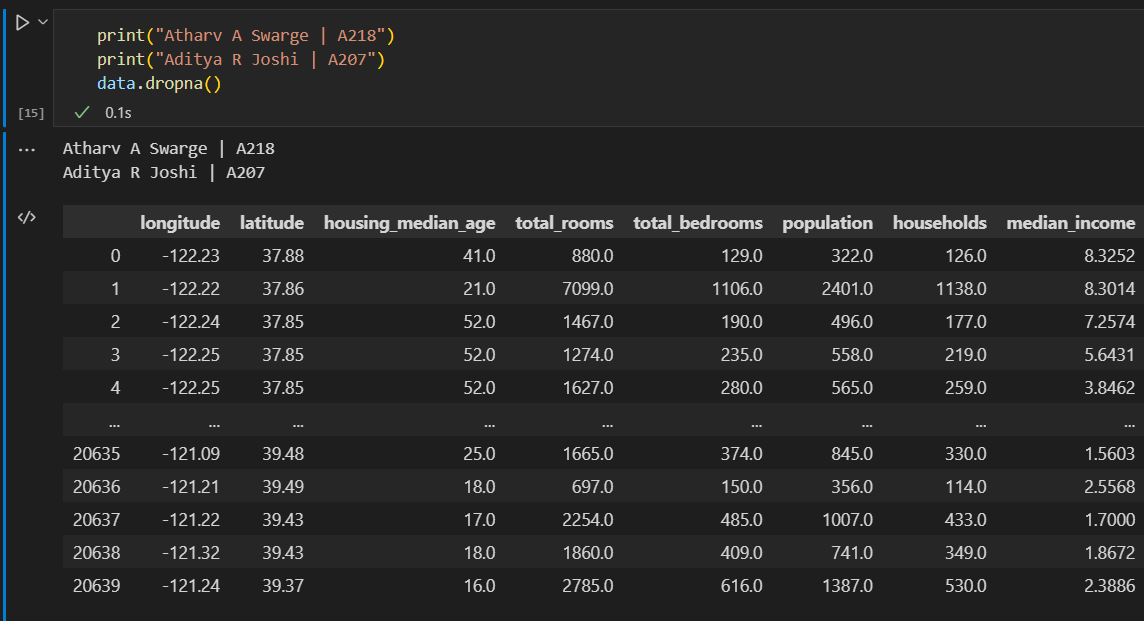
**NMIMS MPSTME, Shirpur.**



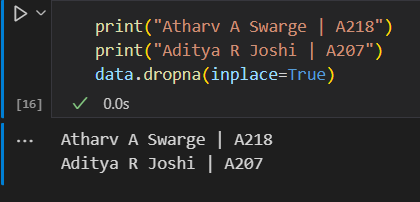
To find non-null values:

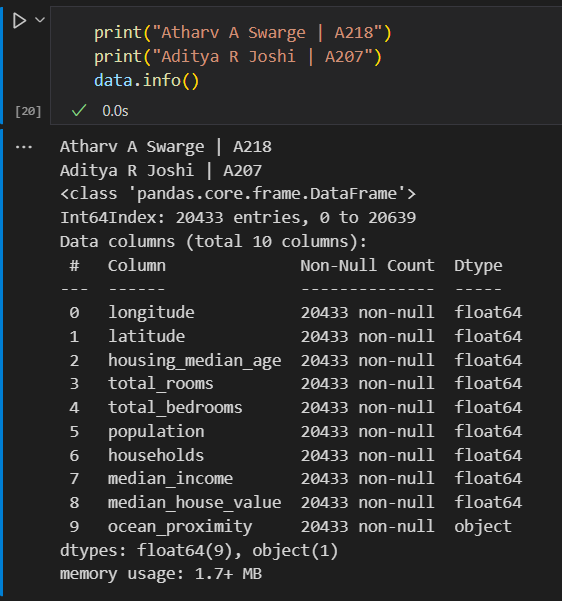


4. total bedrooms have some null values missing, so we will drop these NAN entries (not a number)



If we want these changes to get applied and get saved





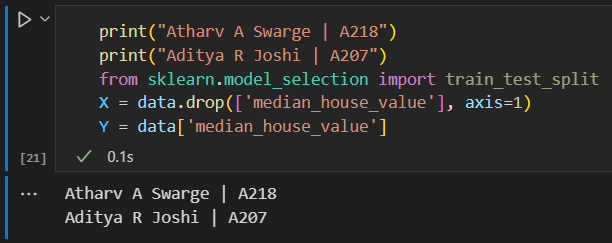
Now all the columns have same amount of non-null values.

Splitting the data into training and testing data and we will split it into X and Y data.

We want to train the model on one set of data and evaluate the model on another set of data.

We use train test split to split the data into X train X test and Y train Y test.

We need to define what X and Y is.

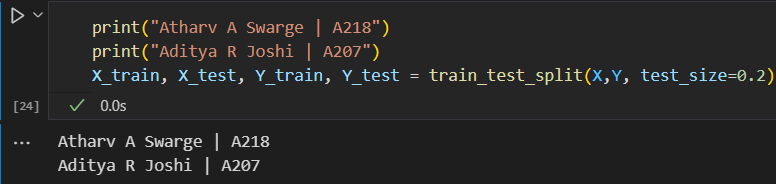


Full Dataframe without one column named as median house value.



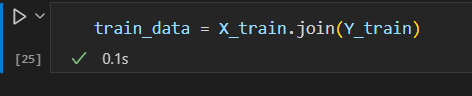
Now we specify how much percent of the data we are going to use for testing.

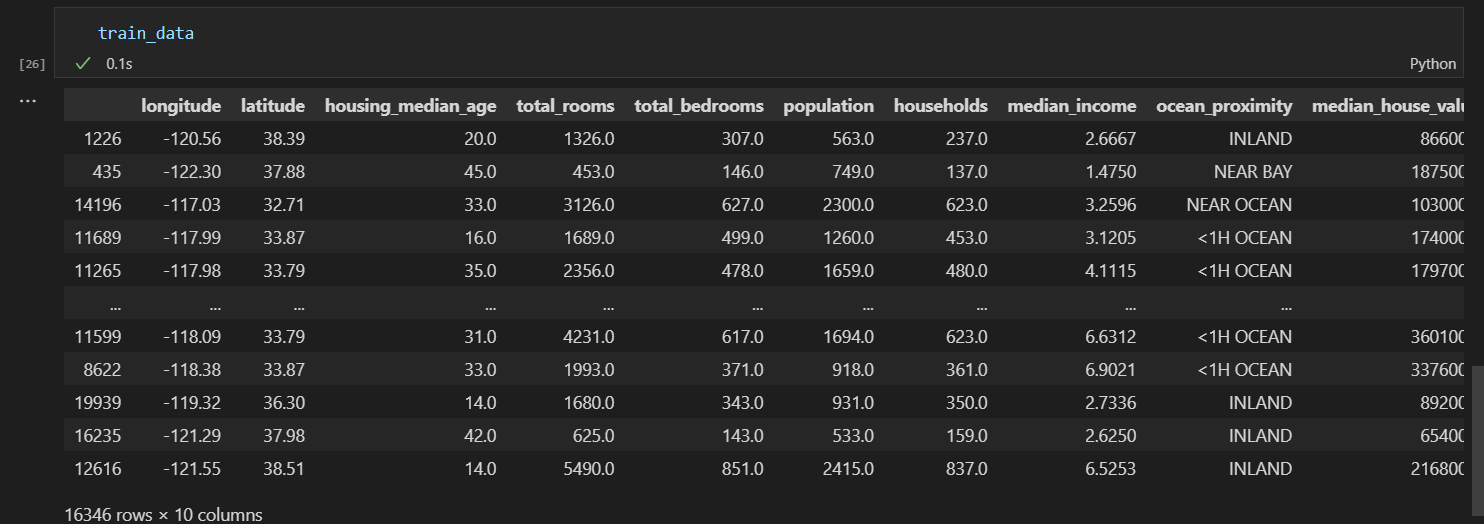
So, 20% of the data will be reserved for evaluating.



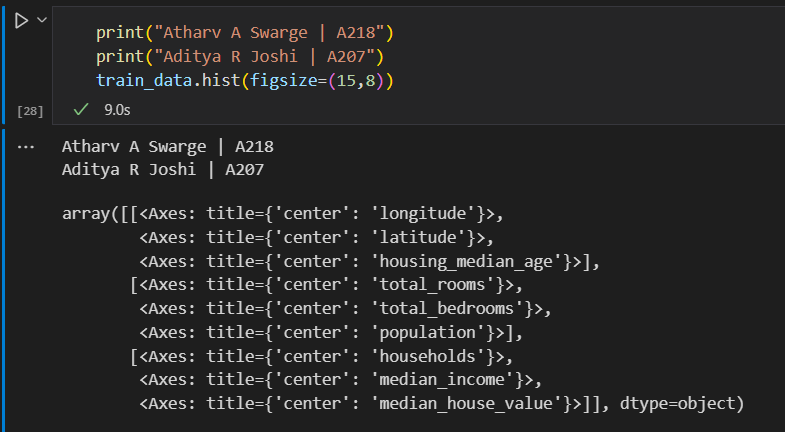
We will use this data only after we complete hyper-parameter tuning and proper training.

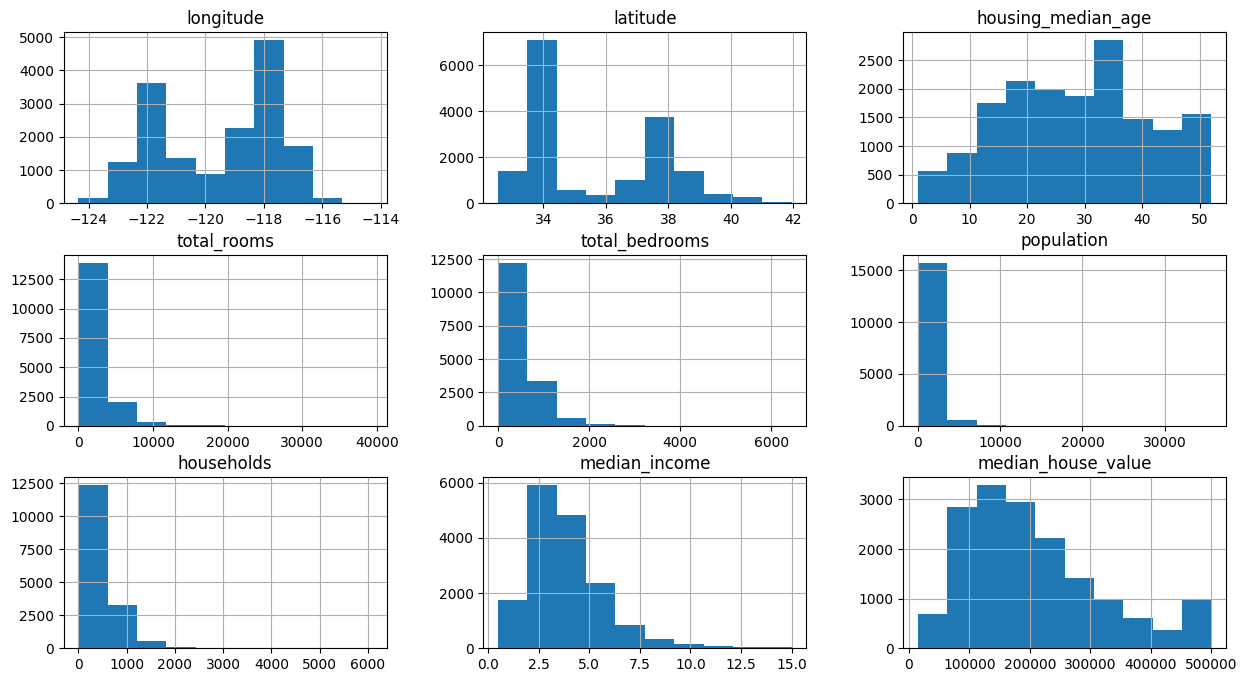
Now, we will try to find co-relation between the X and Y training data by joining them.



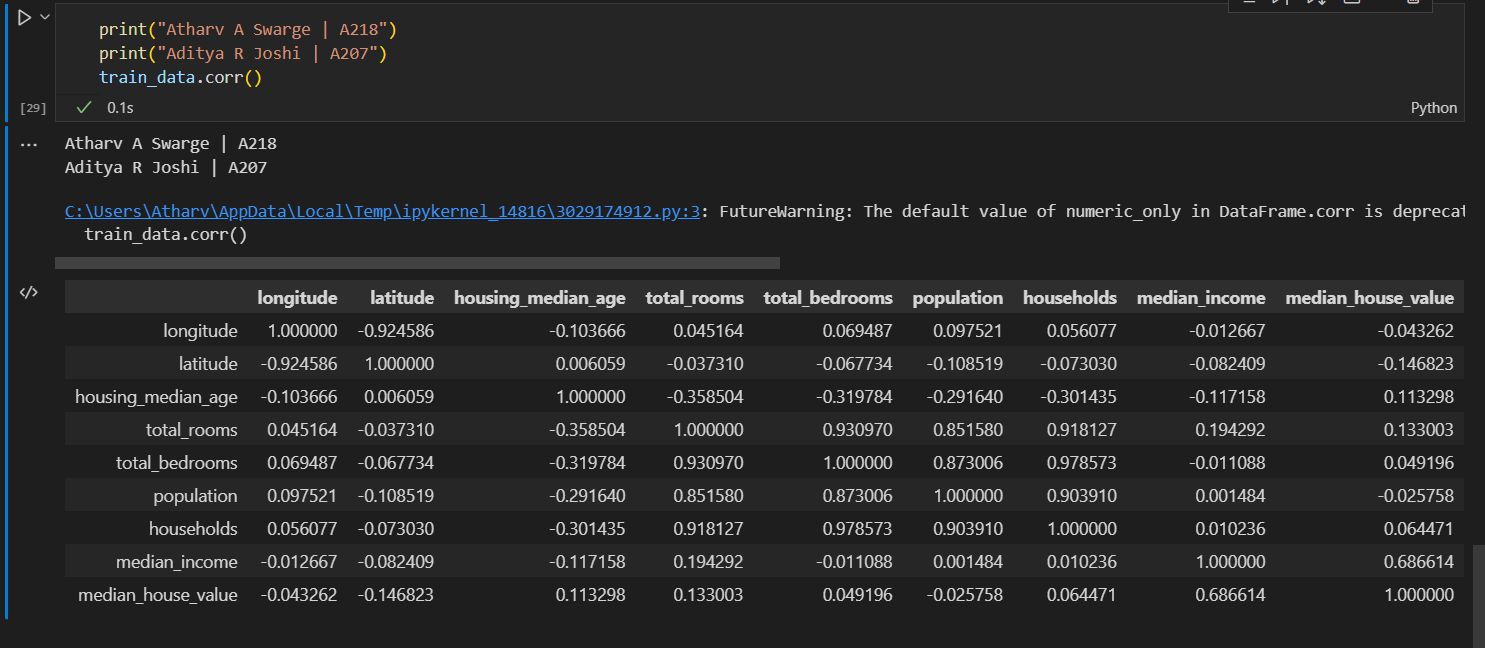


We will get a basic histogram now

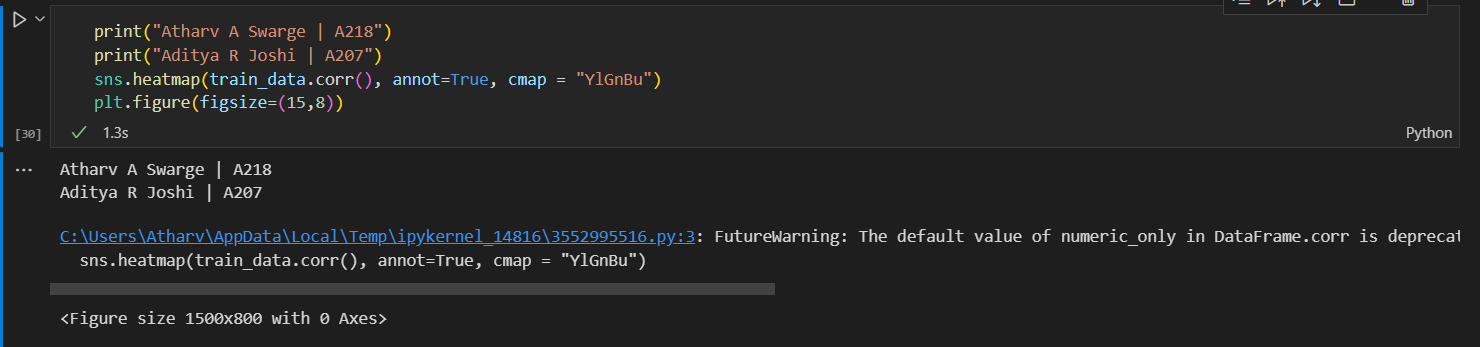


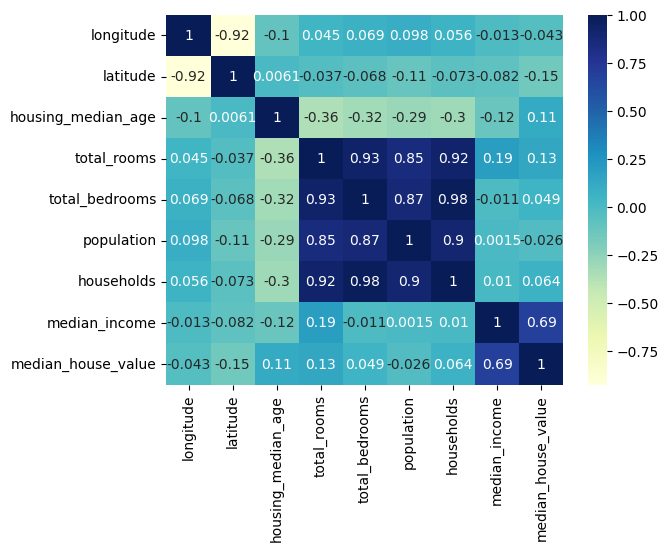


This is how a correlation matrix looks like.

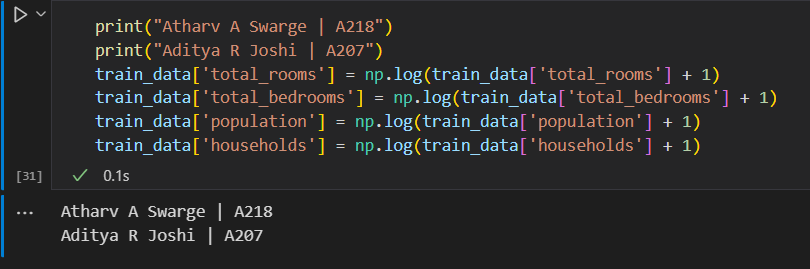


Now we will explore a heatmap to visualize the correlation matrix.

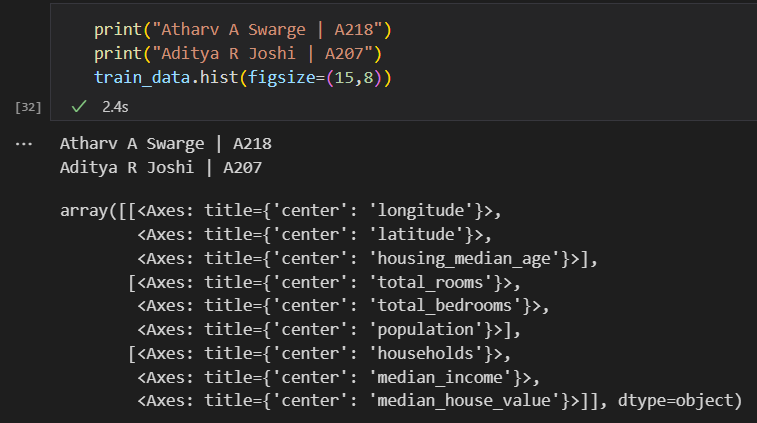


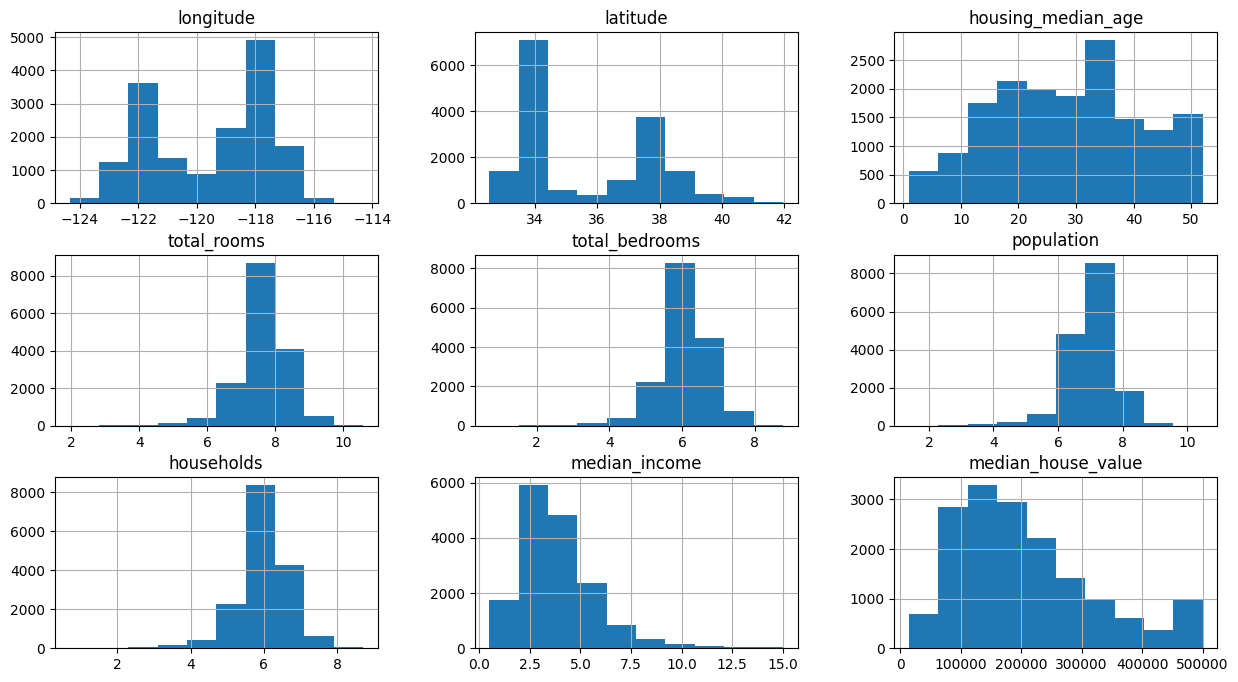


The data is Right Skewed. Its not a Gaussian Curve. We will take log of that features and see what the distribution looks like.



Now via histogram, we can see that the data is now following a gaussian bell curve.

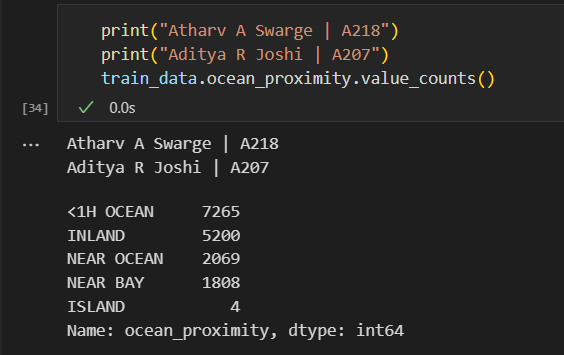




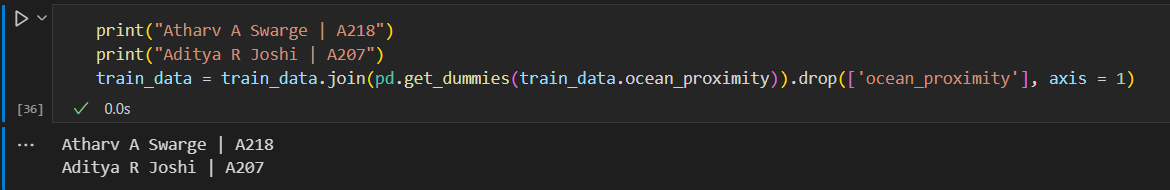
We will use the “Ocean proximity feature too”.

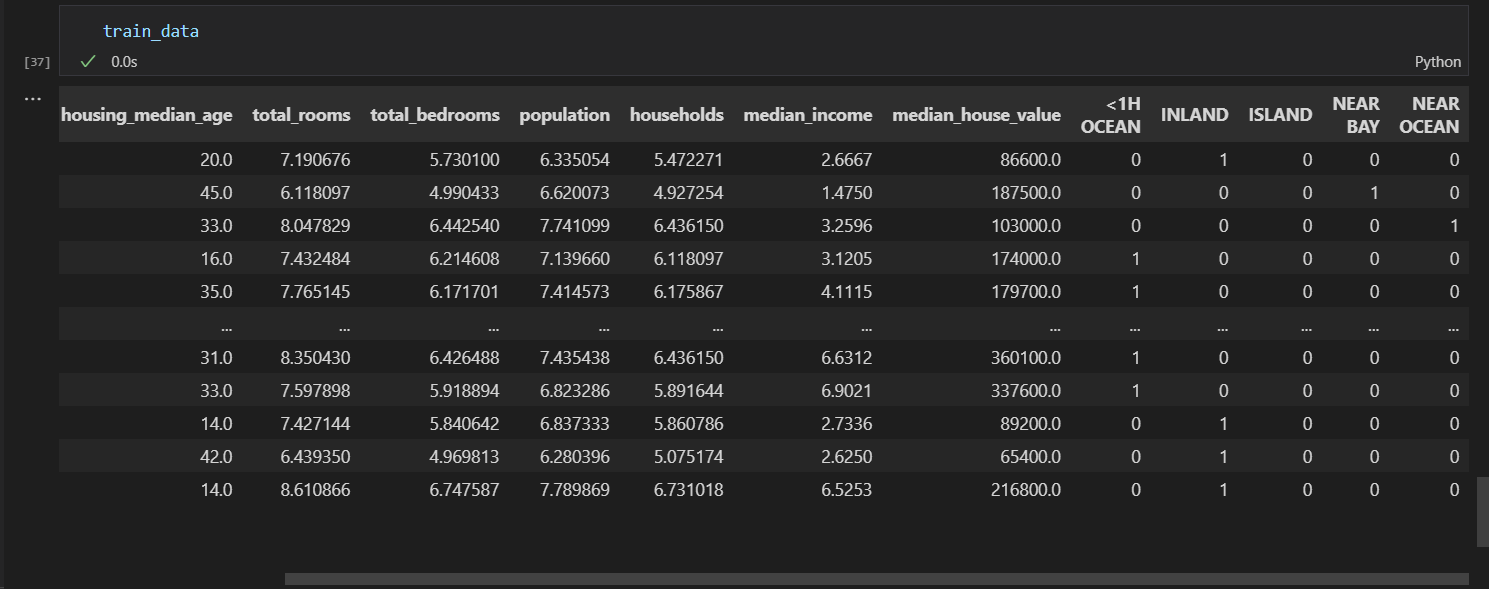
Closer to beach means higher prices, more inland means lower prices.

We have to turn the feature into a numerical value first. We will turn them into binary features.



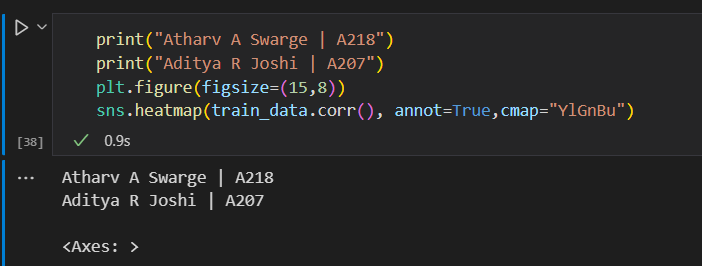
Instead of just assigning numbers to them, we will create a new feature for every single one of those and give them a value of 1 and 0. The first feature will be YES or No instead of 0 and 1.

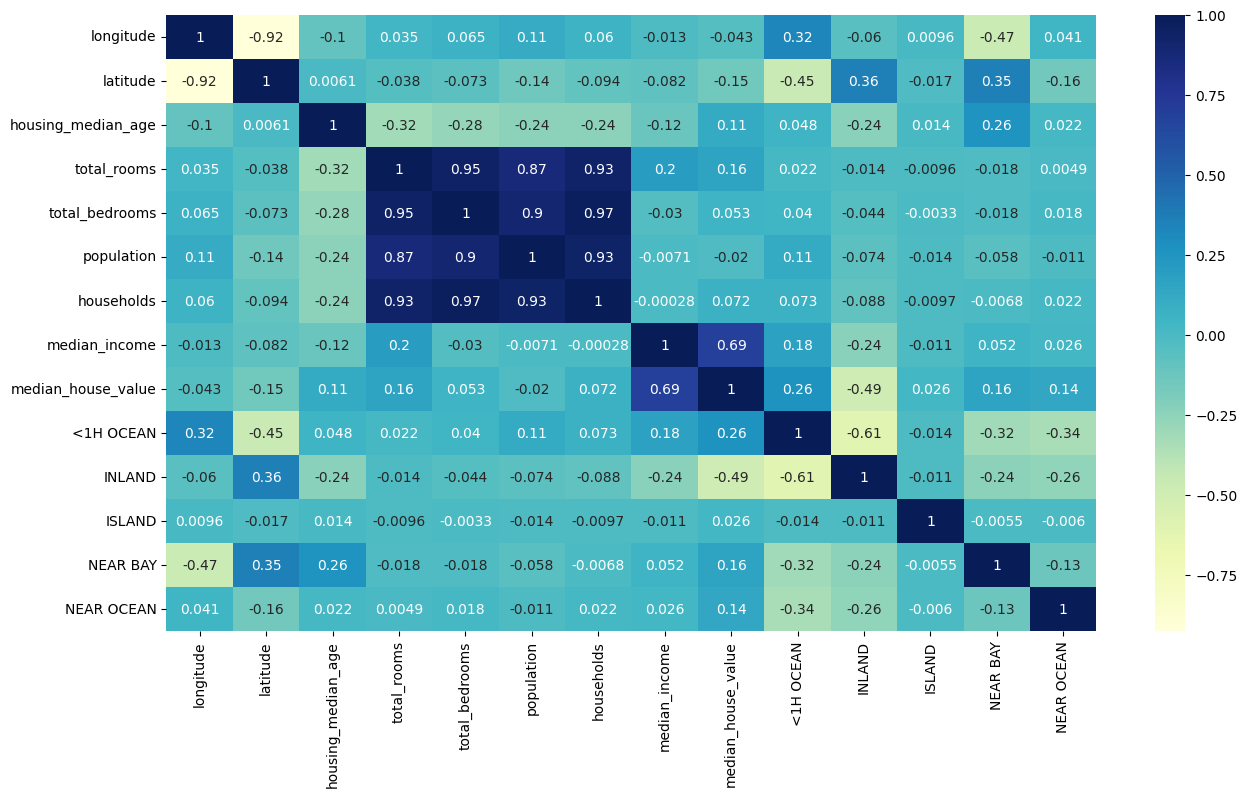




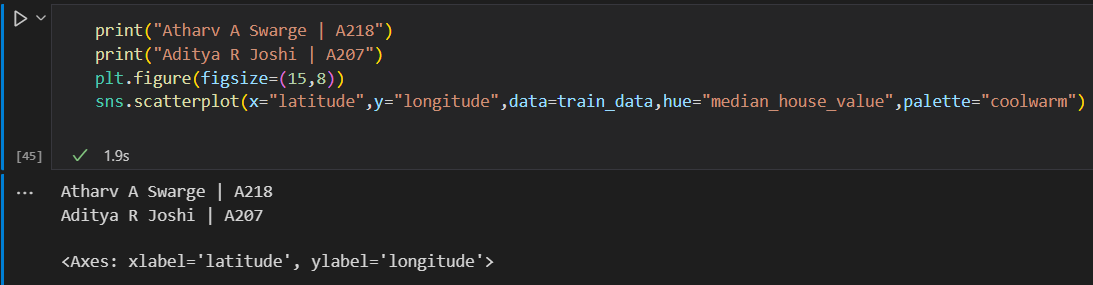
As it is visible, we have created new features and allocated 0 and 1 for each of them.

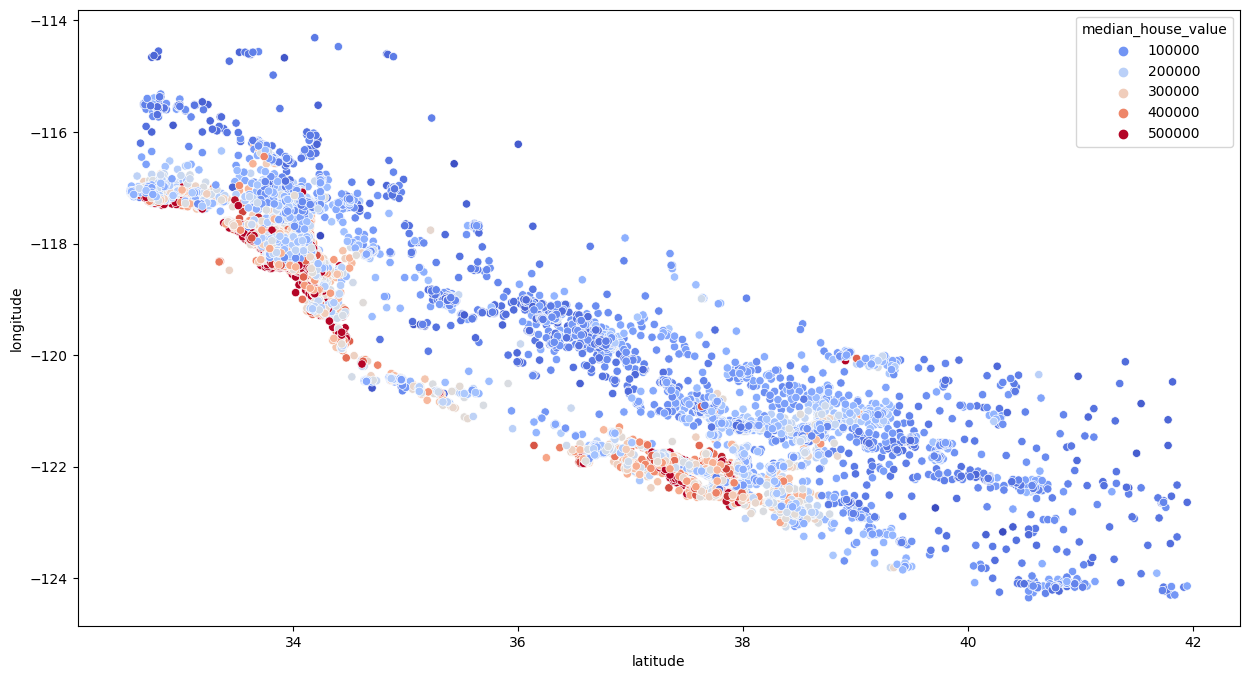
Now, we can also look at the correlation between these factors using Histogram.





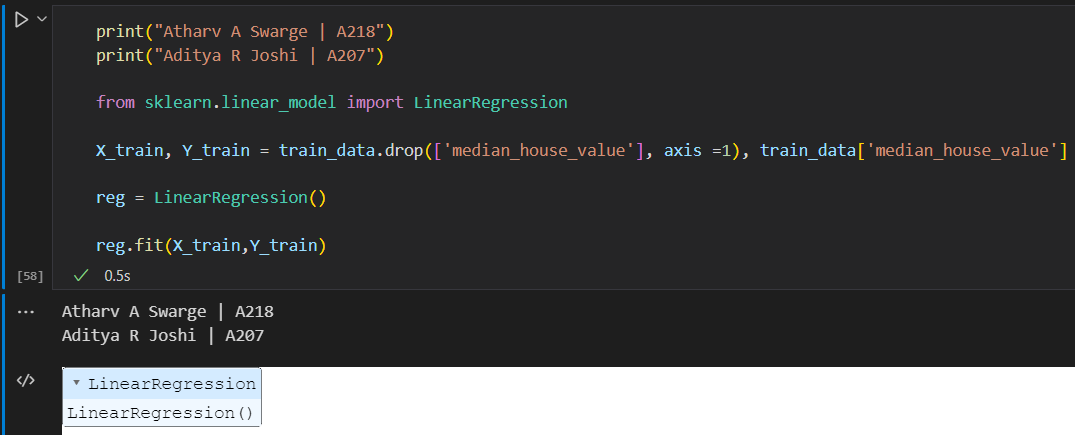
Let’s visualize the coordinates now.

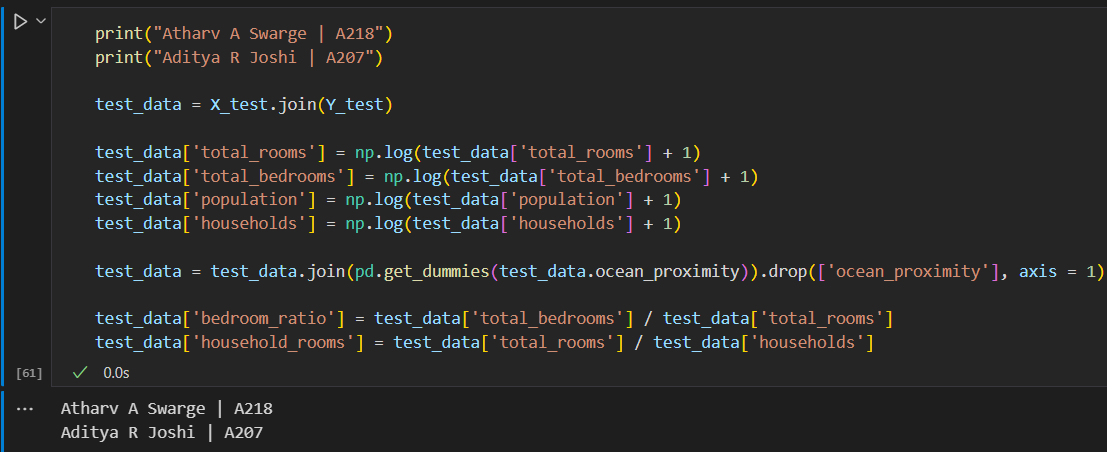


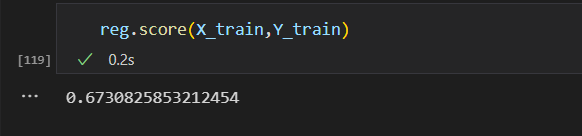


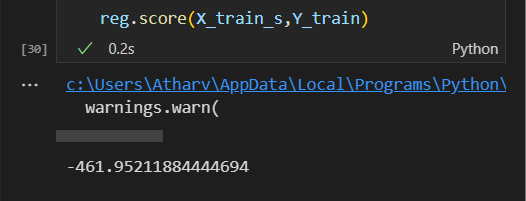
The redder it gets, the more expensive the house is and the bluer it is the less expensive the house.

Now we will train a linear regression model.

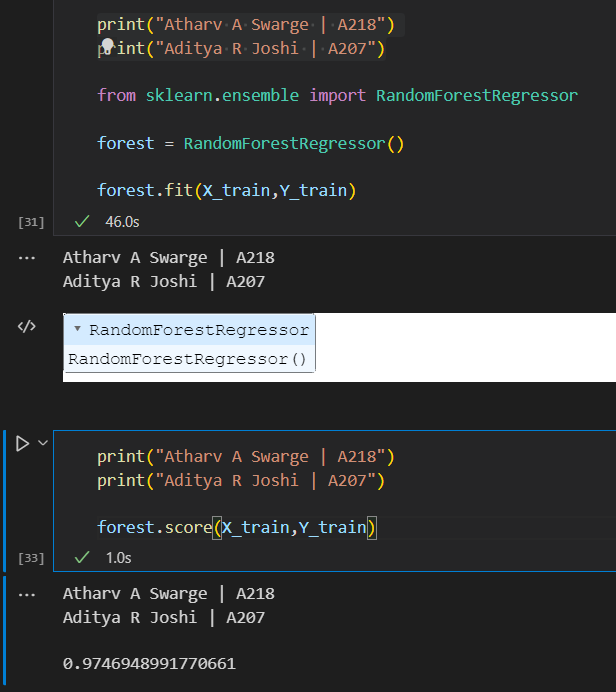




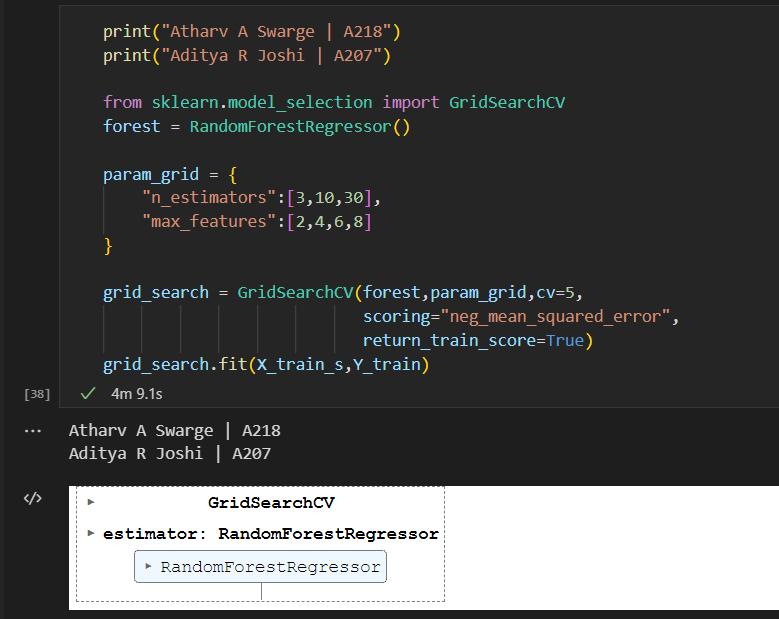




Now, we will use Random Forest Algorithm and we will also perform hyper-parameter tuning.



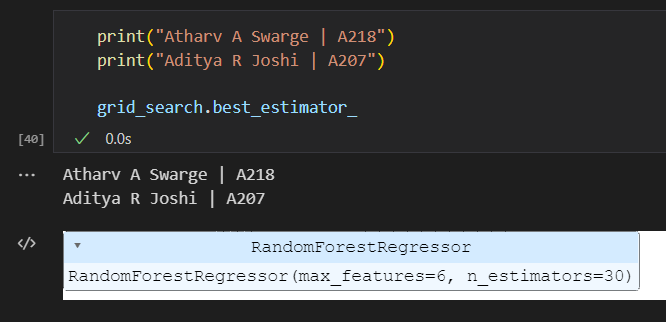
We will now provide a parameter grid for the better accuracy of the result with cross validation.



We will train it on the scaled data. It will take some time but it will give us the optimal model, resulting in the best estimator and the best regressor which can then be used against the test data later on. We will define the forest as a fresh RandomForestRegressor which may take additional time but it should be done to ensure that we don’t have the fitted version from above and it will create a whole entire new and fresh Random Forest for us.

Hyperparameters are parameters whose values control the learning process and determine the values of model parameters that a learning algorithm ends up learning. The prefix ‘hyper\_’ suggests that they are ‘top-level’ parameters that control the learning process and the model parameters that result from it.

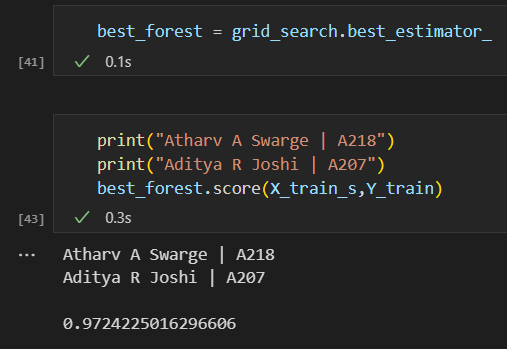
Now to get optimal hyper-parameters, we will do the following:



By this way, we can get the optimal hyper-parameters.

Now we will check if this best estimator actually produces better results or not.

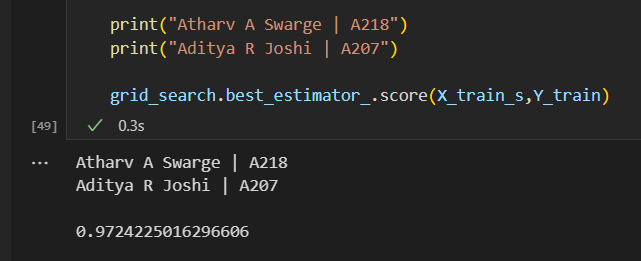
Sometimes, it may also even happen that the results degrade from good to worse.



And, in this case, we get actually a worse performance. As I mentioned before, it can happen. Usually when you tune the hyper pararmeter, you expect a better result.

But as the change is very minimal, we will ignore it.

But even if we decide to get the accuracy score higher, due to the number of estimators and the depth of the hyper parameters, we cannot:



In this project, we learnt about Exploring a dataset, Preprocessing the data, Scaling the features and then building models and then evaluating them.