**Project Income Qualification**

**Simplilearn Machine Learning Certification**

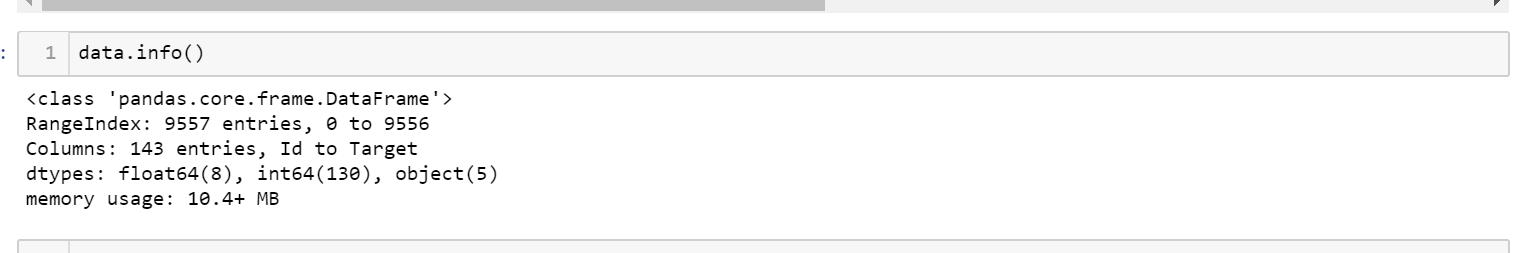
[**gundeep5830@gmail.com**](mailto:gundeep5830@gmail.com)

* **Identify the output Variable:**

In the given data set of 143 columns the last column ‘Target’ is the output and has poverty level classified in 4 classes(1,2,3,4).

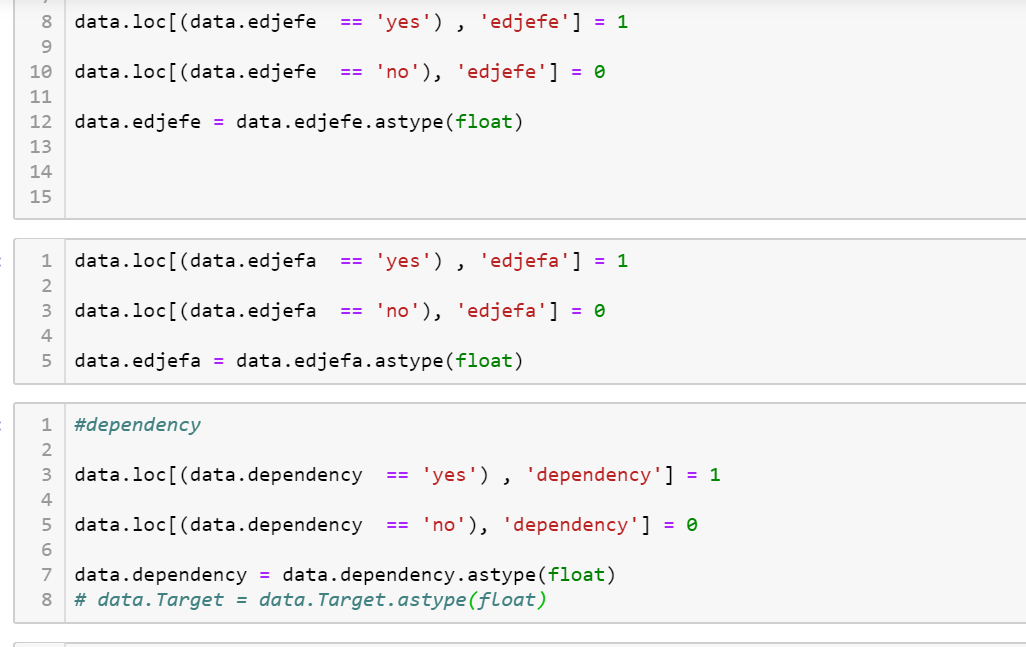
* **Understand the type of data**

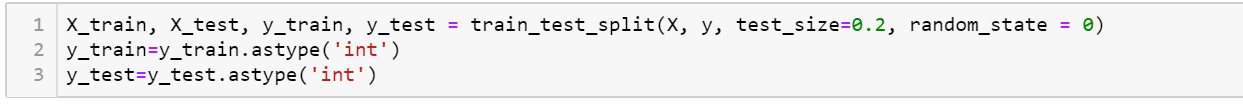
For the understanding of the data we run the following code on our dataset and the observations are also stated below:



From the above output we determine that out of 143 columns, we have 138 as float or int and are thus understandable by the machine. But 5 out of the all are objects(string format) and hence need to be modified for the machine to understand.

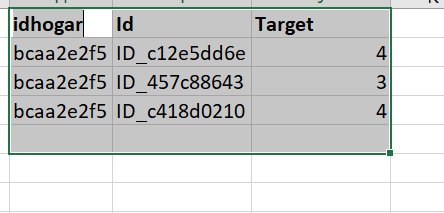
We try to convert the yes/no in 1/0 binary form and for the label(target) of the dataset, the type is yet unknown and thus we specify its type as int while training. This can be shown in the upcoming snippets and screenshots.





* **Check whether all members of the house have the same poverty level.**

Well, if we go through the whole data we come to know that ***the poverty level of all the house members is not same*** and hence to maintain the consistency of the data we make it same for all the house members.



In this above picture , it can be seen that idhogar(which identifies a unique houslehold) is same in all three cases while the target(the poverty level) is different for all three individuals in the house.

* **Check if there is a house without a family head.**

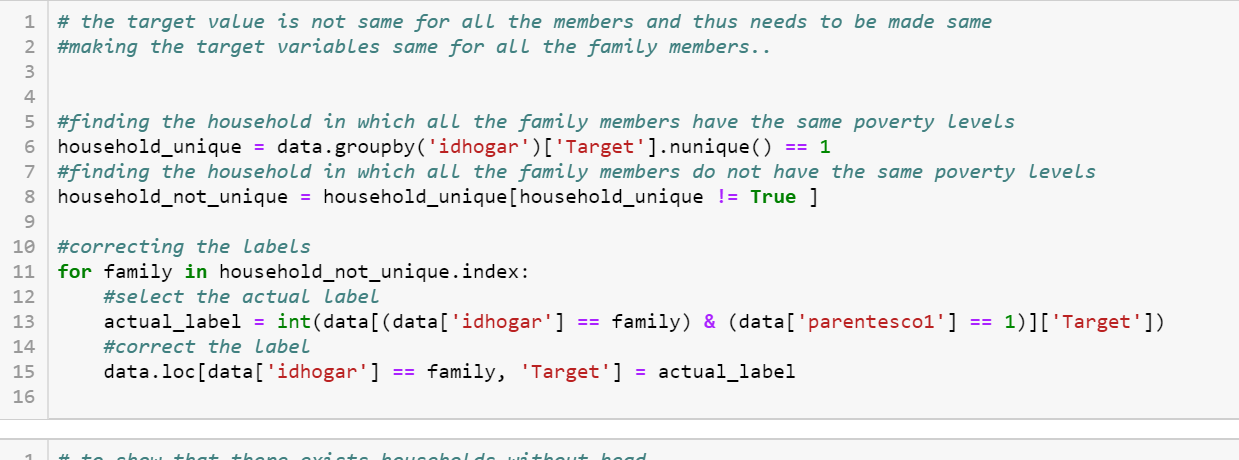
For this we check the variable (‘parentesco1’) of the data set which is 1 if there exists a family head and is 0 otherwise.



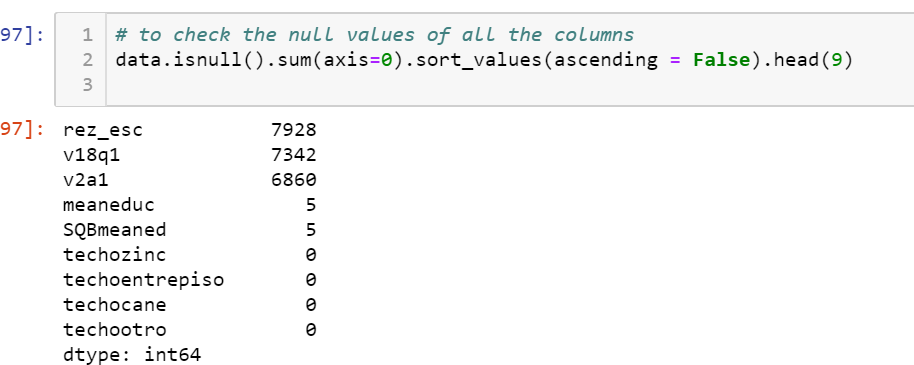
In the above pic we can see that the values of parentesco1 is a mixture of 0/1 so **yeah there exists families with no family head.**

* **Set poverty level of the members and the head of the house within a family.**

So as discussed above that the members and the family heads possess different levels of poverty and hence for the sake of consistency we tend to keep these levels same. This implementation can be seen below:

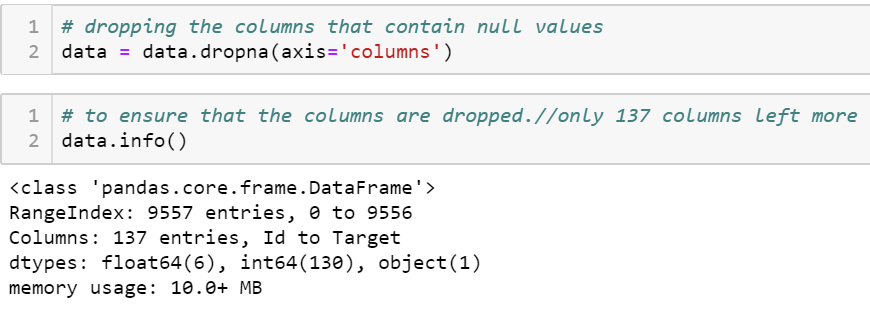


* **Count how many null values are existing in columns.**



The above pic shows the count of null values existing in each column.

* **Remove null value rows of the target variable.**

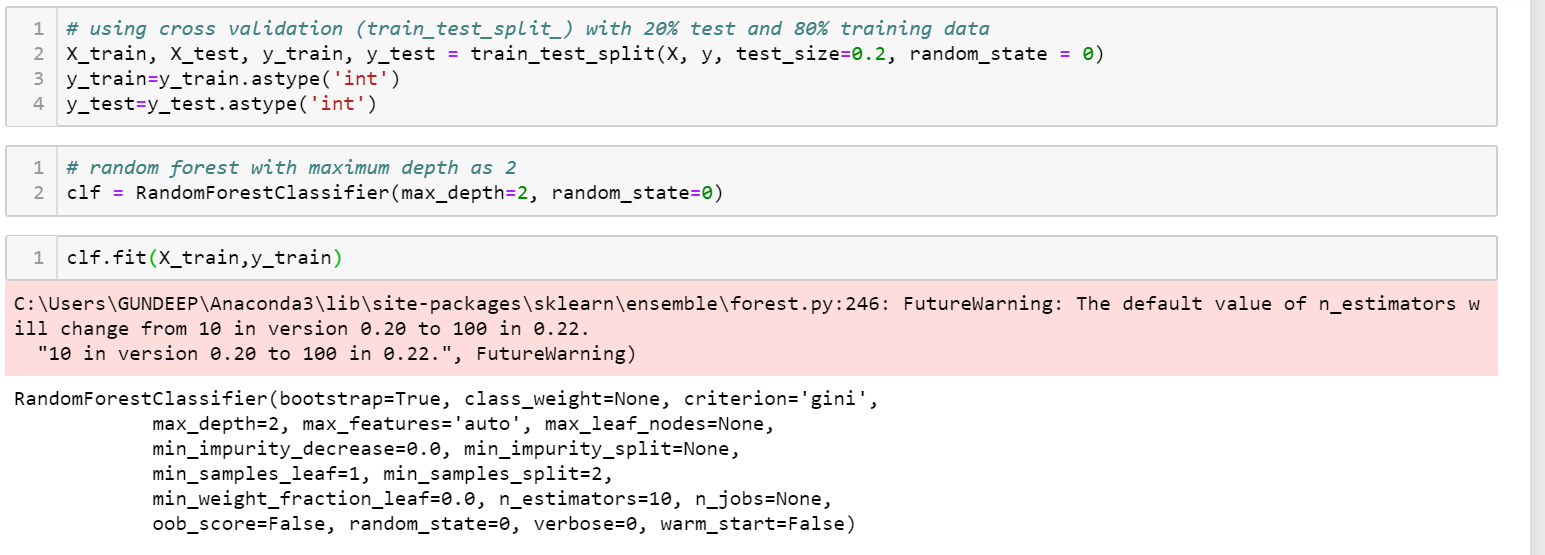


Since we have a lot of entries with null values and deleting a row in target variable will result in lots of data for training the model and will highly affect the accuracy of the model hence rather we delete the 5 features entirely as this reserves a lot of data for training.

This leaves us with just 137 columns including the target(label) as well.

Also before training the model we drop ‘idhogar’ and ‘Id’ columns as they are independent of the predictions and are of string data types so they can be left out before feeding the data for training.

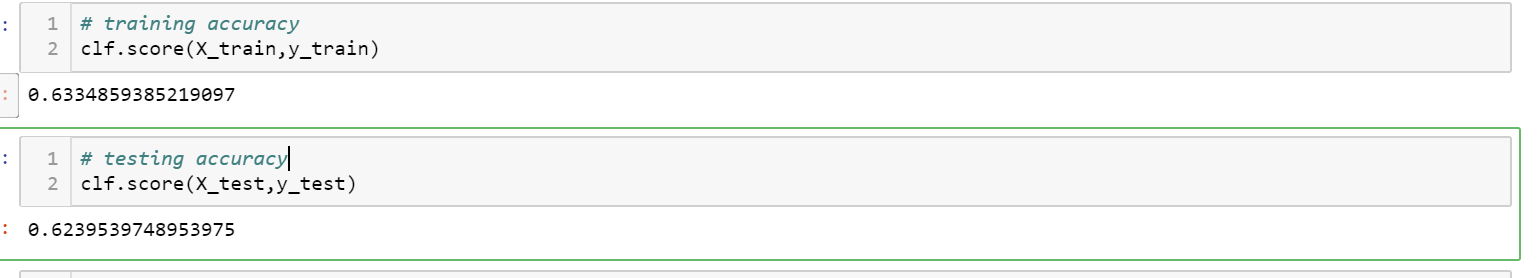
* **Check the accuracy using random forest with cross validation.**



The cross-validation method is used for the train and test split. 80% of the data is kept for training while 20% of the data is kept for testing. Below is the screen shot for the accuracy

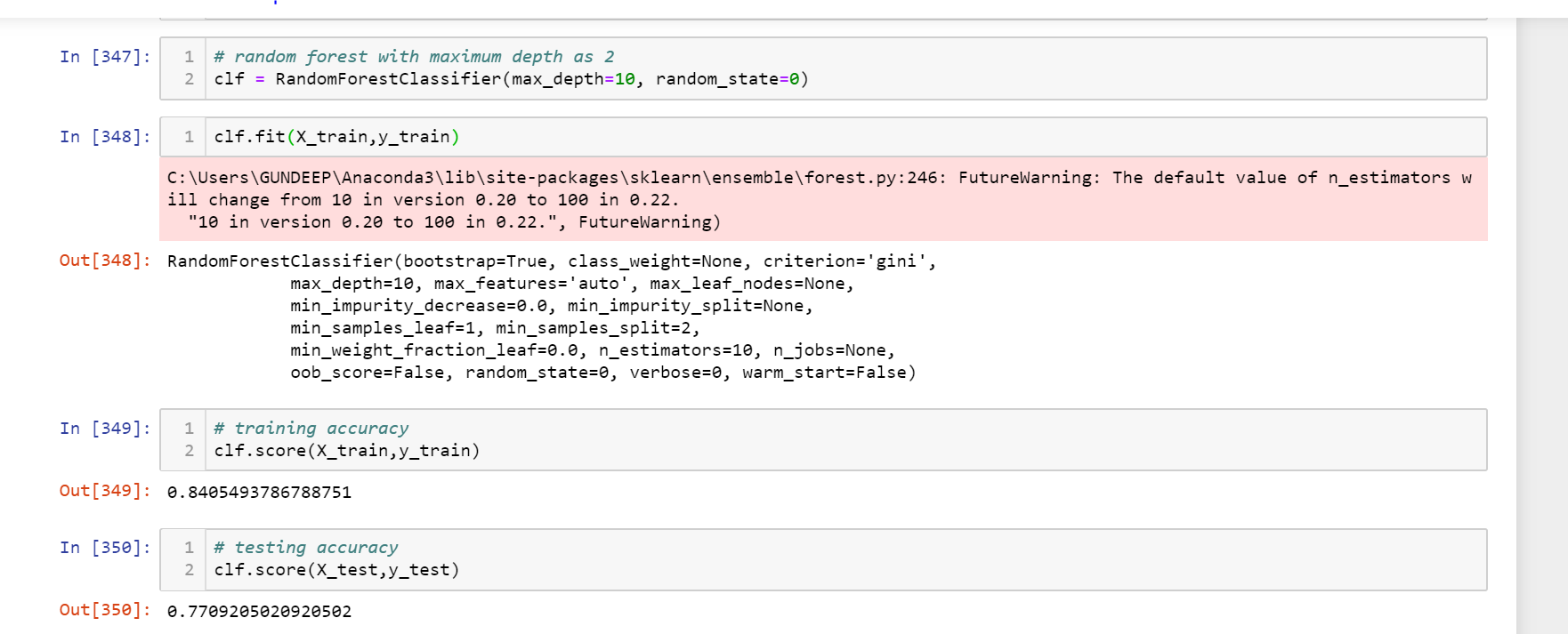
I have used the random forest classifier as our training model with all the parameters kept default except the max\_depth(used to decide the length of the tree) which is being kept 2 for simplicity of the model.

The accuracy of the training and testing data can be seen in the below screenshot:

**`**

It is worth mentioning that if we increase the max\_depth parameter of randomforestClassifier to 10 then the accuracy comes to be as high as 84.05% for training and 77.09% for testing split.

This can be seen in the below picture:



To my knowledge I guess we should not increase the max\_depth parameter beyond this so as to avoid overfitting.