End to end machine learning checklist:

1. Frame the problem
2. Get the data
3. Explore the data to gain insights
4. Prepare the data to better expose the underlying data patterns to machine learning algorithms
5. Explore many different models and shortlist the main ones
6. Fine tune you model and combine them into a great solution
7. Present you solution
8. Launch monitor and maintain your system

Frame the problem:

The goal should be to have a business objective or a solution to a problem. Just building a model should never be the goal. Things to consider

1. How you the output/ result will be used?
   1. It will help us to generate result in the expected way. Otherwise we might need to process the generated output/ result again.
2. What are the current solutions?
   1. It will help us to measure the performance of the new model. It will tell us if the new model is actually worth making.
3. How should we frame the problem?
   1. Like what kind of ML to use – supervised, unsupervised, RL, offline ,online. It all depends on the need or objective of the project.
4. How should we measure the performance of our model?
   1. There a variety of techniques such a loss function. Decide which loss function to use and why
5. Is the performance measure aligned with the business objective?
6. What would be the minimum performance needed to reach the business objective?

Select a performance measure:

The given problem is predicting California housing prices. This is also a multiple regression problem as we will be using multiple features to train the model. Additionally, this is also a univariate regression since we will be predicting a single value. A typical performance measure for regression is RMSE-root mean square error.

Know your data:

Before starting to make decisions to choose a model we must evaluate our data. There are lots of things to consider . Need to learn more about that later. For now consider missing values, normalize values, find out the features to be used for the model etc.

Create a test set:

Data set should be divided into three categories, test, train, validate. There are few ways to divide the data

Randomly we can choose 80% for train and 20% for test. But the problem is if we run the program again, it will generate a different set which might affect the model’s performance. So, to solve that we can set the random number generator seed so that it always generates the same shuffle indices.

Another way is, just put the divide and save the data separately. But it will create problem if we try to fetch the updated data set.

Another way, its good but difficult, is to hash the id( or any other unique colum) to decide which example will go to test and which will go for train.

NOTE: if the dataset is not diverse or large enough we can run into sampling bias problem. For example, as we are doing this randomly, it can occur that, almost all the positive values of a feature went to train set and all the negative value went to test set. So we need to be aware of this and keep this in mind. The solution to this problem is called stratified sample (we will learn more later)