We know in classification, we predict the classes. In case of a binary classification, we have only two classes to predict from 0 or 1. The prediction is done by a classific model on the basis of class probablity set in model.

Lets say that the class propablity (y_p) is set as 0.5. Then anything which is greater than 0.5 will be taken as 1 and less than 0.5 will be taken as 0.
Now we can also change the cuttoff probablity for the classification models. we can set it to more or less depending upon the problem statement.

We can make a list of all the cuttoffs, then we can predict the different values using cuttoffs. Then we can calculate the precision, recall or any other evaluation matrix, and check for which cuttoff we have the least of best accuracy.

Now, there are different meathods using which we can find the best possible optimal threshold.
First, we can find the optimal threshold using the roc_curve.

```
from sklearn.metrics import roc_curve

fpr, tpr, thresholds = roc_curve(y_train, pred_prob)
```

We give the actual values and the probablity of prediction being 1 inside the parameters of roc_curve.

Now, in order to get the best performing model, we want the true positive rate to be maximum & false positive rate to be minimum. Or the difference should be minimum. This can be done using the following function.

```
np. argmax (tpr - fpr)
# return index where tpr-fpr is maximum
```

Now, we want the threshold for this index and it would be the optimal threshold.

```
opt_thres = thresholds[np.argmax (tpr - fpr)]
```