

**Data Analytics with Python**  
**Prof. Ramesh Anbanandam**  
**Department of Management Studies**  
**Indian Institute of Technology – Roorkee**

**Lecture – 43**  
**Performance of Logistic Model – III**

In this lecture, we are going to test the performance of logistic regression model. We use Python and I will show you a demo how to check the performance of a logistic regression model.

**(Refer Slide Time: 00:41)**

### Agenda

Python demo for accuracy prediction in logistic regression model using Receiver operating characteristics curve

The agenda for this lecture is Python demo for accuracy prediction in logistic regression model using ROC curve.

**(Refer Slide Time: 01:11)**

### Sensitivity and Specificity

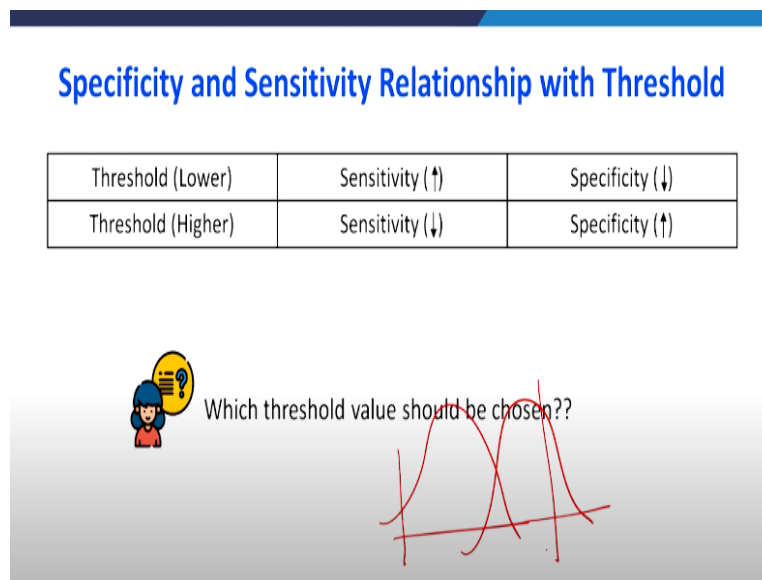
- For checking, what type of error we are making; we use two parameters-

1. Sensitivity =  $tp/(tp+fn)$   $\longrightarrow$  True Positive Rate (tpr)

2. Specificity =  $tn/(tn+fp)$   $\longrightarrow$  True Negative Rate (tnr)

There are two terms, one is Sensitivity and another one is Specificity. For checking what type of error we are making, we use 2 parameter. One is sensitivity. The another name for sensitivity is True Positive Rate. This also I have shown you in your previous lecture tp divided true positive by false negative. Specificity is a true negative rate, that is true negative divided by true negative plus false positive.

**(Refer Slide Time: 01:20)**



In this lecture, I am going to stay the connection between sensitivity and specificity for different threshold value. The first case is, when the threshold value is low, when the threshold value is suppose this way, suppose when you put threshold value here, we will increase the sensitivity, but decrease our specificity. When the threshold value is higher, suppose if you keep the threshold value here, what will happen specificity will increase, sensitivity will decrease. So, which threshold value should be chosen? That is the problem. That I will show you with the help of Python programming.

**(Refer Slide Time: 02:04)**

## Measuring Accuracy, Specificity and Sensitivity

```
In [20]: 1 Accuracy = (tp + tn) / (tp + tn + fp + fn)
          2 print("Accuracy {:.2f}".format(Accuracy))

Accuracy 0.76

In [21]: 1 Specificity = tn/(tn+fp)
          2 print("Specificity {:.2f}".format(Specificity))

Specificity 0.94

In [22]: 1 Sensitivity = tp/(tp+fn)
          2 print("Sensitivity {:.2f}".format(Sensitivity))

Sensitivity 0.44
```

$t=0.5$

First, we will check what accuracy is. Accuracy is true positive plus true negative divided by true positive plus true negative plus false positive plus false negative. So, the accuracy for our problem is 0.76. Then specificity, true negative divided by true negative plus false positive. For our problem, it was 0.94. Then sensitivity is true positive divided by true positive plus false negative. For our problem, sensitivity is 0.44.

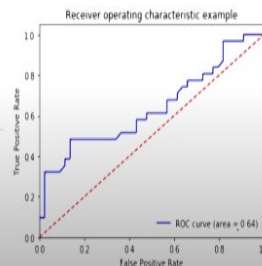
We got this specificity and sensitivity by taking threshold is 0.5. That is our default value, but the question will come, if it goes above 0.5 or below 0.5, what will happen and what should be the correct value.

**(Refer Slide Time: 02:59)**

## ROC Curve for Training dataset

```
In [23]: 1 from sklearn.metrics import roc_auc_score
          2 from sklearn.metrics import roc_curve, auc
          3 log_RUC_AUC1 = roc_auc_score(y_train, y_predict_train)
          4 fpr1, tpr1, thresholds1 = roc_curve(y_train, y_prob_train)
          5 roc_auc1 = auc(fpr1, tpr1)

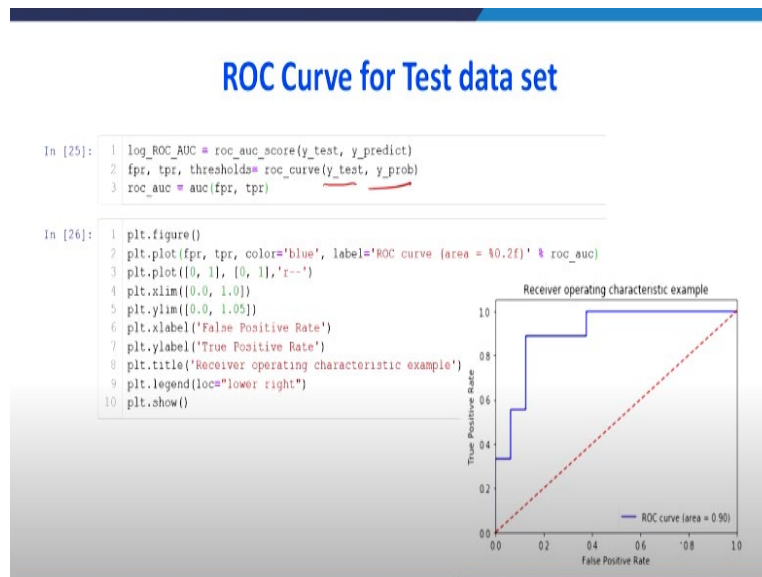
In [24]: 1 plt.figure()
          2 plt.plot(fpr1, tpr1, color='blue', label='ROC curve (area = %.2f)' % roc_auc1)
          3 plt.plot([0, 1], [0, 1], 'r--')
          4 plt.xlim([0.0, 1.0])
          5 plt.ylim([0.0, 1.05])
          6 plt.xlabel('False Positive Rate')
          7 plt.ylabel('True Positive Rate')
          8 plt.title('Receiver operating characteristic example')
          9 plt.legend(loc='lower right')
          10 plt.show()
```



Now we will draw the ROC curve for training dataset. So, from `sklearn.metrics` import `roc_auc_score`, from `sklearn.metrics` import `roc_curve`, `auc`, log of RUC\_AUC1 is equal to

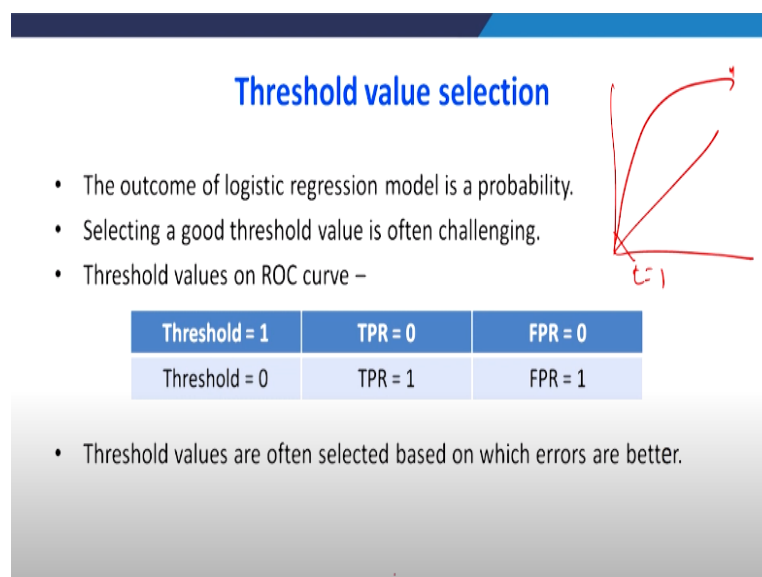
roc\_auc\_score, y\_train, y\_predict\_train. Then fpr, tpr. Find out false positive rate and true positive rate and threshold also we are going to test it. So, when you plot it, fpr and tpr, threshold 1 equal to roc\_curve(y\_train, y\_prob\_train), so roc\_auc1 is equal to, so we are going to draw auc curve under fpr1 is false positive and true positive rate.

(Refer Slide Time: 04:29)



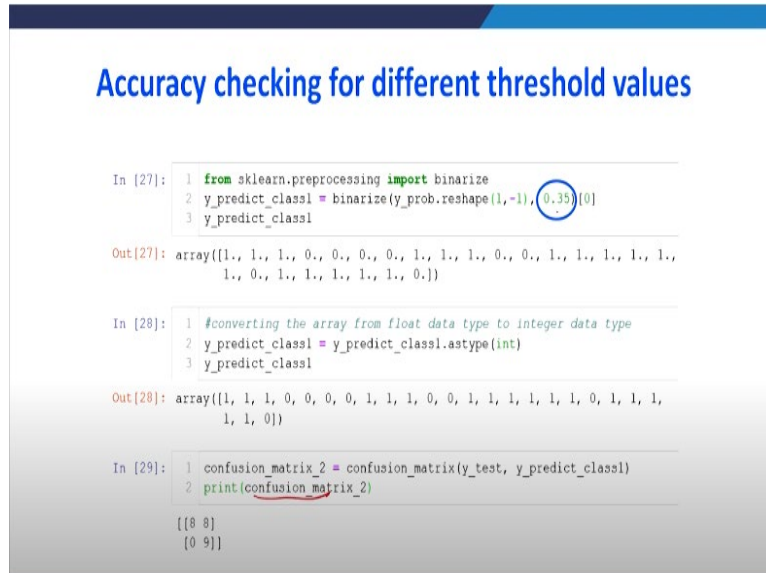
When you plot this, we are getting for different combination of fpr and tpr, we are getting for default ROC curve. So, the area under AUC for this model is 0.64. Now, let us draw the ROC curve for the Test data set. See only these things are changed, when we draw ROC curve for Test data set, what happened when compared to this, here the AUC, area under curve is increased by 0.9. So what we infer from the previous one, this one is, this model is performing well for the y test dataset because the AUC is 0.9.

(Refer Slide Time: 05:03)



How to select threshold value? The outcome of logistic regression model is a probability. Selecting a good threshold value is often challenging. The threshold value on ROC curve, you can take it is 1, if the threshold is 1, what will happen the true positive rate is 0 and false positive rate also 0. This situation. So, this is the place where T equal to 1. When threshold is 0, so true positive rate is 1, this point and false positive rate also 1. So, threshold values are often selected based on which error is better.

**(Refer Slide Time: 05:51)**



```
In [27]: 1 from sklearn.preprocessing import binarize
2 y_predict_class1 = binarize(y_prob.reshape(1,-1), 0.35)[0]
3 y_predict_class1

Out[27]: array([1., 1., 1., 0., 0., 0., 0., 1., 1., 1., 0., 0., 1., 1., 1., 1.,
1., 0., 1., 1., 1., 1., 0.])

In [28]: 1 #converting the array from float data type to integer data type
2 y_predict_class1 = y_predict_class1.astype(int)
3 y_predict_class1

Out[28]: array([1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1,
1, 1, 0])

In [29]: 1 confusion_matrix_2 = confusion_matrix(y_test, y_predict_class1)
2 print(confusion_matrix_2)

[[8 8]
 [0 9]]
```

At present, randomly we choose some threshold value 0.35. Let us compare by changing different threshold value and verify which threshold value is right choice for us. For that, `y_predict_class1` equal to `binarize y_prob.reshape 1, -1` by taking 0.35. So, if you take 0.35, this was our predicted values, but if we want to get to the integer value to use `y_predict_class1.astype integer`, you are getting the integer value and from that, you are getting the confusion matrix.

So, what it says that by keeping threshold values 0.35, so true positive is 9 and true negative is 8. So, we can change different value, for example, for that value, we can check our recall and precision. Now, by taking threshold value as 0.5, let us verify what has happened that confusion matrix value is increased, so the true negative is 15, true positive is 4.

**(Refer Slide Time: 07:14)**

# Accuracy checking for different threshold values

```
In [35]: 1 from sklearn.preprocessing import binarize  
        2 y_predict_class3 = binarize(y_prob.reshape(1,-1), 0.70)0]  
        3 y_predict_class3
```

```
Out[35]: array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
                0., 0., 0., 0., 0., 0.]])
```

```
In [36]: 1 confusion_matrix_4 = confusion_matrix(y_test, y_predict_class3)  
        2 print(confusion_matrix_4)
```

```
[16  0]  
[ 9  0]]
```

```
In [37]: 1 from sklearn.metrics import classification_report  
        2 print(classification_report(y_test, y_predict_class3))
```

	precision	recall	f1-score	support
0	0.64	1.00	0.78	16
1	0.00	0.00	0.00	9
micro avg	0.64	0.64	0.64	25
macro avg	0.32	0.50	0.39	25
weighted avg	0.41	0.64	0.50	25

**(Refer Slide Time: 07:45)**

## Calculating Optimal Threshold Value

```
In [38]: 1 from sklearn.metrics import roc_curve, auc

In [39]: 1 fpr, tpr, thresholds= roc_curve(y_test, y_prob)
2 roc_auc = auc(fpr, tpr)

In [40]: 1 print("Area under the ROC curve : %f" % roc_auc)

In [41]: 1 import numpy as np
2 i = np.arange(len(tpr)) # index for df
3 roc = pd.DataFrame({'fpr' : pd.Series(fpr, index=i), 'tpr' : pd.Series(tpr, index = i),
4                    '1-fpr' : pd.Series(1-fpr, index = i), 'tf' : pd.Series(tpr - (1-fpr), index = i),
5                    'thresholds' : pd.Series(thresholds, index = i)})
6 roc.iloc[(roc.tf-0).abs().argsort()[:1]]

Out[41]:
```

	tpr	tpr	1-fpr	tf	thresholds
7	0.125	0.888889	0.875	0.013889	0.457033

But the question will come how to choose the right threshold value. We have seen 0.3, 0.5, and 0.7. The appropriate T value that we can get by looking at this table. For example, the output of this program says that the threshold value should be 0.457 is most appropriate for us. Now, Optimal Threshold Value in ROC curve. What has happened here in x-axis, you take 1 minus false positive rate, in y-axis, you take true positive rate. So, at the intersection

point where the true positive rate and true negative rates are intersecting, so that point is considered as the optimal value of threshold value.

(Refer Slide Time: 08:36)

## Classification Report using Optimal Threshold Value

```
In [43]: 1 from sklearn.preprocessing import binarize
2 y_predict_class4 = binarize(y_prob.reshape(1,-1), 0.45)[0]
3 y_predict_class4

Out[43]: array([1., 1., 1., 0., 0., 0., 0., 1., 0., 0., 0., 0., 1., 0., 0., 1.,
1., 0., 1., 0., 1., 1., 0., 0.])

In [44]: 1 confusion_matrix_5 = confusion_matrix(y_test, y_predict_class4)
2 print(confusion_matrix_5)

[[14  2]
 [ 1  8]]

In [45]: 1 from sklearn.metrics import classification_report
2 print(classification_report(y_test, y_predict_class4))
```

	precision	recall	f1-score	support
0	0.93	0.88	0.90	16
1	0.80	0.89	0.84	9
micro avg	0.88	0.88	0.88	25
macro avg	0.87	0.88	0.87	25
weighted avg	0.89	0.88	0.88	25

So, here Classification Report using Optimal Threshold Value, this was our program output. So, here I use binarize y\_prob. reshape, so this is our predicted value. So, we got confusion matrix, here true negative is 14, true positive is 8. For that we got the Classification Report also. Here it says that specificity and sensitivity, both are little higher. So, it shows that this is the optimal threshold value.

(Refer Slide Time: 09:13)

```
0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
1, 1, 0, 0, 0, 0, 1, 1, 0, dtype=int64)

In [17]: M y_prob_train = Lreg.predict_proba(x_train)[:,:1]
y_prob_train.reshape(1,-1)

Out[17]: array([[0.49622117, 0.32880793, 0.44329114, 0.33320924, 0.41456465,
0.32890329, 0.3975043 , 0.60921229, 0.25844531, 0.63672372,
0.29274886, 0.28466974, 0.5159296 , 0.41992276, 0.24342956,
0.528514 , 0.47965107, 0.52805789, 0.33191449, 0.27457435,
0.49179296, 0.63261616, 0.24690181, 0.47089452, 0.27842076,
0.41663875, 0.36155602, 0.49978327, 0.23621636, 0.37860052,
0.48809323, 0.28877877, 0.28563859, 0.37231882, 0.65309742,
0.43807264, 0.33638478, 0.40406607, 0.23431177, 0.37282384,
0.49970327, 0.39768396, 0.32880793, 0.25782472, 0.47393834,
0.42878861, 0.26520939, 0.33120924, 0.54682409, 0.45446086,
0.44326597, 0.4965167 , 0.60065954, 0.38989654, 0.49149447,
0.27414424, 0.27785686, 0.67464141, 0.28195004, 0.48593427,
0.38633222, 0.31373499, 0.42810085, 0.27418723, 0.44371771,
0.41629601, 0.642004 , 0.6571001 , 0.44068025, 0.28195004,
0.40217015, 0.43807264, 0.58977653, 0.57944626, 0.2904233 ]])

In [ ]: M y_prob = Lreg.predict_proba(x_test)[:,:1]
y_prob.reshape(1,-1)
w.mrnh
```

Now, with help of Python I am going to run the code. I am going to explain how to choose the correct threshold value. So, important necessary library, imported pandas, and imported matplotlib.pyplot then this was the dataset. This dataset I have already discussed with you.

There are two independent variable, and one depend variable, that is Coupon. So, we will describe this dataset. It will give a basic statistics of all the columns.

For spending, there are 100 dataset. The mean is 3.3, standard deviation is 1.74, minimum is 1. So, 25th percentile, 50th percentile, and 75th percentile, maximum is 7.07. For Card, we can look at only the count value, because there is no meaning for mean and standard deviation. Similarly, for Coupon, because both the variables are binary variables. Now, we look at what is the value in the Column coupon.

There are two values there one is 0 and 1, 0 mean that customer did not use the coupon, and 1 means that the customer has coupon. Now, let us see how many 0s and how many 1s by using `value_counts` function. So, there are 60 people did not use the Coupon and 40 people used the Coupon. So, the baseline method 0.6. Now, we will go for building the LogisticRegression model. I have imported `linear_model`, `sklearn.model_selection`.

I have imported `train_test_split`, there also I have imported LogisticRegression. Then, we will split the dataset by the ratio of 25 percentage dataset for the training, and the remaining 25 percent dataset is for testing. So, let us see how much training dataset, how much test dataset. So, for x variable, the training dataset is 75, for y variable, the training data is 75. For test dataset, x is 25, the test dataset for y is 25. Now, we will construct a LogisticRegression. So, we use the solver `lbfgs`.

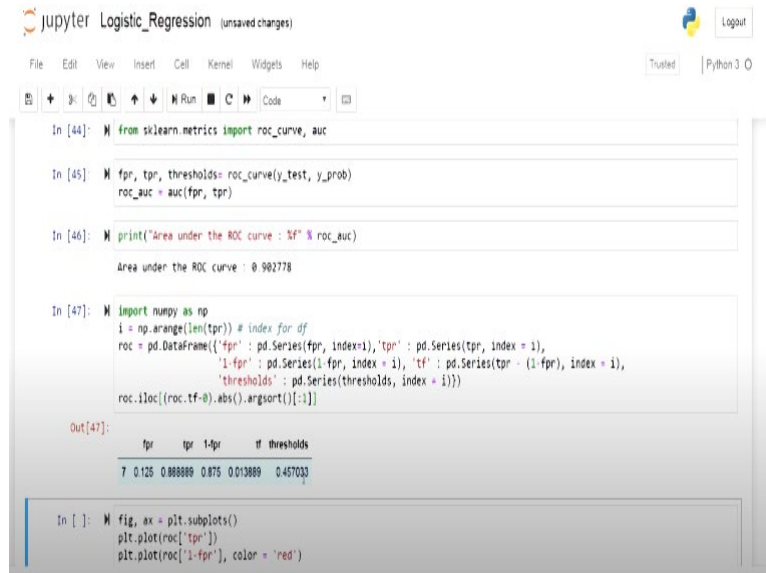
Then, we predict our constructor model with the help of test dataset. In our model, after substituting x values, this was our predicted y value. We can get to know there are different solvers, when you use the LogisticRegression? You can get to know there are different cases. For example, this is help function. You see the multinomial option is supported only by the `lbfgs`. There are some more method, `sag` method and `newton-cg` method that we are not using.

So that is why we have used `lbfgs` solver for getting this LogisticRegression output. Now, we will predict our model with test data set. This was our predicted y value. Here the input is test data set. Now, we will take the training dataset, then we will predict the model. Because in the training dataset, there are 75 dataset. Here only 25 was there. So, this was our predicted output for the training dataset. Now, we will get the probability value for this.



So, this is the probability for our training dataset. So, there will be 75 dataset is there. There is a different probability. Our question comes what should be the threshold value or the cutoff value. So that we can classify this is 1 or 0. Now for the test dataset also, there should be 25 dataset, we can get the probability. Now, we will run the regression model.

**(Refer Slide Time: 13:16)**



```

In [44]: from sklearn.metrics import roc_curve, auc

In [45]: fpr, tpr, thresholds = roc_curve(y_test, y_prob)
roc_auc = auc(fpr, tpr)

In [46]: print("Area under the ROC curve : %f" % roc_auc)

Area under the ROC curve : 0.982778

In [47]: import numpy as np
i = np.arange(len(tpr)) # index for df
roc = pd.DataFrame({'fpr' : pd.Series(fpr, index=i), 'tpr' : pd.Series(tpr, index = i),
                    '1-fpr' : pd.Series(1-fpr, index = i), 'tf' : pd.Series(tpr - (1-fpr), index = i),
                    'thresholds' : pd.Series(thresholds, index = i)})
roc.iloc[(roc.tf-0).abs().argsort()[:1]]

Out[47]:
   fpr    tpr  1-fpr  tf thresholds
7  0.125  0.888889  0.875  0.013889  0.457033

In [ ]: fig, ax = plt.subplots()
plt.plot(roc['tpr'])
plt.plot(roc['1-fpr'], color = 'red')

```

This is our predicted model, but here we have to show that what should be the threshold value. First, we will check the accuracy of the model. For knowing the accuracy of the model, we have to import `accuracy_score`. So, the accuracy is 0.76. Next, we can go for constructing a confusion matrix. For that, you have to import a new library called `confusion_matrix`. When you run this, you are getting confusion matrix.

Here the confusion matrix is see default value is 0.5 is taken, this 15 says true negative and 4 says true positive. The 1 is false positive, the 5 is false negative. So, in our dataset, say that the true negative is 15, false positive is 1, false negative is 5, and true positive is 4. Now, we will get to know what is the Classification Report. In classification report, important things you have to remember. One is the recall; another one is the support.

The Recall gives us an idea about what is actually yes and how often does it predict yes. The Recall value 1 is called as sensitivity, the recall value 0 is called specificity. Precision tells us about when it predicts yes, how often is it correct. So, Precision is true positive divided by true positive by false positive for 1. The accuracy is, the diagonal value of the confusion matrix is the tp and true negative, if you add all the cells value, that is our accuracy.

Now the recall is  $tp / (tp + fn)$ , for value of 1. The f measure is giving the balance between precision and our recall. Now, we will go for finding the accuracy. Accuracy is 0.76. Then, we will go for specificity. The specificity is 0.94, here the default value is 0.5. So, we will go for sensitivity that is true positive rate. For true positive rate, it is 0.44. Now, we will for ROC curve. Now, we will plot that ROC curve.

For default value, the ROC curve is a blue one, which says the area under curve is 0.64. Now, we will see different false positive rate and true positive rate, so this was that one. Now, let us see what are the values of false positive rate. First, we will say fpr. These are the different false positive rate. Now, I will display the output of true positive rate, tpr. So, these values are going to be our x and y axis of our ROC curve.

For, different combinations, we may get different ROC values. So, we will plot ROC curve. This is for our test dataset because you see that here the value which I have taken, this is y set dataset and this is ROC curve. When you look at these two curves, here when you look at this dataset, this is for our training dataset. For training dataset, the ROC curve is like this. The area is 0.64. For the test dataset, the value for ROC curve is 0.9.

So, our model is very well for the test dataset. Now, let us randomly give different T value, different threshold value. Let us see how the ROC curve appears. Suppose we have taken the ROC curve value 0.35, this is threshold value, let us see ROC curve for this value. So, I have predicted values, then value I need in an integer form, I have taken integer form, then I am going to draw the confusion matrix.

So, here the confusion matrix true negative is 8, true positive is 9 because here the value is 0.35, so if it is 0.35, see there are higher true positive rate, that is 9. For this value, let us draw the true positive is 9 because our threshold value is low, everything will be predicted as positive. Now, let us get the Classification Report for that. So, this is our Classification Report. The recall when it is 0, it is 0.5, recall when it is 1, it is 1.00.

Now, let us go for another threshold value that is 0.5, when it is 0.5, see that I have changed the value is 0.5, let us predicted y value. This predicted y value, then let us draw the confusion matrix. What has happened, the threshold value has move on right hand side, we

are getting more true negative and less true positive value. For this, let us get the Classification Report. This 0 represent specificity is increasing.

So what has happening here, sensitivity is decreasing. When you move towards right hand side, the threshold value goes towards right hand side, what is happening specificity is increasing, sensitivity is decreasing. The previous curve you see that, when it is low this side, you see that the sensitivity is 1 almost. It is 1 exactly. When we have low value of, low value is 0.35. When it is 0.35, you see the specificity is 0.5, but sensitivity is 0.1.

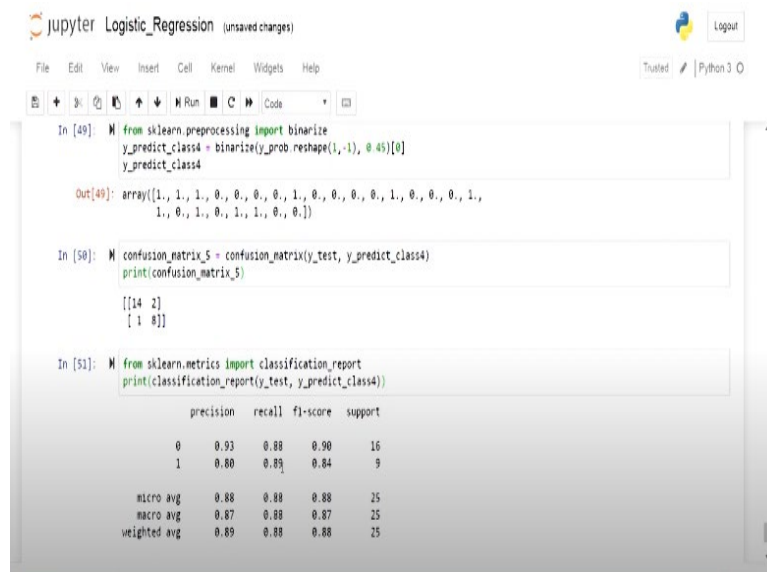
When the threshold value is increasing, what has happened that the specificity increased, but the sensitivity decreased. Let us go for 0.7. When the threshold value is 0.7, this is our predicted value. Let us go for confusion matrix. When the threshold value is high, there are more true negative because it is extremely the right hand side. So what they say, whatever the kind of pathology lab that whoever goes there, they will get a negative report.

So, that is the effect of changing the threshold value from lower side to upper side. Now, let us get the Classification Report for this. Here the specificity is 1 when you are having higher threshold value. So, sensitivity is 0 because you have chosen higher threshold value. Now, the important task in our class, we have to choose what is the optimal cut-off point or cut-off point in the sense, optimal threshold value.

So, we will import this ROC curve, then we will run ROC curve, `y_test` and `y_prob`, then we will print area under ROC curve, that is AUC curve, the area under ROC curve for optimal threshold value is 0.90, so it is the best one, because it is nearer to 1. But we want to know what is the optimal threshold value. For that purpose, we have to run this one. We have to import numpy as np, i equal to np.arange for tpr.

For each tpr value, we have to get roc, roc equal to pd. DataFrame, pd. Series false positive rate, then we are getting different index values, so when you run this command, you are getting a table, which shows the optimal threshold value. So, what is the meaning of this one is, if you take the t values 0.457, that will give you higher area under curve.

**(Refer Slide Time: 22:15)**



```
In [49]: from sklearn.preprocessing import binarize
y_predict_class4 = binarize(y_prob.reshape(1,-1), 0.45)[0]
y_predict_class4

Out[49]: array([1., 1., 1., 0., 0., 0., 1., 0., 0., 0., 1., 0., 0., 0., 1.,
        1., 0., 1., 0., 1., 1., 0., 0.])

In [50]: confusion_matrix_5 = confusion_matrix(y_test, y_predict_class4)
print(confusion_matrix_5)

[[14  2]
 [ 1  8]]

In [51]: from sklearn.metrics import classification_report
print(classification_report(y_test, y_predict_class4))
```

	precision	recall	f1-score	support
0	0.93	0.88	0.90	16
1	0.80	0.89	0.84	9
micro avg	0.88	0.88	0.88	25
macro avg	0.87	0.88	0.87	25
weighted avg	0.89	0.88	0.88	25

So, when you run this, you see that the blue line says that true positive rate, the red one shows the 1 minus false positive rate. So, this intersection where you see what happening, the true positive rate high, here 1 minus false positive rate also high. So, this is our optimal value. For that optimal value, let us draw our new ROC curve. Here, I have taken 0.45, I have drawn the confusion matrix, you see that here.

Here the confusion matrix 14, the true negative is very high, true positive also very high. So, when you take threshold values 0.45, you are getting higher true negative value and higher true positive value. When you look at Classification Report, you see the specificity value is 0.88, the sensitivity value is 0.89. In this lecture, I have taken a sample problem. With the help of sample problem, I have explained to you how to construct a confusion matrix and how to choose the correct T value, correct threshold value.

We have chosen different threshold value, for example, we have taken threshold value 0.35, we plotted the ROC curve. Then we have taken threshold value 0.5, then we plotted the ROC curve. Next, we have taken threshold value above 0.5 that is 0.7, then we have plotted different threshold value. Then when compared, when we improved or when we increase the threshold value, how the ROC curve differs.

At the end, we have chosen the optimal threshold value. In this problem, we got it as 0.45. Then for that optimal threshold value, we found the AUC, the area under curve, that also very high. So, this is the way to choose the correct threshold value for checking the quality of our classification matrix or Regression models. Thank you.