

Classification Metrics

- How to check the Classification you have applied is accurate or not by using Classification Metrics
- video
- github

- Types
 - Accuracy
 - Confusion Matrix
 - Precision
 - Recall
 - F1 score

Accuracy

$$\text{Accuracy} = \frac{\text{Number of correct Prediction}}{\text{Number of total prediction}}$$

code

```
from sklearn.metrics import accuracy_score

# Initialize the logistic regression model
logreg = LogisticRegression()

# Train the model on the training data
logreg.fit(X_train, y_train)

# Make predictions on the testing data
y_pred = logreg.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)

print("Accuracy:", accuracy)
```

How much Accuracy is good It depends on problem

Problem with Accuracy It tell us your system is 90% accurate but does not tell what type of error is coming in rest of the 10%

When accuracy is misleading For imbalance data set

Confusion Matrix

For binary Class Classification (output have only 2 class -> 0,1)

code

```
from sklearn.metrics import confusion_matrix

y_pred = logreg.predict(X_test)

array([[26, 6],
       [8, 29]])

print(confusion_matrix(y_test, y_pred))
```

output of confusion matrix

	Prediction 1	Prediction 0
Actual 1	True Positive (TP)	False Negative (FN)
Actual 0	False Positive (FP)	True Negative (TN)

Type 2 error = False Negative

Type 1 error = False Positive

For Multi Class Classification ((output have more then 2 class -> 0,1,3))

Actual\Predicted	Cat	Dog	Rabbit
Cat	42	8	2
Dog	10	35	5
Rabbit	3	9	45

output of confusion matrix

Precision , Recall and F1 score

For binary Class Classification (output have only 2 class -> 0,1)

Precision & Recall

Precision

Out of all the positive prediction , how much of them truly belong their

Formula

$$\text{Precision} = \frac{TP}{TP+FP}$$

Here TP=True Positive FP=False Positive

Recall

How much of a categories cases get correctly classified

Formula

$$\text{Recall} = \frac{TP}{TP+FN}$$

Here TP=True Positive FN=False Positive

When to use Precision and Recall

Case 1 - Email spam classifier

Let suppose accuracy of the two model you are comparing are the same , but they have difference in Type 1 and Type 2 error

Type 1 error : You have predicted a non spam mail as spam

Type 2 error : You have predicted a spam mail as non spam

Here type 1 error is more dangerous so we use precision

Case 2 - Cancer Detector

Let suppose accuracy of the two model you are comparing are the same , but they have difference in Type 1 and Type 2 error

Type 1 error : You have predicted a non cancer patients as cancer

Type 2 error : You have predicted a cancer patients as non patients

Here type 2 error is more dangerous so we use Recall

code

```
from sklearn.metrics import recall_score, precision_score

y_pred = logreg.predict(X_test)

print("Precision - ", precision_score(y_test, y_pred))
print("Recall - ", recall_score(y_test, y_pred))
```

F1 score

Formula

$$F1 = \frac{2PR}{P+R}$$

P= Precision R= Recall

When to use

When both type 1 and type 2 are equally important -> Example = Dog and cat classifier

code

```
from sklearn.metrics import f1_score

y_pred = logreg.predict(X_test)

print("F1 score - ", f1_score(y_test, y_pred))
```

Implementation

For Multi Class Classification ((output have more then 2 class -> 0,1,3))

code

way 1

weighted

```
precision_score(y_test, y_pred, average='weighted')
recall_score(y_test, y_pred, average='weighted')
f1_score(y_test, y_pred, average='weighted')
```

macro

```
precision_score(y_test, y_pred, average='macro')
recall_score(y_test, y_pred, average='macro')
f1_score(y_test, y_pred, average='macro')
```

way 2 (Better way)

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

Interpretation

	precision	recall	f1-score	support
0	0.95	0.96	0.96	621
1	0.95	0.96	0.96	962
2	0.95	0.96	0.95	829
3	0.96	0.92	0.94	690
4	0.93	0.94	0.94	550
5	0.97	0.94	0.96	779
6	0.95	0.94	0.95	529
7	0.95	0.93	0.94	879
8	0.96	0.92	0.94	979
9	0.93	0.92	0.93	824
accuracy			0.95	6900
macro avg	0.95	0.95	0.95	6900
weighted avg	0.95	0.95	0.95	6900