

T72 - ROC

March 6, 2021

1 ROC and AUC

1.1 Setting up

- Breast cancer data
- 2 classes
- 30 features

```
[ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline

# Breast cancer data
from sklearn.datasets import load_breast_cancer

# Load data
dataObj = load_breast_cancer()
X = dataObj.data
y = dataObj.target

# Remove some data (to make ROC curve looks more interesting)
X = X[:, [1, 2]]

# Splitting data
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.20,
                                                    stratify=y,
                                                    random_state=1)
```

1.2 ROC Curve

```
[ ]: from sklearn.metrics import plot_roc_curve

# Parameters
```

```

#C = 1e-5
#C = 1e-2
C = 1
#C = 10

# Estimator
pipe_lr = Pipeline([('scl', StandardScaler()),
                    ('clf', LogisticRegression(random_state=0, C=C))])

# Training
pipe_lr.fit(X_train, y_train)

#Plot ROC curve
plot_roc_curve(pipe_lr, X_test, y_test)

```

1.3 Details

```

[ ]: from sklearn.metrics import roc_curve

# Make prediction
y_pred = pipe_lr.predict(X_test)

# Calculate probability
proba = pipe_lr.predict_proba(X_test)

# Display using dataframe
comb = np.concatenate((y_pred.reshape(-1,1), proba, y_test.reshape(-1,1)),
                      ↪axis=1)
df = pd.DataFrame(comb, columns=['y_pred', 'Prob(y=0)', 'Prob(y=1)', 'y_test'])
display(df)

[ ]: # Calculate FPR, TPR, thresholds values
fpr, tpr, thresholds = roc_curve(y_true=y_test, y_score=proba[:,1], pos_label=1)

# Display in dataframe
comb = np.stack((thresholds, fpr, tpr), axis=1)
df = pd.DataFrame(comb, columns=['Threshold', 'FPR', 'TPR'])
display(df)

```

1.4 Calculate AUC

```

[ ]: from sklearn.metrics import roc_auc_score

#AUC Value
auc_score = roc_auc_score(y_true=y_test, y_score=proba[:,1])
print(f"AUC:{auc_score:6.3f}")

```

1.5 Predicting class with different threshold

```
[ ]: # Let assume that we need TPR to be very high, we can lower the threshold  
y_pred2 = np.where(proba[:,1] > 0.2, 1, 0)  
print(y_pred2)
```