Evaluasi Model Klasifikasi Lanjutan

Pada hari sebelumnya, kita telah membahas tentang evaluasi model klasifikasi menggunakan metrik-metrik seperti akurasi, presisi, recall, F1-score, dan kurva ROC-AUC. Hari ini, kita akan membahas beberapa teknik evaluasi lanjutan yang dapat memberikan wawasan lebih dalam tentang kinerja model Anda.

# Cross-Validation untuk Evaluasi Model yang Lebih Robust

Seperti yang telah kita bahas sebelumnya, cross-validation adalah teknik yang sangat berguna untuk mendapatkan estimasi kinerja model yang lebih akurat dan menghindari overfitting. Dalam konteks klasifikasi, kita dapat menggunakan cross-validation untuk menghitung rata-rata dan standar deviasi dari metrik evaluasi pada beberapa fold data.

# Stratified K-Fold Cross-Validation

Ketika dataset Anda tidak seimbang (jumlah sampel per kelas tidak sama), penting untuk menggunakan stratified k-fold cross-validation. Teknik ini memastikan bahwa setiap fold memiliki distribusi kelas yang sama dengan dataset asli, sehingga hasil evaluasi lebih representatif.

# Analisis Kurva Precision-Recall

Kurva precision-recall (PR curve) adalah alternatif berbagai dari kurva ROC. Kurva ini memplot precision (sumbu y) terhadap recall (sumbu x) pada berbagai ambang batas klasifikasi. Kurva PR ini sangat berguna ketika Anda lebih peduli tentang meminimalkan false positif atau false negative.

# Average Precision (AP)

Average Precision (AP) adalah metrik tunggal yang merangkum kinerja model pada seluruh kurva PR. AP dihitung sebagai rata-rata dari precision pada setiap tingkat recall.

# Calibration Curve

Calibration Curve adalah plot yang menunjukkan seberapa baik probabilitas yang diprediksi oleh model sesuai dengan probabilitas sebenarnya. Model yang terkalibrasi dengan baik akan kurva yang mendekati garis diagonal.

**Tugas Hari 34**

Menggunakan dataset“Breast Cancer Wiscoint (Diagnostic)” yang tersedia di scikit-learn.

**Tugas**

1. **Memuat dan Menampilkan Data:**

* Muat dataset “Breast Cancer Wisconsin” dari scikit-learn.
* Tampilkan 5 baris pertama dari dataset.

1. **Membuat Model Klasifikasi:**

* Pilih algoritma klasifikasi yang Anda inginkan (misalnya, Logistic Regression, Decision Tree, Random Forest, atau SVM).
* Bagi data menjadi data latih (80%) dan data uji (20%).
* Latih Model pada data latih.

1. **Evaluasi Model:**

* Lakukan 5-fold stratified cross-validation pada model.
* Hitung rata-rata dan standar deviasi dari akurasi, presisi, recall, dan F1-score pada setiap fold.
* Buat kurva ROC dan hitung AUC.
* Buat kurva precision-recall dan hitung average precision.
* Buat calibration curve.
* Analisis semua hasil evaluasi.

**Contoh Kode (Scikit-Learn dan Matplotlib)**

from sklearn.datasets import load\_breast\_cancer

from sklearn.model\_selection import StratifiedKFold

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score, roc\_curve, auc, precision\_recall\_curve, average\_precision\_score, plot\_calibration\_curve

# ... (kode untuk memuat dan mempersiapkan dataset, membagi data, dan membuat model)

# Cross-validation

cv = StratifiedKFold(n\_splits=5, shuffle=True, random\_state=42)

scores = []

for train\_idx, test\_idx in cv.split(X, y):

X\_train, X\_test = X[train\_idx], X[test\_idx]

y\_train, y\_test = y[train\_idx], y[test\_idx]

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

scores.append({

'accuracy': accuracy\_score(y\_test, y\_pred),

'precision': precision\_score(y\_test, y\_pred),

'recall': recall\_score(y\_test, y\_pred),

'f1': f1\_score(y\_test, y\_pred)

})

# ... (kode untuk menghitung rata-rata dan standar deviasi dari skor)

# ... (kode untuk membuat kurva ROC dan menghitung AUC)

# ... (kode untuk membuat kurva precision-recall dan menghitung average precision)

# ... (kode untuk membuat calibration curve)

**Catatan:**

Coba gunakan beberapa algoritma klasifikasi yang berbeda dan bandingkan kinerjanya menggunakan metrik-metrik evaluasi.

**Tugas**

1. **Memuat dan Menampilkan Data**

**import numpy as np # Import NumPy**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**from sklearn.datasets import load\_breast\_cancer**

**from sklearn.model\_selection import train\_test\_split, StratifiedKFold**

**from sklearn.linear\_model import LogisticRegression**

**from sklearn.metrics import (**

**accuracy\_score, precision\_score, recall\_score, f1\_score,**

**roc\_curve, auc, precision\_recall\_curve, average\_precision\_score,**

**ConfusionMatrixDisplay**

**)**

**from sklearn.calibration import calibration\_curve**

**# 1. Memuat dan Menampilkan Data**

**data = load\_breast\_cancer()**

**df = pd.DataFrame(data.data, columns=data.feature\_names)**

**df['target'] = data.target**

**print(df.head().to\_markdown(index=False, numalign="left", stralign="left"))**

1. **Membuat Model Klasifikasi**

**# 2. Memisahkan Fitur (X) dan Target (y)**

**X = df.drop(columns=['target'])**

**y = df['target']**

**# 3. Membagi Data menjadi Data Latih dan Data Uji**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42, stratify=y)**

**# 4. Membuat Model Klasifikasi (Logistic Regression)**

**model = LogisticRegression()**

1. **Evaluasi Model**

**# 5. Evaluasi Model**

**cv = StratifiedKFold(n\_splits=5, shuffle=True, random\_state=42)**

**accuracy\_scores = []**

**precision\_scores = []**

**recall\_scores = []**

**f1\_scores = []**

**tprs = []**

**aucs = []**

**mean\_fpr = np.linspace(0, 1, 100)**

**fig, ax = plt.subplots()**

**# Inisialisasi list untuk menyimpan y\_test dan y\_proba dari semua fold**

**y\_tests = []**

**y\_probas = []**

**for i, (train, test) in enumerate(cv.split(X, y)):**

**model.fit(X.iloc[train], y.iloc[train])**

**y\_proba = model.predict\_proba(X.iloc[test])[:, 1]**

**y\_probas.extend(y\_proba)**

**y\_pred = model.predict(X.iloc[test])**

**y\_tests.extend(y.iloc[test])**

**accuracy\_scores.append(accuracy\_score(y.iloc[test], y\_pred))**

**precision\_scores.append(precision\_score(y.iloc[test], y\_pred))**

**recall\_scores.append(recall\_score(y.iloc[test], y\_pred))**

**f1\_scores.append(f1\_score(y.iloc[test], y\_pred))**

**fpr, tpr, \_ = roc\_curve(y.iloc[test], y\_proba)**

**tprs.append(np.interp(mean\_fpr, fpr, tpr))**

**tprs[-1][0] = 0.0**

**roc\_auc = auc(fpr, tpr)**

**aucs.append(roc\_auc)**

**ax.plot(fpr, tpr, lw=1, alpha=0.3,**

**label='ROC fold %d (AUC = %0.2f)' % (i, roc\_auc))**

**# Menghitung dan plot rata-rata ROC curve (setelah loop selesai)**

**mean\_tpr = np.mean(tprs, axis=0)**

**mean\_tpr[-1] = 1.0**

**mean\_auc = auc(mean\_fpr, mean\_tpr)**

**std\_auc = np.std(aucs)**

**ax.plot(mean\_fpr, mean\_tpr, color='b',**

**label=r'Mean ROC (AUC = %0.2f $\pm$ %0.2f)' % (mean\_auc, std\_auc),**

**lw=2, alpha=.8)**

**# Tambahkan garis diagonal (random classifier)**

**ax.plot([0, 1], [0, 1], linestyle='--', lw=2, color='r',**

**label='Random (AUC = 0.5)', alpha=.8)**

**# Styling plot**

**ax.set(xlim=[-0.05, 1.05], ylim=[-0.05, 1.05],**

**title="ROC Curve untuk 5-Fold Cross Validation")**

**ax.legend(loc="lower right")**

**plt.show()**

**# Cetak hasil evaluasi rata-rata**

**print("\nHasil Evaluasi Rata-rata (5-fold Cross Validation):")**

**print(f'Akurasi: {np.mean(accuracy\_scores):.2f} (+/- {np.std(accuracy\_scores):.2f})')**

**print(f'Presisi: {np.mean(precision\_scores):.2f} (+/- {np.std(precision\_scores):.2f})')**

**print(f'Recall: {np.mean(recall\_scores):.2f} (+/- {np.std(recall\_scores):.2f})')**

**print(f'F1-Score: {np.mean(f1\_scores):.2f} (+/- {np.std(f1\_scores):.2f})')**

**# Precision-Recall curve dan Average Precision**

**precision, recall, \_ = precision\_recall\_curve(y\_tests, y\_probas) # Menggunakan semua y\_proba**

**average\_precision = average\_precision\_score(y\_tests, y\_probas) # Menggunakan semua y\_proba**

**plt.figure()**

**plt.step(recall, precision, color='b', alpha=0.2, where='post')**

**plt.fill\_between(recall, precision, step='post', alpha=0.2, color='b')**

**plt.xlabel('Recall')**

**plt.ylabel('Precision')**

**plt.ylim([0.0, 1.05])**

**plt.xlim([0.0, 1.0])**

**plt.title(f'Precision-Recall curve: AP={average\_precision:.2f}')**

**plt.show()**

**# Calibration curve**

**prob\_true, prob\_pred = calibration\_curve(y\_tests, y\_probas, n\_bins=10)**

**plt.figure()**

**plt.plot([0, 1], [0, 1], linestyle='--') # Garis referensi (diagonal)**

**plt.plot(prob\_pred, prob\_true, marker='.')**

**plt.xlabel("Mean predicted value")**

**plt.ylabel("Fraction of positives")**

**plt.title("Calibration Curve")**

**plt.show()**

**Penjelasan Tambahan:**

Berikut adalah beberapa penjelasan tambahan tentang metrik evaluasi dan visualisasi yang Anda gunakan dalam kode:

* **Stratified K-Fold Cross-Validation:**
  + Digunakan untuk memastikan setiap fold memiliki distribusi kelas yang sama dengan dataset asli, sehingga hasil evaluasi lebih representatif, terutama pada dataset yang tidak seimbang.
* **ROC Curve (Receiver Operating Characteristic Curve):**
  + Menggambarkan kinerja model pada berbagai ambang batas probabilitas (thresholds).
  + True Positive Rate (TPR) vs. False Positive Rate (FPR) diplot untuk setiap threshold.
  + Semakin tinggi kurva ROC, semakin baik kinerja model.
  + Area di bawah kurva ROC (AUC) adalah metrik tunggal yang merangkum kinerja model. Nilai AUC yang ideal adalah 1.0.
* **Precision-Recall Curve:**
  + Mirip dengan ROC curve, tetapi memplot precision (presisi) vs. recall (recall) pada berbagai ambang batas.
  + Berguna ketika Anda lebih peduli tentang false positive atau false negative.
  + Average Precision (AP) adalah metrik tunggal yang merangkum kinerja model pada seluruh kurva precision-recall. Nilai AP yang ideal adalah 1.0.
* **Calibration Curve:**
  + Menunjukkan seberapa baik probabilitas yang diprediksi oleh model sesuai dengan probabilitas sebenarnya.
  + Jika model terkalibrasi dengan baik, kurva akan mendekati garis diagonal.

**Kesimpulan:**

Dengan menggunakan berbagai metrik evaluasi dan visualisasi ini, Anda telah memperoleh pemahaman yang lebih komprehensif tentang kinerja model klasifikasi Logistic Regression pada dataset "Breast Cancer Wisconsin". Anda dapat menggunakan informasi ini untuk memilih model terbaik, menyesuaikan ambang batas keputusan, dan bahkan mencoba algoritma klasifikasi lain untuk melihat apakah ada yang lebih baik.