MODEL PERFORMANCE ANALYTICS

MODULE

DATA SCIENCE TRACK

Table of Contents

TUJUAN DAN DESKRIPSI PEMBELAJARAN

Deskripsi Pembelajaran
Peserta akan mengikuti pembelajaran *Live Session* dimana peserta akan belajar mengenai konsep dasar statistika pada *Data Science*. Peserta juga wajib menyelesaikan tugas mandiri dimana peserta akan diberikan beberapa soal mengenai materi yang sudah dijelaskan dan peserta diharapkan dapat menjawab soal tersebut dengan baik dan benar. Pada materi ini, ada beberapa sub materi yang akan dibahas: 1. *The science of predictive modeling* 2. *Learning curve* 3. *ROC analysis*
Tujuan Pembelajaran
- Peserta memahami apa itu pemodelan prediktif dan mampu menerapkannya pada masalah ilmu data untuk memprediksi perilaku masa depan - Peserta dapat menggunakan kurva pembelajaran untuk mendiagnosis kinerja model pembelajaran mesin - Peserta dapat menggunakan ROC untuk mendiagnosis kinerja model pembelajaran mesin
Silabus
Adapun materi dan sub materi yang akan dipelajari peserta ialah sebagai berikut: **4. 1 The science of predictive modeling** Memberikan pemahaman kepada mahasiswa tentang apa itu predictive modeling dan bagaimana penerapannya untuk memprediksi perilaku masa depan **4.2 Learning curves** Memberikan pemahaman kepada mahasiswa tentang cara menggunakan learning curves untuk mendiagnosis performa model machine learning **4.3 ROC analysis** Memberikan pemahaman kepada mahasiswa tentang cara menggunakan ROC Analysis untuk mendiagnosis performa model machine learning.
Durasi Pembelajaran
90 jam

Referensi

No	SubTopik	Referensi
1	The Science of Predictive Modeling	Article: • Predictive Modeling • What is Predictive Analytics • Model Complexity & Overfitting in Machine Learning
2	Learning Curve	Article: • Why you should be plotting learning curves in your next machine learning project • Plotting the Learning Curve with a Single Line of Code • Model Complexity & Overfitting in Machine Learning
3	ROC Analysis	Article: How to Plot a ROC Curve in Python (Step-by-Step)

The Science of Predictive Modeling

Predictive modeling adalah proses matematis yang digunakan untuk memprediksi kejadian mendatang atau output dengan menganalisis pola - pola dalam data input. Predictive modeling sendiri adalah salah satu komponen penting dalam *predictive analytics*, yaitu salah satu tipe *data analytics* yang menggunakan data historis dan data saat ini untuk meramalkan aktivitas, kebiasaan, maupun tren. Selain itu, *predictive modeling* dapat memprediksi kebutuhan di masa depan atau membantu proses analisis *what if*.

Beberapa bentuk *predictive modeling* yang paling umum digunakan antara lain:

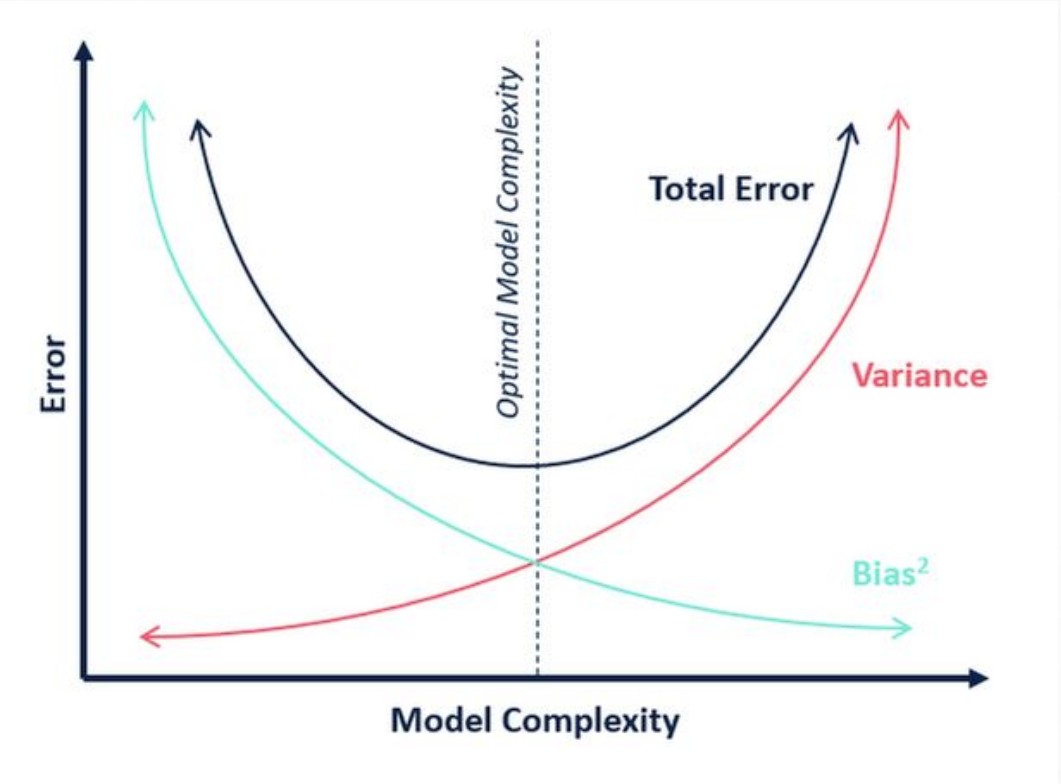
- *Classification Model*
  Model klasifikasi adalah salah satu cabang dari *supervised learning*. Model ini mengkategorikan data berdasarkan data historis dengan mendeskripsikan relasi yang ada dalam *dataset*. Beberapa model klasifikasi antara lain *logistic regression*, *decision tree*, *random forest*, dll.
- *Clustering Model*
  Model *clustering* adalah salah satu cabang dari *unsupervised learning*. Model ini mengelompokkan data berdasarkan atribut - atribut yang sama. Beberapa model *clustering* antara lain *k - means*, *mean - shift*, *hierarchical*, dll.

The Science of Predictive Modeling

- *Time Series Model*
  Model *time series* menggunakan berbagai jenis data input dalam frekuensi waktu yang spesifik seperti harian, mingguan, tahunan, dsb. Umumnya, model ini mampu mendeteksi tren, *seasonality*, dan *cyclical behavior*. Beberapa model *time series* antara lain *moving* average, ARMA, ARIMA, dll.
- *Regression model*
  model regresi merupakan fungsi yang menggambarkan hubungan antara satu atau lebih variabel independen dan variabel respons, dependen, atau target. tipe model regresi meliputi tipe linear, multiple, non linear, dan *stepwise* regression *modeling.*

Salah satu penerapan *predictive modeling* terdapat pada bidang *marketing and sales*. *Predictive modeling* memungkinkan perusahaan untuk lebih proaktif dalam proses pendekatan terhadap klien. Sebagai contoh, model *churn prediction* dapat membantu divisi *sales* untuk mengidentifikasi konsumen yang tidak puas secara lebih cepat. Selain itu, *predictive modeling* dapat membantu tim *marketing* dalam melaksanakan *cross sell strategies*. Hal ini biasanya diimplementasikan dalam bentuk *recommendation engine* dalam *website* perusahaan.

The Science of Predictive Modeling



Model Complexity dan *overfitting* adalah dua masalah utama yang dapat terjadi dalam *machine learning*, khususnya dalam *predictive modeling*. *Model Complexity* dapat menghasilkan model yang terlalu kompleks dan tidak dapat digeneralisasikan dengan baik ke data baru, sedangkan *overfitting* dapat menyebabkan model bekerja dengan baik pada data pelatihan tetapi buruk pada data baru. Ada beberapa cara untuk mencegah masalah ini, termasuk menggunakan model yang lebih sederhana, menggunakan teknik regularisasi, memisahkan data menjadi set pelatihan dan set pengujian, penghentian awal, dan validasi silang. Penting untuk memantau kinerja model saat sedang dilatih dan menyesuaikan parameternya.

The Science of Predictive Modeling

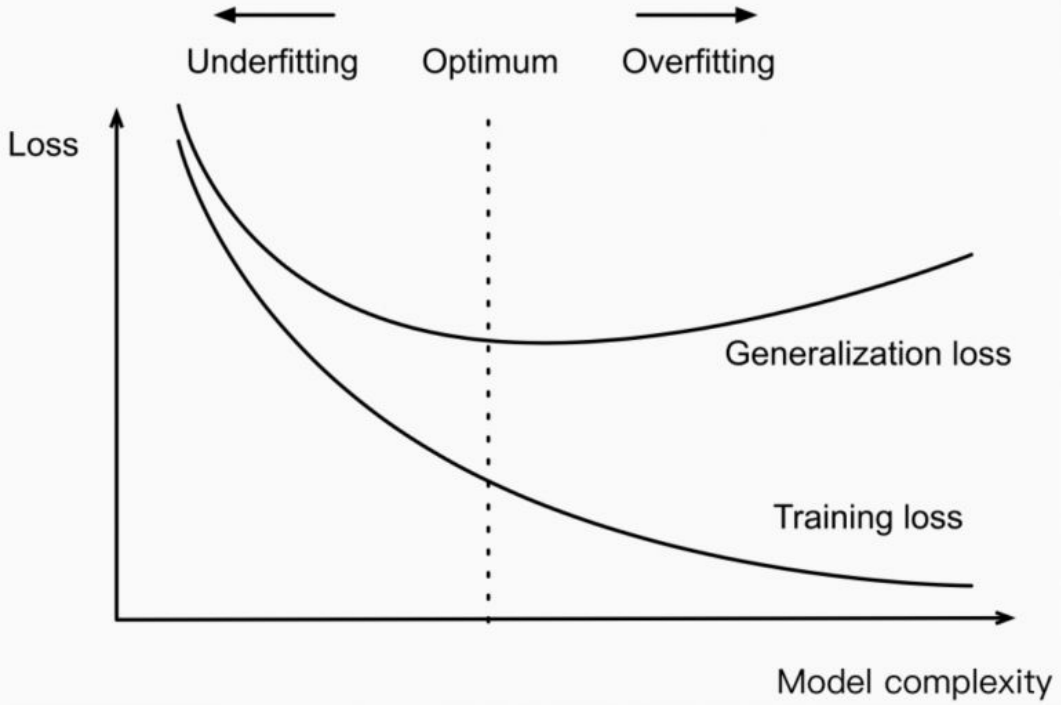
Ada beberapa cara untuk menghindari masalah ini:

- Gunakan model yang lebih sederhana
  Model yang lebih sederhana seringkali lebih kuat dan lebih baik digeneralisasikan ke data yang lebih baru. Salah satu cara untuk membuat model yang lebih sederhana adalah dengan menghindari terlalu banyak fitur. Penting untuk hanya memilih fitur yang paling relevan untuk model data.
- Gunakan teknik regularisasi
  Teknik regulasi membantu menghindari pembuatan model yang terlalu rumit dengan menghindari nilai parameter yang berlebihan. Teknik regularisasi yang umum termasuk regularisasi L1 (Lasso) dan L2 (Ridge). Misalnya, regresi Lasso adalah jenis regresi linier yang menggunakan regularisasi untuk mengurangi *Model Complexity* dan mencegah *overfitting*.
- Pisahkan data menjadi satu set pelatihan dan satu set pengujian
  Memungkinkan model untuk dilatih pada satu set data dan kemudian diuji pada set data lainnya. Ini dapat membantu mencegah *overfitting* dengan memastikan bahwa model digeneralisasikan dengan baik ke data baru.
- Gunakan penghentian lebih awal
  Penghentian lebih awal adalah teknik lain yang dapat digunakan untuk mencegah *overfitting*. Ini melibatkan pelatihan model sampai kesalahan validasi mulai meningkat dan kemudian menghentikan proses pelatihan. Ini memastikan bahwa model tidak terus sesuai dengan data pelatihan setelah model mulai *overfit*.

The Science of Predictive Modeling

- Gunakan validasi silang
  Validasi silang adalah teknik yang dapat digunakan untuk mengurangi *overfitting* dengan membagi data menjadi beberapa set dan melatih setiap set secara bergantian. Hal ini memungkinkan model dilatih pada data yang berbeda dan mencegahnya dari *over fitted* pada kumpulan data tertentu.
- Pantau kinerja model saat *training* dan sesuaikan parameternya.

Learning Curves



Seperti yang telah dijelaskan di modul sebelumnya, dalam *machine learning*, tujuan utama suatu model *machine learning* adalah mengoptimalkan *class-assignment probability* atau probabilitas masing - masing kelas untuk suatu *input*. Kemampuan model untuk mengoptimalkan *class-assignment probability* ini diukur dalam evaluasi kinerja model. Selain *cross validation* dan *bootstrapping*, tool lain yang dapat digunakan adalah *learning curves*.

Learning curves menunjukkan hubungan antara ukuran *training dataset*. Dengan metode evaluasi yang digunakan (RMSE, akurasi, dll) terhadap *training dataset* dan *validation* (atau *cross validation*) *dataset*. Melalui *learning curves*, kita dapat mendeteksi apakah model yang sedang dibangun memiliki bias atau varian.

Learning Curves

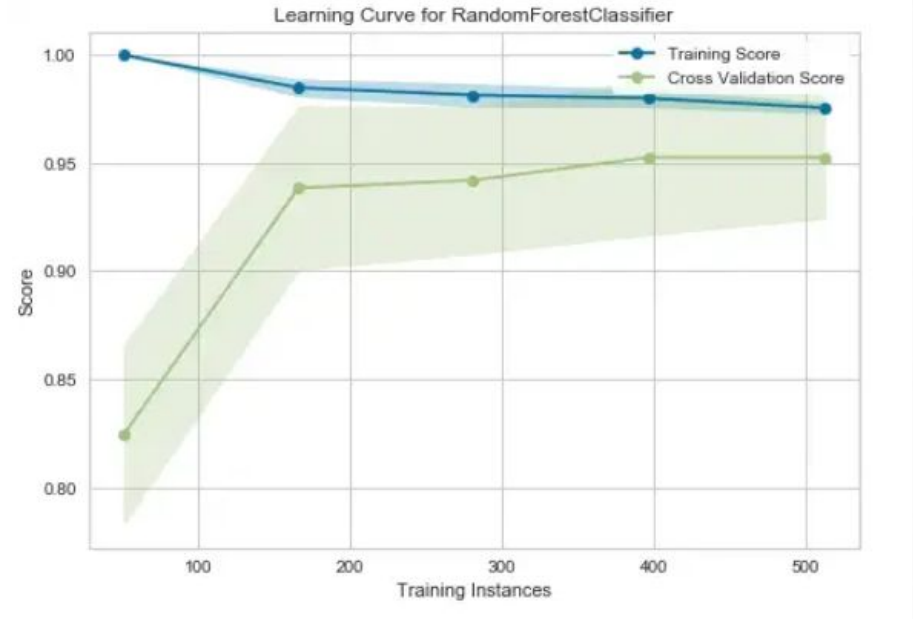
Apabila *learning curves* menunjukkan hasil di sebelah kiri garis optimum, hal ini berarti model yang sedang dibangun memiliki tingkat bias yang tinggi. Tingkat *error* dalam *training* dan *generalization* sangat tinggi. Selain itu, *learning curve* di atas juga menunjukkan bahwa model mengalami *underfitting*.

Apabila *learning curves* menunjukkan hasil di sebelah kanan garis *optimum*, hal ini berarti model yang sedang dibangun memiliki tingkat varian yang tinggi. Dalam grafik tersebut, tingkat *error* pada *generalization* jauh lebih tinggi dari tingkat *error* pada *training*. Selain itu, learning curve di atas juga menunjukkan bahwa model mengalami *overfitting*.

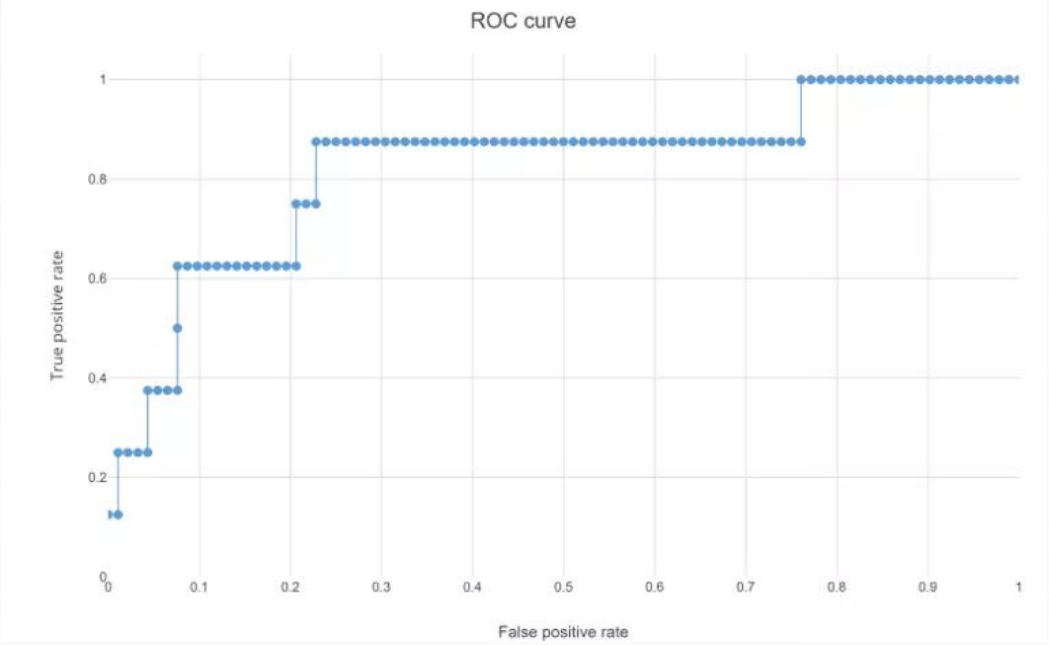
Berikut adalah contoh penggunaan *learning curves* pada model *random forest*.

```
1  from sklearn.ensemble import RandomForestClassifier
2  from sklearn.datasets import load_breast_cancer
3  from yellowbrick.model_selection import learning_curve
4
5  cancer = load_breast_cancer()
6  X = cancer.data
7  y = cancer.target
8
9  rfc = RandomForestClassifier(n_estimators=100, max_depth=3, random_state=0)
10 print(learning_curve(rfc, X, y, cv=10, scoring='accuracy'))
```

Learning Curves



ROC Analysis



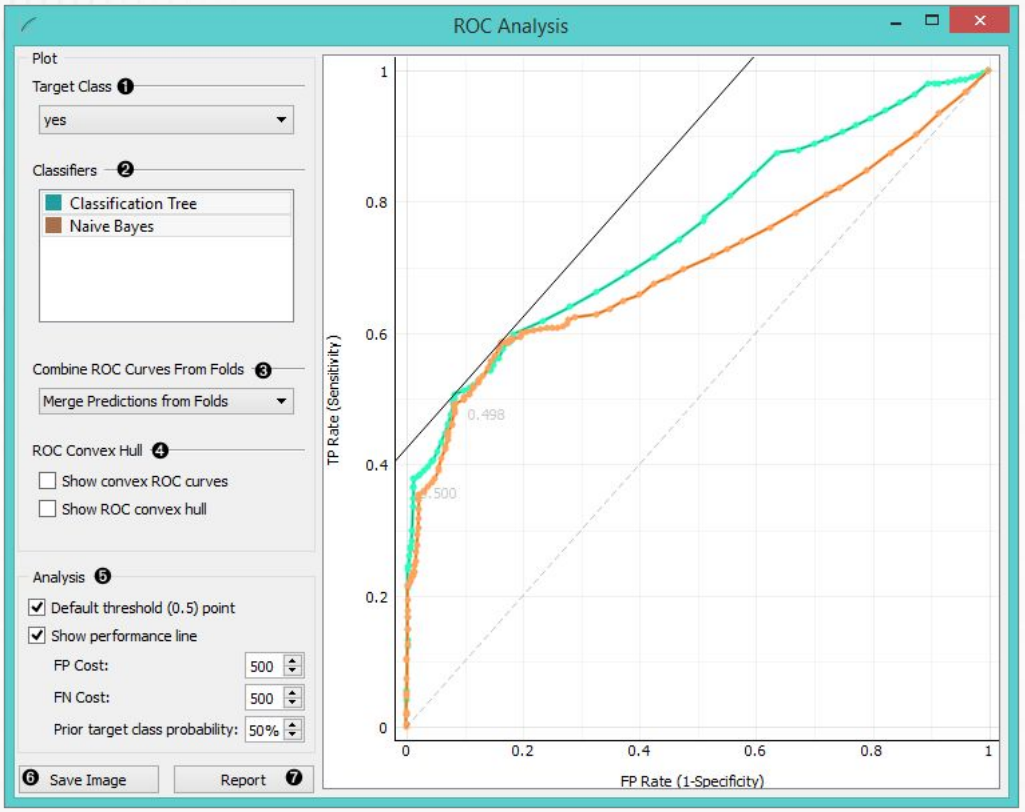
Receiver Operating Characteristic (ROC) adalah cara yang tepat untuk menilai keakuratan prediksi model dengan memplot sensitivitas versus (1-spesifisitas) uji klasifikasi (karena ambang bervariasi pada seluruh rentang hasil uji diagnostik). Area di bawah kurva ROC tertentu, atau AUC, merumuskan statistik penting yang mewakili probabilitas bahwa prediksi akan berada dalam urutan yang benar ketika variabel uji diamati (untuk satu subjek dipilih secara acak dari kelompok kasus, dan subjek lainnya dipilih secara acak. dipilih dari kelompok kontrol). Analisis ROC mendukung inferensi mengenai satu kurva AUC, *precision-recall* (PR), dan memberikan opsi untuk membandingkan dua kurva ROC yang dihasilkan dari kelompok independen atau subjek berpasangan.

ROC Analysis

Kurva ROC juga merupakan plot grafis yang digunakan untuk menunjukkan kemampuan diagnostik dari pengklasifikasi biner. Hal ini pertama kali digunakan dalam teori deteksi sinyal namun sekarang digunakan di banyak bidang lain seperti kedokteran, radiologi, bahaya alam, dan machine learning.

Widget menampilkan kurva ROC untuk model yang diuji dan *convex hull* yang sesuai. Hal ini berfungsi sebagai sarana perbandingan antara model klasifikasi. Kurva memplot laju positif palsu pada sumbu x (spesifisitas 1; probabilitas bahwa target=1 bila nilai sebenarnya=0) terhadap laju positif sejati pada sumbu y (sensitivitas; probabilitas bahwa target=1 bila nilai sebenarnya= 1). Semakin dekat kurva mengikuti batas kiri dan batas atas ruang ROC, semakin akurat pengklasifikasi. Mengingat biaya positif palsu dan negatif palsu, *widget* juga dapat menentukan pengklasifikasi dan ambang optimal.

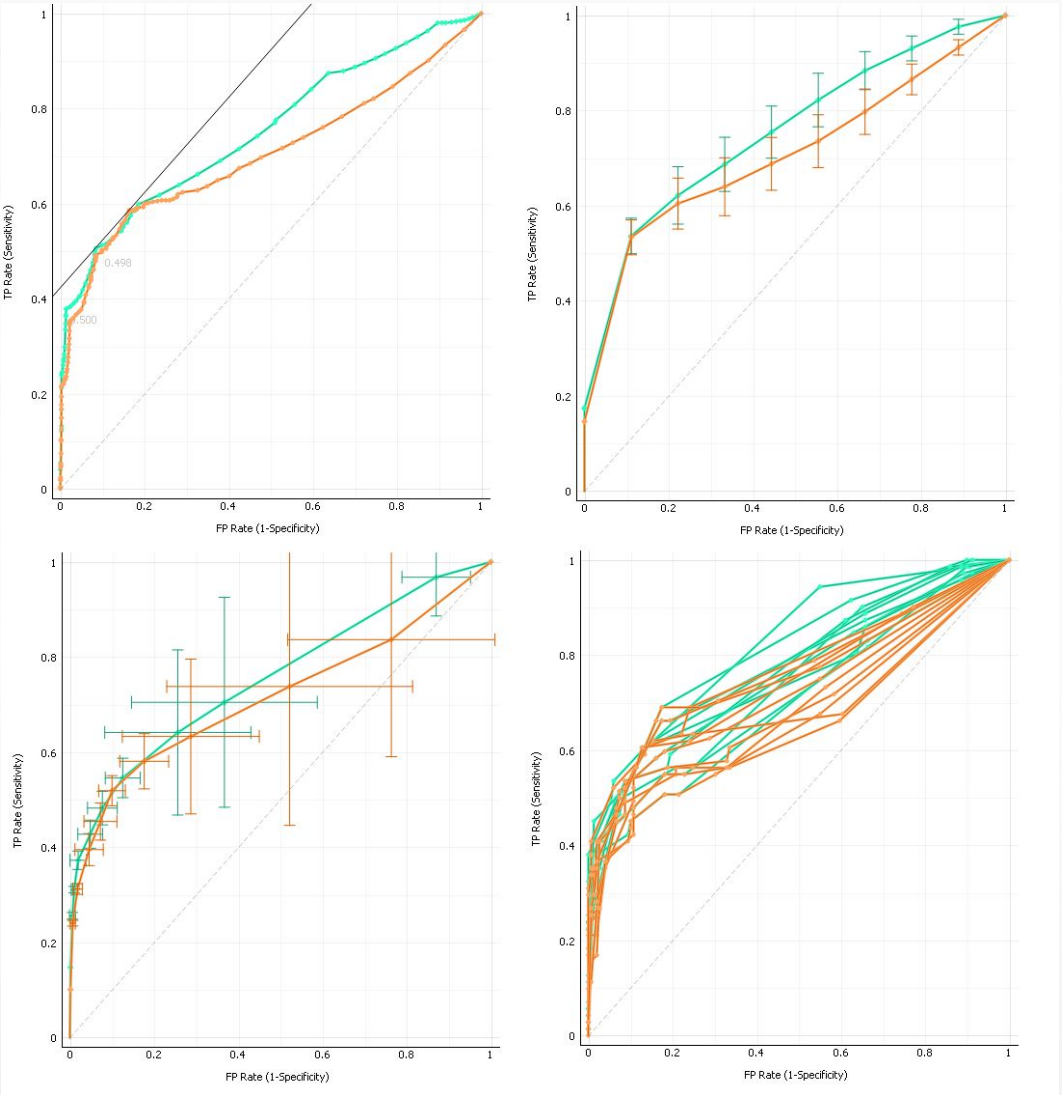
ROC Analysis



Berikut adalah tahapan - tahapan untuk melakukan ROC Analysis menggunakan *Widget* :

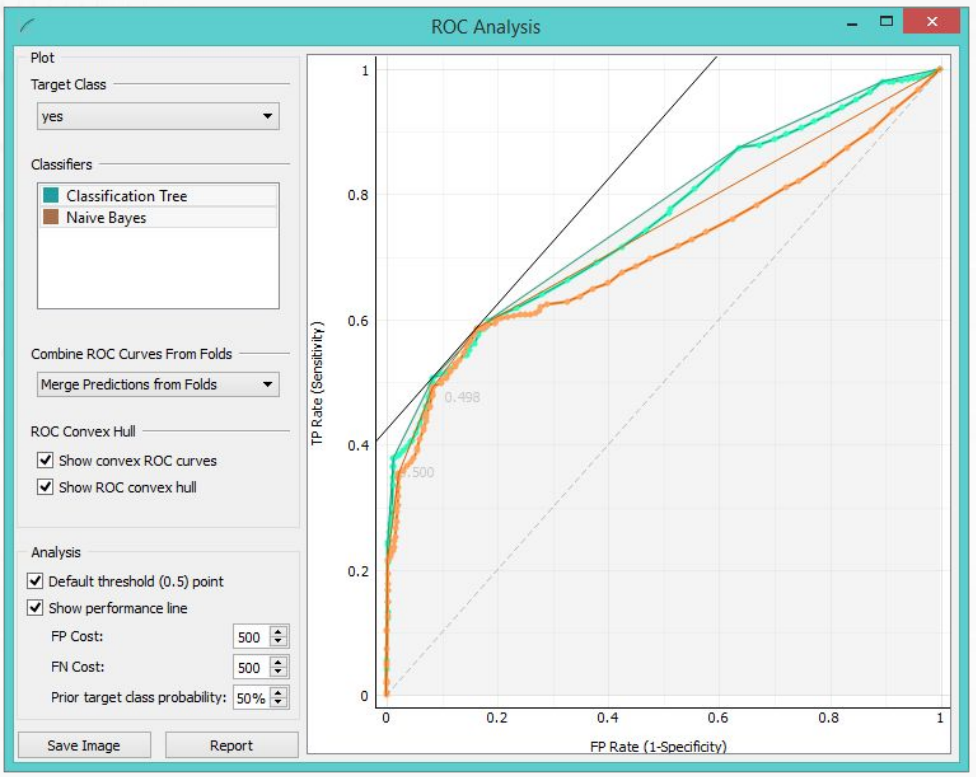
- Pilih kelas target yang diinginkan. Kelas default dipilih berdasarkan abjad.
- Jika hasil tes berisi lebih dari satu pengklasifikasi, pengguna dapat memilih kurva mana yang ingin dilihatnya diplot. Klik pada classifier untuk memilih atau membatalkan pilihan.
- Ketika data berasal dari beberapa iterasi pelatihan dan pengujian, seperti validasi silang k-fold, hasilnya dapat (dan biasanya) dirata-ratakan.
  - ➢ Gabungkan prediksi dari lipatan (kiri atas), yang memperlakukan semua data uji seolah-olah berasal dari satu iterasi

ROC Analysis



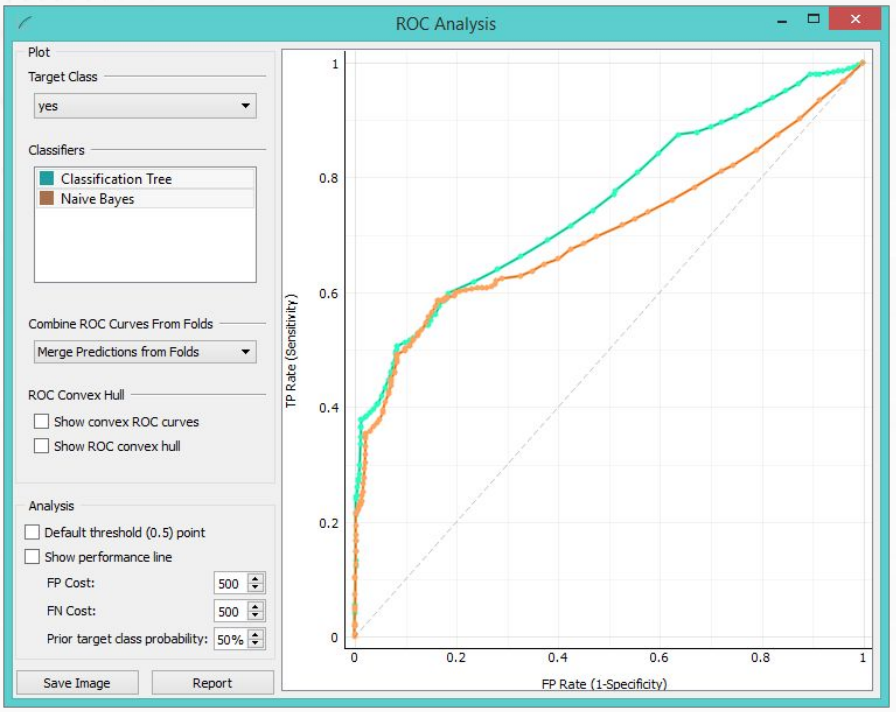
- Tingkat rata-rata TP (kanan atas) rata-rata kurva secara vertikal, menunjukkan *confidence interval* yang sesuai
- TP dan FP pada ambang batas (kiri bawah) melintasi ambang batas, rata-rata posisi kurva dan menunjukkan *confidence interval* horizontal dan vertikal
- Tampilkan kurva individu (kanan bawah)

ROC Analysis



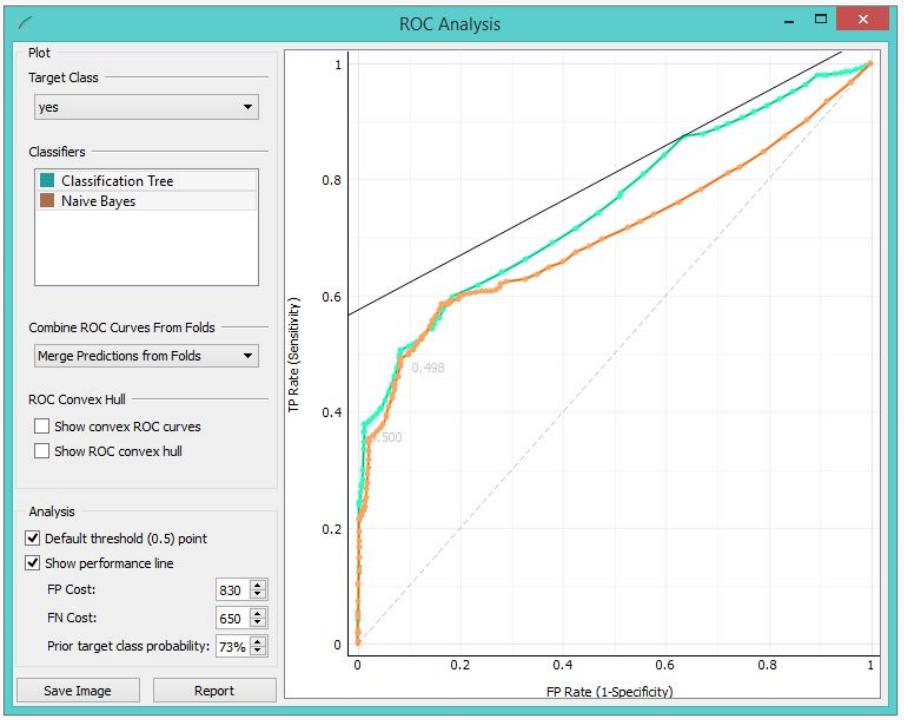
- Opsi Tampilkan kurva ROC cembung mengacu pada kurva cembung di atas setiap *classifier* individu (garis tipis yang diposisikan di atas kurva). Garis putus-putus diagonal mewakili perilaku pengklasifikasi acak. Garis diagonal penuh mewakili kinerja iso. Simbol "A" hitam di bagian bawah grafik menyesuaikan ulang grafik secara proporsional.

ROC Analysis



- Kotak terakhir dilakukan untuk analisis kurva. Pengguna dapat menentukan biaya *false positive* (FP) dan *false negative* (FN), dan probabilitas kelas target sebelumnya. Titik ambang batas *default* (0,5) menunjukkan titik pada kurva ROC yang dicapai oleh pengklasifikasi jika memprediksi kelas target jika probabilitasnya sama atau melebihi 0,5.
  Show performance line menunjukkan *iso-performance* pada ruang ROC sehingga semua titik pada garis memberikan keuntungan/kerugian yang sama. Garis lebih jauh ke kiri atas lebih baik daripada yang ke bawah dan ke kanan. Arah garis tergantung pada biaya dan probabilitas. Jika kita mendorong *iso-performance* lebih tinggi atau lebih ke kiri, titik-titik pada garis *iso-performance* tidak dapat dijangkau oleh pembaca.

ROC Analysis



- Widget memungkinkan pengaturan biaya dari 1 hingga 1000. Satuan tidak penting, begitu juga besarannya. Yang penting adalah hubungan antara kedua biaya tersebut, jadi menyetelnya ke 100 dan 200 akan memberikan hasil yang sama dengan 400 dan 800.
- Tekan Simpan Gambar jika Anda ingin menyimpan gambar yang dibuat ke komputer Anda dalam format .svg atau .png. Dan kemudian menghasilkan laporan.

ROC Analysis

Selain menggunakan widget Test & Score, analisis ROC juga dapat dilakukan menggunakan Python, tahapan - tahapannya antara lain:

- *Import package*

Import Package

```
[ ]  import pandas as pd
     import numpy as np
     from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LogisticRegression
     from sklearn import metrics
     import matplotlib.pyplot as plt
```

- *Import dataset*

import dataset from CSV file on Github

```
[ ]  url = "https://raw.githubusercontent.com/Statology/Python-Guides/main/default.csv"
     data = pd.read_csv(url)
```

```
[ ]  data
```

	default	student	balance	income
0	0	0	729.526495	44361.625074
1	0	1	817.180407	12106.134700
2	0	0	1073.549164	31767.138947
3	0	0	529.250605	35704.493935
4	0	0	785.655883	38463.495879
...	...	...	...	...
9995	0	0	711.555020	52992.378914
9996	0	0	757.962918	19660.721768
9997	0	0	845.411989	58636.156984
9998	0	0	1569.009053	36669.112365
9999	0	1	200.922183	16862.952321

10000 rows × 4 columns

ROC Analysis

- Membuat model

define the predictor variables and the response variable

```
[ ]  X = data[['student', 'balance', 'income']]
     y = data['default']
```

split the dataset into training (70%) and testing (30%) sets

```
[ ]  X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_state=0)
```

instantiate the model

```
[ ]  log_regression = LogisticRegression()
```

fit the model using the training data

```
[ ]  log_regression.fit(X_train,y_train)
```

```
LogisticRegression()
```

Define Metrics

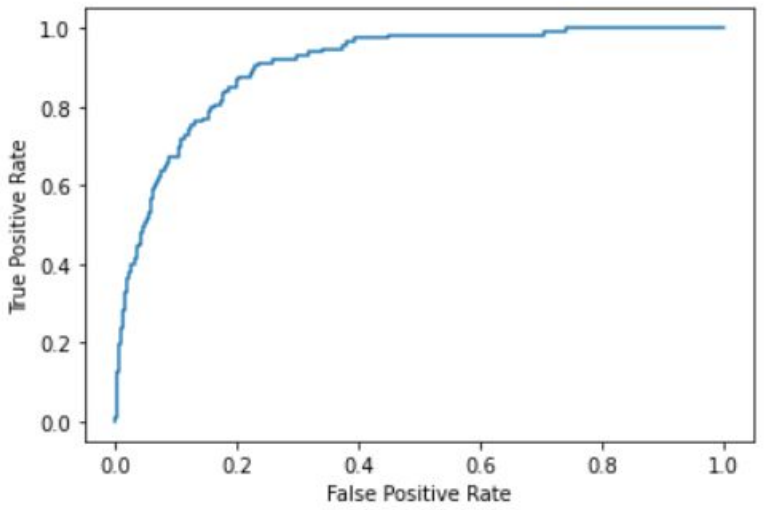
```
[ ]  y_pred_proba = log_regression.predict_proba(X_test)[::,1]
     fpr, tpr, _ = metrics.roc_curve(y_test,  y_pred_proba)
```

ROC Analysis

- Membuat kurva ROC

create ROC curve

```
[ ]  plt.plot(fpr,tpr)
     plt.ylabel('True Positive Rate')
     plt.xlabel('False Positive Rate')
     plt.show()
```



Google Colab dapat diakses pada: [ROC Curve]

EXERCISE

Berikut adalah langkah - langkah untuk mengerjakan *exercise*:

1. Buka *link* Google Colab menggunakan Google Chrome
2. Klik 'Copy to Drive'
3. Kerjakan *exercise* sesuai instruksi yang tertera

Google Colab dapat diakses pada:
Exercise Modul 4 DS

Startup Campus

Kampus Merdeka
INDONESIA JAYA

MODULE

UDATA SCIENCE TRACK