



Artikel Penelitian

Perbandingan Performa Model *Data Mining* untuk Prediksi *Dropout* Mahasiswa

Muchamad Taufiq Anwar¹, Denny Rianditha Arief Permana¹

¹Politeknik STMI Jakarta, Jl Letjend Suprpto No 26, Jakarta Pusat, 10510, Indonesia

INFORMASI ARTIKEL

Diterima : 16 Juli 2021
 Direvisi : 09 Agustus 2021
 Diterbitkan : 28 Agustus 2021

KATA KUNCI

klasifikasi, machine learning, perbandingan performa data mining, pycaret

KORESPONDENSI

E-mail Author Korespondensi:
taufiq@stmi.ac.id
 E-mail Co-Author:
dennyrian76@stmi.ac.id

A B S T R A K

Penentuan teknik/model data mining yang tepat pada sebuah kasus sangat penting untuk mendapatkan model yang baik (tingkat akurat tinggi dan kesesuaiannya dengan masalah yang dipecahkan). Penelitian ini bertujuan untuk membandingkan performa teknik *data mining* untuk diterapkan pada kasus prediksi *dropout* mahasiswa. Perbandingan performa dilakukan menggunakan library *PyCaret* pada Python untuk melakukan pemodelan menggunakan 14 model / teknik data mining yaitu: *Extreme Gradient Boosting*, *Ada Boost Classifier*, *Light Gradient Boosting Machine*, *Random Forest Classifier*, *Gradient Boosting Classifier*, *Extra Trees Classifier*, *Decision Tree Classifier*, *K Neighbors Classifier*, *Naive Bayes*, *Ridge Classifier*, *Linear Discriminant Analysis*, *Logistic Regression*, *SVM - Linear Kernel*, dan *Quadratic Discriminant Analysis*. Metrik evaluasi performa model yang digunakan yaitu *Accuracy*, *AUC*, *Recall*, *Precision*, *F1*, *Kappa*, dan *MCC (Matthews correlation coefficient)*. Hasil eksperimen menunjukkan bahwa kasus prediksi *dropout* mahasiswa lebih tepat jika dimodelkan dengan model berbasis *ensemble learner* dan pohon keputusan dengan akurasi mencapai 99%. Pohon keputusan memiliki keunggulan dibandingkan model lain seperti *SVM - Linear Kernel* dan *Quadratic Discriminant Analysis* karena ia dapat dengan lebih detail dalam memisahkan data ke dalam kedua kelas target. Setelah dilakukan penyesuaian atribut, pembuangan data dengan *missing values*, dan *parameter tuning*, didapatkan hasil akurasi yang mirip dari berbagai model yaitu sebesar 87%. Perbedaan akurasi antar model menjadi sangat kecil di saat atribut data yang digunakan sedikit.

PENDAHULUAN

Teknik *Data Mining* (DM) atau yang secara lebih umum merupakan lingkup teknik *Machine Learning* (ML) merupakan cara-cara untuk memprediksi sebuah keadaan berdasarkan pemodelan terhadap data. Hingga saat ini sudah banyak terdapat teknik DM/ML yang tersedia dan masing-masing memiliki karakteristik tersendiri. Dalam sebuah usaha penyelesaian masalah dengan teknik DM/ML, kita perlu memilih dan menggunakan teknik yang tepat agar model prediktif yang dihasilkan memiliki performa yang baik dan cocok dengan masalah yang dipecahkan. Untuk mengetahui model yang terbaik, dapat dilakukan dengan melakukan beberapa eksperimen untuk menguji setiap model yang ada.

Riset oleh Anwar pada tahun 2021 untuk prediksi kasus *dropout* mahasiswa hanya menggunakan satu model saja, yaitu C4.5 (Muchamad Taufiq Anwar, Heriyanto, & Fanini, 2021). Hal tersebut memunculkan pertanyaan baru

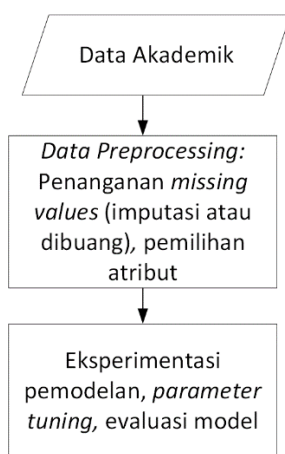
apakah memang model C4.5 merupakan model terbaik untuk diterapkan dalam kasus tersebut, sehingga penelitian ini bertujuan untuk mengetahui teknik DM yang terbaik untuk diterapkan pada kasus prediksi *dropout* mahasiswa. Dengan ditemukannya model yang terbaik, maka model akhir yang digunakan akan memiliki akurasi yang tinggi dalam kemampuan prediksinya. Riset tentang perbandingan teknik DM semacam ini sesungguhnya umum untuk dikerjakan, seperti misalnya riset (M T Anwar, Hadikurniawati, Winarno, & Widiyatmoko, 2020) yang membandingkan performa tiga teknik *data mining* untuk prediksi hujan, riset (Arbain & Balakrishnan, 2019) yang membandingkan enam teknik DM untuk prediksi penyakit hati, riset (Luo et al., 2017) yang membandingkan tiga teknik DM untuk prediksi penyakit jantung bawaan, dan lain sebagainya. Dalam alur kerja pengembangan model prediksi, hendaknya memang diperlukan tahapan perbandingan performa teknik DM untuk menentukan teknik atau model apa yang paling tepat untuk digunakan. Kasus prediksi *dropout* yang digunakan dalam penelitian ini merupakan sebuah masalah / kasus klasifikasi.

Sebanyak 14 teknik DM akan dibandingkan performanya dalam memprediksi kemungkinan seorang mahasiswa untuk *dropout*/lulus. Model yang terbaik dapat diketahui berdasarkan dari metrik evaluasi performa model seperti akurasi dan lain-lain.

METODE

Metode penelitian dalam riset ini ditunjukkan pada Gambar 1. Data yang digunakan dalam penelitian ini merupakan data yang juga digunakan pada riset Anwar tahun 2021 (Muchamad Taufiq Anwar et al., 2021). Data terdiri dari 10 kolom Indeks Prestasi Semester dengan tipe data numerik dan satu kolom kelas tujuan dengan tipe data nominal yang terdiri dari dua kelas yaitu “Lulus” dan “DO”. Secara keseluruhan terdapat 979 baris data dalam *dataset* ini.

Pemrosesan awal data (*data pre-processing*) dilakukan untuk menangani entri data yang terdapat nilai kosong (*missing values*). Proses ini disebut juga dengan istilah imputasi, yaitu pemberian nilai dugaan ke dalam entri data yang sebelumnya kosong. Teknik imputasi yang paling umum diterapkan dalam data numerik adalah dengan menggunakan nilai rata-rata dari masing-masing atribut. Selain imputasi, terdapat opsi lain untuk menangani data dengan *missing values*, yaitu dengan membuang entri data yang memiliki *missing values* tersebut. Proses penanganan *missing values* ini penting untuk dilakukan karena sebagian besar teknik DM tidak dapat memproses data yang terdapat *missing values* di dalamnya. Adapun tahap pemilihan atribut dilakukan dalam eksperimen lanjutan untuk menguji pengaruh atribut terhadap akurasi model.



Gambar 1. Metode penelitian

Data yang telah siap olah kemudian akan diproses oleh 14 macam model DM yaitu: *Extreme Gradient Boosting*, *Ada Boost Classifier*, *Light Gradient Boosting Machine*, *Random Forest Classifier*, *Gradient Boosting Classifier*, *Extra Trees Classifier*, *Decision Tree Classifier*, *K*

Neighbors Classifier, *Naive Bayes*, *Ridge Classifier*, *Linear Discriminant Analysis*, *Logistic Regression*, *SVM - Linear Kernel*, dan *Quadratic Discriminant Analysis*. Dalam tahap awal eksperimen, semua model akan dijalankan dengan menggunakan setelan standar (*default*). Untuk mengusahakan performa yang lebih baik, beberapa model dapat dilakukan ‘*parameter tuning*’, yaitu dengan mengubah-ubah setelan parameter modelnya. Terakhir, evaluasi performa model dilakukan dengan teknik *10-fold cross-validation* di mana 90% data dijadikan sebagai data *training* dan 10% lainnya menjadi data *testing* dan diulang 10 kali untuk setiap bagian yang berbeda. Sementara metrik evaluasi performa model yang digunakan yaitu *Accuracy*, *AUC*, *Recall*, *Precision*, *F1*, *Kappa*, dan *MCC* (*Matthews correlation coefficient*). Proses penyiapan data, eksperimentasi pemodelan, dan pengujian performa ke-14 model DM akan dibantu dengan menggunakan *library* bernama *PyCaret* pada Python. *Pycaret* memungkinkan kita untuk menguji banyak model secara mudah dan cepat.

PyCaret merupakan *library lowcode* untuk ML dalam Python (Ali, 2020). *Lowcode* berarti ia hanya membutuhkan sangat sedikit kode dalam penggunaannya. *PyCaret* sendiri terinspirasi dari *package* bernama *caret* (*Classification And REgression Training*) pada bahasa R yang memiliki fungsi yang mirip, yaitu menyederhanakan proses pembuatan model ML. *PyCaret* pada dasarnya merupakan *wrapper* yang memanfaatkan beberapa *library* dan *framework* ML seperti *scikit-learn*, *XGBoost*, *Microsoft LightGBM*, *spaCy* dan lain-lain. *PyCaret* versi stabil pertama (versi 1.0.0) dipublikasikan oleh Moez Ali pada April 2020. Saat ini sudah ada beberapa riset yang menggunakan *library* ini untuk berbagai keperluan seperti klasifikasi deformasi jembatan (Avendano, Otero, & Otero, 2021), klasifikasi citra satelit (Larsen, Noever, MacVittie, & Lilly, 2021), prediksi fraksi tak-terikat dalam plasma manusia (Mulpuru & Mishra, 2021), prakiraan kasus covid-19 di India (Hota, Handa, & Shrivastava, 2021), dan deteksi anomali data curah hujan harian (Herho, Fajary, & Irawan, 2021).

Proses yang dapat dilakukan oleh *PyCaret* antara lain adalah penyiapan data, pemodelan, perbandingan model, *parameter tuning*, dan abstraksi model. Dari banyaknya kemampuan *PyCaret* ini, ada kalanya sebuah riset hanya menggunakan sebagian saja dari proses yang terdapat dalam *PyCaret* tersebut. Misalnya dalam sebuah riset, *PyCaret* digunakan hanya untuk keperluan memilih model ML yang terbaik berdasarkan akurasinya (Kuhn & George, 2021). *PyCaret* juga ada kalanya khusus digunakan untuk menyiapkan data untuk kemudian diproses oleh algoritma ML, selain juga digunakan untuk menguji performa berbagai teknik ML (Aguilar-Dominguez, Ejeh, Dunbar, & Brown, 2021). Selain itu, *PyCaret* juga dapat digunakan untuk *hyperparameter-tuning* dengan menggunakan teknik

grid search (Terragni, Fersini, Galuzzi, Tropeano, & Candelieri, 2021). PyCaret dapat juga dikustomisasi untuk keperluan tertentu seperti yang dilakukan oleh (do Nascimento et al., 2021) untuk keperluan analisis prediktif pada laboratorium kesehatan. PyCaret merupakan salah satu dari banyak *framework autoML* lainnya seperti *PyTorch*, *Theano*, dan lain sebagainya (Gain & Hotti, 2021). Menurut riset (Gain & Hotti, 2021), PyCaret memiliki keunggulan karena ia kompatibel dengan *Google Colaboratory* (Colab) dan aplikasi BI (*business intelligence*). PyCaret sendiri memiliki beberapa modul yaitu klasifikasi, regresi, *clustering*, *Association Rule Mining*, deteksi anomali, dan *Natural Language Processing*. Dalam kasus ini, PyCaret digunakan untuk klasifikasi. Versi PyCaret yang digunakan dalam eksperimen ini adalah versi 2.3.1.

HASIL DAN DISKUSI

Pada percobaan pertama, seluruh data digunakan dan pada entri data yang terdapat *missing values* dilakukan imputasi dengan cara mengambil rerata untuk masing-masing atribut. Dalam penelitian ini juga tidak dilakukan penyeimbangan kelas. Selengkapnya, setelan eksperimen pertama dengan PyCaret ditunjukkan pada Tabel 1. Hasil perbandingan performa ke-14 model ditunjukkan pada Tabel 2. Waktu yang diperlukan untuk melakukan keseluruhan eksperimen pada 14 model berkisar antara 20-30 detik dengan spesifikasi komputer i3 *dual-core* 2.0 GHz dan RAM 8 GB.

Tabel 1. Setelan eksperimen PyCaret dengan menggunakan seluruh data dan imputasi untuk data dengan *missing values*

Description	Value
session_id	3082
Target	StatusTerakhir
Target Type	Binary
Label Encoded	DO: 0, Lulus: 1
Original Data	(979, 11)
Missing Values	TRUE
Numeric Features	10
Categorical Features	0
Ordinal Features	FALSE
High Cardinality Features	FALSE
High Cardinality Method	None
Transformed Train Set	(685, 10)
Transformed Test Set	(294, 10)
Shuffle Train-Test	TRUE
Stratify Train-Test	FALSE
Fold Generator	StratifiedKFold
Fold Number	10
CPU Jobs	-1

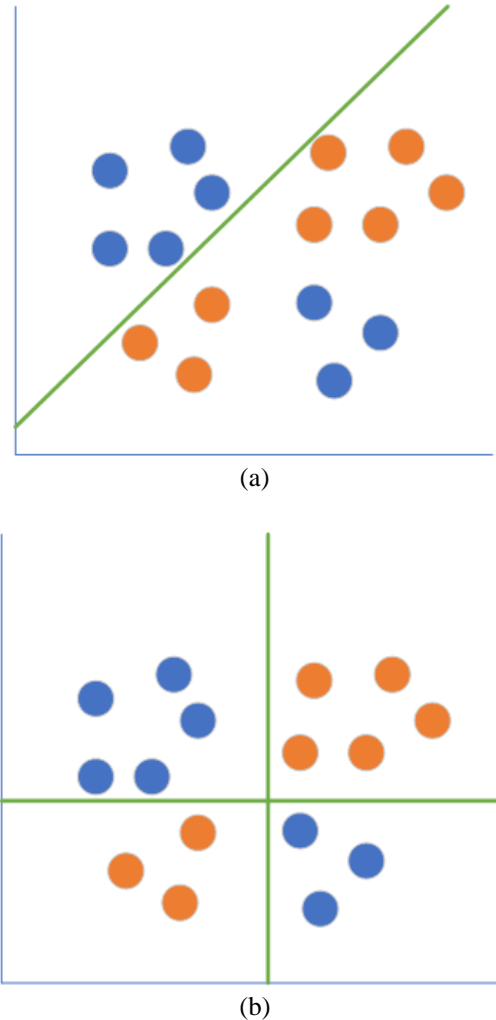
Use GPU	FALSE
Log Experiment	FALSE
Experiment Name	clf-default-name
USI	d7af
Imputation Type	simple
Iterative Imputation Iteration	None
Numeric Imputer	mean
Iterative Imputation Numeric Model	None
Categorical Imputer	constant
Iterative Imputation Categorical Model	None
Unknown Categoricals Handling	least_frequent
Normalize	FALSE
Normalize Method	None
Transformation	FALSE
Transformation Method	None
PCA	FALSE
PCA Method	None
PCA Components	None
Ignore Low Variance	FALSE
Combine Rare Levels	FALSE
Rare Level Threshold	None
Numeric Binning	FALSE
Remove Outliers	FALSE
Outliers Threshold	None
Remove Multicollinearity	FALSE
Multicollinearity Threshold	None
Clustering	FALSE
Clustering Iteration	None
Polynomial Features	FALSE
Polynomial Degree	None
Trigonometry Features	FALSE
Polynomial Threshold	None
Group Features	FALSE
Feature Selection	FALSE
Feature Selection Method	classic
Features Selection Threshold	None
Feature Interaction	FALSE
Feature Ratio	FALSE
Interaction Threshold	None
Fix Imbalance	FALSE
Fix Imbalance Method	SMOTE

Tabel 2. perbandingan performa 14 model ML dengan menyertakan seluruh data dan imputasi *misisng values*

Model	Accu racy	AU C	Re cal l	Pr ec. e.	F1	Ka pp a	M CC
<i>Extreme Gradient Boosting</i>	0.99	0.9	1	0.9	0.9	0.9	0.9
<i>Decision Tree Classifier</i>	71	98	6	8	9	932	93
<i>Random Forest Classifier</i>	0.99	0.9	1	0.9	0.9	0.9	0.9
<i>Extra Trees Classifier</i>	56	94	4	9	8	898	9
<i>Ada Boost Classifier</i>	0.99	0.9	0.9	0.9	0.9	0.9	0.9
<i>Light Gradient Boosting Machine</i>	42	98	6	9	6	865	86
<i>Gradient Boosting Classifier</i>	0.99	0.9	0.9	0.9	0.9	0.9	0.9
<i>K Neighbors Classifier</i>	0.95	0.9	0.9	0.9	0.9	0.8	0.8
<i>Naive Bayes</i>	0.89	0.9	0.8	0.9	0.9	0.7	0.7
<i>Logistic Regression</i>	0.88	0.9	0.9	0.8	0.9	0.7	0.7
<i>Linear Discriminant Analysis</i>	0.87	0.9	0.9	0.8	0.9	0.6	0.7
<i>Ridge Classifier</i>	0.86	0	0.9	0.8	0.9	0.6	0.6
<i>SVM - Linear Kernel</i>	0.81	0	0.8	0.9	0.8	0.5	0.6
<i>Quadratic Discriminant Analysis</i>	0.73	0.6	0.9	0.7	0.8	0.2	0.2

Perlu diperhatikan bahwa hasil eksperimen bisa sedikit berbeda ketika dijalankan pada waktu yang lain (meski perbedaannya sangat kecil) karena beberapa teknik ML seperti *Random Forest* dan *Extreme Gradient Boosting* menggunakan inisiasi acak. Selain itu, data *training* dan data *testing* juga dipilih secara acak. Hasil eksperimen menunjukkan bahwa kasus prediksi *dropout* mahasiswa lebih tepat jika dimodelkan dengan model berbasis pohon keputusan dengan akurasi mencapai 99%. Di sisi lain, model seperti *Linear SVM* dan *QDA* kurang tepat untuk digunakan karena teknik tersebut memiliki jumlah dan bentuk *hyperplane* yang terbatas dalam membagi “ruang” data menjadi kedua kelas – sementara model berbasis pohon keputusan dapat memiliki lebih banyak garis

pembatas yang membuat pemisahan antara kedua kelas menjadi lebih akurat seperti diilustrasikan pada Gambar 2. Dengan demikian, disimpulkan bahwa kasus klasifikasi mahasiswa *dropout* lebih tepat jika diprediksi dengan model berbasis pohon keputusan karena ia lebih sesuai dengan karakteristik masalah.



Gambar 2. Ilustrasi perbandingan garis pemisah (*hyperplane*) (garis warna hijau) pada kasus (a) *Linear SVM* dan (b) Pohon keputusan (*Decision Tree*)

Pola lain yang tampak adalah bahwa teknik *ensemble learning* seperti *Extreme Gradient Boosting* dan *Random Forest* juga memiliki keunggulan dibandingkan dengan teknik “tradisional” seperti *Naïve Bayes*. Namun patut diperhatikan juga bahwa pohon keputusan tunggal juga memiliki akurasi yang sangat baik dan sebanding dengan teknik *ensemble learning*. Hal ini dapat dijelaskan karena jumlah atribut yang tidak cukup banyak (sepuluh atribut) dan jumlah data yang sekitar 1000 yang membuat perbedaan akurasi antara teknik *ensemble decision tree* dan *single decision tree* tidak terpaat banyak.

Perlu diperhatikan bahwa PyCaret melakukan imputasi terhadap data dengan *missing values* karena ia harus menguji 14 model ML sekaligus, sementara sebagian besar teknik ML tidak dapat memproses data dengan *missing values*. Hal ini menjadi masalah dalam eksperimen pertama. Imputasi ini menyebabkan mahasiswa yang DO pada semester 2 yang seharusnya tidak memiliki nilai IPS 3 sampai IPS8 menjadi memiliki nilai IPS untuk semester setelah ia DO. Imputasi semacam ini tentunya tidak tepat untuk dilakukan dalam kasus ini. Percobaan kedua dilakukan dengan hanya menggunakan atribut IPS1 dan IPS2 dan menghilangkan baris data yang memiliki *missing values*. Eksperimen ini sekaligus untuk membandingkan dengan penelitian sebelumnya yang menemukan bahwa atribut yang paling berpengaruh dalam prediksi DO adalah IPS1 dan IPS2 (Muchamad Taufiq Anwar et al., 2021).

Penanganan missing values, penyesuaian atribut, dan parameter tuning

Ketika hanya digunakan dua atribut yaitu IPS1 dan IPS2 dan menghilangkan data dengan *missing values* sehingga menghasilkan 927 data, hasil uji performanya adalah seperti ditunjukkan pada Tabel 3. Hasil eksperimen kedua ini menunjukkan bahwa secara umum nilai akurasi yang didapat lebih rendah dari eksperimen pertama. Hal ini dapat dipahaminya karena pada eksperimen pertama digunakan lebih banyak atribut. Perlu dicatat bahwa pada eksperimen perbandingan model, hal tersebut hanya digunakan untuk mengetahui kemampuan dasar dari masing-masing teknik ML / DM. Hal ini perlu dilanjutkan dengan *parameter tuning* untuk mendapatkan performa terbaik dari sebuah model. *Parameter tuning* ini juga dapat dilakukan dengan PyCaret dengan perintah “*tune model*”. Di sini dilakukan *parameter tuning* untuk beberapa model, yaitu *Logistic Regression* (LR), *Extreme Gradient Boosting* (XGB), *Decision Tree* (DT), dan *SVM - Linear Kernel* (L-SVM). Perbandingan performa keempat model tersebut antara sebelum dan sesudah *parameter tuning* ditunjukkan pada Tabel 4.

Tabel 3. Perbandingan performa 14 model ML dengan menyertakan hanya data lengkap (tanpa *missing values*), tanpa imputasi, dan hanya menggunakan atribut IPS1 dan IPS2

Model	Accuracy	AUC	Recall	Precision	F1	Kappa	MC
Logistic Regression	0.875	0.845	0.927	0.865	0.912	0.625	0.609
Naive Bayes	0.673	0.438	0.687	0.679	0.675	0.128	0.36
Ada Boost Classifier	0.672	0.014	0.729	0.655	0.157	0.085	0.37
Gradient Boosting Classifier	0.642	0.109	0.583	0.713	0.123	0.126	0.321
K Neighbors Classifier	0.611	0.234	0.5	0.738	0.098	0.086	0.248

DOI: 10.52330/jtm.v19i2.34

Linear Discriminant Analysis	0.8	0.8	0.9	0.8	0.9	0.5	0.6
SVM - Linear Kernel	0.8	0	0.9	0.8	0.9	0.5	0.6
Quadratic Discriminant Analysis	0.8	0.8	0.9	0.8	0.9	0.5	0.5
Ridge Classifier	0.8	0	0.9	0.8	0.9	0.5	0.5
Extreme Gradient Boosting	0.8	0.7	0.9	0.8	0.8	0.5	0.5
Random Forest Classifier	0.8	0.8	0.9	0.8	0.8	0.5	0.5
Light Gradient Boosting Machine	0.8	0.8	0.9	0.8	0.8	0.5	0.5
Extra Trees Classifier	0.8	0.7	0.9	0.8	0.8	0.5	0.5
Decision Tree Classifier	0.7	0.7	0.8	0.8	0.8	0.4	0.4

Tabel 4. Perbandingan performa 4 model ML antara model dasar dan model setelah dilakukan *parameter tuning*

Mode l	Base Model			Tuned-Up Model		
	Accuracy	Recall	Precision	Accuracy	Recall	Precision
LR	0.875	0.9854	0.865	0.8765	0.9854	0.8665
XGB	0.8457	0.9271	0.873	0.8657	0.9771	0.861
DT	0.781	0.8353	0.8648	0.8703	0.9812	0.8629
L-SVM	0.8550	0.9750	0.8516	0.8612	0.9771	0.8566

Hasil *parameter tuning* menunjukkan bahwa model *Decision Tree* (DT) yang memiliki akurasi terendah dalam perbandingan model dasar, kini setelah dilakukan *tuning*, akurasi bisa mencapai 0.8703 – sudah setara dengan model terbaik. Sementara model terbaik yaitu *Logistic Regression* ketika di-*tuning* hanya terjadi peningkatan minimal menjadi 0.8765. Begitu pun *Extreme Gradient Boosting* akurasinya hanya naik sedikit menjadi 0.8657. Ini berarti DT dapat menerima manfaat dari *parameter tuning* dibandingkan model lain (bahkan untuk model yang lebih kompleks seperti RF dan XGB sekalipun yang mana mereka seharusnya bisa lebih mengambil manfaat dari *parameter tuning* (M T Anwar et al., 2020; Muchamad Taufiq Anwar, 2021; Hadikurniawati, Anwar, Marlina, & Kusumo, 2021)). Setelah dilakukan *tuning*, hasil akhir akurasi antar model menjadi mirip, yaitu sebesar 87%. Dengan demikian dapat disimpulkan bahwa pengaruh pilihan model terhadap akurasi adalah sangat kecil.

Ketika dilakukan *parameter tuning* pada DT, *Recall* mengalami perbaikan, sementara *precision* tidak. Hal ini berarti bahwa *parameter tuning* pada DT mengakibatkan pengurangan pada prediksi negatif yang salah (*False-*

Negatif) yaitu ketika mahasiswa yang sebenarnya “DO” justru diprediksi sebagai “Lulus”. Dalam kasus seperti prediksi DO ini, *Recall* menjadi metrik yang penting karena ia mengukur keberhasilan model dalam memprediksi secara benar apakah seorang mahasiswa masuk pada kelas DO yang mana hal itu menjadi hal yang perhatian dalam penanganan masalah DO. Hasil pohon keputusan dan *confusion matrix* untuk model DT ditunjukkan pada Gambar 3 dan Gambar 4. Kelas 0 adalah kelas “DO” sementara kelas 1 adalah “Lulus”. Hasil akurasi dari model DT ini sedikit berbeda dari riset sebelumnya karena ada sedikit perbedaan pada data, teknik *splitting* dalam pembuatan pohon keputusan (riset sebelumnya menggunakan *entropy*, sementara riset ini menggunakan *gini index*), dan kedalaman pohon. Hasil *tune-up* untuk model DT ditunjukkan pada Tabel 5. Dari 10 eksperimen untuk *10-fold cross-validation* terlihat bahwa variasi akurasi (ataupun metrik performa lainnya) dari ke-10 model tersebut sangat kecil.

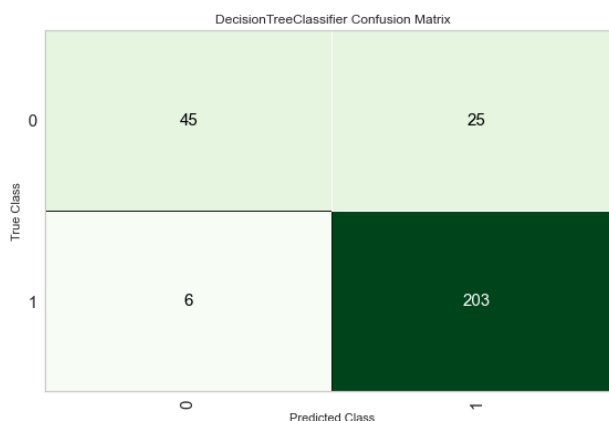
```

|--- feature_0 <= 2.61
|   |--- feature_0 <= 1.74
|   |   |--- class: 0
|   |--- feature_0 > 1.74
|   |   |--- feature_1 <= 2.57
|   |   |   |--- class: 0
|   |   |--- feature_1 > 2.57
|   |   |   |--- class: 1
|--- feature_0 > 2.61
|   |--- feature_0 <= 3.11
|   |   |--- feature_1 <= 2.01
|   |   |   |--- class: 0
|   |   |--- feature_1 > 2.01
|   |   |   |--- class: 1
|   |--- feature_0 > 3.11
|   |   |--- class: 1
    
```

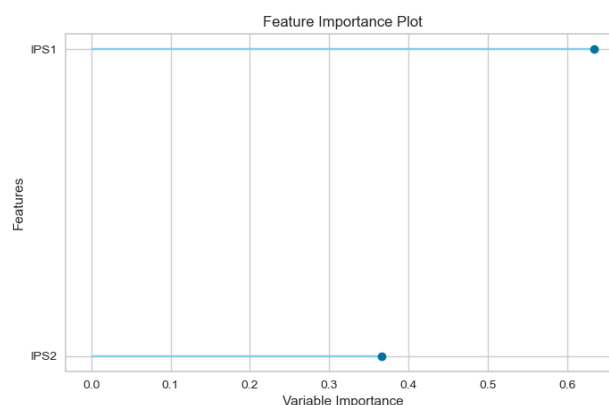
Gambar 3. Hasil pohon keputusan untuk model DT yang telah dilakukan *parameter tuning*

Tabel 4. Perbandingan performa 4 model ML antara model dasar dan model setelah dilakukan *parameter tuning*

Fold	Accu racy	AUC	Reca ll	Prec.	F1	Kap pa	MC C
0	0.876	0.764	1	0.857	0.923	0.624	0.673
	9	7		1	1	3	6
1	0.861	0.754	0.979	0.854	0.912	0.586	0.619
	5	3	2	5	6	6	5
2	0.923	0.852	1	0.905	0.950	0.78	0.799
	1	9		7	5		6
3	0.876	0.764	1	0.857	0.923	0.624	0.673
	9	7		1	1	3	6
4	0.892	0.813	0.979	0.886	0.930	0.691	0.709
	3	1	2	8	7	9	3
5	0.876	0.764	1	0.857	0.923	0.624	0.673
	9	7		1	1	3	6
6	0.876	0.802	0.958	0.884	0.92	0.655	0.665
	9	7	3	6		2	1
7	0.815	0.704	0.937	0.833	0.882	0.460	0.478
	4		5	3	4	6	3
8	0.859	0.760	0.958	0.867	0.910	0.581	0.597
	4	4	3	9	9	4	8
9	0.843	0.705	1	0.824	0.903	0.506	0.582
	8	9		6	8	9	7
Mean	0.870	0.768	0.981	0.862	0.918	0.613	0.647
	3	8	2	9		5	3
SD	0.027	0.043	0.021	0.023	0.016	0.085	0.080
	1	2	8	3	9	1	9



Gambar 4. *Confusion Matrix* untuk model DT



Gambar 5. *Feature importance* untuk model DT

Meski hanya menggunakan dua atribut, model DT terbaik dalam kasus ini memiliki pohon keputusan yang sedikit lebih kompleks dibanding riset Anwar (Muchamad Taufiq Anwar et al., 2021). Dibandingkan dengan riset sebelumnya, titik batas pada IPS1 mirip sedangkan pada IPS2 berbeda. *Feature importance* untuk model DT ditunjukkan pada Gambar 5. Sama seperti riset sebelumnya, IPS1 lebih berpengaruh dalam prediksi DO dibanding IPS2.

Dalam kasus jumlah atribut sangat sedikit, maka performa model cenderung identik. Perbedaan performa baru muncul ketika jumlah atributnya banyak, yang dalam kasus ini hingga menimbulkan perbedaan akurasi sebesar 20% untuk data dengan 10 atribut. Secara umum, performa

model yang baik adalah yang model *ensemble learning* dan berbasis pohon keputusan, sementara model berbasis *hyperplane* seperti Linear SVM dan QDA kurang baik dalam memisahkan kedua kelas dari 10 atribut yang ada. Jadi, dalam kasus klasifikasi, model yang baik adalah model yang mampu memisahkan kedua / beberapa kelas dengan teknik *ensemble learning* maupun pohon keputusan. Hasil eksperimen yang ketiga ini (dengan menggunakan hanya dua atribut dan hanya menggunakan data lengkap, serta dilakukan *parameter tuning*) menunjukkan bahwa berbagai model DM memiliki akurasi yang mirip yaitu 87%. Ketika membandingkan performa beberapa model, riset-riset biasanya akan mendapatkan hasil akurasi yang mirip, sehingga dapat disimpulkan bahwa akurasi prediksi lebih bergantung kepada data daripada pilihan model, seperti yang pernah disebutkan juga dalam riset Anwar (M T Anwar et al., 2020).

Gambar 5 menunjukkan bahwa IPS1 merupakan prediktor DO yang lebih kuat daripada IPS2. Untuk melengkapi eksperimen sebelumnya, dilakukan juga eksperimen tambahan dengan hanya menyertakan satu atribut, yaitu IPS1. Ketika hanya menggunakan atribut IPS1, akurasi model adalah 81,8%. Hasil ini berarti bahwa DO atau tidaknya mahasiswa dapat diprediksi sejak setelah semester 1 selesai / setelah nilai IPS1 diketahui (dengan akurasi 81,8%) atau setelah semester 2 selesai / setelah nilai IPS2 diketahui (dengan akurasi 90,6%).

KESIMPULAN

Penelitian ini bertujuan untuk membandingkan performa teknik data mining yang diterapkan pada kasus prediksi dropout mahasiswa. Perbandingan performa dilakukan menggunakan library PyCaret pada Python menggunakan 14 model / teknik data mining yaitu: *Extreme Gradient Boosting*, *Ada Boost Classifier*, *Light Gradient Boosting Machine*, *Random Forest Classifier*, *Gradient Boosting Classifier*, *Extra Trees Classifier*, *Decision Tree Classifier*, *K Neighbors Classifier*, *Naive Bayes*, *Ridge Classifier*, *Linear Discriminant Analysis*, *Logistic Regression*, *SVM - Linear Kernel*, dan *Quadratic Discriminant Analysis*. Hasil eksperimen menunjukkan bahwa kasus prediksi *dropout* mahasiswa lebih tepat jika dimodelkan dengan model berbasis *ensemble learner* dan pohon keputusan dengan akurasi mencapai 99%. Pohon keputusan memiliki keunggulan dibandingkan model lain seperti *Linear SVM* dan QDA karena ia dapat dengan lebih detil dalam memisahkan data ke dalam kedua kelas target. Setelah dilakukan penyesuaian atribut, pembuangan data dengan *missing values*, dan *parameter tuning*, didapatkan hasil akurasi yang mirip dari berbagai model yaitu sebesar 87%. Perbedaan akurasi antar model menjadi sangat kecil di saat atribut data yang digunakan sedikit. Penelitian berikutnya dapat menambahkan lebih banyak lagi teknik

data mining untuk diperbandingkan performanya dan diterapkan pada data yang memiliki beberapa atribut tambahan yang dapat memprediksi kemungkinan DO seperti misalnya keadaan sosial ekonomi mahasiswa, nilai ketika SMA / sederajat, dan lain sebagainya.

DAFTAR PUSTAKA

- Aguilar-Dominguez, D., Egeh, J., Dunbar, A. D. F., & Brown, S. F. (2021). Machine learning approach for electric vehicle availability forecast to provide vehicle-to-home services. *Energy Reports*, 7, 71–80.
- Ali, M. (2020). *PyCaret: An open source, low-code machine learning library in Python*. Retrieved from <https://www.pycaret.org>
- Anwar, M T, Hadikurniawati, W., Winarno, E., & Widiyatmoko, W. (2020). Performance Comparison of Data Mining Techniques for Rain Prediction Models in Indonesia. *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 83–88. <https://doi.org/10.1109/ISRITI51436.2020.9315460>
- Anwar, Muchamad Taufiq. (2021). Automatic Complaints Categorization Using Random Forest and Gradient Boosting. *Advance Sustainable Science, Engineering and Technology*, 3(1), 210106.
- Anwar, Muchamad Taufiq, Heriyanto, L., & Fanini, F. (2021). Model Prediksi Dropout Mahasiswa Menggunakan Teknik Data Mining. *Jurnal Informatika Upgris*, 7(1).
- Arbain, A. N., & Balakrishnan, B. Y. P. (2019). A comparison of data mining algorithms for liver disease prediction on imbalanced data. *International Journal of Data Science and Advanced Analytics (ISSN 2563-4429)*, 1(1), 1–11.
- Avendano, J. C., Otero, L. D., & Otero, C. (2021). Application of Statistical Machine Learning Algorithms for Classification of Bridge Deformation Data Sets. *2021 IEEE International Systems Conference (SysCon)*, 1–7.
- do Nascimento, C. F., Dos Santos, H. G., de Moraes Batista, A. F., Roman Lay, A. A., Duarte, Y. A. O., & Chiavegatto Filho, A. D. P. (2021). Cause-specific mortality prediction in older residents of São Paulo, Brazil: a machine learning approach. *Age and Ageing*.
- Gain, U., & Hotti, V. (2021). Low-code AutoML-augmented Data Pipeline--A Review and Experiments. *Journal of Physics: Conference Series*, 1828(1), 12015.
- Hadikurniawati, W., Anwar, M. T., Marlina, D., & Kusumo, H. (2021). Predicting tuberculosis drug resistance using machine learning based on DNA sequencing data. *Journal of Physics: Conference Series*, 1869(1), 12093.
- Herho, S. H. S., Fajary, F. R., & Irawan, D. E. (2021). *On the statistical learning analysis of rain gauge data over the Natuna Islands*.
- Hota, H. S., Handa, R., & Shrivastava, A. K. (2021). COVID-19 pandemic in India: Forecasting using machine learning techniques. In *Data Science for COVID-19*

- (pp. 503–525). Elsevier.
- Kuhn, S., & George, N. (2021). Performance of OpenBCI EEG Binary Intent Classification with Laryngeal Imagery. *ArXiv Preprint ArXiv:2107.00045*.
- Larsen, E., Noever, D., MacVittie, K., & Lilly, J. (2021). Overhead-MNIST: Machine Learning Baselines for Image Classification. *ArXiv Preprint ArXiv:2107.00436*.
- Luo, Y., Li, Z., Guo, H., Cao, H., Song, C., Guo, X., & Zhang, Y. (2017). Predicting congenital heart defects: A comparison of three data mining methods. *PloS One*, 12(5), e0177811.
- Mulpuru, V., & Mishra, N. (2021). In Silico Prediction of Fraction Unbound in Human Plasma from Chemical Fingerprint Using Automated Machine Learning. *ACS Omega*, 6(10), 6791–6797.
- Terragni, S., Fersini, E., Galuzzi, B. G., Tropeano, P., & Candelieri, A. (2021). OCTIS: Comparing and Optimizing Topic models is Simple! *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, 263–270.