

CLASSIFICATION OF VOCATIONAL HIGH SCHOOL GRADUATES' ABILITY IN INDUSTRY USING EXTREME GRADIENT BOOSTING (XGBOOST), RANDOM FOREST, AND LOGISTIC REGRESSION

Afikah Agustiningih^{*1}, Yulian Findawati², Irwan Alnarus Kautsar³

^{1,2,3}Informatics, Faculty of Science and Technology, Universitas Muhammadiyah Sidoarjo, Indonesia
Email: ¹afikahagustiningih0@gmail.com, ²yulianfindawati@umsida.ac.id, ³irwan@umsida.ac.id

(Article received: March 05, 2023; Revision: March 26, 2023; published: August 31, 2023)

Abstract

The education world is one of the main sources in producing Human Resources. Vocational High School (SMK) is one level of school that presents various majors that are ready to compete in the industrial world. Therefore, a school institution needs to have a system to determine the quality of education provided to students so that they can compete in the industrial world. This study designs a system that is capable of classifying SMK student graduates as an evaluation for the school institution. The goal is to enable the school to devise strategies for producing better student quality in the following year. There are four classes in this study, namely those who work, those who are not working yet, those who are in college, and those who are entrepreneurs. There are several stages in building the classification system, including pre-processing, processing, and evaluation. This research uses three machine learning algorithms, namely XGBoost, Random Forest, and Logistic Regression. The results of the three methods obtained a training score of 91.70%, a test score of 66.88%, and an accuracy score of 67% generated by the XGBoost algorithm. The Random Forest algorithm produced a training score of 97.36%, a test score of 68.71%, and an accuracy score of 67%. Meanwhile, Logistic Regression produced a training score of 51.14%, a test score of 50.43%, and an accuracy score of 50%.

Keywords: Classification, graduate quality, machine learning, vocational high school (SMK).

KLASIFIKASI KEMAMPUAN LULUSAN SMK DI INDUSTRI MENGGUNAKAN EXTREME GRADIENT BOOSTING (XGBOOST), RANDOM FOREST, DAN LOGISTIC REGRESSION

Abstrak

Dunia pendidikan merupakan salah satu sumber utama dalam menghadirkan Sumber Daya Manusia. Sekolah Menengah Kejuruan (SMK) merupakan salah satu tingkatan sekolah yang menghadirkan berbagai jurusan yang siap untuk bersaing di dunia industri. Oleh karena itu suatu lembaga sekolah perlu mempunyai sistem untuk mengetahui seberapa besar kualitas pendidikan yang diberikan ke peserta didik agar mampu bersaing di dunia industri. Penelitian ini melakukan perancangan sistem yang mampu untuk melakukan klasifikasi lulusan peserta didik SMK sebagai evaluasi bagi pihak lembaga sekolah. Tujuannya adalah agar lembaga sekolah dapat menyusun strategi agar menghasilkan kualitas peserta didik yang lebih baik di tahun berikutnya. Pada penelitian ini terdapat 4 kelas yaitu bekerja, belum bekerja, mahasiswa, dan wirausaha. Terdapat beberapa tahapan dalam membangun sistem klasifikasi diantaranya yaitu tahap preprocessing, processing dan evaluasi. Penelitian ini menggunakan tiga algoritma machine learning yaitu XGBoost, Random Forest, dan Logistic Regression. Hasil dari ketiga metode tersebut mendapatkan skor train 91.70%, skor test 66.88% dan skor akurasi 67% yang dihasilkan oleh algoritma XGBoost, skor train 97.36%, skor test 68.71% dan skor akurasi 67% Oleh Random Forest, dan skor train 51.14%, skor test 50.43% dan skor akurasi 50% Oleh Logistic Regression.

Kata kunci: Klasifikasi, Kualitas lulusan, machine learning, SMK

1. PENDAHULUAN

Sumber Daya Manusia (SDM) merupakan aset yang paling penting dalam kemajuan pembangunan suatu Negara, khususnya pada suatu

organisasi/lembaga pendidikan, sehingga sangat penting agar sumber daya manusia yang ada dapat direncanakan sebaik-baiknya, demi terwujudnya tujuan organisasi atau lembaga. Sumber Daya Manusia disadari sepenuhnya memiliki dampak yang

sangat besar dan dianggap sebagai kunci utama dalam meningkatkan mutu pendidikan [1]. Salah satu penyebab baik tidaknya SDM suatu Negara adalah dengan melihat kualitas pendidikan.

Pendidikan memegang peranan penting dalam kemajuan suatu Negara, dimana hal ini menjadi suatu kepentingan bagi negara yang ingin berkembang, maju serta mampu bersaing secara global [2].

Pemerintah mempersiapkan Sekolah Menengah Kejuruan (SMK) untuk mengatasi permasalahan pengangguran yang ada dan meningkatkan taraf hidup serta mendorong masyarakat untuk memiliki harapan karir yang lebih menonjol [3]. Oleh karena itu diperlukan adanya sistem yang dapat membantu pihak sekolah untuk melakukan klasifikasi lulusan peserta didik sebagai bahan untuk evaluasi lulusan di tahun berikutnya.

Beberapa penelitian terdahulu yang mengkaji mengenai kemampuan lulusan SMK pernah dilakukan sebelumnya. Diantaranya adalah penelitian yang dilakukan oleh Lianny Wydiastuty Kusuma dengan judul “Prediksi kemampuan lulusan SMK untuk dapat bersaing di dunia kerja dengan menggunakan Naïve Bayes: Studi kasus SMK Buddhi Tangerang”. Pengujian ini menggunakan confusion matrix dan kurva ROC dengan target berupa belum mampu bersaing dan mampu bersaing. Berdasarkan hasil evaluasi dan validasi yang telah dilakukan, algoritma naïve bayes memiliki akurasi dan performa baik dengan nilai akurasi 98% dan nilai AUC 0,980 [4].

Penelitian yang dilakukan oleh Yufika Septiani, Pipin Farida Ariyani dengan judul “Penerapan Algoritma Naïve Bayes Menentukan Klasifikasi Tingkat Kelulusan Siswa SMK Media Informatika Jakarta” pada tahun 2022. Data yang digunakan adalah data arsip nilai pihak sekolah sebanyak 200 data. Hasil dari penelitian ini mendapatkan perhitungan persentase tingkat akurasi sebesar 90%, recall sebesar 94,44 dan presisi sebesar 94,4% [5].

Penelitian yang dilakukan oleh Esty Purwaningsih, Ela Nurelasari dengan judul “Penerapan K-Nearest Neighbor untuk klasifikasi tingkat kelulusan pada siswa” pada tahun 2021. Data yang diuji menggunakan metode K-Nearest Neighbor dan dataset dibagi menjadi 2 yaitu data training dan data testing menggunakan fold cross validation. Dalam penelitian ini menggunakan rapidminer sebagai pengolahan data. Hasil dari prediksi tingkat kelulusan siswa dengan KNN mendapatkan rata-rata akurasi sebesar 96,49% [6].

Untuk melakukan klasifikasi, penelitian ini menggunakan tiga algoritma machine learning diantaranya adalah Extreme Gradient Boosting (XGBoost), Random Forest dan Logistic Regression. XGBoost merupakan salah satu algoritma machine learning yang mampu mengatasi permasalahan regresi dan klasifikasi berdasarkan Boosting Decision Tree (GBDT) serta dapat membangun boosted trees secara efisien dan beroperasi secara parallel. Pada

dasarnya algoritma ini merupakan metode ensemble yang didasarkan pada gradient boosting tree [7]. Umumnya, metode boosting bekerja dengan cara membuat model baru secara bertahap dengan mempertimbangkan kesalahan model sebelumnya, lalu menambahkan model baru dengan model yang ada. Hal ini dilakukan untuk meningkatkan kompleksitas model akhir [8].

Random Forest merupakan algoritma machine learning dalam decision tree atau pohon keputusan yang digunakan untuk pengklasifikasian dataset dalam jumlah besar atau big data. Algoritma ini digunakan untuk membuat pohon keputusan yang terdiri dari root node, internal node, dan leaf node dengan memilih atribut dan kata secara acak sesuai dengan aturan yang diterapkan. Root node digunakan untuk mengumpulkan data, Internal node yang terdapat di root node berisi pertanyaan tentang data, sedangkan leaf node digunakan untuk memecahkan masalah serta membuat keputusan [9].

Sedangkan Algoritma Logistic Regression merupakan jenis analisis statistik yang sering digunakan untuk pemodelan prediktif. Algoritma ini memprediksi pada saat variable dependen (y) atau output suatu data berupa biner atau berbagai opsi lain [10].

Berdasarkan latar belakang yang telah dipaparkan diatas, penelitian ini mengkaji tentang sebuah sistem yang dapat melakukan klasifikasi lulusan SMK dengan empat kelas yaitu bekerja, belum bekerja, mahasiswa dan wirausaha. Sehingga penelitian ini diharapkan dapat membantu pihak lembaga sekolah untuk melakukan evaluasi dan membuat strategi agar lulusan peserta didik menjadi lebih baik di tahun berikutnya.

2. METODE PENELITIAN

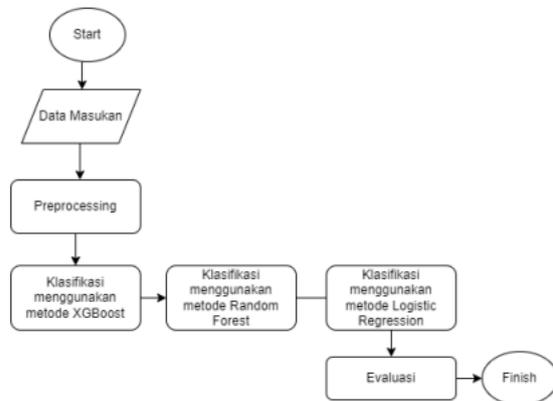
2.1. Tahapan Penelitian

Tahapan penelitian merupakan gambaran umum mengenai alur penelitian yang akan dilakukan. Gambar 1 merupakan alur penelitian yang digunakan untuk klasifikasi dari ketiga metode penelitian yang dimulai dari menginputkan data yang sudah diperoleh dan melakukan *preprocessing* untuk membuat model data dan setelah itu melakukan pengklasifikasian oleh ketiga metode yaitu metode XGBoost, Random Forest dan Logistic Regression. Pada tahapan terakhir yaitu mengevaluasi menggunakan confusion matrix untuk pengukuran akurasi dari masing-masing metode.

2.2. Pengumpulan Data

Data yang digunakan pada penelitian merupakan data credential yang diambil dari SMK Negeri 1 Bangil. Data tersebut terdiri dari 1126 record dengan beberapa indikator diantaranya adalah tahun kelulusan, jurusan, USBN pengetahuan, USBN keterampilan, rata-rata rapor, NUS dan NS.

Sedangkan data status dari tracer study merupakan target perbandingan hasil klasifikasi. Penelitian ini menggunakan data peserta didik kelas 12 tahun ajaran 2020 dan 2021. Berikut merupakan sample dataset yang digunakan dalam penelitian.



Gambar 1. Alur Penelitian

Tabel 1. Sample Data Penelitian

TK	JR	UP	UK	RR	NU	NS	ST
2020	MM	88.8	90.5	85	90	87	B
2020	MM	86.6	83	84	84	84	B
2020	MM	88.2	94.25	86	92	88	B
2020	PSPT	93	89	85	90	87	B
2020	PSPT	94.4	88.88	85	91	87	B
2020	PSPT	93	90.4	87	91	88	M
2020	RPL	93	90.5	88	91	89	B
2020	RPL	95	91.5	89	93	90	W
2020	TB	90	86.8	84	88	85	W
2020	TB	95	90.4	87	92	89	B
2020	TEI	80	90	87	88	87	M
2020	TEI	80	95.25	89	91	90	B
2020	TITL	97.8	90.70	86	89	86	M
2020	TITL	98	90.73	88	86	85	B
2020	TKJ	88.8	85.6	79	87	81	B
2020	TO	83	84.3	82	84	83	B
2021	MM	80	81	83	81	82	M
2021	PSPT	90	89	84	89	86	B
2021	RPL	88	82.5	82	84	83	B
2021	TB	83	83.75	83	84	83	W

Tabel 1. menunjukkan kategori nilai dari tahun ajaran 2020 dan 2021. Berikut merupakan keterangan atribut yang digunakan dalam tabel tersebut :

- TK : Tahun Kelulusan
- JR : Jurusan
- UP : USBN Pengetahuan
- UK : USBN Keterampilan
- RR : Rata-rata Rapor
- NU : Nilai Ujian Sekolah
- NS : Nilai Semester
- ST : Status/Target

Berikut merupakan keterangan dari kolom atribut status/target yang digunakan pada tabel 1.

- B : Bekerja
- BK : Belum Bekerja
- M : Mahasiswa
- W : Wirausaha

2.3. Preprocessing

Data mentah tidak dapat dilakukan *processing* secara langsung. Perlu adanya tahapan *preprocessing* terlebih dahulu. Tahap *preprocessing* bertujuan untuk memodifikasi data guna meningkatkan kualitas data yang digunakan. Terdapat beberapa sub tahapan pada tahap *preprocessing*, diantaranya adalah encoding categorical data, handling *imbalanced* data, normalisasi data dengan minmax scaler.

2.3.1. Encoding Categorical Data

Encoding Categorical Data merupakan tahapan yang harus dilakukan jika data yang digunakan terdapat data yang berjenis kategorik. Pada penelitian ini, peneliti menggunakan teknik one hot encoder untuk melakukan encoding data nominal. Data nominal merupakan data yang menghasilkan satu jenis kategori saja. Selain itu, peneliti juga melakukan teknik label encoder untuk melakukan encoding data ordinal. Data ordinal merupakan data yang memiliki urutan atau tingkatan tertentu. Biasanya urutan ini dapat berupa dari yang terendah hingga tertinggi atau sebaliknya [11].

2.3.2. SMOTE

Ketika melakukan klasifikasi data, salah satu masalah yang sering terjadi adalah ketidakseimbangan jumlah data antara kelas yang berbeda. Apabila ketidakseimbangan tersebut sangat ekstrim, maka masalah ini dikenal dengan sebutan *rare* atau *imbalanced* data [7].

Synthetic Minority Oversampling Technique (SMOTE) merupakan metode yang diterapkan untuk menangani ketidakseimbangan kelas. Teknik ini bekerja dengan cara menyeimbangkan jumlah distribusi data sampel pada kelas minoritas dengan menyeleksi data sampel tersebut sampai jumlah data sample menjadi seimbang dengan jumlah sampel pada kelas mayoritas [12].

2.3.3. Normalisasi Data

Beberapa dataset terdapat perbedaan rentang nilai pada setiap atribut. Perbedaan ini dapat mengakibatkan atribut yang memiliki nilai jauh lebih kecil dibandingkan dengan atribut lainnya menjadi tidak berfungsi. Oleh karena itu, diperlukan *transformasi* data dengan melakukan normalisasi agar rentang nilai pada setiap atribut dapat disamakan dengan skala tertentu [13]. Pada penelitian ini peneliti menggunakan teknik minmax scaler dimana teknik ini

merupakan teknik normalisasi data yang mengubah nilai numerik dalam dataset ke skala umum tanpa mendistorsi perbedaan dalam rentang nilai. Rentang nilai yang biasa digunakan adalah 0 hingga 1. Bila dituliskan kedalam rumus *minmax scaler* terlihat pada persamaan 1.

$$X_{new} = \frac{X_{old} - X_{min}}{X_{max} - X_{min}} \quad (1)$$

Dimana :

X_{new} : Nilai data yang telah dinormalisasi.

x_{old} : Nilai data pada atribut sebelum dilakukan normalisasi.

x_{min} : Nilai terkecil pada atribut.

x_{max} : Nilai terbesar pada atribut.

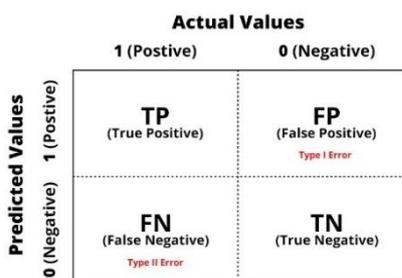
2.4. Processing

Setelah melalui seluruh tahapan *preprocessing*, tahapan selanjutnya yaitu *processing* dimana pada tahapan ini dilakukan penerapan algoritma machine learning. Sebelum melakukan penerapan algoritma machine learning, peneliti melakukan pembagian data menjadi dua bagian yaitu data train dan data test dengan persentase 80% untuk data train dan 20% untuk data test.

Untuk mendapatkan parameter yang paling optimal, peneliti menggunakan teknik *randomized search cross validation* dimana teknik ini dapat melakukan *hyperparameter* tuning lebih cepat dibandingkan *grid search cross validation* [14]. Setelah mendapatkan parameter terbaik, selanjutnya dilakukan fitting model dengan tiga algoritma machine learning diantaranya yaitu XGBoost, Random Forest dan Logistic Regression.

2.5. Evaluasi

Pada tahapan evaluasi, peneliti menggunakan confusion matrix untuk mengukur performa model yang telah dibuat. Confusion matrix berbentuk tabel matrix yang menggambarkan kinerja model klasifikasi pada serangkaian data uji yang sebenarnya diketahui. Confusion matrix digunakan untuk membandingkan kelas prediksi dengan kelas data yang sebenarnya [15]



Gambar 2. Confusion Matrix

Pada Gambar 2. Pengukuran confusion matrik menggunakan rumus *accuracy*, *recall*, *precision* dan *micro average F1-score*.

1. *Accuracy* digunakan untuk mengevaluasi performa dari suatu model atau algoritma dalam melakukan prediksi atau klasifikasi. Berikut merupakan rumus *accuracy* dalam persamaan 2:

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \quad (2)$$

2. *Recall* digunakan untuk mengevaluasi seberapa baik ketepatan jumlah prediksi terhadap suatu kelas yang berhasil diprediksi dengan benar dari total jumlah data yang memiliki label kelas tersebut. Berikut merupakan rumus *recall* dalam persamaan 3 :

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

3. *Precision* digunakan untuk mengevaluasi seberapa baik ketepatan jumlah prediksi dengan benar. Berikut merupakan rumus *precision* dalam persamaan 4 :

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

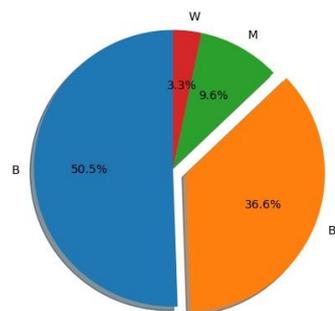
4. *Micro Average F1-Score* adalah kombinasi pengukuran terhadap *precision* dan *recall* yang digunakan untuk mengukur kombinasi nilai yang telah dihasilkan. Berikut merupakan rumus *F1-score* dalam persamaan 5 :

$$F1 - score = 2 \times \frac{Prec \times Rec}{Prec + Rec} \quad (5)$$

3. HASIL DAN PEMBAHASAN

3.1. Dataset Analysis

Sebelum mengawali proses machine learning, langkah awal yang perlu dilakukan adalah dengan menganalisis dataset yang akan digunakan. Hal ini akan membantu memahami karakteristik dan kualitas data. Langkah pertama yang dilakukan pada penelitian ini yaitu mengelompokkan data perjurusan dengan target sehingga memudahkan untuk mengetahui perbandingan kelas tiap jurusan. Selanjutnya pengelompokan semua jurusan pada tahun 2020-2021 untuk dibandingkan dengan target sehingga menghasilkan persentase perbandingan kelas.



Gambar 3. Visualisasi Perbandingan Dataset

Pada Gambar 3. Menunjukkan bahwa lulusan yang bekerja mendapatkan persentase paling tinggi yaitu 50.5%, belum bekerja mendapat persentase 36.6%, mahasiswa mendapat persentase 9.6% dan wirausaha mendapatkan persentase 3.3%.

3.2. Data Preprocessing

Pada tahapan *preprocessing* atau modelling data diantaranya yaitu:

3.2.1. One Hot Encoder dan Label Encoder

Data jurusan merupakan data kategorikal, pada *preprocessing* ini akan digunakan teknik one hot encoder untuk mengubah data kategorikal menjadi vector biner atau one-hot, dimana masing-masing fitur hanya dapat bernilai 0 atau 1 dan hanya satu fitur yang bernilai 1 untuk setiap data.

Tabel 2. One Hot Encoder

Status	MM	PSPT	RPL	TB
B	1	0	0	0
BK	1	0	0	0
BK	1	0	0	0
B	0	1	0	0
B	0	1	0	0
M	0	1	0	0
B	0	0	1	0
W	0	0	1	0
W	0	0	0	1
B	0	0	0	1
BK	0	0	0	1

Tabel 2. Menunjukkan bahwa kolom jurusan MM bernilai '1' maka jurusan lainnya akan bernilai '0', begitupun seterusnya.

Data status yang awalnya merupakan data kategorikal, pada *preprocessing* data digunakan teknik label encoder untuk mengubah data kategorikal menjadi angka yang unik. Dikarenakan data status ini memiliki jenjang maka data status di ubah menjadi data ordinal.

Tabel 3. Label Encoder

Status	MM	PSPT	RPL	TB
0	1	0	0	0
1	1	0	0	0
1	1	0	0	0
0	0	1	0	0
0	0	1	0	0
2	0	1	0	0
0	0	0	1	0
3	0	0	1	0
3	0	0	0	1
0	0	0	0	1
1	0	0	0	1

Pada Tabel 3. Menunjukkan data status 'bekerja' atau 'B' di ubah menjadi '0', status 'belum bekerja' atau 'BK' menjadi '1', status 'mahasiswa' atau 'M' menjadi '2', dan status 'wirausaha' atau 'W' menjadi '3'.

3.2.2. Handling Imbalanced Data

Berdasarkan Gambar 3. Visualisasi perbandingan dataset dapat diketahui bahwa terdapat ketidakseimbangan jumlah kelas, artinya data tidak seimbang atau terdapat suatu kondisi dimana sebuah himpunan data terdapat satu kelas yang memiliki jumlah kecil dibandingkan dengan jumlah kelas yang lain. Oleh karena itu diperlukan adanya tahapan handling imbalanced data.

Tahapan ini melakukan SMOTE dimulai dengan menghitung jarak antara data pada data minoritas, kemudian menentukan nilai persentase SMOTE dan terakhir menetapkan jumlah data terdekat [12].

Tabel 4. Perbandingan Imbalanced Data

Before	Counter :	{{0:569,1:412,2:10
Oversampling		8,3:37}}
After	Counter :	{{0:569,1:569,2:56
Oversampling		9,3:569}}

Pada Tabel 4. Menunjukkan setelah dilakukan handling *imbalanced* data dengan SMOTE, jumlah data pada tiap kelas menjadi sama, sehingga mengatasi ketidakseimbangan jumlah data pada tiap kelas sebelumnya.

3.2.3. MinMax Scaler

Tabel 5. Hasil Minmax Scaler

TK	UP	UK	RR	NU	NS
0	1	0.7	0.7	0.6	0.6
0	0.3	0.8	0.7	0.7	0.8
1	0.7	0.5	0.6	0.6	0.6
0	1	0.8	0.7	0.6	0.7

Tabel 5. menunjukkan bahwa dalam proses normalisasi, nilai x old dikurangi dengan nilai x min, hasilnya dibagi dengan selisih nilai x max dan x min. Hal ini menghasilkan nilai baru yang berada pada rentang 0 hingga 1, sehingga nilai-nilai pada atribut memiliki skala yang sama dan dapat dibandingkan dengan mudah.

3.3. Klasifikasi Algoritma

Sebelum dilakukan klasifikasi, dilakukan pembagian data antara data X dan data Y yang mana data X merupakan kolom fitur dan data Y merupakan kolom target. Setelah melakukan pembagian data X

dan Y, dilakukan pembagian data train dan data test pada data X menggunakan modul scikit-learn yaitu `train_test_split`.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,
y, test_size=0.20, random_state=5)
```

Gambar 4. Pembagian dataset

Pada Gambar 4. Pembagian data dibagi menjadi 2 dengan persentase data train 80% dan data test 20%. Selanjutnya melakukan modelling menggunakan algoritma XGBoost, Random Forest dan Logistic Regression dengan parameter tuning.

Tabel 6. XGBoost Hyperparameter Tuning

Hyperparameter	Value
<code>algo__max_depth</code>	Integer(low=1, high=10)
<code>algo__learning_rate</code>	Real(low=-2, high=0, prior='log-uniform')
<code>algo__n_estimators</code>	Integer(low=100, high=200)
<code>algo__subsample</code>	Real(low=0.3, high=0.8, prior='uniform')
<code>algo__gamma</code>	Integer(low=1, high=10)
<code>algo__colsample_bytree</code>	Real(low=0.1, high=1, prior='uniform')
<code>algo__reg_alpha</code>	Real(low=-3, high=1, prior='log-uniform')
<code>algo__reg_lambda</code>	Real(low=-3, high=1, prior='log-uniform')

Tabel 7. Random Forest Hyperparameter Tuning

Hyperparameter	Value
<code>algo__n_estimators</code>	Integer(low=100, high=200)
<code>algo__max_depth</code>	Integer(low=20, high=80)
<code>algo__max_features</code>	Real(low=0.1, high=1, prior='uniform')
<code>algo__min_samples_leaf</code>	Integer(low=1, high=20)

Tabel 8. Logistic Regression Hyperparameter Tuning

Hyperparameter	Value
<code>algo__fit_intercept</code>	[True, False]
<code>algo__C</code>	Real(low=-3, high=3, prior='log-uniform')

Pada Tabel 6, Tabel 7 dan Tabel 8. Merupakan parameter tuning yang disediakan oleh *library jcompil* dengan mengimport fungsi *random_search_params*.

Randomized Search Cross Validation adalah bagian dari modul scikit-learn yang bertujuan secara otomatis dan sistematis melakukan validasi beberapa model dan setiap *hyperparameter*. Ketika proses *running* randomized search cross validation sudah selesai, maka akan didapatkan model beserta skor

train dan skor test. Proses tuning dan modelling ditunjukkan pada tabel 9, tabel 10 dan tabel 11 dibawah ini :

Tabel 9. Proses Tuning dan Modelling Metode XGBoost

```
numerical_pipeline = Pipeline([
('scaler', MinMaxScaler())
])
preprocessor = ColumnTransformer([
('numeric', numerical_pipeline, X_train.columns)
])
pipeline = Pipeline([
('prep', preprocessor),
('algo', XGBClassifier())
])
model = RandomizedSearchCV(pipeline, rsp.xgb_params,
cv=5, n_iter = 50, n_jobs=-1, verbose=-1, random_state=42)
model.fit(X_train, y_train)
print(model.score(X_train,
y_train),model.best_score_,model.score(X_test, y_test))
print(model.score(X_train,y_train)*100, model.score(X_test,
y_test)*100)
```

Tabel 10. Proses Tuning dan Modelling Metode Random Forest

```
numerical_pipeline = Pipeline([
('imputer', SimpleImputer(strategy='mean')),
('scaler', MinMaxScaler())
])
preprocessor = ColumnTransformer([
('numeric', numerical_pipeline, X_train.columns)
])
pipeline = Pipeline([
('prep', preprocessor),
('algo', RandomForestClassifier())
])
model = RandomizedSearchCV(pipeline, rsp.rf_params, cv=5,
n_iter = 50, n_jobs=-1, verbose=-1, random_state=42)
model.fit(X_train, y_train)
print(model.score(X_train,
y_train),model.best_score_,model.score(X_test, y_test))
print(model.score(X_train,y_train)*100, model.score(X_test,
y_test)*100)
```

Tabel 11. Proses Tuning dan Modelling Metode Logistic Regression

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.20, random_state=5)
numerical_pipeline = Pipeline([
('imputer', SimpleImputer(strategy='mean')),
('scaler', MinMaxScaler())
])
preprocessor = ColumnTransformer([
('numeric', numerical_pipeline, X_train.columns)
])
```

```

pipeline = Pipeline([
    ('prep', preprocessor),
    ('algo', LogisticRegression())
])
model = RandomizedSearchCV(pipeline, rsp_rf_params, cv=5,
n_iter = 50, n_jobs=-1, verbose=-1, random_state=42)
model.fit(X_train, y_train)
print(model.score(X_train,
y_train),model.best_score_,model.score(X_test, y_test))
print(model.score(X_train,y_train)*100, model.score(X_test,
y_test)*100)
    
```

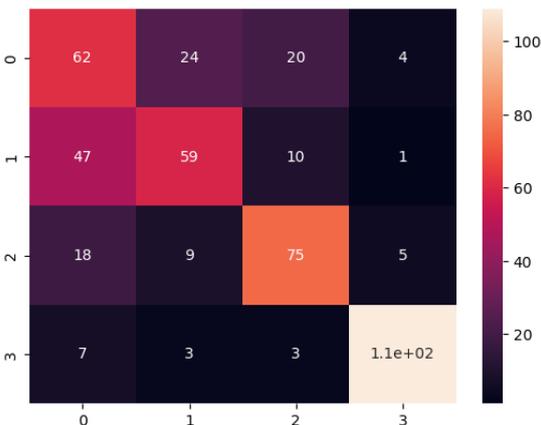
Tabel 12. Perbandingan Best Skor

Algoritma	Score Train	Score Test
XGBoost	91.70%	66.88%
Random Forest	97.36%	68.71%
Logistic Regression	51.14%	50.43%

Berdasarkan Tabel 12. Dapat diketahui bahwa terdapat skor test yang paling tinggi yaitu 68.71% yang dihasilkan oleh algoritma Random Forest. Namun dari hasil tersebut mengalami overfitting dengan gap antara skor test dan skor train yang tinggi yaitu 28.65%.

3.4. Evaluasi

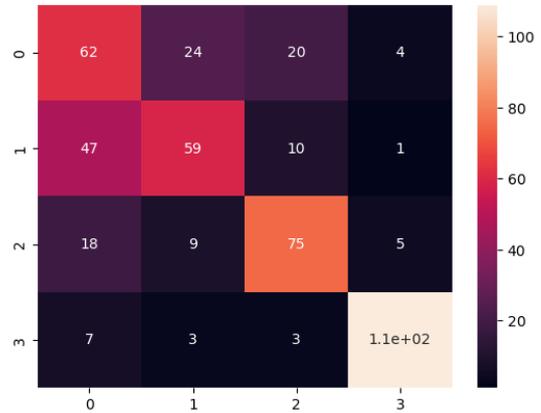
Setelah proses klasifikasi selesai, dilakukan evaluasi model untuk mengukur performa model. Dalam hal ini peneliti menggunakan confusion matrix.



Gambar 5. Confusion Matrix Algoritma XGBoost

Pada Gambar 5 dan Gambar 6. merupakan hasil dari pengukuran model confusion matrix pada algoritma XGBoost dan Random Forest. Kedua algoritma tersebut mendapatkan hasil pengukuran model yang sama. Untuk kelas 0 terdapat 62 data yang diklasifikasikan dengan benar, 24 data diklasifikasikan sebagai kelas 1, 20 data diklasifikasikan sebagai kelas 2 dan 4 data diklasifikasikan sebagai kelas 3

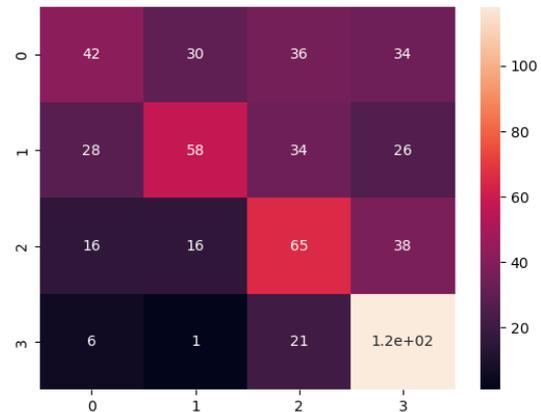
Untuk kelas 1 terdapat 59 data yang diklasifikasikan dengan benar, 47 data diklasifikasikan sebagai kelas 0, 10 data diklasifikasikan sebagai kelas 2, dan 1 data diklasifikasikan sebagai kelas 3.



Gambar 6. Confusion Matrix Algoritma Random Forest

Untuk kelas 2 terdapat 75 data yang diklasifikasikan dengan benar, 18 data diklasifikasikan sebagai kelas 0, 9 data diklasifikasikan sebagai kelas 1 dan 5 data diklasifikasikan sebagai kelas 3.

Untuk kelas 3 terdapat 110 data yang diklasifikasikan dengan benar, 7 data diklasifikasikan sebagai kelas 0, 3 data diklasifikasikan sebagai kelas 1 dan 3 data diklasifikasikan sebagai kelas 2.



Gambar 7. Confusion Matrix Algoritma Logistic Regression

Pada Gambar 7. merupakan hasil dari pengukuran model confusion matrix pada algoritma Logistic Regression. Untuk kelas 0 terdapat 42 data yang diklasifikasikan dengan benar, 30 data diklasifikasikan sebagai kelas 1, 36 data diklasifikasikan sebagai kelas 2 dan 34 data diklasifikasikan sebagai kelas 3

Untuk kelas 1 terdapat 58 data yang diklasifikasikan dengan benar, 28 data diklasifikasikan sebagai kelas 0, 34 data diklasifikasikan sebagai kelas 2, dan 26 data diklasifikasikan sebagai kelas 3.

Untuk kelas 2 terdapat 65 data yang diklasifikasikan dengan benar, 16 data diklasifikasikan sebagai kelas 0, 16 data diklasifikasikan sebagai kelas 1 dan 38 data diklasifikasikan sebagai kelas 3.

Untuk kelas 3 terdapat 120 data yang diklasifikasikan dengan benar, 6 data diklasifikasikan sebagai kelas 0, 1 data diklasifikasikan sebagai kelas 1 dan 21 data diklasifikasikan sebagai kelas 2.

Tabel 13. Evaluasi Perbandingan Confusion Matrix

Algoritma	Accuracy	precision	recall	F1 score
XGBoost	67%	67%	67%	67%
Random Forest	67%	67%	67%	67%
Logistic Regression	50%	50%	50%	50%

Pada tabel 13. dapat diketahui hasil dari pengukuran tiga metode machine learning dalam confusion matrix skor akurasi, precision, recall dan F1-score paling tinggi yaitu 67% yang diperoleh dari algoritma XGBoost dan Random Forest, sedangkan untuk algoritma Logistic Regression mendapatkan skor 50% untuk skor akurasi, precision, recall dan F1-score.

4. DISKUSI

Dari ke tiga penelitian terdahulu yang sudah dipaparkan pada bab pendahuluan, peneliti melakukan penelitian yang sama dengan penelitian yang dilakukan oleh Lianny Wydiastuty Kusuma dengan judul "Prediksi kemampuan lulusan SMK untuk dapat bersaing di dunia kerja dengan menggunakan naive bayes : Study kasus SMK Buddhi Tangerang". Pada penelitian ini menggunakan 100 dataset.

Pada penelitian yang dilakukan oleh Yufika Septiani, Pipin Farida Ariyani dengan judul "Penerapan Algoritma Naïve Bayes Menentukan Klasifikasi Tingkat Kelulusan Siswa SMK Media Informatika Jakarta" pada tahun 2022. Data yang digunakan dalam penelitian ini sebanyak 200 data.

Serta penelitian yang dilakukan oleh Esty Purwaningsih, Ela Nurelasari dengan judul "Penerapan K-Nearest Neighbor untuk klasifikasi tingkat kelulusan pada siswa" pada tahun 2021. Penelitian ini menggunakan dataset sebanyak 104 data dengan atribut sebanyak 5.

Pada ketiga penelitian terdahulu tersebut mendapatkan hasil akurasi yang termasuk kategori tinggi. Hal ini disebabkan oleh beberapa faktor, salah satunya adalah kualitas dataset yang digunakan dan pemilihan algoritma serta parameter yang tepat, namun penelitian tersebut memiliki beberapa keterbatasan seperti jumlah data yang relative kecil sehingga perlu dilakukan penelitian lebih lanjut dengan data yang lebih banyak untuk memperoleh hasil yang lebih *reliable*.

Penelitian ini jelas memiliki perbedaan yang cukup banyak sehingga terdapat keterbaruan dari penelitian sebelumnya. Penelitian ini menggunakan tiga algoritma machine learning diantaranya yaitu XGBoost, Random Forest dan Logistic Regression. Dalam hal ini bertujuan sebagai perbandingan. Parameter tuning yang digunakan adalah randomized search cross validation dan evaluasi pengukuran menggunakan confusion matrix.

Penelitian ini terdapat empat kelas yaitu bekerja, belum bekerja, mahasiswa dan wirausaha sehingga dapat dikatakan sebagai klasifikasi multiclass.

5. KESIMPULAN

Berdasarkan penelitian yang sudah dilakukan, maka dapat ditarik kesimpulan bahwa :

1. Teknik SMOTE membantu mengatasi permasalahan pada ketidakseimbangan kelas yang terjadi saat *preprocessing*.
2. Randomized Search Cross Validation bekerja dengan baik untuk menemukan parameter terbaik dari ketiga metode penelitian.
3. Pengujian dilakukan menggunakan confusion matrix dengan dataset sebanyak 1126 dengan pembagian data train 80% dan data test 20%.
4. Algoritma XGBoost dan Random Forest mendapatkan hasil akurasi yang sama yaitu 67%. sedangkan Logistic Regression menghasilkan skor akurasi 50%. dari ketiga metode ini mengalami overfitting.

UCAPAN TERIMA KASIH

Puji dan syukur kami panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, kami dapat menyelesaikan penelitian ini. Peneliti juga mengucapkan terimakasih kepada Prodi Informatika Universitas Muhammadiyah Sidoarjo yang telah memberikan bimbingan dan fasilitas yang dibutuhkan dalam melakukan penelitian ini. Peneliti juga ingin berterimakasih kepada BRIN yang telah memberikan dukungan dana untuk melakukan penelitian ini. Dan peneliti mengucapkan terimakasih kepada SMKN 1 Bangil yang telah memberikan kesempatan dan membantu dalam melakukan pengumpulan data untuk penelitian ini.

DAFTAR PUSTAKA

- [1] Y. Sangsurya, M. Muazza, and R. Rahman, "Perencanaan Sumber Daya Manusia Dalam Peningkatan Mutu Pendidikan Di Sd Islam Mutiara Al Madan Kota Sungai Penuh," *J. Manaj. Pendidik. Dan Ilmu Sos.*, vol. 2, no. 2, pp. 766–778, 2021, doi: 10.38035/jmpis.v2i2.644.
- [2] S. A. Nurfatimah, S. Hasna, and D. Rostika, "Membangun Kualitas Pendidikan di Indonesia dalam Mewujudkan Program Sustainable Development Goals (SDGs)," *J.*

- Basicedu*, vol. 6, no. 4, pp. 6145–6154, 2022, doi: 10.31004/basicedu.v6i4.3183.
- [3] H. Priyono, R. Sari, and T. Mardiana, “Klasifikasi Pemilihan Jurusan Sekolah Menengah Kejuruan Menggunakan Gradient Boosting Classifier,” *J. Inform.*, vol. 9, no. 2, pp. 131–139, 2022, doi: 10.31294/inf.v9i2.12654.
- [4] L. W. Kusuma, “Prediksi Kemampuan Lulusan SMK untuk Dapat Bersaing Di Dunia Kerja dengan Menggunakan Naïve Bayes: Studi Kasus SMK Buddhi Tangerang,” *Prediksi Kemamp. Lulusan SMK untuk Dapat Bersaing Di Dunia Kerja dengan Menggunakan Naïve Bayes Stud. Kasus SMK Buddhi Tangerang*, vol. 1, pp. 56–63, 2019.
- [5] Y. Septiani and P. F. Ariyani, “Penerapan Algoritma Naive Bayes Menentukan Klasifikasi Tingkat Kelulusan Siswa SMK Media Informatika Jakarta Application of The Naive Bayes Algorithm Determining Classification of Students’ Graduation Level of Jakarta Media Informatika Vocational School,” no. September, pp. 607–613, 2022.
- [6] E. Purwaningsih and E. Nurelasari, “Penerapan K-Nearest Neighbor Untuk Klasifikasi Tingkat Kelulusan Pada Siswa,” *Syntax J. Inform.*, vol. 10, no. 01, pp. 46–56, 2021, doi: 10.35706/syji.v10i01.5173.
- [7] I. Muslim and K. Karo, “Implementasi Metode XGBoost dan Feature Importance untuk Klasifikasi pada Kebakaran Hutan dan Lahan,” *J. Softw. Eng. Inf. Commun. Technol.*, vol. 1, no. 1, pp. 10–16, 2020.
- [8] M. Rizky Mubarak, Muliadi, and R. Herteno, “Hyper-Parameter Tuning pada XGBoost Untuk Prediksi Keberlangsungan Hidup Pasien Gagal Jantung,” *Kumpul. J. Ilmu Komput.*, vol. 9, no. 2, pp. 391–401, 2022.
- [9] P. R. Sihombing and I. F. Yuliati, “Penerapan Metode Machine Learning dalam Klasifikasi Risiko Kejadian Berat Badan Lahir Rendah di Indonesia,” *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 20, no. 2, pp. 417–426, 2021, doi: 10.30812/matrik.v20i2.1174.
- [10] A. K. Santoso, A. Noviriandini, A. Kurniasih, B. D. Wicaksono, and A. Nuryanto, “Klasifikasi Persepsi Pengguna Twitter Terhadap Kasus Covid-19 Menggunakan Metode Logistic Regression,” *JIK (Jurnal Inform. dan Komputer)*, vol. 5, no. 2, pp. 234–241, 2021.
- [11] A. Tjalla and M. Mahdiyah, “Data Kategorik dalam Penelitian : Review Bibliometrik,” vol. 9, no. 1, pp. 796–802, 2023, doi: 10.58258/jime.v9i1.4814/http.
- [12] A. N. Kasanah, M. Muladi, and U. Pujiyanto, “Penerapan Teknik SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Objektivitas Berita Online Menggunakan Algoritma KNN,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 3, no. 2, pp. 196–201, 2019, doi: 10.29207/resti.v3i2.945.
- [13] D. A. Nasution, H. H. Khotimah, and N. Chamidah, “Perbandingan Normalisasi Data untuk Klasifikasi Wine Menggunakan Algoritma K-NN,” *Comput. Eng. Sci. Syst. J.*, vol. 4, no. 1, p. 78, 2019, doi: 10.24114/cess.v4i1.11458.
- [14] E. Agustin, A. Eviyanti, and N. L. Azizah, “Deteksi Penyakit Epilepsi Melalui Sinyal EEG Menggunakan Metode DWT dan Extreme Gradient Boosting,” vol. 7, pp. 117–127, 2023, doi: 10.30865/mib.v7i1.5412.
- [15] M. Noveanto, H. Sastypratiwi, and H. Muhardi, “Uji Akurasi Klasifikasi Emosi Pada Lirik Lagu Bahasa Indonesia Emotion Classification Accuracy Test in Indonesian Song Lyrics,” vol. 10, no. 3, pp. 311–318, 2022, doi: 10.26418/justin.v10i3.56804.