

# Klasifikasi Performa Akademik Siswa Menggunakan Multinomial Logistic Regression dan Support Vector Machine Multiclass

**Nama:** Ilham Akbar

## 1. Pendahuluan

### 1.1 Latar Belakang

Dalam dunia pendidikan modern, pemanfaatan data untuk memahami dan memprediksi performa akademik siswa menjadi semakin penting. Dengan semakin berkembangnya sistem pembelajaran digital dan pelaporan terstruktur, institusi pendidikan kini memiliki akses ke beragam data siswa yang dapat dianalisis guna mendukung pengambilan keputusan berbasis bukti. Salah satu pendekatan yang populer dalam menganalisis data semacam ini adalah penggunaan algoritma machine learning untuk klasifikasi tingkat performa siswa.

Studi ini memanfaatkan dataset xAPI-Edu-Data.csv yang terdiri dari 480 observasi siswa dengan berbagai fitur yang mencerminkan karakteristik demografis, aktivitas pembelajaran, dan interaksi siswa dengan sistem pembelajaran digital. Target variabel dalam penelitian ini adalah Class, yaitu kategori performa akademik siswa yang diklasifikasikan menjadi tiga kelas: rendah (L), sedang (M), dan tinggi (H).

Untuk melakukan klasifikasi multikelas terhadap variabel target tersebut, digunakan dua metode machine learning yang umum: Regresi Logistik Multinomial dan Support Vector Machine (SVM) Multiclass. Regresi Logistik bekerja dengan menghitung probabilitas keanggotaan kelas menggunakan fungsi logit, sementara SVM

memisahkan kelas menggunakan hyperplane optimal dengan pendekatan One-vs-Rest serta pemanfaatan kernel nonlinier seperti RBF.

Proses pembangunan model dilakukan dengan mengikuti praktik terbaik, termasuk validasi silang 5-fold untuk menghindari overfitting, serta tuning parameter melalui Grid Search guna mengoptimalkan performa model. Selain itu, dilakukan analisis interpretabilitas model melalui eksplorasi fitur penting, untuk memahami variabel mana yang paling berpengaruh dalam proses klasifikasi.

Dengan pendekatan ini, penelitian bertujuan untuk mengevaluasi efektivitas kedua model dalam memprediksi performa akademik siswa, serta memberikan rekomendasi model yang paling sesuai untuk implementasi di lingkungan pendidikan. Selain menilai akurasi dan keseimbangan prediksi, aspek efisiensi dan interpretabilitas model juga menjadi pertimbangan utama dalam analisis.

---

## 1.2 Rumusan Masalah

Dalam upaya meningkatkan pemahaman terhadap pemodelan klasifikasi performa akademik siswa, penelitian ini mengangkat beberapa pertanyaan utama yang menjadi dasar analisis. Adapun rumusan masalah yang ingin dijawab adalah sebagai berikut:

1. Bagaimana model Regresi Logistik Multinomial dan SVM Multiclass dapat digunakan untuk memprediksi performa akademik siswa (kategori L, M, H)?
2. Model mana yang memberikan performa klasifikasi terbaik?
3. Model mana yang memberikan performa prediksi yang lebih seimbang pada semua kelas?
4. Model mana yang lebih efisien dari sisi komputasi

## 1.3 Tujuan

Penelitian ini bertujuan untuk membandingkan model Regresi Logistik Multinomial dan SVM Multiclass dalam memprediksi performa akademik siswa (kategori Low, Middle, High). Tujuan khusus meliputi:

1. Menganalisis bagaimana model Regresi Logistik Multinomial dan SVM Multiclass dapat diterapkan dalam mengklasifikasikan tingkat performa akademik siswa berdasarkan atribut-atribut yang tersedia dalam dataset pendidikan, seperti tingkat kehadiran, partisipasi, latar belakang siswa, dan aktivitas belajar lainnya.
2. Membandingkan performa klasifikasi dari kedua model berdasarkan metrik evaluasi seperti akurasi, precision, recall, dan f1-score untuk menilai efektivitas masing-masing algoritma dalam memetakan kategori performa siswa.
3. Menilai keseimbangan prediksi kedua model terhadap seluruh kelas target, dengan menekankan pada evaluasi macro average dan distribusi f1-score per kelas, guna mengidentifikasi model yang paling adil dalam mengenali kategori L, M, dan H.
4. Mengevaluasi efisiensi komputasi dari masing-masing model, termasuk waktu pelatihan dan kompleksitas tuning parameter, untuk menentukan algoritma mana yang lebih hemat sumber daya dan waktu dalam implementasi praktis.

## 2. Metodologi

### 2.1 Data dan Deskripsinya

Dataset yang digunakan adalah xAPI-Edu-Data.csv yang terdiri dari 480 data siswa. Berikut adalah deskripsi tiap variabel:

- gender: Jenis kelamin (M, F) NationalITy:
- Kewarganegaraan siswa
- PlaceofBirth: Tempat lahir
- StageID: Jenjang pendidikan (lowerlevel, MiddleSchool, HighSchool)
- GradeID: Tingkatan kelas (G-01 hingga G-12)
- SectionID: Kode kelas (A, B, C)
- Topic: Mata pelajaran (Math, IT, dll.)
- Semester: Semester pembelajaran (F, S) Relation:
- Hubungan wali (Father, Mum) raisedhands: Jumlah mengangkat tangan (interaksi)
- VisITedResources: Kunjungan ke sumber belajar
- AnnouncementsView: Akses pengumuman
- Discussion: Partisipasi diskusi online
- ParentAnsweringSurvey: Orang tua mengisi survei (Yes, No)
- ParentschoolSatisfaction: Kepuasan orang tua (Good, Bad)
- StudentAbsenceDays: Ketidakhadiran (Under-7, Above-7) Class:
- Target variabel performa siswa (L, M, H)

Variabel Target akan dikodekan menjadi angka:

- $L \rightarrow 0$  (Low)
- $M \rightarrow 1$  (Middle)
- $H \rightarrow 2$  (High)

---

## 2.2 Regresi Logistik Multinomial

Logistic Regression merupakan salah satu metode klasifikasi yang umum digunakan dalam machine learning untuk memodelkan hubungan antara variabel prediktor dengan variabel target kategorik. Untuk kasus klasifikasi multikelas, digunakan pendekatan multinomial logistic regression (softmax regression) yang memperluas regresi logistik biner agar mampu memetakan input ke lebih dari dua kelas (Hosmer et al., 2013).

Model ini bekerja dengan mengestimasi probabilitas setiap kelas berdasarkan fungsi logit dan memilih kelas dengan probabilitas tertinggi sebagai prediksi. Logistic Regression banyak digunakan dalam konteks pendidikan karena interpretasinya yang sederhana dan kemampuannya menangani data kategorik dan numerik secara bersamaan (Zhang, 2004).

---

## 2.3 SVM Multiclass

SVM adalah algoritma pembelajaran mesin berbasis margin yang bekerja dengan mencari hyperplane optimal yang memisahkan kelas-kelas data. Untuk menangani klasifikasi multikelas, digunakan pendekatan One-vs-Rest (OvR), di mana satu model SVM dilatih untuk membedakan satu kelas terhadap gabungan semua kelas lainnya (Rifkin & Klautau, 2004).

Selain itu, untuk mengatasi hubungan non-linier antar fitur, digunakan fungsi kernel, seperti radial basis function (rbf). Kernel ini memungkinkan pemetaan data ke ruang berdimensi lebih tinggi di mana pemisahan antar kelas lebih mungkin dilakukan (Cortes & Vapnik, 1995). SVM dikenal memiliki performa tinggi pada dataset dengan dimensi besar dan sangat cocok untuk klasifikasi yang kompleks.

---

## 2.4 Validasi Model (Cross-Validation)

Untuk memastikan bahwa model yang dibangun memiliki generalizability yang baik terhadap data baru, digunakan teknik validasi silang (cross-validation). Dalam studi ini digunakan 5fold cross-validation, di mana data dibagi menjadi lima bagian, dan proses pelatihan dan pengujian dilakukan sebanyak lima kali, masing-masing dengan satu bagian sebagai data uji dan empat sisanya sebagai data latih (Kohavi, 1995).

Teknik ini membantu menghindari overfitting dan memberikan estimasi performa model yang lebih stabil dibanding hanya menggunakan satu pembagian data. Cross-validation juga merupakan praktik yang umum dalam evaluasi model machine learning modern.

---

## 2.5 Eksperimen Variasi Parameter (Tuning)

Untuk mengoptimalkan performa kedua model, digunakan teknik Grid Search, yaitu metode pencarian sistematis terhadap kombinasi parameter terbaik berdasarkan performa model melalui cross-validation. Proses ini dilakukan untuk masingmasing model:

1. Logistic Regression: disesuaikan parameter seperti nilai regularisasi (C), jenis solver ('lbfgs', 'liblinear'), dan jumlah iterasi maksimum (max\_iter).
2. SVM: diuji kombinasi parameter seperti C (regularisasi), gamma (untuk kernel RBF), serta jenis kernel (linear, rbf).

Grid Search dikombinasikan dengan cross-validation, sehingga setiap kombinasi parameter diuji pada lima subset berbeda, memastikan hasil yang optimal dan tidak bias terhadap pembagian data tertentu (Bergstra & Bengio, 2012).

---

## 2.6 Eksplorasi Fitur Penting

Setelah diperoleh model terbaik dari proses tuning, langkah berikutnya adalah menganalisis fitur mana yang paling berkontribusi terhadap keputusan klasifikasi.

Pada Logistic Regression, koefisien dari model digunakan sebagai ukuran pentingnya fitur. Nilai absolut koefisien dihitung dan dinormalisasi untuk mengetahui pengaruh relatif tiap fitur terhadap prediksi.

Pada SVM, digunakan pendekatan Permutation Importance, yaitu mengacak nilai suatu fitur dan mengukur dampaknya terhadap

performa model. Fitur yang menyebabkan penurunan akurasi besar dianggap penting (Breiman, 2001; Molnar, 2022).

Pendekatan interpretasi model ini penting untuk memahami bagaimana model mengambil keputusan dan memberikan wawasan tambahan mengenai variabel yang relevan dalam konteks klasifikasi pada data pendidikan.

### 3. Kode Implementasi Awal

Berikut adalah implementasi kode tahapan analisis, beserta

#### 3.1 Memuat Library dan Dataset

Langkah awal dari analisis adalah memuat pustaka (library) Python yang diperlukan dan membaca dataset. Hal ini penting untuk memastikan semua fungsi siap digunakan dan data dapat diproses dengan benar.

```
# Import Library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder, StandardScale
from sklearn.model_selection import train_test_split, GridSea
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_
from sklearn.inspection import permutation_importance
from sklearn.pipeline import make_pipeline

# Load Dataset
```

	Gender	NationalITY	PlaceofBirth	StageID	GradeID
0	M	KW	KuwaIT	lowerlevel	G-04

6/18/25, 12:22 PM

Multinomial Logistic Regression & SVM Multiclass Clasification - Colab

df =	1	M	KW	KuwaIT	lowerlevel	G-04
	2	M	KW	KuwaIT	lowerlevel	G-04
	3	M	KW	KuwaIT	lowerlevel	G-04
	4	M	KW	KuwaIT	lowerlevel	G-04

pd.read\_csv('/content/xAPI-Edu-Data.csv') df.head()

Next steps:

Generate code withdf

View recommended plots

New

### 3.2 Eksplorasi dan Pra-pemrosesan Data

Tahapan ini dimulai dengan meninjau struktur data untuk memastikan tidak ada nilai kosong dan mengidentifikasi tipe data. Sebagian besar fitur bertipe kategorik, sehingga dilakukan Label Encoding agar dapat digunakan dalam model. Data kemudian dipisahkan menjadi fitur (X) dan target (y), lalu dibagi menjadi data latih dan uji. Terakhir, fitur numerik dinormalisasi menggunakan StandardScaler untuk memastikan skala yang seragam dalam proses pelatihan model.

```
# Cek info df.info()
df['Class'].value_counts()

# Encode semua fitur kategorik categorical_cols =
df.select_dtypes(include='object').columns df_encoded =
df.copy()

# Label Encoding untuk semua kolom kategorik
label_encoders = {} for col in
categorical_cols:
```




```
le = LabelEncoder()      df_encoded[col] =
le.fit_transform(df_encoded[col])
label_encoders[col] = le

# Feature & Target
X = df_encoded.drop('Class', axis=1)
y = df_encoded['Class']

# Split data latih dan uji
X_train, X_test, y_train, y_test = train_test_split(X, y, tes

# Normalisasi fitur numerik
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```



<class 'pandas.core.frame.DataFrame'>				
RangeIndex: 480 entries, 0 to 479 Data				
columns (total 17 columns):				
#	Column	Non-Null Count		Dtype
---	-----	-----		-----
0	gender	480	non-null	object
1	NationalITy	480	non-null	object
2	PlaceofBirth	480	non-null	object
3	StageID	480	non-null	object
4	GradeID	480	non-null	object
5	SectionID	480	non-null	object
6	Topic	480	non-null	object
7	Semester	480	non-null	object
8	Relation	480	non-null	object
9	raisedhands	480	non-null	int64
10	VisITedResources	480	non-null	int64
11	AnnouncementsView	480	non-null	int64
12	Discussion	480	non-null	int64
13	ParentAnsweringSurvey	480	non-null	object
14	ParentschoolSatisfaction	480	non-null	object
15	StudentAbsenceDays	480	non-null	object
	Class	480	non-null	object
dtypes: int64(4), object(13) memory usage: 63.9+ KB				

Dataset terdiri dari 480 observasi dan 17 fitur, tanpa adanya nilai kosong, yang berarti data dalam kondisi lengkap dan siap untuk diproses lebih lanjut. Sebanyak 13 fitur memiliki tipe data kategorik (bertipe object), seperti gender, NationalITy, dan Class, sedangkan 4

fitur lainnya merupakan data numerik bertipe int64, yaitu raisedhands, VisITedResources, AnnouncementsView, dan Discussion.

### 3.3 Cross-Validation Model Dasar

Tahapan ini dilakukan untuk memperoleh gambaran awal performa model Logistic Regression dan SVM sebelum dilakukan tuning. Metode cross-validation 5-fold diterapkan agar evaluasi lebih stabil dan tidak bias terhadap pembagian data tertentu.

```
# Model dasar lr_model = make_pipeline(StandardScaler(),  
LogisticRegression svm_model =  
make_pipeline(StandardScaler(), SVC())  
  
# Cross-validation lr_cv =  
cross_val_score(lr_model, X, y, cv=5) svm_cv =  
cross_val_score(svm_model, X, y, cv=5)  
  
print("Akurasi Logistic Regression (CV):", lr_cv.mean())  
print("Akurasi SVM (CV):", svm_cv.mean())
```



```
Akurasi Logistic Regression (CV): 0.7  
Akurasi SVM (CV): 0.6541666666666666
```

Berdasarkan hasil, Logistic Regression menghasilkan akurasi rata-rata sebesar 70%, sedangkan SVM menghasilkan akurasi rata-rata sebesar 65,4%. Perbedaan ini menunjukkan bahwa model Logistic Regression secara default memiliki kemampuan yang lebih baik dalam menangkap pola klasifikasi pada data dibandingkan SVM, walaupun keduanya masih memiliki potensi untuk ditingkatkan lebih lanjut melalui tuning parameter.

### 3.4 Grid Search untuk Tuning Parameter

Pada tahap ini, dilakukan proses tuning parameter menggunakan teknik Grid Search untuk mencari kombinasi hiperparameter terbaik bagi masing-masing model, yaitu Logistic Regression dan Support Vector Machine (SVM). Grid Search memanfaatkan validasi silang

(cross-validation) untuk menguji setiap kombinasi parameter dan memilih konfigurasi yang menghasilkan performa terbaik pada data latih. Untuk Logistic Regression, parameter yang diuji meliputi nilai regularisasi C, metode solver, dan jumlah iterasi maksimum. Sementara itu, SVM diuji dengan variasi nilai C, jenis kernel (linear dan rbf), serta parameter gamma. Hasil dari grid search menunjukkan konfigurasi optimal yang akan digunakan pada proses pelatihan akhir dan evaluasi model.

```
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC

# Grid Search Logistic Regression
lr_params = {
    'C': [0.01, 0.1, 1, 10],
    'solver': ['lbfgs', 'newton-cg'],
    'max_iter': [200, 500]
}

# Grid Search SVM
svm_params = {
    'C': [0.1, 1, 10],
    'kernel': ['linear', 'rbf'],
    'gamma': ['scale', 'auto']
}

# Grid search lr_grid = GridSearchCV(LogisticRegression(),
lr_params, cv=5) svm_grid =
GridSearchCV(SVC(probability=True), svm_params, cv

# Training
lr_grid.fit(X_train, y_train)
svm_grid.fit(X_train, y_train)

print("Best Params Logistic Regression:", lr_grid.best_params_
print("Best Params SVM:", svm_grid.best_params_)
```



Best Params Logistic Regression: {'C': 10, 'max\_iter': 20  
Best Params SVM: {'C': 1, 'gamma': 'scale', 'kernel': 'r

Grid search menghasilkan parameter terbaik bagi masingmasing model.

Logistic Regression bekerja optimal dengan  $C=10$ , solver lbfgs, dan  $\text{max\_iter}=200$ , yang menunjukkan preferensi terhadap regularisasi lemah dan efisiensi komputasi.

Sementara itu, SVM mencapai performa terbaik dengan  $C=1$ , kernel rbf, dan  $\text{gamma}=\text{'scale'}$ , kombinasi yang seimbang untuk menangani pola non-linear secara otomatis. Parameterparameter ini digunakan untuk pelatihan model akhir.

## 3.5 Evaluasi Akhir dan Confusion Matrix

Setelah model Logistic Regression dan SVM disempurnakan melalui Grid Search, langkah selanjutnya adalah mengevaluasi kinerja keduanya terhadap data uji. Evaluasi ini dilakukan dengan dua pendekatan utama:

1. Laporan klasifikasi (classification report) yang mencakup metrik seperti akurasi, precision, recall, dan F1-score untuk masing-masing kelas.
2. Confusion Matrix, yang memberikan gambaran visual mengenai jumlah prediksi benar dan salah pada setiap kelas target (High, Middle, Low).

Analisis ini bertujuan untuk melihat seberapa baik model mengklasifikasikan data yang belum pernah dilihat sebelumnya, dan apakah model cenderung bias terhadap salah satu kelas tertentu. Hasil evaluasi ini juga menjadi dasar untuk membandingkan performa akhir antara kedua algoritma.

```
# Prediksi y_pred_lr =  
lr_grid.predict(X_test) y_pred_svm =  
svm_grid.predict(X_test)  
  
# Laporan Klasifikasi print("Logistic Regression:\n",  
classification_report(y_test, print("SVM:\n",  
classification_report(y_test, y_pred_svm))  
  
# Confusion Matrix  
fig, ax = plt.subplots(1, 2, figsize=(12, 5))  
ConfusionMatrixDisplay.from_predictions(y_test, y_pred_lr, ax  
ax[0].set_title("Confusion Matrix Logistic Regression")  
  
ConfusionMatrixDisplay.from_predictions(y_test, y_pred_svm, a  
ax[1].set_title("Confusion Matrix SVM")  
  
plt.tight_layout()  
plt.show()
```



Logistic Regression:  
f1-score      support

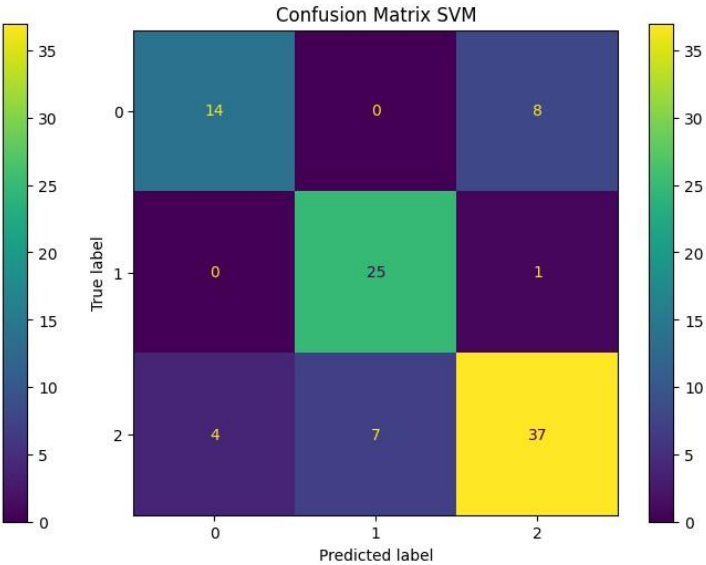
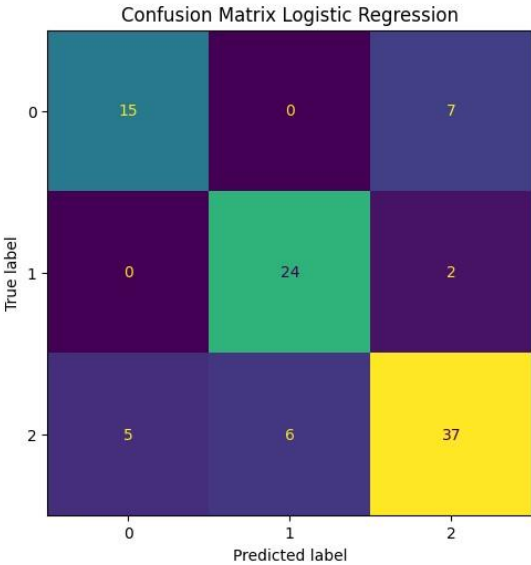
precision      recall

0	0.75	0.68	0.71	22	
1	0.80	0.92	0.86	26	2
	0.80	0.77	0.79	48	
accuracy				0.79	96
macro avg		0.78	0.79	0.79	96
weighted avg		0.79	0.79	0.79	96

SVM:

precision      recall      f1-score  
support

0	0.78	0.64	0.70	22	
1	0.78	0.96	0.86	26	2
	0.80	0.77	0.79	48	
accuracy				0.79	96
macro avg		0.79	0.79	0.78	96
weighted avg		0.79	0.79	0.79	96



Evaluasi akhir dilakukan untuk menilai performa prediktif dari dua model: Logistic Regression dan Support Vector Machine (SVM). Hasil menunjukkan bahwa keduanya mencapai akurasi yang sama, yaitu 79%, menandakan kemampuan klasifikasi yang cukup baik secara keseluruhan.

Pada kelas 1 (Middle), kedua model tampil sangat baik dengan nilai recall tinggi—92% pada Logistic Regression dan 96% pada SVM—yang menunjukkan bahwa sebagian besar siswa dalam kategori ini berhasil dikenali dengan benar. Sementara itu, kelas 0 (Low) menjadi kelas yang paling sulit diprediksi, dengan recall hanya 68% dan 64% masing-masing pada Logistic Regression dan SVM. Kelas 2 (High)

menunjukkan performa yang seimbang pada kedua model, dengan f1-score sekitar 0.79.

Confusion matrix memperlihatkan bahwa sebagian besar kesalahan terjadi antara kelas 0 dan 2, kemungkinan karena pola fitur yang mirip. Logistic Regression memberikan hasil yang lebih seimbang antar kelas, sedangkan SVM sedikit lebih tajam dalam mengenali kelas dominan. Dengan performa yang hampir setara, pemilihan model akhir bisa disesuaikan dengan kebutuhan:

Logistic Regression unggul dalam interpretabilitas, sedangkan SVM lebih kuat dalam menangani data kompleks dan non-linier.

## 3.6 Eksplorasi Fitur Penting

Tahap analisis yang terakhir, kita ingin memahami pengaruh masing-masing fitur terhadap keputusan klasifikasi yang dihasilkan oleh kedua model, Logistic Regression dan SVM.

1. Untuk Logistic Regression, kita memanfaatkan nilai koefisien regresi dari masing-masing kelas target. Koefisien ini menggambarkan arah dan kekuatan hubungan fitur terhadap peluang masuk ke setiap kelas. Semakin besar nilai absolut koefisien, semakin besar kontribusi fitur tersebut. Untuk mendapatkan pengaruh global sebuah fitur, dihitung rata-rata dari nilai absolut koefisien fitur tersebut di semua kelas. Agar bisa dibandingkan dengan metode lain, nilai importance ini kemudian dinormalisasi terhadap nilai maksimum, menghasilkan skor antara 0–1.
2. Sedangkan karena SVM tidak memiliki koefisien eksplisit maka digunakanlah metode Permutation Importance. Pendekatan ini menghitung penurunan performa model saat satu fitur diacak nilainya. Jika performa turun drastis, maka fitur tersebut dianggap penting. Skor importance juga dinormalisasi terhadap

skor maksimum, agar skala sebanding dengan Logistic Regression.

Analisis ini bermanfaat untuk menafsirkan model secara lebih transparan, serta memberikan wawasan tambahan terkait faktor-faktor penting dalam menentukan kelas output pada dataset pendidikan ini.

```
# 1. Eksplorasi Fitur Penting: Logistic Regression

# Ambil koefisien dari model logistic regression terbaik
coeffs = pd.DataFrame(
    lr_grid.best_estimator_.coef_,      columns=X.columns )

# Ambil absolute mean coef untuk tiap fitur dari logit
logit_coef_mean = np.mean(np.abs(coeffs), axis=0)

# 2. Eksplorasi Fitur Penting: SVM dengan Permutation Importa

# Hitung permutation importance untuk SVM
```



```
result = permutation_importance(
    svm_grid.best_estimator_,
    X_test,      y_test,
    n_repeats=10,
    random_state=42,      n_jobs=-1
)

# Simpan hasil importance
svm_importances = pd.Series(result.importances_mean, index=X
svm_importances_sorted = svm_importances.sort_values()

print("Koefisien Logistic Regression:")
display(coeffs)

print("Tabel Permutation Importance untuk SVM:")
display(svm_importances.sort_values(ascending=False))

# Normalisasi agar skala bisa dibandingkan
logit_coef_norm = logit_coef_mean / np.max(logit_coef_mean)
svm_importance_norm = svm_importances / np.max(svm_importance

# Gabungkan comparison_df =
pd.DataFrame({
    'Fitur': X.columns,
    'Logistic_Importance': logit_coef_norm,
    'SVM_Importance': svm_importance_norm
}).set_index('Fitur')

# Visualisasi gabungan
comparison_df.sort_values('Logistic_Importance', ascending=Fa
plt.title("Perbandingan Feature Importance: Logistic Regressi
plt.xlabel("Skor Normalisasi")
plt.tight_layout()
plt.show()

# Ambil Top 5 Fitur dari Logistic Regression dan SVM
top5_logit = comparison_df['Logistic_Importance'].sort_values
top5_svm = comparison_df['SVM_Importance'].sort_values(ascend
```



Koefisien Logistic Regression:

**gender NationalITy PlaceofBirth**

**StageID**

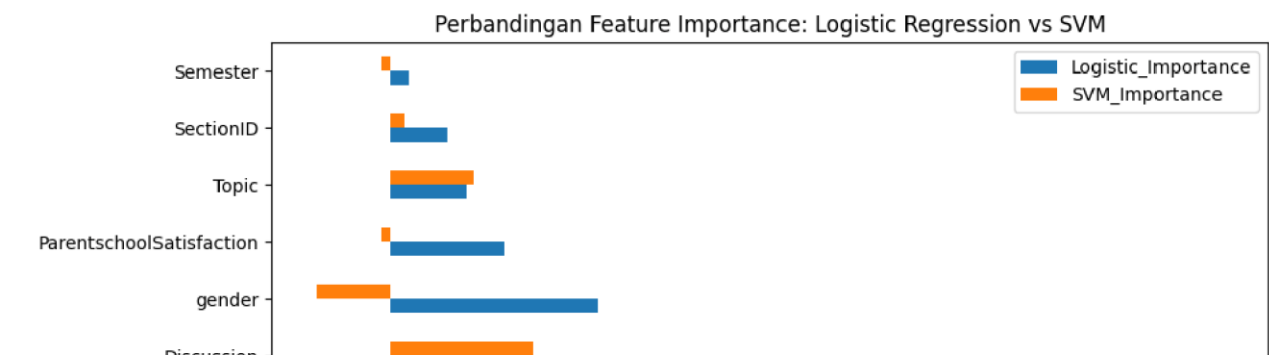
**Grade**

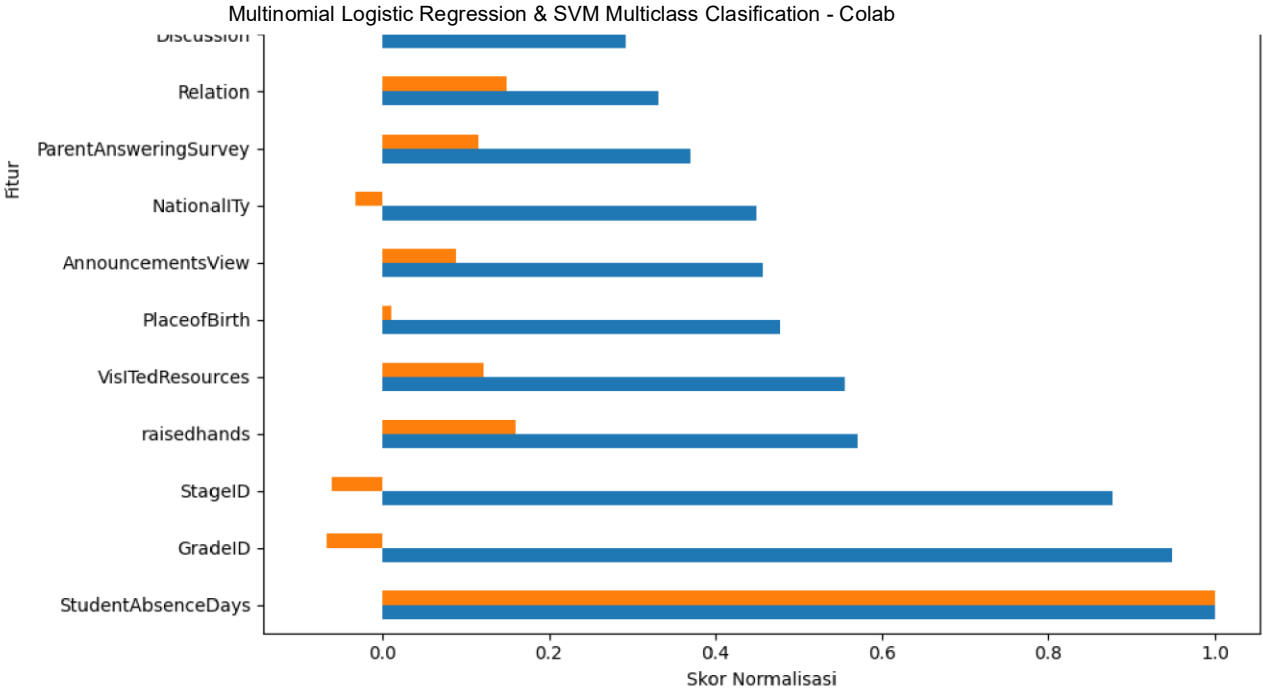
0	-0.345939	-0.473760	0.522826	-0.815922	-0.9721
1	0.344778	0.624746	-0.664574	1.218555	1.3174
2	0.001161	-0.150987	0.141748	-0.402633	-0.3452

Tabel Permutation Importance untuk SVM:

	0
StudentAbsenceDays	0.188542
Discussion	0.032292
raisedhands	0.030208
Relation	0.028125
VisITedResources	0.022917
ParentAnsweringSurvey	0.021875
Topic	0.018750
AnnouncementsView	0.016667
SectionID	0.003125
PlaceofBirth	0.002083
Semester	-0.002083
ParentschoolSatisfaction	-0.002083
NationalITy	-0.006250
StageID	-0.011458
GradeID	-0.012500
gender	-0.016667

dtype: float64





Next  
steps:

[Generate code with coeffs](#)[View recommended plots](#)

```
print(" Top 10 Fitur Terpenting versi Logistic Regression (mu
for i, (feat, val) in enumerate(top5_logit.items(), 1):
print(f"{i}. {feat} (skor: {val:.3f})")

print("\n Top 10 Fitur Terpenting versi SVM (Permutation Impo
for i, (feat, val) in enumerate(top5_svm.items(), 1):
    print(f"{i}. {feat} (skor: {val:.3f})")
```



Top 10 Fitur Terpenting versi Logistic Regression (mult

1. StudentAbsenceDays (skor: 1.000)
2. GradeID (skor: 0.948)
3. StageID (skor: 0.877)
4. raisedhands (skor: 0.571)
5. VisITedResources (skor: 0.555)
6. PlaceofBirth (skor: 0.478)
7. AnnouncementsView (skor: 0.457)
8. NationalITy (skor: 0.450)
9. ParentAnsweringSurvey (skor: 0.370)
10. Relation (skor: 0.331)

Top 10 Fitur Terpenting versi SVM (Permutation Importan

1. StudentAbsenceDays (skor: 1.000)
2. Discussion (skor: 0.171)
3. raisedhands (skor: 0.160)
4. Relation (skor: 0.149)
5. VisITedResources (skor: 0.122)
6. ParentAnsweringSurvey (skor: 0.116)
7. Topic (skor: 0.099)
8. AnnouncementsView (skor: 0.088)
9. SectionID (skor: 0.017)
10. PlaceofBirth (skor: 0.011)

Berdasarkan output Feature Importance dari kedua model, kedua model secara konsisten menempatkan StudentAbsenceDays sebagai fitur paling berpengaruh. Artinya, tingkat kehadiran siswa merupakan indikator utama performa akademik, yang secara logis sejalan dengan asumsi bahwa siswa yang sering hadir lebih aktif dan mengikuti pembelajaran.

Pada model Logistic Regression, fitur-fitur seperti GradeID dan StageID juga muncul sebagai penting. Ini dapat dijelaskan karena tingkatan kelas dan jenjang pendidikan siswa memang sangat menentukan kompleksitas materi dan penilaian, yang pada akhirnya memengaruhi performa akhir.

Sementara itu, model SVM menyoroti Discussion dan Relation sebagai fitur penting, selain StudentAbsenceDays. Hal ini menunjukkan bahwa model SVM mampu menangkap interaksi non-linier antar fitur yang tidak terlalu terlihat secara linear, seperti pengaruh komunikasi siswa (Discussion) atau peran orang tua (Relation).

## 4. Hasil dan Pembahasan

Model Regresi Logistik Multinomial dan SVM Multiclass digunakan untuk memprediksi performa akademik siswa ke dalam tiga kategori: rendah (L), sedang (M), dan tinggi (H), berdasarkan fitur seperti kehadiran, partisipasi kelas, dan karakteristik demografis. Regresi Logistik menghitung probabilitas tiap kelas menggunakan fungsi logit, sedangkan SVM menggunakan pendekatan satu-vs-rest dengan hyperplane optimal. Kedua model dilatih menggunakan validasi silang 5-fold guna meningkatkan keandalan hasil dan mencegah overfitting.

Hasil awal menunjukkan Regresi Logistik memiliki akurasi rata-rata lebih tinggi (0.700) dibandingkan SVM (0.654). Setelah tuning parameter melalui Grid Search, akurasi kedua model meningkat menjadi sama, yaitu 0.79. Meskipun performa akhir setara, keunggulan awal Regresi Logistik menunjukkan kemampuannya dalam mengenali pola data lebih konsisten sejak awal pelatihan.

Dari sisi pemerataan performa antar kelas, Regresi Logistik juga lebih unggul. F1-score model ini cukup merata di ketiga kelas, dengan skor terbaik pada kelas 1 (0.86), serta performa solid pada kelas 0 (0.71) dan kelas 2 (0.79). Sebaliknya, SVM menunjukkan ketimpangan

prediksi dengan recall rendah (0.64) pada kelas 0. Secara keseluruhan, macro average f1-score Regresi Logistik (0.79) sedikit lebih tinggi dibanding SVM (0.78), menandakan distribusi prediksi yang lebih seimbang.

Dalam hal efisiensi, Regresi Logistik lebih ringan secara komputasi. Model ini memiliki waktu pelatihan lebih singkat dan tidak melibatkan kernel kompleks seperti SVM dengan RBF. Selain itu, tuning parameter pada Regresi Logistik lebih sederhana dan cepat, sehingga cocok untuk dataset berukuran sedang dengan fitur terbatas.

Berdasarkan evaluasi menyeluruh, Regresi Logistik Multinomial terbukti sebagai model terbaik untuk memprediksi performa akademik siswa. Selain akurasi awal yang lebih tinggi dan keseimbangan antar kelas yang lebih baik, model ini juga lebih efisien dan mudah diinterpretasikan. Meskipun SVM dapat mencapai akurasi serupa setelah tuning, Regresi Logistik menawarkan keunggulan praktis dan metodologis yang menjadikannya lebih layak untuk implementasi pada studi serupa.

## 5. Kesimpulan

### 1. Efektivitas Masing-masing model

Regresi Logistik Multinomial terbukti lebih efektif dibandingkan SVM Multiclass, dengan akurasi awal lebih tinggi dan performa prediksi yang lebih merata di semua kelas. Setelah tuning, kedua model mencapai akurasi akhir yang sama (0.79), namun Regresi Logistik tetap unggul dalam stabilitas dan konsistensi prediksi. Sebaliknya, SVM cenderung kurang stabil pada kelas minoritas dan memerlukan proses tuning yang lebih kompleks.

### 2. Rekomendasi Penggunaan Model di Lingkungan Sekolah

Regresi Logistik lebih direkomendasikan untuk digunakan di sekolah karena mudah dipahami, cepat dilatih, dan memberikan hasil prediksi

yang seimbang. Model ini cocok untuk mendukung pengambilan keputusan berbasis data dalam pemantauan performa akademik siswa tanpa membutuhkan keahlian teknis tinggi.

### 3. Kelebihan dan Keterbatasan Pendekatan yang Digunakan

Kelebihan pendekatan ini terletak pada penggunaan validasi silang dan tuning parameter yang meningkatkan keandalan hasil, serta evaluasi komprehensif dengan metrik akurasi dan f1-score. Namun, model hanya menggunakan fitur terbatas seperti kehadiran, partisipasi, dan demografi, sehingga belum mencakup faktor eksternal lain yang juga memengaruhi performa siswa.

## Daftar Pustaka

Bergstra, J., & Bengio, Y. (2012). Random search for hyperparameter optimization. *Journal of Machine Learning Research*, 13(Feb), 281–305.

<https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf>

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297. <https://doi.org/10.1007/BF00994018>

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied*