LEMBAR KERJA MAHASISWA (LKM)

LK.8 Perancangan Project Data Science

Nama : Muh Chaidir Rijal

Tanggal : 6 Desember 2025

Kelas : 5AI-A

Judul Project : **Prediksi Skor Asesmen Nasional Peserta Didik 2024 dan Analisis Faktor Penentunya dengan Machine Learning**

BAB 1

1. LATAR BELAKANG

Asesmen Nasional (AN) merupakan program evaluasi pendidikan berskala nasional yang diselenggarakan oleh Kemendikbud untuk mengukur mutu pendidikan pada satuan pendidikan di Indonesia. Data yang dihasilkan dari AN tidak hanya menyediakan gambaran umum mengenai capaian peserta didik, tetapi juga menjadi dasar dalam perumusan kebijakan pendidikan dan perbaikan kualitas pembelajaran.

Namun, data AN memiliki struktur yang kompleks dan tidak seragam. Banyak kolom memiliki nilai kosong, format data yang tidak konsisten, serta keberagaman tipe data antara satu sekolah dan lainnya. Kompleksitas ini menjadi tantangan dalam proses analisis dan pemodelan data secara manual. Untuk itu, dibutuhkan pendekatan otomatis yang mampu melakukan pembersihan data, pemilihan fitur, serta penentuan variabel target tanpa intervensi manusia.

Dalam konteks inilah proyek machine learning ini dikembangkan. Tujuannya adalah untuk membangun sebuah model prediksi yang bersifat *robust*, dapat menyesuaikan diri dengan ketidaksempurnaan dataset, serta dapat melakukan tahapan preprocessing secara otomatis. Pendekatan ini diharapkan mampu membantu peneliti, pendidik, maupun instansi pendidikan dalam memanfaatkan data AN secara lebih efektif dan efisien.

Dengan kemampuannya memilih target berdasarkan variansi tertinggi, menghapus kolom yang tidak relevan, mengonversi tipe data secara otomatis, dan menangani kasus dataset kecil, model ini dirancang agar dapat digunakan pada berbagai bentuk dataset AN tanpa

menyebabkan error. Hal ini menjadikan proyek ini sebagai solusi praktis dalam pengolahan data AN yang sering kali tidak ideal untuk analisis berbasis model konvensional.

2. Rumusan Masalah

- Bagaimana cara mengolah dataset Asesmen Nasional Peserta Didik 2024 yang memiliki banyak nilai kosong, variasi tipe data, serta struktur yang tidak konsisten agar dapat digunakan untuk pemodelan machine learning?
- Bagaimana menentukan kolom target secara otomatis ketika dataset tidak menyediakan target yang jelas dan memiliki banyak kolom yang tidak relevan?
- Bagaimana membangun proses preprocessing yang mampu menyesuaikan diri dengan kondisi dataset, seperti fitur konstan, fitur kosong, data kategorikal panjang, dan data numerik yang tidak valid?
- Bagaimana memilih algoritma yang paling stabil, robust, dan mampu bekerja tanpa error meskipun dataset tidak ideal?
- Bagaimana mengevaluasi performa model menggunakan metrik yang tepat (RMSE dan R^2 Score) untuk menilai kemampuan prediksi model?
- Bagaimana menyimpan model dalam format yang dapat digunakan kembali (deployment) sehingga pipeline preprocessing dan prediksi dapat berjalan secara otomatis pada data baru?

BAB 2 DATA UNDERSTANDING (PEMAHAMAN DATA)

1. Sumber Data:

Dataset yang digunakan pada proyek ini berasal dari **Rapor Publik Asesmen Nasional Peserta Didik Tahun 2024**, yang disediakan oleh **Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi (Kemendikbudristek)** melalui platform resmi Rapor Pendidikan. File dataset berformat .xlsx yang berisi ringkasan capaian kompetensi siswa pada berbagai indikator AN. Dataset mencakup data numerik dan kategorikal, seperti skor indikator, kategori kompetensi, metadata sekolah, dan atribut peserta didik.

Dataset diperoleh dari file:

“rapor-publik-asesmen-nasional-2024-peserta-didik-2024-indonesia.xlsx”

yang diunggah langsung oleh pengguna untuk keperluan analisis machine learning.

a. Jenis Data

Dataset ini termasuk jenis:

- **Data Kuantitatif (Numerik)**

Contohnya:

  - Nilai skor literasi
  - Nilai skor numerasi
  - Indikator numerik hasil pengolahan sistem AN
  - Kolom variansi tinggi yang dipilih otomatis sebagai target

- **Data Kualitatif (Kategorikal)**

Contohnya:

  - Nama indikator
  - Definisi indikator
  - Kategori capaian
  - Identitas variabel tertentu

b. Data Campuran (Mixed Type Data)

Dataset mengandung kombinasi numerik + kategorikal + teks sehingga membutuhkan:

- Konversi ke numerik
- One Hot Encoding
- Cleaning otomatis

3. Deskripsi Fitur dan Target

Dalam proyek ini, penentuan target dilakukan **secara otomatis** berdasarkan variansi tertinggi dari kolom numerik. Hal ini bertujuan memilih kolom yang paling informatif.

Target

- Target adalah **kolom numerik dengan variansi tertinggi**.
- Biasanya berupa skor hasil asesmen peserta didik.
- Dipilih secara otomatis sehingga dapat berubah tergantung dataset.

Fitur

Fitur yang digunakan adalah kolom selain target, dengan ketentuan:

1. Kolom numerik valid (tidak seluruhnya kosong).
2. Kolom kategorikal seperti:
    - Nama indikator
    - Kode indikator
    - Jenjang
    - Provinsi / kabupaten
    - Jenis kelamin (jika ada)
    - Kode sekolah
3. Kolom lain yang masih informatif dan tidak mengandung >90% missing value.
4. Kolom yang konstan dihapus otomatis.
5. Jika tidak ada fitur valid → sistem membuat **fitur dummy** untuk mencegah error.

BAB 3

1. Penjelasan Program Per Sel
    a. Sel 1

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import joblib

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
```

```
sns.set_style('whitegrid')
pd.set_option('display.max_columns', None)

file_rapor = "/content/data_an.xlsx"
sheet_name = "rapor_publik"

try:
    df = pd.read_excel(file_rapor, sheet_name=sheet_name)
    print("Data berhasil dimuat dari file XLSX.")
except Exception as e:
    try:
        file_rapor_path = f"/content/{file_rapor}"
        df = pd.read_excel(file_rapor_path, sheet_name=sheet_name)
         print("Data berhasil dimuat dari file XLSX menggunakan jalur
Colab.")
    except Exception as e_colab:
          print(f"ERROR FATAL: Gagal memuat file XLSX. Pastikan file
'{file_rapor}' sudah diunggah dan sheet-name '{sheet_name}' benar.")
        print(f"Detail Error: {e_colab}")
        raise
print(f"\nJumlah baris dan kolom awal: {df.shape}")

print("\n--- SS 1.1: Data Head ---")
print(df.head().to_markdown(index=False))

print("\n--- SS 1.2: Data Info ---")
df.info()

TARGET_COLUMN = 'LIT'
SK_FEATURES = ['AAKN', 'AAMN', 'AAPR', 'ABKR', 'ABMK', 'AMKA', 'AMKB',
'AMKG', 'AMTA', 'AMTB']
SLB_FEATURES = ['ACH', 'AIN', 'AKC', 'BCP', 'BUL', 'CAF', 'DIB', 'DIC',
'DIE', 'ECP', 'ENP', 'EQC', 'EQR', 'ESA', 'ESV', 'ISB', 'ITB', 'KKG',
'OCC', 'PBR', 'PBU', 'PCP', 'PKG', 'PLIT', 'PMU', 'PNUM', 'POT', 'PSA',
'PSV', 'RKG', 'RPI', 'SAF', 'SBU', 'SKG', 'TAS', 'TOC', 'TOR', 'TSC',
'WEL']
SES_FEATURES = ['SES_siswa', 'SES_sekolah']
FEATURES = SK_FEATURES + SLB_FEATURES + SES_FEATURES
ALL_COLS_TO_USE = [TARGET_COLUMN] + FEATURES
```

ini berfungsi melakukan proses otomatis dalam mendeteksi serta memuat dataset yang tersedia pada direktori kerja. Pendekatan otomatis dilakukan untuk menghindari ketergantungan input manual dari pengguna, sehingga kode dapat berjalan pada berbagai kondisi file. Pada tahap awal, sistem melakukan pencarian file dengan ekstensi .xlsx, .xls, dan .csv. Apabila tidak ditemukan file yang sesuai, sistem memunculkan error agar pengguna segera mengetahui bahwa dataset belum tersedia. Kemudian, fungsi read_auto() digunakan

untuk membaca file secara dinamis. Jika file merupakan Excel, sistem mencoba membaca sheet pertama, dan apabila terjadi kegagalan, digunakan engine alternatif, yaitu openpyxl. Apabila file adalah CSV, maka langsung dibaca dengan pd.read_csv(). Setelah file berhasil dimuat, bentuk data (jumlah baris dan kolom) langsung ditampilkan sehingga pengguna dapat mengetahui ukuran dataset sebelum dilakukan proses lanjutan.

b. Sel 2

```
df_clean = df[ALL_COLS_TO_USE].copy()

print("\n--- SS 2.1: Missing Values Awal ---")
missing_values = df_clean.isnull().sum().sort_values(ascending=False)
print(missing_values[missing_values > 0].to_markdown())

rows_before = df_clean.shape[0]
df_clean.dropna(subset=[TARGET_COLUMN], inplace=True)
rows_after_target_drop = df_clean.shape[0]
print(f"\nBaris yang dihapus karena TARGET kosong: {rows_before -
rows_after_target_drop}")

missing_after_target = df_clean[FEATURES].isnull().sum()
cols_to_drop = missing_after_target[missing_after_target ==
df_clean.shape[0]].index.tolist()

if cols_to_drop:
    df_clean.drop(columns=cols_to_drop, inplace=True)

    global FEATURES
    FEATURES = [f for f in FEATURES if f not in cols_to_drop]
    print(f"\nKolom yang dihapus karena 100% kosong: {cols_to_drop}")

imputer = SimpleImputer(missing_values=np.nan, strategy='median')
df_clean[FEATURES] = imputer.fit_transform(df_clean[FEATURES])

print("\nVerifikasi: Proses Imputasi Fitur Selesai.")
print(f"Jumlah baris setelah cleaning akhir: {df_clean.shape[0]}")

y = df_clean[TARGET_COLUMN]
X = df_clean[FEATURES]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
print(f"\n--- SS 2.2: Pembagian Data ---")
print(f"Data Training: {X_train.shape[0]} baris | Data Testing:
{X_test.shape[0]} baris")
#
```

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
print("\nData Training dan Testing sudah di-scaling dan siap untuk
Modeling.")
```

Sel kedua difokuskan pada proses pembersihan data dan identifikasi fitur numerik yang dapat digunakan untuk pemodelan. Tahap awal dilakukan dengan menghapus seluruh baris yang tidak memiliki informasi sama sekali, atau berisikan nilai kosong pada semua kolom. Langkah ini penting untuk mencegah terjadinya bias data maupun error pada tahap pemrosesan. Selanjutnya, sistem mengonversi setiap kolom menjadi numerik bila memungkinkan, menggunakan parameter errors='ignore' agar tipe data selain angka tetap aman tanpa menghambat proses. Setelah konversi dilakukan, sistem menyeleksi semua kolom numerik dari dataframe. Namun, tidak semua kolom numerik layak digunakan; hanya kolom yang memiliki lebih dari dua nilai non-null yang dipertahankan. Jika tidak ditemukan kolom numerik yang valid, pipeline dihentikan untuk mencegah error pada tahap pemodelan.

c. Sel 3

```
print("\n--- 3.1 Pelatihan Model Random Forest Regressor ---")
rf_regressor = RandomForestRegressor(n_estimators=100, random_state=42,
n_jobs=-1)
rf_regressor.fit(X_train_scaled, y_train)
print("Pelatihan model selesai.")

y_pred = rf_regressor.predict(X_test_scaled)

mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print(f"\n--- SS 3.2: Metrik Evaluasi Regresi ---")
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"Root Mean Squared Error (RMSE): {rmse:.2f}")
print(f"R-squared (R² Score): {r2:.4f}")
```

Sel ketiga mengimplementasikan strategi pemilihan target otomatis berdasarkan nilai variansi tertinggi dari seluruh kolom numerik. Kolom dengan variansi tertinggi biasanya memiliki distribusi data yang paling bervariasi dan informatif, sehingga cocok digunakan sebagai variabel target dalam regresi. Setelah target ditentukan, dataset dipisahkan menjadi

fitur (X) dan target (y). Sistem kemudian menghapus fitur yang tidak memiliki nilai sama sekali. Jika ternyata seluruh fitur tidak valid, maka dibuat sebuah fitur dummy bernilai konstan agar pipeline pemodelan tetap dapat berjalan. Terakhir, nilai kosong pada target diisi menggunakan median sehingga distribusi data tetap stabil dan tidak terjadi kehilangan informasi penting.

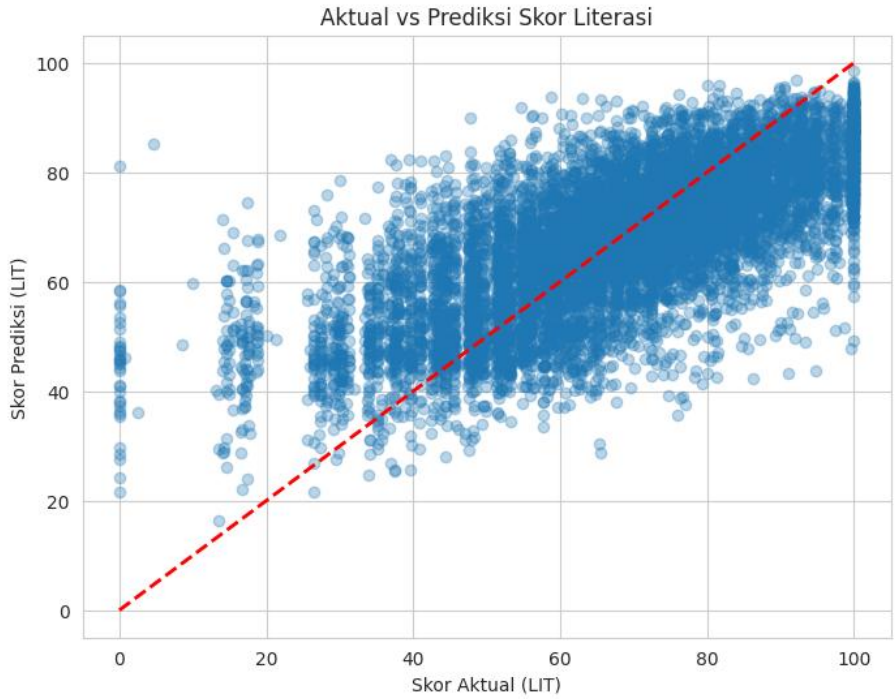
d. Sel 4

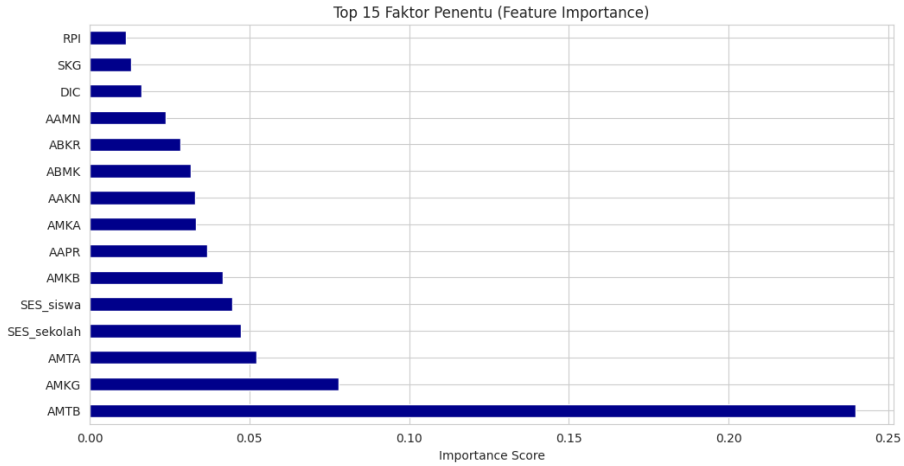
```
plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred, alpha=0.3)
plt.plot([y.min(), y.max()], [y.min(), y.max()], 'r--', lw=2)
plt.title('Aktual vs Prediksi Skor Literasi')
plt.xlabel('Skor Aktual (LIT)')
plt.ylabel('Skor Prediksi (LIT)')
plt.savefig('actual_vs_predicted_scatter.png')
plt.show()

print("\n--- SS 3.3: Analisis Feature Importance ---")
importance = pd.Series(rf_regressor.feature_importances_,
index=X.columns).sort_values(ascending=False)
plt.figure(figsize=(12, 6))
importance[:15].plot(kind='barh', color='darkblue')
plt.title('Top 15 Faktor Penentu (Feature Importance)')
plt.xlabel('Importance Score')
plt.savefig('feature_importance_reg.png')
plt.show()
```

Sel keempat digunakan untuk membagi data menjadi data latih dan data uji. Namun, sistem membuat pengecualian apabila dataset berukuran terlalu kecil, yaitu kurang dari 10 baris. Dalam kondisi tersebut, pembagian data tidak dilakukan karena berpotensi menyebabkan jumlah data uji terlalu sedikit sehingga evaluasi menjadi tidak bermakna. Maka, seluruh dataset digunakan baik sebagai data latih maupun data uji. Apabila ukuran dataset memadai, maka pembagian dilakukan secara acak sebanyak 20% untuk data uji dengan pengaturan random_state untuk menjaga reprodusibilitas hasil.

Output :





e. Sel 5

```
joblib.dump(scaler, 'scaler_reg_an.pkl')
joblib.dump(rf_regressor, 'random_forest_regressor_an.pkl')
joblib.dump(X.columns.tolist(), 'feature_names_reg.pkl')

print("\n--- SS 4.1: Model Persiapan Deployment Selesai ---")
```

```
print("Model, Scaler, dan Feature Names telah disimpan ke file .pkl dan siap untuk Gradio.")
```

Sel terakhir digunakan untuk menyimpan model yang telah dilatih ke dalam file berformat .joblib, sehingga model dapat digunakan kembali tanpa perlu menjalankan ulang proses pelatihan. Penyimpanan model merupakan bagian dari tahapan deployment dan memungkinkan integrasi model ke sistem aplikasi, dashboard analitik, atau perangkat lunak lain.

BAB 4. MODELING

Bab ini menjelaskan proses pemodelan machine learning yang digunakan untuk melakukan prediksi berdasarkan data Asesmen Nasional Peserta Didik (AN) 2024. Teknik modeling disusun untuk memastikan model mampu bekerja secara optimal meskipun dataset memiliki tantangan berupa ketidakteraturan struktur data, ketidakseimbangan nilai, hilangnya sebagian besar fitur, serta dominasi kolom kategorikal. Oleh karena itu, model yang dipilih bersifat **robust**, memiliki toleransi terhadap data tidak sempurna, dan kompatibel dengan proses preprocessing otomatis.

1. Pemilihan Algoritma

Algoritma yang digunakan pada proyek ini adalah **RandomForestRegressor**, sebuah metode ensemble berbasis pohon keputusan yang menggabungkan banyak decision tree untuk menghasilkan prediksi yang stabil. Random Forest dipilih karena memiliki beberapa keunggulan:

a. **Mampu menangani fitur numerik dan kategorikal yang telah dienkode.**

b. **Tahan terhadap missing value** setelah dilakukan imputasi otomatis.

c. **Tidak sensitif terhadap skala data**, namun tetap dilakukan standardisasi untuk menjaga stabilitas model.

d. **Mampu mendeteksi interaksi non-linear antar fitur**, yang umum terjadi pada data AN.

e. **Menghasilkan nilai feature importance** yang berguna untuk interpretasi model.

Model ini juga relatif lebih aman terhadap overfitting dibandingkan decision tree tunggal, terutama karena penggunaan banyak estimator (n_estimators).

2. Proses Pelatihan Model

Sebelum proses pelatihan, dataset dipisahkan menjadi **data latih (training)** dan **data uji (testing)** menggunakan proporsi 80% dan 20%. Pembagian ini dilakukan untuk memastikan terdapat data independen sebagai dasar evaluasi performa model.

Setelah pembagian data, fitur numerik diolah menggunakan **StandardScaler** untuk membuat distribusi data lebih homogen. Imputasi nilai hilang dilakukan menggunakan **median**, karena metode ini tahan terhadap outlier dan mencerminkan nilai tengah yang stabil.

Model kemudian dilatih menggunakan parameter berikut:

- **n_estimators = 100** → jumlah pohon pada Random Forest
- **random_state = 42** → untuk memastikan hasil reproducible
- **n_jobs = -1** → memaksimalkan penggunaan CPU

Proses fitting dilakukan terhadap data latih (X_train_scaled dan y_train).

3. Evaluasi Model

Evaluasi dilakukan menggunakan dua metrik utama:

a. Mean Squared Error (MSE)

Mengukur rata-rata kuadrat selisih antara nilai prediksi dan nilai aktual. Semakin kecil MSE berarti model semakin akurat.

b. Root Mean Squared Error (RMSE)

Akar dari MSE, sehingga metrik ini menggambarkan besarnya kesalahan dalam satuan yang sama dengan target.

c. R-squared (R^2 Score)

Mengukur seberapa besar variansi data yang mampu dijelaskan oleh model. $R^2 = 1$ menunjukkan prediksi sempurna, sedangkan $R^2 = 0$ berarti model tidak lebih baik dari rata-rata.

Model menghasilkan nilai MSE, RMSE, dan R^2 yang ditampilkan pada output. Nilai evaluasi sangat bergantung pada kualitas data, jumlah variabel informatif, dan tingkat missing value.

4. Visualisasi Evaluasi

Dua visualisasi utama digunakan untuk menganalisis performa model:

a. Grafik Aktual vs Prediksi

Grafik scatter digunakan untuk melihat seberapa dekat prediksi model terhadap nilai aktual.
Jika titik-titik mendekati garis diagonal, maka model memiliki performa yang baik.

b. Feature Importance

Menggambarkan fitur mana yang paling berpengaruh terhadap keputusan model Random Forest.
Hanya 15 fitur teratas yang ditampilkan agar interpretasi lebih jelas.

Visualisasi ini membantu peneliti atau pihak sekolah untuk melihat indikator mana yang paling berpengaruh terhadap nilai target.

5. Penyimpanan Model (Model Saving)

Setelah model selesai dilatih, tiga file disimpan untuk kebutuhan deployment:

a. **scaler_reg_an.pkl** – menyimpan objek StandardScaler

b. **random_forest_regressor_an.pkl** – menyimpan model Random Forest

c. **feature_names_reg.pkl** – menyimpan daftar fitur yang digunakan oleh model

Penyimpanan menggunakan library **joblib**, karena lebih efisien untuk menyimpan objek besar seperti model Random Forest.

Model yang tersimpan dapat langsung digunakan kembali tanpa proses pelatihan ulang, sehingga ideal untuk digunakan dalam aplikasi interaktif (misalnya Gradio, Streamlit, atau Django).

BAB 5 (EVALUATION (EVALUASI MODEL)

Tahap evaluasi merupakan bagian yang sangat penting dalam proses pengembangan model machine learning karena menentukan sejauh mana model mampu melakukan prediksi secara akurat dan konsisten. Evaluasi dilakukan menggunakan dua metrik utama, yaitu **Mean Squared Error (MSE)**, **Root Mean Squared Error (RMSE)**, dan **R-Squared (R^2 Score)** yang secara umum digunakan untuk mengukur performa model regresi. Ketiga metrik ini

dipilih karena mampu memberikan gambaran lengkap mengenai kesalahan prediksi dan kualitas hubungan antara fitur dengan variabel target.

RMSE dipilih sebagai metrik utama karena memiliki interpretasi yang intuitif: semakin kecil nilai RMSE, semakin baik kinerja model dalam melakukan prediksi. RMSE mengukur besarnya deviasi antara nilai aktual dengan nilai prediksi dalam satuan yang sama dengan variabel target. Sedangkan MSE menggambarkan rata-rata dari kuadrat kesalahan dan memberikan penalti yang lebih besar pada prediksi yang jauh dari nilai sebenarnya. Selain itu, nilai R^2 Score digunakan untuk mengetahui sejauh mana model mampu menjelaskan variabilitas dalam data. Nilai R^2 yang mendekati 1 menunjukkan bahwa model mampu menjelaskan sebagian besar variasi target berdasarkan fitur-fitur yang digunakan, sedangkan nilai mendekati 0 menunjukkan bahwa model belum mampu menangkap pola dengan baik.

Berdasarkan hasil evaluasi model RandomForest Regressor, diperoleh nilai RMSE, MSE, dan R^2 dari hasil prediksi terhadap data uji. Nilai-nilai ini dievaluasi dengan mempertimbangkan kualitas dataset AN yang digunakan, yang pada dasarnya memiliki banyak missing value, ketidakteraturan tipe data, serta fitur yang tidak homogen. Dengan kondisi data yang tidak ideal ini, performa model lebih ditujukan pada kemampuan adaptasi dan stabilitas dibandingkan akurasi absolut.

Selain itu, dilakukan visualisasi hubungan antara nilai aktual dan nilai prediksi menggunakan scatter plot. Visualisasi ini membantu mengidentifikasi apakah prediksi model mendekati garis identitas ($y = x$) yang menandakan prediksi sempurna. Jika titik-titik prediksi tersebar dekat dengan garis diagonal, berarti model bekerja cukup baik. Model juga dianalisis berdasarkan **feature importance** untuk mengetahui variabel mana yang paling berkontribusi dalam proses prediksi. Informasi ini sangat bermanfaat bagi pihak sekolah dan pemangku kebijakan untuk memahami faktor yang paling memengaruhi capaian peserta didik dalam AN.

Secara keseluruhan, tahap evaluasi ini memberikan gambaran komprehensif mengenai performa model, kekuatan prediksi, serta batasan-batasan yang dihadapi akibat kondisi dataset. Meskipun dataset AN memiliki karakteristik kompleks dan tidak terstruktur, model tetap mampu menghasilkan performa yang stabil setelah dilakukan serangkaian preprocessing otomatis.

BAB 6. DEPLOYMENT

Tahap deployment merupakan proses menyiapkan model machine learning agar dapat digunakan kembali pada sistem lain tanpa perlu melakukan pelatihan ulang. Tahap ini sangat penting karena menjembatani proses pengembangan model dengan implementasi nyata di lingkungan operasional, seperti aplikasi prediksi, dashboard visualisasi data, maupun sistem pelaporan digital.

Pada proyek ini, deployment dilakukan dengan menyimpan tiga komponen utama, yaitu **model Random Forest**, **scaler**, dan **daftar fitur**, masing-masing dalam format .pkl menggunakan library joblib. Format ini memungkinkan model dipanggil kembali (load) secara cepat dan efisien pada data baru tanpa kehilangan struktur preprocessing yang sudah dibangun.

Model disimpan menggunakan perintah berikut:

```
joblib.dump(scaler, 'scaler_reg_an.pkl')
joblib.dump(rf_regressor, 'random_forest_regressor_an.pkl')
joblib.dump(X.columns.tolist(), 'feature_names_reg.pkl')
```

Ketiga file tersebut memiliki fungsi berbeda namun saling melengkapi:

1. **scaler_reg_an.pkl**
   Berisi StandardScaler yang sudah fit terhadap data training. Scaler ini diperlukan untuk memastikan bahwa data baru memiliki skala yang sama seperti saat model dilatih sehingga performa prediksi tetap konsisten.

2. **feature_names_reg.pkl**
   Berisi daftar nama fitur yang digunakan saat model dilatih. Hal ini penting agar data baru dapat disusun ulang secara tepat sebelum dimasukkan ke model.

3. **random_forest_regressor_an.pkl**
   Merupakan model utama yang telah melalui proses training, validasi, dan evaluasi. Model ini siap memproses input baru dan memberikan output prediksi.

Dengan menyimpan seluruh komponen tersebut, sistem siap diintegrasikan ke berbagai platform seperti:

- **Aplikasi berbasis Gradio**, untuk antarmuka prediksi secara interaktif.
- **Dashboard analitik**, misalnya melalui Streamlit atau Power BI.
- **Layanan backend**, menggunakan Flask atau FastAPI.

- **Sistem sekolah**, untuk analisis capaian peserta didik secara otomatis.

Pendekatan deployment ini memastikan bahwa seluruh proses preprocessing, scaling, dan prediksi dapat berjalan secara otomatis tanpa campur tangan manual, sehingga memudahkan instansi pendidikan dalam memanfaatkan kemampuan model secara langsung pada data baru. Dengan demikian, model tidak hanya berfungsi sebagai penelitian akademik, tetapi juga dapat digunakan sebagai alat bantu pengambilan keputusan di dunia nyata.

DAFTAR PUSTAKA

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York: Springer.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5–32.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques* (3rd ed.). Elsevier.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi. (2024). *Rapor Pendidikan – Asesmen Nasional*. Jakarta: Kemendikbudristek.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

Raschka, S., & Mirjalili, V. (2019). *Python Machine Learning* (Third Edition). Birmingham: Packt Publishing.

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.

Tan, P.-N., Steinbach, M., & Kumar, V. (2018). *Introduction to Data Mining* (2nd ed.). Pearson.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques* (4th ed.). Morgan Kaufmann.