NAMA : RYAN AIDIL

NPM : 237006416026

MATKUL : MACHINE LEARNING K.01

LINK GITHUB : <https://github.com/Nezyrx/UAS-Machine-Learning_Ryan-Aidil_237006416026.git>

LINK GDrive : <https://drive.google.com/drive/folders/1tmZPtwO5byizJGbZTg88F7zdZWOpA-Sc?usp=sharing>

1. **TUJUAN ANALISIS**

Tujuan utama dari analisis dataset ini adalah untuk membangun sebuah model prediktif yang mampu memperkirakan luas area (dinyatakan dalam hektar) yang terbakar akibat kebakaran hutan. Prediksi ini didasarkan pada serangkaian data spasial, temporal, dan meteorologis. Dengan model ini, diharapkan dapat membantu upaya mitigasi dan penanggulangan bencana dengan mengidentifikasi faktor-faktor kunci yang paling berpengaruh terhadap besarnya area kebakaran.

1. **Algoritma yang Digunakan: Random Forest Regressor**

Algoritma yang dipilih untuk tugas ini adalah **Random Forest Regressor**. Ini adalah sebuah metode *ensemble learning* yang bekerja dengan membangun sejumlah besar *decision trees* (pohon keputusan) pada saat pelatihan dan menghasilkan prediksi berupa rata-rata (regresi) dari prediksi masing-masing pohon.

**Alasan Pemilihan Algoritma:**

* **Kemampuan Menangani Hubungan Non-Linear**: Hubungan antara kondisi cuaca (seperti suhu, angin, kelembapan) dan luas area kebakaran hutan bersifat kompleks dan tidak selalu linear. Random Forest sangat efektif dalam menangkap pola-pola rumit seperti ini.
* **Robust terhadap Overfitting**: Dengan menggabungkan hasil dari banyak pohon keputusan yang berbeda, Random Forest cenderung lebih tahan terhadap overfitting dibandingkan dengan satu pohon keputusan tunggal.
* **Mengukur Pentingnya Fitur (Feature Importance)**: Algoritma ini menyediakan mekanisme untuk mengukur dan memberi peringkat pada setiap variabel (fitur) berdasarkan kontribusinya terhadap akurasi prediksi.  Hal ini sangat berguna untuk memahami faktor apa yang paling dominan dalam mempengaruhi luas kebakaran, seperti suhu atau kelembapan.
* Stabilitas: Algoritma ini stabil dan memberikan hasil yang konsisten, bahkan dengan adanya data yang noisy atau nilai yang hilang (meskipun dataset ini tidak memiliki nilai yang hilang).

1. **Analisis dan Program Python**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

# ----- 1. Memuat Data -----

try:

df = pd.read\_csv('forestfires.csv')

print("Dataset berhasil dimuat.")

print("Dimensi data: ", df.shape)

print("\nContoh data awal:")

print(df.head())

except FileNotFoundError:

print("Error: Pastikan file 'forestfires.csv' berada di direktori yang sama dengan skrip ini.")

exit()

# ----- 2. Pra-pemrosesan Data -----

# Transformasi logaritmik pada target variabel 'area' untuk menormalkan distribusi

# np.log1p(x) adalah log(x+1), ini berguna untuk menangani nilai area = 0

df['area'] = np.log1p(df['area'])

# Mengubah kolom kategorikal (bulan dan hari) menjadi numerik menggunakan one-hot encoding

df\_processed = pd.get\_dummies(df, columns=['month', 'day'], drop\_first=True)

# Memisahkan fitur (X) dan target (y)

X = df\_processed.drop('area', axis=1)

y = df\_processed['area']

# ----- 3. Pembagian Data Latih dan Uji -----

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

print(f"\nData dibagi menjadi: {X\_train.shape[0]} data latih dan {X\_test.shape[0]} data uji.")

# ----- 4. Melatih Model Random Forest Regressor -----

print("Melatih model Random Forest Regressor...")

model = RandomForestRegressor(n\_estimators=100, random\_state=42, n\_jobs=-1)

model.fit(X\_train, y\_train)

print("Model selesai dilatih.")

# ----- 5. Evaluasi Model -----

print("\nMengevaluasi performa model pada data uji...")

y\_pred\_log = model.predict(X\_test)

# Mengembalikan prediksi ke skala semula dengan eksponensial (kebalikan dari log)

y\_pred\_original = np.expm1(y\_pred\_log)

y\_test\_original = np.expm1(y\_test)

# Hitung metrik evaluasi

mae = mean\_absolute\_error(y\_test\_original, y\_pred\_original)

mse = mean\_squared\_error(y\_test\_original, y\_pred\_original)

r2 = r2\_score(y\_test\_original, y\_pred\_original)

# ----- 6. Menampilkan Output Hasil -----

output\_text = f"""

=======================================

HASIL EVALUASI MODEL PREDIKSI

=======================================

- Mean Absolute Error (MAE) : {mae:.4f}

- Mean Squared Error (MSE) : {mse:.4f}

- R-squared (R²) : {r2:.4f}

=======================================

"""

print(output\_text)

# ----- 7. Analisis Faktor Penting (Feature Importance) -----

feature\_importances = pd.Series(model.feature\_importances\_, index=X.columns).sort\_values(ascending=False)

print("\n10 FAKTOR PALING BERPENGARUH:\n")

print(feature\_importances.head(10))

# Menyimpan output ke file .txt

with open("output\_analisis.txt", "w") as f:

f.write(output\_text)

f.write("\n10 FAKTOR PALING BERPENGARUH:\n\n")

f.write(feature\_importances.head(10).to\_string())

# ----- 8. Visualisasi Hasil -----

plt.figure(figsize=(10, 6))

sns.scatterplot(x=y\_test\_original, y=y\_pred\_original, alpha=0.6)

plt.plot([y\_test\_original.min(), y\_test\_original.max()], [y\_test\_original.min(), y\_test\_original.max()], '--r', linewidth=2)

plt.title('Perbandingan Area Aktual vs. Area Prediksi')

plt.xlabel('Area Aktual yang Terbakar (ha)')

plt.ylabel('Area Prediksi yang Terbakar (ha)')

plt.grid(True)

plt.savefig("prediksi\_vs\_aktual.png")

print("\nPlot visualisasi telah disimpan sebagai 'prediksi\_vs\_aktual.png'")

plt.show()

OUTPUT

