



PREDIKSI NILAI ECOLOGICAL FOOTPRINT SUATU NEGARA

Pacmann Course: Machine Learning Process

Nuzula Elfa Rahma
elfarahma@gmail.com

I. Rumusan Masalah

Tema dari proyek ini merupakan kelanjutan dari proyek kelas sebelumnya yaitu *Introduction to Machine Learning*, yaitu prediksi nilai *ecological footprint* suatu negara berdasarkan nilai HDI (Human Development Index). Sehingga secara umum, latar belakang proyek kelas *Machine Learning Process* tidak banyak berbeda, seperti yang dikutip di bawah ini:

Seiring dengan semakin mencuatnya isu perubahan iklim akibat emisi gas rumah kaca dari berbagai aktivitas manusia, konsep Malthus ini mulai mendapat banyak pertentangan dan sanggahan. Diskursus mengenai siapa yang patut disalahkan dalam isu kerusakan lingkungan terutama perubahan lingkungan mulai bergeser dan tertuju pada peran negara-negara maju (Post et al., 2019). Sejalan dengan semakin menonjolnya konsep jejak ekologis (ecological footprint) dan jejak karbon (carbon footprint), kontribusi manusia dalam membebani kapasitas lingkungan kini sudah dapat dikuantifikasi, sehingga tidak lagi bersifat retorik. Adapun ecological footprint merupakan satuan (Global Hectare) yang mencerminkan banyaknya sumberdaya yang dikonsumsi oleh seseorang atau kumpulan masyarakat, dalam melangsungkan hidupnya baik itu dalam bentuk aktivitas pemenuhan kebutuhan pangan, sandang, listrik, transportasi, hiburan, komunikasi, hingga limbah yang dihasilkan. Sehingga upaya mitigasi yang lebih terukur dengan menasar kepada fenomena, aktivitas dan gaya hidup yang benar-benar menjadi kontributor dampak lingkungan sudah dapat dilakukan. Salah satu upaya yang dapat dilakukan adalah dengan memanfaatkan model sebagai instrumen untuk melakukan simulasi ataupun prediksi magnitude dari aktivitas manusia di suatu wilayah terhadap lingkungannya berdasarkan parameter yang relevan.

Dengan demikian objektif dari *final project* ini juga adalah Metriks bisnis yang dapat diambil adalah upaya dan waktu untuk menganalisa sumber-sumber kontribusi peningkatan jejak korban dalam rangka menekan laju kerusakan lingkungan dapat dipermudah dan dipersingkat.

II. Pendekatan Lebih Lanjut

Dari project sebelumnya dengan melakukan perbandingan antara model regresi linear dan regresi polynomial, didapatkan bahwa regresi polynomial tidak dapat merepresentasikan hubungan antara fitur HDI, lokasi suatu negara (continent) dengan nilai *ecological footprint per kapita* (EFConsPerCap) suatu negara. Untuk itu, pada final project ini, pendekatan yang diambil adalah dengan melihat kinerja tiga

model regresi sebagai baseline yaitu regresi linear, random forest regressor, dan decision tree regressor. Dari ketiga model tersebut akan ditentukan model yang kinerjanya paling baik sesuai dengan metrik yang digunakan.

Pendekatan di atas akan diaplikasikan dalam suatu rangkaian (*pipeline*) yang dihubungkan oleh simpul-simpul kompartemen yang saling bertautan. Di mana masing-masing kompartemen akan memiliki fungsinya masing-masing. Output dari suatu kompartemen akan menjadi input bagi kompartemen selanjutnya. Sehingga di ujung alur akan dihasilkan sebuah model machine learning yang siap digunakan dan merupakan model dengan performa yang terbaik.

Selain itu, machine learning pipeline pada final project ini juga dirancang agar user lain dapat ikut serta melakukan berbagai kegiatan baik terkait melakukan prediksi, training, maupun pengembangan model. Ini dilakukan dengan memanfaatkan container yang berada pada suatu platform server yang dapat diakses secara *remote* dari mana saja. Kontainer berperan layaknya lingkungan virtual dengan konfigurasi sistem pendukung yang sama dari manapun lingkungan tersebut diakses.

III. Target dan Performa

Target dari final project ini adalah menemukan model regresi dengan performa terbaik. Untuk itu, metrik yang digunakan untuk mengukur performa dari model regresi yang diaplikasikan adalah metrik MSE (*Mean Square Error*) dan R-square (R^2). Kedua metrik ini lazim digunakan pada aplikasi model regresi. Selain itu waktu yang dibutuhkan untuk melatih suatu model (training time) juga menjadi pertimbangan dalam menentukan performa terbaik. Semakin singkat waktu yang diperlukan maka semakin baik performanya.

- MSE (*Mean Square Error*) adalah metrik yang dihitung dengan menentukan rata-rata dari kumpulan selisih antara nilai prediksi suatu model dengan nilai aslinya, yang kemudian dikuadratkan. Atau bisa dilihat sebagai berikut:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

dimana:

n : jumlah data yang diamati

y_i : nilai sebenarnya dari data ke- i

\bar{y} : nilai rata-rata dari seluruh data

Σ : penjumlahan dari 1 ke-1 hingga n .

Selisih ini menggambarkan rentang error yang dibuat oleh model yang bersangkutan. Sehingga semakin kecil nilai MSE maka performa model dinilai semakin baik.

- R-Square adalah metrik yang menggambarkan tingkat akurasi suatu garis/kurva dalam model regresi merepresentasikan data yang sebenarnya. Tingkat akurasi ini didefinisikan sebagai jarak antara titik data sebenarnya ke garis regresi. Rentang nilai R-Squared adalah antara 0 dan 1 atau dalam bentuk persentase. Nilai 1 mengindikasikan 100% titik data berimpitan dengan garis/kurva model. Dengan demikian semakin tinggi nilai R-squared, semakin baik juga performa suatu model. Rumusnya dapat dilihat sebagai berikut:

$$R^2 = 1 - \frac{RSS}{TSS}$$

Dimana:

RSS : jumlah kuadrat antara nilai prediksi model dan nilai yang diamati

TSS : jumlah kuadrat antara nilai yang diamati dan mean dari variabel dependen

IV. Solusi Utama

4.1. Dataset

Dataset yang digunakan dalam proyek berikut merupakan dataset time series tahunan dengan rentang waktu dari tahun 2000 sampai dengan 2014 yang mencakup data profil negara-negara seluruh dunia. Total entry yang termaktub dalam dataset tersebut adalah 2156 poin data. Dataset merupakan data repository Github yang diolah dengan menggabungkan data dari World Bank dan data dari Global Footprint Network (Shropshire, 2019). Profil tersebut terdiri dari beberapa variabel yaitu:

- *Country:*
Mencakup 146 negara di seluruh dunia.
- *Continent:*
Mencakup 6 wilayah benua yaitu Asia, Europe, Africa, South America, Oceania, dan North America.
- *HDI (Human Development Index)*
Parameter kependudukan yang merangkum tingkat kesejahteraan masyarakat suatu wilayah dari aspek kesehatan, pendidikan, dan taraf hidup masyarakat, dengan skala 0 – 1.
- *Life Expectancy*
Rata-rata harapan hidup penduduk suatu negara.
- *Population*
Jumlah penduduk suatu negara pada tahun ke-n.
- *Ecological Footprint per Capita (EFConsPerCap)*
Rata-rata besaran porsi kapasitas lingkungan per tahun yang dihabiskan per penduduk suatu negara dalam melangsungkan kehidupannya, mencakup semua kebutuhan baik primer, sekunder, ataupun tersier (Global Hectare per capita).

- Total Ecological Footprint dalam Global Hectare (EFConsTotGHA).
Total besaran porsi kapasitas lingkungan yang dihabiskan suatu negara (Total GHA).
- Biocapacity per Capita (BiocapPerCap)
Kapasitas lingkungan suatu negara dalam penyediaan sumberdaya dalam pemenuhan gaya hidup per penduduk (GHA per capita).
- Total Biocapacity dalam Global Hectare (BiocapTotGHA)
Total kapasitas lingkungan suatu negara dalam penyediaan sumberdaya dalam pemenuhan gaya hidup per penduduk (GHA per capita).

Unnamed: 0	country	year	hdi	life_expectancy	population	continent	EFConsPerCap	EFConsTotGHA	BiocapPerCap	BiocapTotGHA
0	Afghanistan	2000	0.340	55.482	20093756.0	Asia	0.648085	12768498.74	0.509767	1.004337e+07
1	Afghanistan	2001	0.341	56.044	20966463.0	Asia	0.605820	12438205.89	0.500063	1.026689e+07
2	Afghanistan	2002	0.373	56.637	21979923.0	Asia	0.704061	15128227.46	0.570591	1.226035e+07
3	Afghanistan	2003	0.381	57.250	23064851.0	Asia	0.708856	15954511.65	0.591033	1.330261e+07
4	Afghanistan	2004	0.396	57.875	24118979.0	Asia	0.619414	14556109.86	0.502825	1.181630e+07
...
2152	Zimbabwe	2010	0.452	52.975	14086317.0	Africa	1.241330	17346227.76	0.619689	8.659478e+06
2153	Zimbabwe	2011	0.464	54.800	14386649.0	Africa	1.306024	18618155.97	0.580174	8.270734e+06
2154	Zimbabwe	2012	0.488	56.516	14710826.0	Africa	1.173113	17086982.45	0.587819	8.561875e+06
2155	Zimbabwe	2013	0.498	58.053	15054506.0	Africa	1.140062	16984754.24	0.558680	8.323270e+06
2156	Zimbabwe	2014	0.507	59.360	15411675.0	Africa	1.088133	16589564.28	0.540845	8.245675e+06

2157 rows x 11 columns

Gambar 1. Cuplikan dataset yang digunakan

Pada proyek ini, target yang difokuskan adalah pada variabel Ecological Footprint per Capita saja.

4.2. Algoritma yang digunakan

Dari hasil final project sebelumnya didapatkan bahwa model regresi linear adalah model dengan performa yang baik serta representatif. Dalam kesempatan kali ini, dataset yang sama akan digunakan untuk mengevaluasi performa tiga model regresi (baseline). Model yang memberikan performa terbaik kemudian akan dipilih sebagai model produksi.

a) Regresi linear

Regresi linear adalah salah satu model dasar dalam *machine learning* yang digunakan untuk membuat prediksi berdasarkan satu atau beberapa input variabel (fitur). Model ini mengasumsikan adanya hubungan linear (garis lurus) antara fitur dan target prediksi.

Regresi linear dapat dinotasikan sebagai berikut:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$

Di mana,

y	:	target yang ingin diprediksi
x_1, x_2, \dots, x_n	:	fitur yang digunakan sebagai input
$\beta_1, \beta_2, \dots, \beta_n$:	koefisien yang dicari saat melakukan proses <i>learning</i>
β_0	:	intercept (nilai default yang tetap tercapai meski tidak ada input)

b) Random Forest Regressor

Merupakan algoritma machine learning (supervised) yang bekerja dengan membangun beberapa “decision tree” (Wackernagel et al., 2017), secara acak berdasar dari fitur-fitur yang memiliki data numerikal. Dari pohon-pohon ini, kemudian algoritma ini akan melakukan agregasi dari kumpulan hasil prediksi tersebut untuk mendapatkan hasil prediksi final. Kelemahan dari algoritma ini adalah dapat memakan waktu latih yang lebih lama dibandingkan algoritma lainnya.

c) Decision Tree Regressor

Algoritma ini mirip dengan random forest, namun hanya menggunakan satu decision tree terpilih. Kelebihannya tentu saja lebih mudah untuk dikomputasi, dan tidak memerlukan waktu latih yang cenderung panjang seperti *random forest* (Ferreira, 2022) .

V. Seleksi Fitur

Pemilihan fitur pada dataset di atas lebih banyak didasari oleh pertimbangan *domain knowledge*. Untuk memprediksi nilai *ecological footprint* (EF) suatu negara, fitur yang digunakan adalah HDI dan benua (continent) di mana suatu negara itu berlokasi. Fitur negara tidak dimasukkan karena jumlah kelas yang terlalu besar, sehingga akan menyebabkan bengkaknya jumlah fitur apabila ditransformasi dengan *one hot encoding*. Fitur tahun perolehan data (*year*) tidak dimasukkan karena rentang yang terlalu singkat yaitu hanya 14 tahun, sedangkan signifikansi jangka waktu pada studi terkait (Tiezzi et al., 2012) baru terlihat setelah beberapa dekade. Terakhir, fitur harapan hidup juga tidak dimasukkan karena merupakan salah satu variabel pada perhitungan HDI (Mahajan, 2013).

VI. Workflow

Tahapan dalam membangun infrastruktur machine learning dalam *final project* ini adalah sebagai berikut:

1. Persiapan Data

Dalam tahapan ini, aktivitas yang dilakukan adalah sebagai berikut

- Pengumpulan data (lebih detail dapat dilihat pada Bab 4.1),
- Pendefinisian data, yaitu menentukan lingkup dan batasan nilai data yang meliputi rentang nilai pada kolom numerikal (HDI, EFConsPerCap), tipe data (str, int, float, dll), serta batasan kelas pada kolom kategorikal (continent).

- Validasi data, yakni memastikan bahwa setiap entry pada dataset sudah sesuai dengan batasan yang ditentukan dalam pendefinisian data.
 - Data defense, mekanisme warning apabila ada entry data dari API yang tidak sesuai dengan pendefinisian data.
 - Data splitting, membagi dataset untuk tujuan training, test, dan validasi. Test size yang digunakan adalah 20%.
2. Exploratory Data Analysis (EDA)
- Tahapan EDA dilakukan untuk menganalisa apakah dataset yang ada memiliki kecenderungan yang akan menjadi penghambat atau mengurangi performa model, untuk kemudian dianalisa modifikasi apa yang tepat untuk data tersebut. Hal-hal yang dianalisis meliputi: adanya null-data, skewness, dan imbalance pada data.
3. Preprocessing
- Proses yang dilakukan dalam tahap ini adalah penanganan terhadap null-data dengan melakukan imputasi berdasarkan hasil EDA, dan feature engineering. Untuk *feature engineering*, proses yang dilakukan adalah:
- Transformasi data kategorikal (continent) dengan menggunakan *one hot encoding*.
 - Standardisasi data.
 - Penanganan imbalance data dengan membuat 3 dataset dengan tehnik yang berbeda yaitu undersampling, oversampling, dan SMOTE.
4. Modeling
- Proses yang dilakukan dalam tahapan ini adalah:
- Training dan evaluasi model, dimana di akhir proses ini ditentukan model yang terbaik yang akan digunakan sebagai model produksi sesuai dengan metrik MSE (nilai terendah), R-Square (nilai tertinggi), training time (tersingkat).
 - Optimalisasi dengan melakukan hyper-parameter tuning, untuk kemudian dilakukan kembali tahap training dan evaluasi model, serta pemilihan model produksi.
 - Dokumentasi hasil training dan evaluasi model ke dalam *training log*.
5. Deployment
- Deployment merupakan proses kloning infrastruktur machine learning dari environment pada perangkat computer host ke environment baru agar dapat diakses oleh pengguna lain dan dari mana saja dengan menggunakan API. Deployment pada final project ini dilakukan pada server AWS, via *instances EC2*.

VII. Project Timeline

Adapun tahapan kegiatan yang dilakukan dalam final project ini adalah sebagai berikut:

Tabel 1. Project timeline

No	Kegiatan	Minggu
----	----------	--------

		1	2	3	4	5	6	7	8
1.	Pengumpulan Data								
2.	Desk Study								
3.	Persiapan Data								
4.	EDA								
5.	Preprocessing								
6.	Modeling								
7.	Pengujian								
8.	Deployment								
9.	Reporting								

VIII. Review

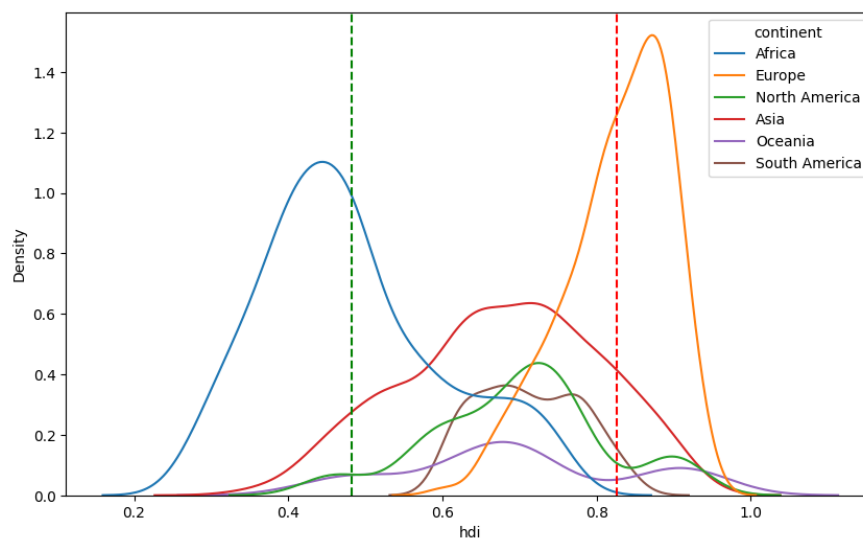
8.1. Exploratory data analysis

- Data Null

Dataset yang digunakan pada project ini sudah berupa *clean data* dari sumbernya, sehingga tidak memiliki data-null. Meski demikian, penanganan imputasi untuk data-null tetap dilakukan untuk mengantisipasi data latih hasil input baru (misalnya dari API).

- Skewness

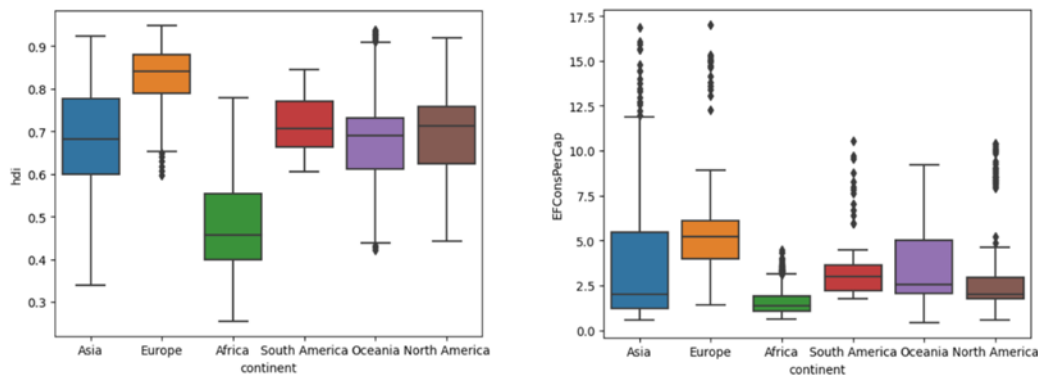
Fitur “continent” menunjukkan pola distribusi yang cukup mencolok pada beberapa kelas. Dapat dilihat pada gambar berikut ada dua kelas yang menunjukkan pola *skewness* yang kentara. Kelas Africa menunjukkan pola *skewed* ke kiri, sedangkan Europe menunjukkan pola *skewed* ke kanan. Pengujian t-test juga mengkonfirmasi bahwa nilai rata-rata (mean) pada kedua kelas ini memang berbeda secara signifikan. Sementara itu, kelas continent lainnya masih menunjukkan pola distribusi yang mendekati distribusi normal.



Gambar 1. Distribusi data dari setiap kelas pada fitur “continent”

- Outlier

Outlier pada dataset ini sangat terkait dengan fitur “continent”. Pengklasifikasian wilayah ke dalam kategori benua adalah pengkerucutan dari kategori negara. Namun pengekerucutan ini dapat dikatakan terlalu sederhana dan tidak representatif. Dapat dilihat pada boxplot (Gambar 2) berikut, fitur benua pada kategori Eropa menunjukkan 70% data berada pada ranah HDI yang tinggi, tetapi tidak berdistribusi normal dan condong (skewing) sebelah kanan. Ini menunjukkan adanya outlier pada sebelah kiri. Hal serupa juga ditunjukkan pada hubungan antara fitur benua dan nilai EF per kapita, terlihat bahwa semua kategori memiliki kecenderungan condong ke sebelah kiri.

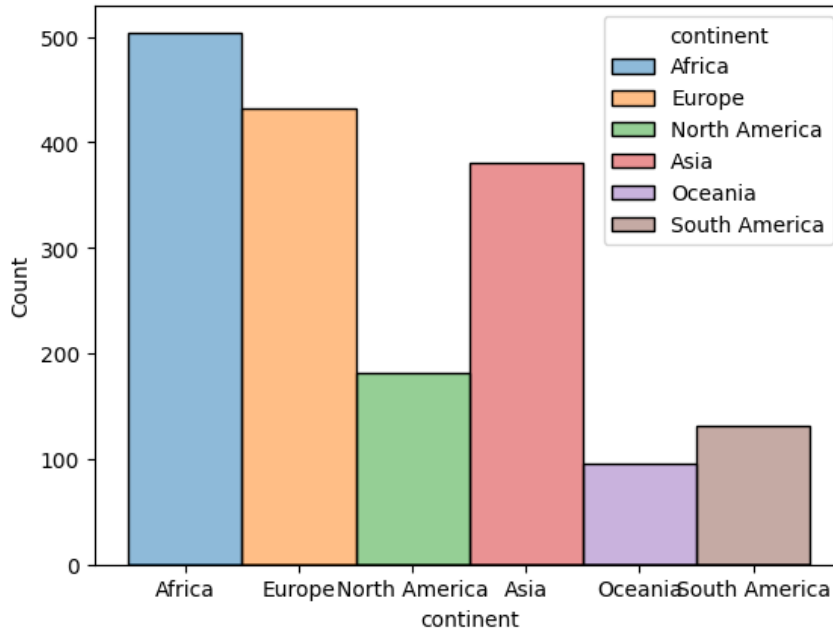


Gambar 2. Hubungan antara fitur benua dengan HDI dan EF per kapita

Dari pola tersebut dapat dilihat bahwa penanganan outlier pada dataset ini tidak dapat dilakukan dengan melakukan imputasi. Karena penyebabnya cukup fundamental yakni fitur benua terlalu luas untuk dapat menangkap diversitas negara-negara yang ada di dalamnya. Negara-negara maju di Asia Timur tentu saja akan memiliki kecenderungan yang berbeda dengan negara-negara berkembang di Asia Tenggara. Sehingga solusi yang lebih tepat untuk penanganan outlier pada dataset ini adalah dengan memperkecil lingkup *machine learning* dari skala dunia menjadi skala Regional (Asia Timur, Asia Tengah, Eropa Barat, dst). Dengan demikian, diharapkan jumlah kelas tidak akan terlalu besar namun tetap representatif.

- Data imbalance

Dari Gambar di bawah dapat dilihat bahwa terdapat data imbalance berdasarkan kelas pada fitur continent. Hal ini disebabkan karena beberapa *continent* memang hanya memiliki beberapa negara, sehingga ketika diagregasi jumlah datanya juga menjadi timpang. Dengan demikian maka akan dilakukan tehnik undersampling, oversampling, dan SMOTE untuk menyeimbangkan jumlah sampel masing-masing kelas.



Gambar 3. Proporsi data pada masing-masing kelas kategori “continent”

8.2. Deployment

Tahapan deployment dilakukan dengan melakukan *push Github repository* berikut https://github.com/elfarahma/ml_process-main.git ke server AWS via instances EC2 serta dengan menggunakan docker container. Proses deployment berhasil dilakukan. Akan tetapi kendala terakhir yang belum terselesaikan adalah meski API sudah dapat dijalankan pada server, proses input data belum dapat dieksekusi.

```

System load: 0.0          Processes:              101
Usage of /:  70.3% of 7.57GB Users logged in:        0
Memory usage: 34%        IPv4 address for docker0: 172.17.0.1
Swap usage:  0%          IPv4 address for eth0:  172.31.30.210

* Ubuntu Pro delivers the most comprehensive open source security and
compliance features.

https://ubuntu.com/aws/pro

* Introducing Expanded Security Maintenance for Applications.
Receive updates to over 25,000 software packages with your
Ubuntu Pro subscription. Free for personal use.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

ubuntu@ip-172-31-30-210:~$ sudo docker images
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
elfarahma/ml_process_api    latest      83add87c665f  27 hours ago  1.75GB
elfarahma/ml_process_streamlit latest      ba183d54e2fe  27 hours ago  1.75GB
ubuntu@ip-172-31-30-210:~$ cd
ubuntu@ip-172-31-30-210:~$ ls
ml_process-main
ubuntu@ip-172-31-30-210:~$ cd ml_process-main
ubuntu@ip-172-31-30-210:~/ml_process-main$ ls
ML_Process.jpg  api.py  data      docker-compose.yaml  eda.ipynb  modeling.py  preprocessing.py  unit test.py
README.md       assets  data_pipeline.py  dockerfile           log         models       requirements.txt  util.py
__pycache__     config  docker      docs                 main.py     notebooks   streamlit.py
ubuntu@ip-172-31-30-210:~/ml_process-main$

```

Gambar 4. Proses deployment Github Repository menuju server AWS dengan instances EC2

8.3. Imputasi Data-Null

Dari hasil tinjauan tersebut, dibuatlah aturan terkait imputasi data-null. Untuk kelas Africa dan Europe, maka data-null pada fitur HDI dan EFConsPerCap akan diimputasi dengan nilai median kelas. Sedangkan untuk kelas continent lainnya, data-null akan diimputasi dengan nilai *mean* kelas. Selanjutnya, apabila data-null ditemukan pada fitur continent, maka data pada baris tersebut akan dihapus.

8.4. Evaluasi Model

Setelah dilakukan sesi training dan evaluasi dengan tiga model regresi. Didapat bahwa model regresi dengan performa terbaik adalah *Random Forest Regressor* dengan metode SMOTE untuk penanganan *imbalance data*. Dari Tabel 1 di bawah, nampak model training dengan *hyper-parameter tuning* menunjukkan performa dari metrik MSE dan R-square yang lebih baik daripada yang tidak. Namun demikian waktu latih tercatat mengalami penambahan cukup signifikan.

Tabel 2. Performa model Random Forest Regressor dari berbagai metrik.

	Performa		
	MSE	R-Square	Training Time
Tidak	0.33	0.67	0.3 menit
Ya	0.25	0.74	2.25 menit

8.5. Requirements

Final project ini dibangun dengan bahasa pemrograman Python versi 3.10. Dengan didukung oleh modul-modul berikut:

Tabel 3. Modul yang dibutuhkan dalam setiap tahapan

Data Pipeline	<ul style="list-style-type: none">• Train_test_split - sklearn.model_selection• Tqdm• Pandas• Os• Copy
Preprocessing	<ul style="list-style-type: none">• Pandas• Numpy• RandomUnderSampler - imblearn.under_sampling• RandomOverSampler, SMOTE -imblearn.over_sampling• StandardScaler• StandardScaler - sklearn.preprocessing -• OneHotEncoder - sklearn.preprocessing
Modeling	<ul style="list-style-type: none">• Joblib• Json• Pandas

	<ul style="list-style-type: none"> • Copy • Hashlib • Os • LinearRegression – sklearn.linear_model • RandomForestRegressor – sklearn.ensemble • DecisionTreeRegressor – sklearn.tree • Mean_squared_error -sklearn.metrics • R2_score -sklearn.metrics • RandomizedSearch – sklearn.model_selection
API	<ul style="list-style-type: none"> • Fastapi • BaseModel – pydantic • Uvicorn • pandas

Sedangkan untuk dataset, untuk input harus berformat CSV, dan memiliki variable nilai HDI, benua, dan nilai *ecological footprint per capita*. Dengan batasan sebagai berikut:

	<i>Nama kolom</i>	<i>Range</i>	<i>Tipe Data</i>
<i>HDI</i>	"hdi"	0 - 1	float
<i>Lokasi benua suatu negara</i>	"continent"	"Asia", "Europe", "Africa", "South America", "North America", "Oceania"	object
<i>nilai ecological footprint per capita suatu negara</i>	"EFConsPerCap"	0 - 20	float

References

- Ferreira, J. A. (2022). Models under which random forests perform badly; consequences for applications. *Computational Statistics*, 37(4), 1839–1854. <https://doi.org/10.1007/s00180-021-01182-4>
- Mahajan, S. (2013). *Human Development Index — Measurements, changes and evolution*. 1–5. <https://doi.org/10.1109/NUiCONE.2013.6780196>
- Post, S., Kleinen-von Königslöw, K., & Schäfer, M. S. (2019). Between Guilt and Obligation: Debating the Responsibility for Climate Change and Climate Politics in the Media. *Environmental Communication*, 13(6), 723–739. <https://doi.org/10.1080/17524032.2018.1446037>
- Shropshire, A. (2019). *HDI vs Ecology Plotly*. Github. https://github.com/as6140/hdi_v_ecology_plotly_blog

- Tiezzi, E., Pulselli, F., & Cristina, C. (2012). Biocapacity vs Ecological Footprint of world regions: A geopolitical interpretation. *Ecological Indicators*, 16, 23–30. <https://doi.org/10.1016/j.ecolind.2011.09.002>
- Wackernagel, M., Hanscom, L., & Lin, D. (2017). Making the sustainable development goals consistent with sustainability. *Frontiers in Energy Research*, 5(JUL). <https://doi.org/10.3389/fenrg.2017.00018>