Laporan Proyek Machine Learning: Predictive Analytics On Adults Income - Vian Sebastian Bromokusumo

Domain Proyek

Latar Belakang

Menurut survei oleh Badan Pusat Statistik Indonesia yang dipublikasikan pada 6 November 2023, tingkat pengangguran di Indonesia adalah sebesar 5,32 persen [1]. Meskipun hal ini merupakan penurunan dari tahun-tahun sebelumnya, hal ini tetap menjadi isu yang perlu terus diselesaikan. Hasil penelitian Pramudjasi dan Juliansyah, 2019 dalam Jurnal FEB Unmul [2] menyatakan bahwa jumlah penduduk berpengaruh positif terhadap tingkat pengangguran secara signifikan. Hal ini dikuatkan pula dalam penelitian Sari dan Pangestuty , 2022 dalam [3] menyatakan bahwa pertambahan jumlah penduduk berdampak pada kenaikan Tingkat Pengangguran Terbuka. Berbagai survei dan penemuan dari jurnal akademik ini menjadi latar belakang Penulis dalam mencoba menganalisa faktor-faktor apa saja yang memiliki keterkaitan kuat terhadap pengangguran, pekerjaan, terutama faktor yang memiliki keterkaitan terhadap pendapatan yang tinggi.

Proyek ini difokuskan pada analisis dataset "Adults Income" dengan tujuan untuk mengidentifikasi dan memprediksi faktor-faktor yang memiliki korelasi tinggi, atau bahkan berkontribusi terhadap pekerjaan terutama jumalah pendapatan seseorang. Dalam hal ini, target fitur dari dataset ini adalah apakah pendapatan per tahun seseorang <50.000 dolar atau >= 50.000 dolar per tahun, sehingga proyek ini akan menyelesaikan kasus klasifikasi. Meskipun dataset ini diambil dari data sensus 1996, dataset ini masih memiliki relevansi yang signifikan di masa kini, karena sifat umum dari variabel-variabel di dataset ini dan tidak adanya perubahan yang drastis pada komponen-komponen penentu penghasilan.

Menurut Jepchumba dari Microsoft [4], *machine learning* merupakan teknik yang menggunakan matematika tingkat tinggi dan ilmu statistika untuk mengenali pola pada data yang tidak ada secara eksplisit, dan dapat memprediksi sesuai dengan hasil pola tersebut. Dengan beragamnya faktor-faktor (variabel) yang terlibat dalam proyek ini, machine learning menjadi solusi yang terbaik. Identifikasi dan prediksi faktor-faktor akan dilakukan dengan cara mengaplikasikan teknik-teknik *data analysis* seperti *Exploratory Data Analysis* (EDA) dan menggunakan algoritma-algoritma *Machine Learning* seperti *Random Forest, K-Nearest Neighbors,* dan *Boosting*.

Proyek ini menjadi sarana kecil untuk membantu menyelesaikan masalah sulitnya mencari kerja, dengan menganalisis dinamika faktor-faktor pendapatan per tahun yang tinggi. Hasil dari proyek ini diharapkan dapat membantu Pemerintah dan individu-individu di usia produktif sebagai tambahan *insight* terkait masalah ini, dan membantu pihak-pihak tersebut untuk mengembangkan masyarakat dan diri sendiri untuk peningkatan kualitas hidup.

Business Understanding

Stakeholder dan sasaran:

1. Pemerintah Sebagai organisasi tingkat tertinggi di sebuah negara, pemerintah dapat membuat kebijakan-kebijakan dan perubahan yang baik, guna meningkatkan kualitas hidup rakyatnya. Salah satu

caranya ialah membuat/memperbaiki sistem di negaranya untuk mendorong kemajuan sumber daya manusianya.
2. Individu Pada tingkat individu, diharapkan hasil proyek ini dapat memberikan insight terhadap faktor-faktor penting yang dapat meningkatkan kualitas hidupnya, melalui pekerjaan dan pekerjaan dengan pendapatan per tahun yang tinggi.

Problem Statements

1. Dari berbagai fitur, apa yang paling berpengaruh terhadap income (pendapatan)?
2. Dengan karakteristik tertentu, apakah income dapat diprediksi?

*income merujuk pada pendapatan per tahun.

Predictive Modelling Goals

1. Mengetahui fitur-fitur yang memiliki kaitan yang tinggi terhadap income.
2. Dapat memprediksi income dengan akurasi di atas 90%.

Solution Statements (Metodologi)

1. Target feature pada dataset ini merupakan variable boolean antara >50k dan <= 50k, sehingga kasus ini merupakan kasus prediksi Klasifikasi.

2. Melakukan Exploratory Data Analysis untuk mendapatkan informasi berguna dalam data dan mengetahui dinamika fitur-fitur.

3. Melakukan uji perbedaan teknik-teknik *missing values handling*, dan dampaknya terhadap akurasi model machine learning.

4. Membuat model machine learning yang dapat memprediksi income dengan akurasi di atas 90%

5. Menggunakan metrik evaluasi Accuracy, Precision, Recall, F1-Score, dan Confusion Matrix untuk mengevaluasi performa model.

Data Understanding

Dataset: https://archive.ics.uci.edu/dataset/2/adult ***Dataset Overview***

Deskripsi variabel pada dataset adalah sebagai berikut:

1. age (int64) : umur sampel
2. workclass (object) : status pekerjaan sampel
3. fnlwgt (int64) : ID sampel
4. education (object) : pendidikan terakhir sampel
5. educational-num (int64) : pendidikan terakhir sampel dalam angka
6. marital-status (object) : status pernikahan sampel
7. occupation (object) : pekerjaan/industri pekerjaan sampel
8. relationship (object) : status hubungan sampel
9. race (object) : ras sampel
10. gender (object) : jenis kelamin sampel
11. capital-gain (int64) : pemasukan kapital

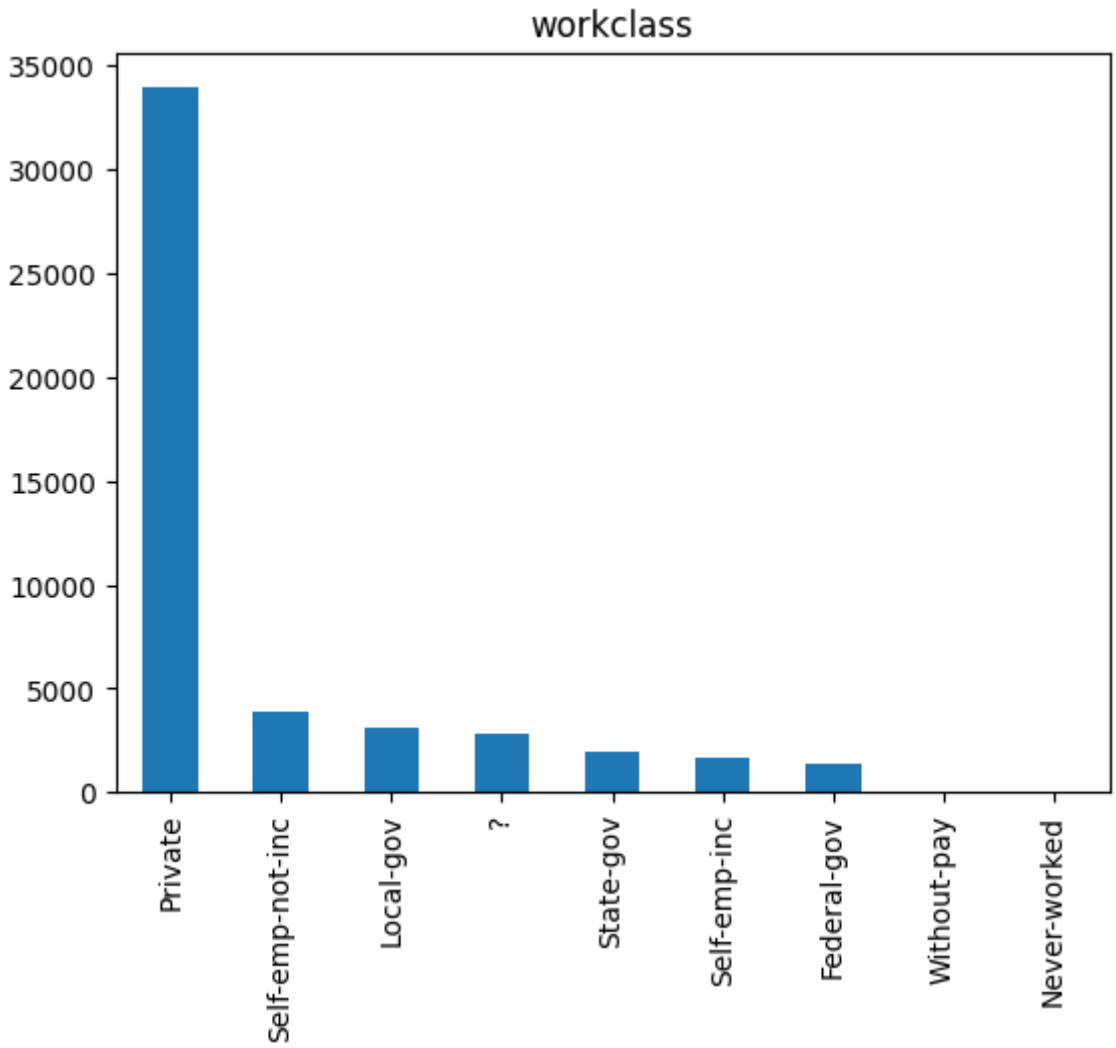
12. capital-loss (int64) : kerugian kapital
13. hours-per-week (int64) : jam kerja sampel dalam seminggu
14. native-country (object) : negara asal/buyut sampel
15. income (object) : pendapatan sampel

Deskripsi rinci dari dataset ini adalah sebagai berikut

1. Terdapat 48.842 sampel data
2. Terdapat dua tipe data yang berbeda, antara lain 'object' (kategorikal), dan 'int64' (numerik).
3. Dari 15 kolom variabel, terdapat 9 kolom kategorikal dan 6 kolom numerik.
4. Pembagiannya adalah 14 kolom fitur dan 1 kolom target.
5. Pada dataset ini, missing values, dtiandai dengan tanda '?', dan berjumlah sekitar 7% dari seluruh dataset.

Deskripsi dari distribusi data dapat dicermati pada bagian *Univariate Analysis* dan *Multivariate Analysis* di bawah ini.

Univariate Analysis



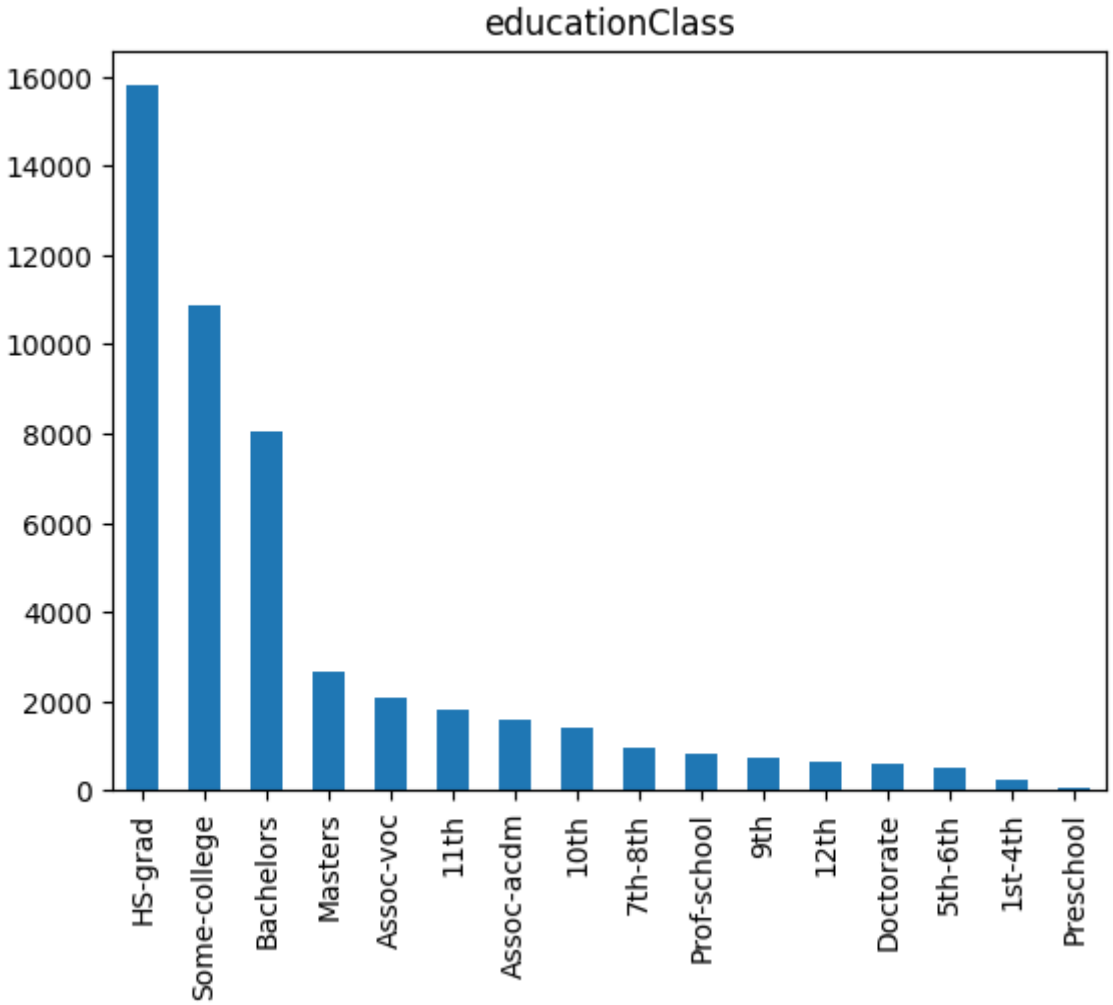
Gambar 1.0: Univariate analysis workclass vs income

Keterangan dari distribusi workclass dapat dicermati dari tabel berikut.

Tabel 1.0: Distribusi Workclass

workclass	sampe count	percentage
Private	33906	69.4
Self-emp-not-inc	3862	7.9
Local-gov	3136	6.4
?	2799	5.7
State-gov	1981	4.1
Self-emp-inc	1695	3.5
Federal-gov	1432	2.9
Without-pay	21	0.0
Never-worked	10	0.0

Mencermati tabel 1.0, dapat dilihat bahwa kategori modus dari workclass adalah Private, dengan selisih lebih dari 60%.



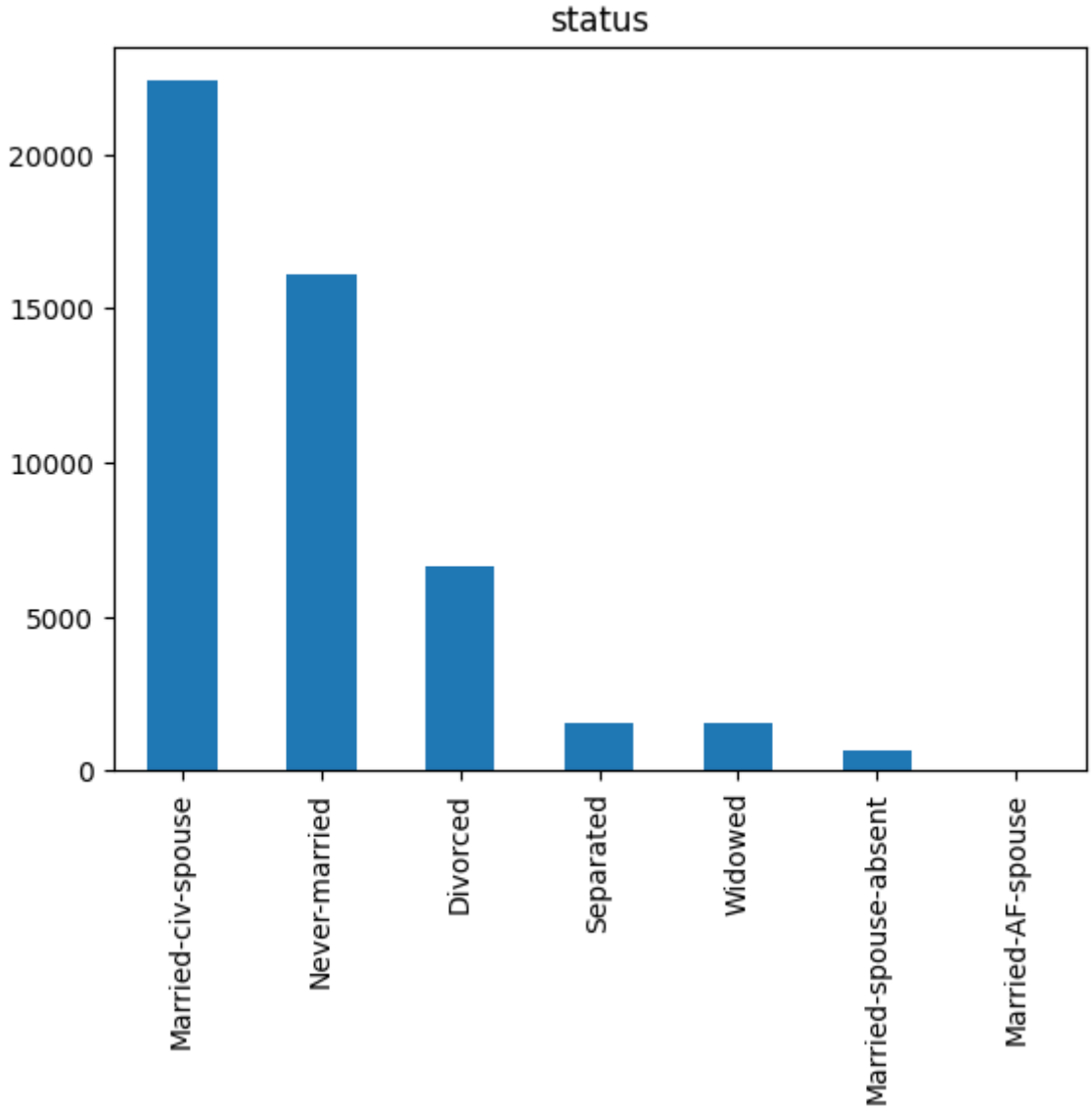
Gambar 1.1: Univariate analysis educationClass vs income

Keterangan dari distribusi educationClass dicermati dari tabel berikut.

Tabel 1.1: Distribusi educationClass

educationClass	**sampe count**	**percentage**
HS-grad	15784	32.3
Some-college	10878	22.3
Bachelors	8025	16.4
Masters	2657	5.4
Assoc-voc	2061	4.2
11th	1812	3.7
Assoc-acdm	1601	3.3
10th	1389	2.8
7th-8th	955	2.0
Prof-school	834	1.7
9th	756	1.5
12th	657	1.3
Doctorate	594	1.2
5th-6th	509	1.0
1st-4th	247	0.5
Preschool	83	0.2

Pada tabel 1.1, dapat dilihat bahwa kategori modus jatuh pada HS-grad (lulusan SMA), namun distribusi terlihat lebih seimbang daripada kategori Workclass, dengan tren *Right-skewed (positive) distribution*.



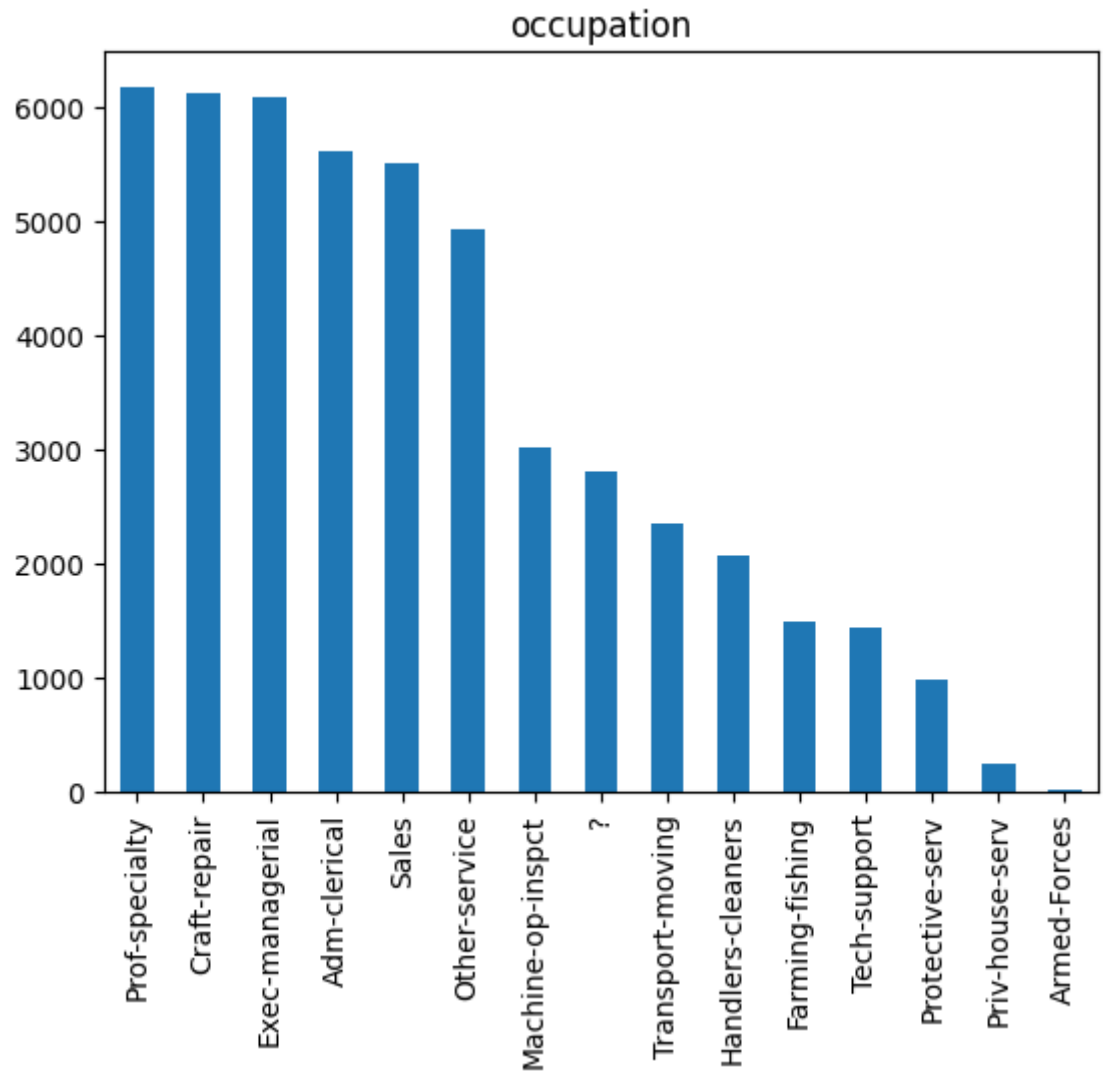
Gambar 1.2: Univariate analysis status vs income

Keterangan dari distribusi status dapat dicermati dari tabel berikut.

Tabel 1.2: Distribusi status

status	sampe count	percentage
Married-civ-spouse	22379	45.8
Never-married	16117	33.0
Divorced	6633	13.6
Separated	1530	3.1
Widowed	1518	3.1
Married-spouse-absent	628	1.3
Married-AF-spouse	37	0.1

Berdasarkan tabel 1.2, mirip dengan educationClass, distribusi terlihat lebih seimbang dan tidak tinggi pada satu kategori, dengan tren *Right-skewed distribution*



Gambar 1.3: Univariate analysis occupation vs income

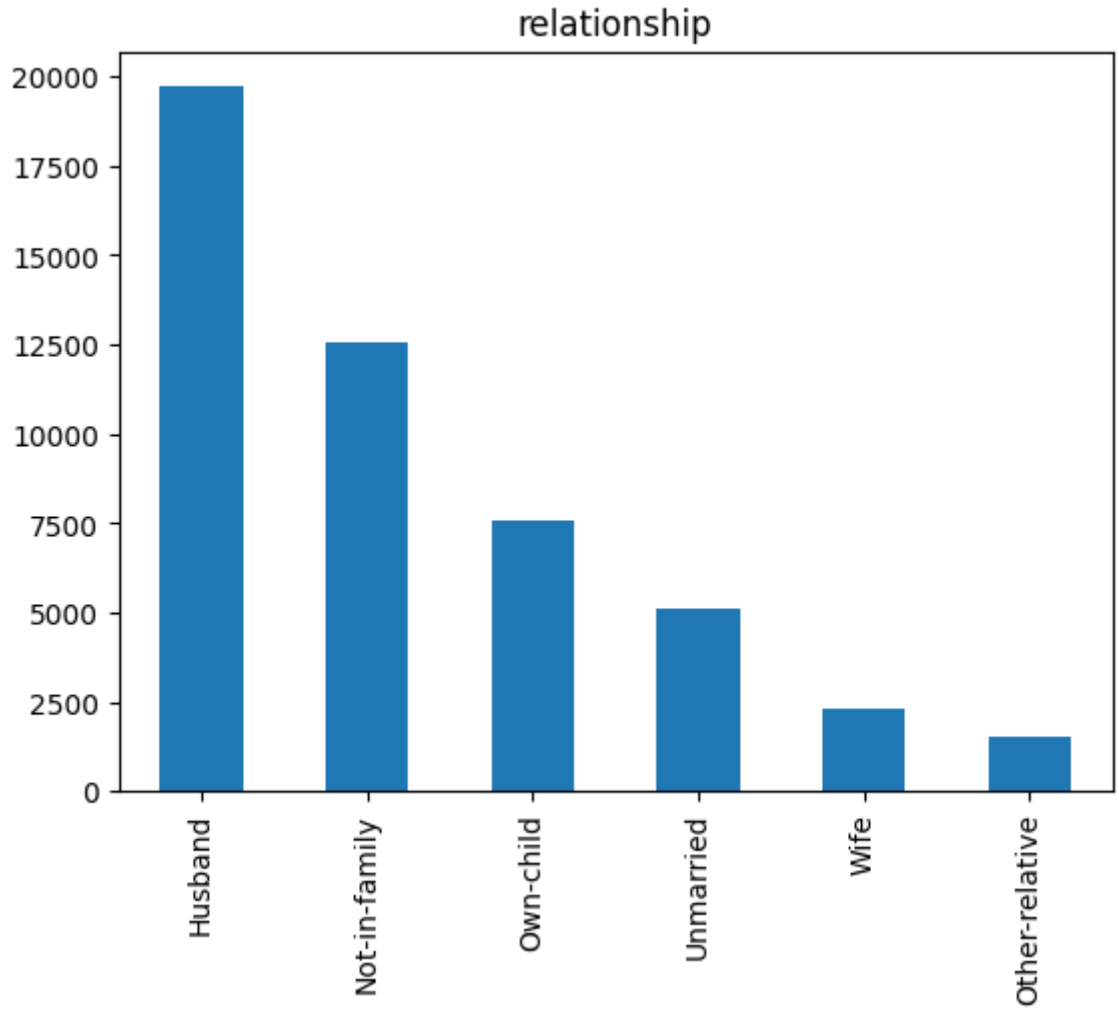
Keterangan dari distribusi occupation dapat dicermati dari tabel berikut.

Tabel 1.3: Distribusi occupation

occupation	sampe count	percentage
Prof-specialty	6172	12.6
Craft-repair	6112	12.5
Exec-managerial	6086	12.5
Adm-clerical	5611	11.5
Sales	5504	11.3
Other-service	4923	10.1
Machine-op-inspct	3022	6.2
?	2809	5.8
Transport-moving	2355	4.8

occupation	sampe count	percentage
Handlers-cleaners	2072	4.2
Farming-fishing	1490	3.1
Tech-support	1446	3.0
Protective-serv	983	2.0
Priv-house-serv	242	0.5
Armed-Forces	15	0.0

Pada tabel 1.3, distribusi semakin normal, dengan tren ringan *Right-skewed distribution*.



Gambar 1.4: Univariate analysis relationship vs income

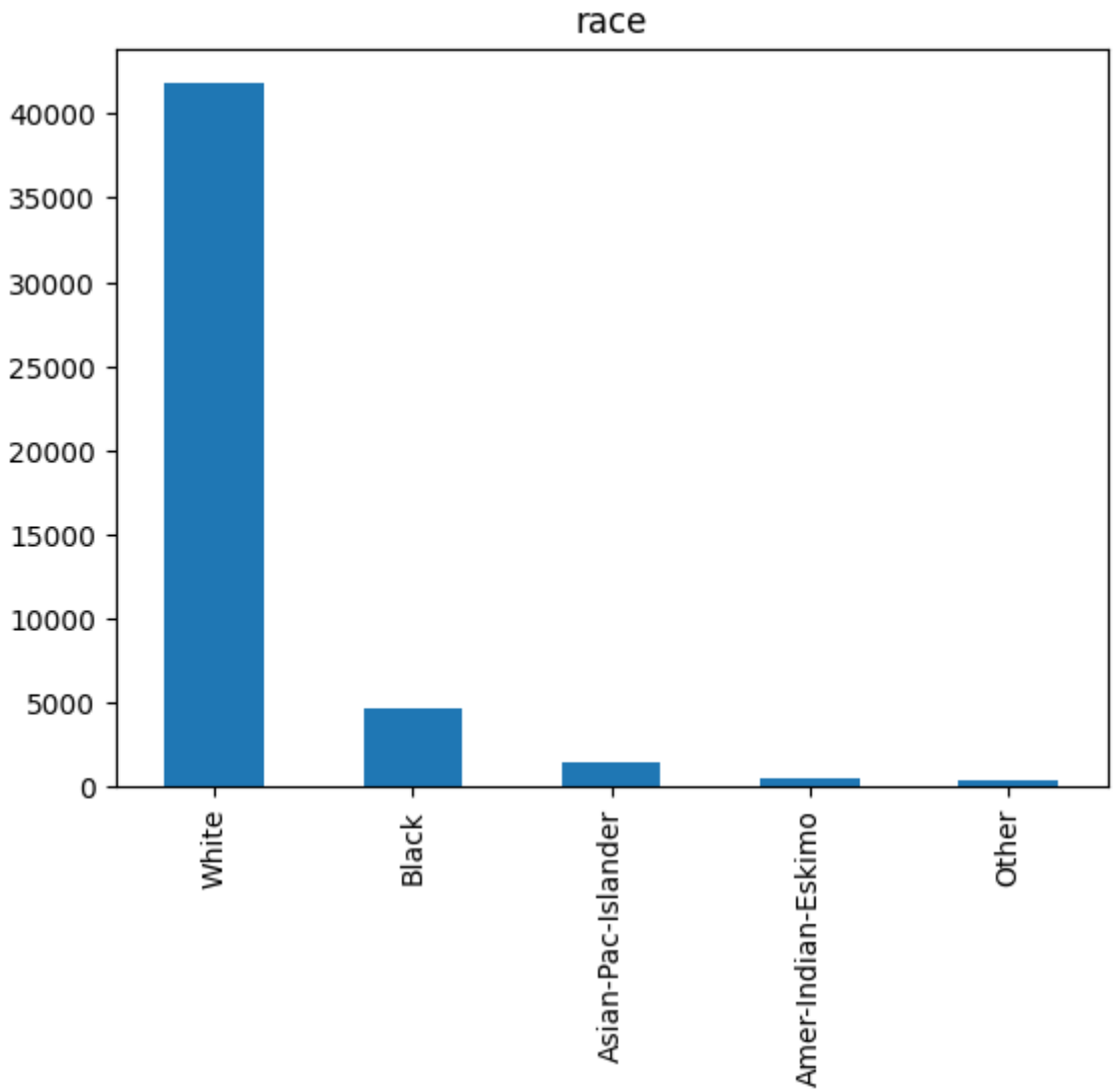
Keterangan dari distribusi relationship dapat dicermati dari tabel berikut.

Tabel 1.4: Distribusi relationship

relationship	sample count	percentage
Husband	19716	40.4

relationship	sample count	percentage
Not-in-family	12583	25.8
Own-child	7581	15.5
Unmarried	5125	10.5
Wife	2331	4.8
Other-relative	1506	3.1

Kemudian pada tabel 1.4, dapat dicermati bahwa tren dataset cukup seimbang, dengan nilai tertinggi Husband. Hasil ini dapat menyiratkan bahwa distribusi Gender akan lebih tinggi pada pria daripada wanita. Hal ini akan dikonfirmasikan pada Univariate Analysis pada fitur Gender.



Gambar 1.5: Univariate analysis race vs income

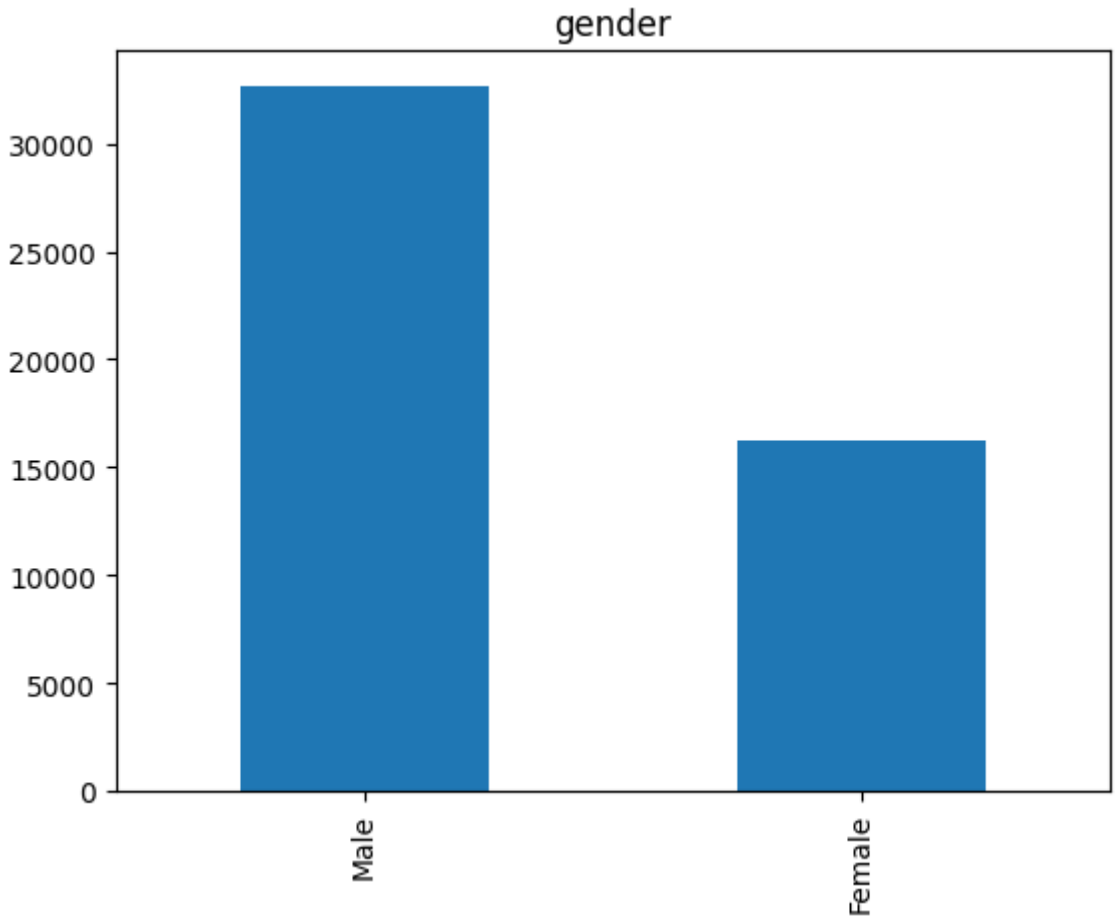
Keterangan dari distribusi race dapat dicermati dari tabel berikut.

Tabel 1.5: Distribusi race

race	sample count	percentage

race	sample count	percentage
White	41762	85.5
Black	4685	9.6
Asian-Pac-Islander	1519	3.1
Amer-Indian-Eskimo	470	1.0
Other	406	0.8

Dari tabel 1.5, dapat terlihat bahwa modus yang sangat jelas jatuh pada kategori White, dengan perbedaan hingga 75,9%.



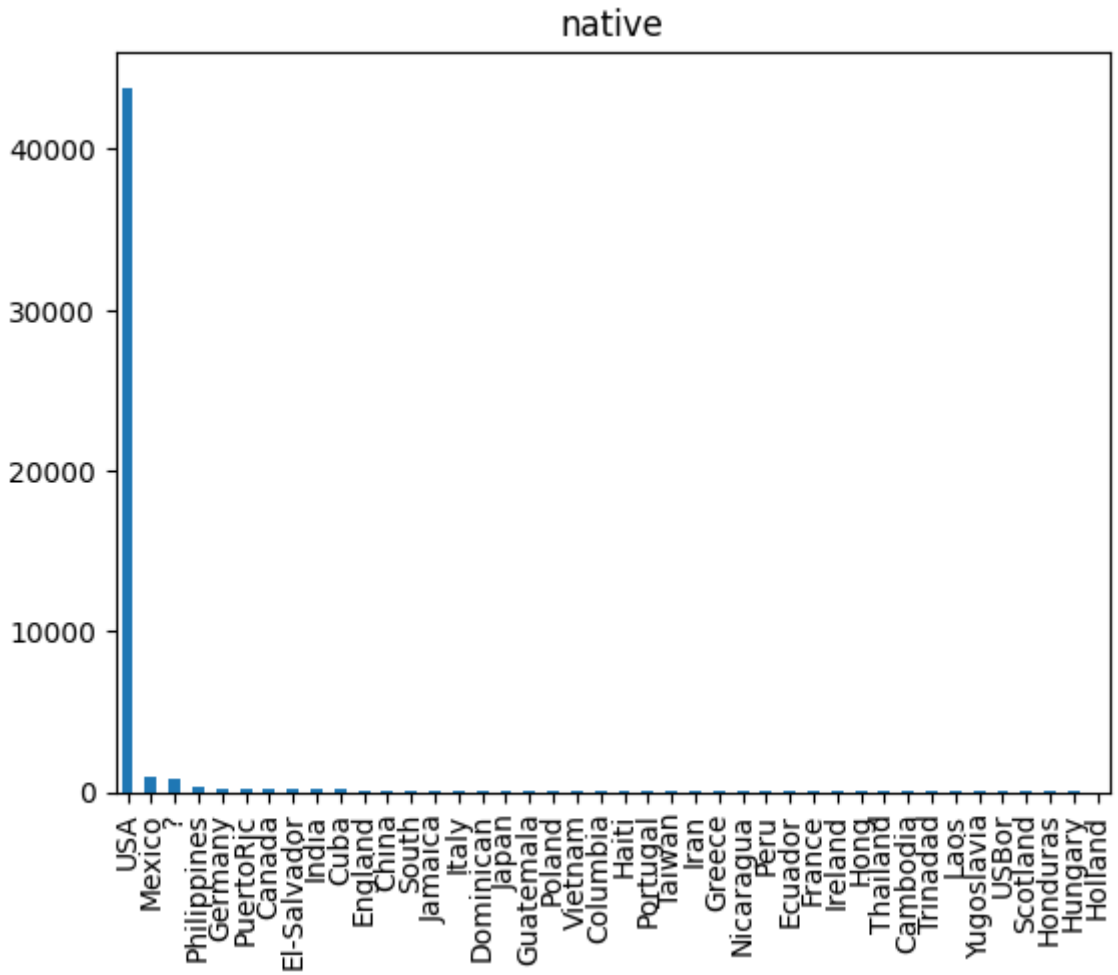
Gambar 1.6: Univariate analysis gender vs income

Keterangan dari distribusi gender dapat dicermati dari tabel berikut.

Tabel 1.6: Distribusi gender

gender	sample count	percentage
Male	32650	66.8
Female	16192	33.2

Hasil dari tabel 1.6 membenarkan teori dari tabel 1.4, dimana distribusi Male (pria) lebih besar dari Female (wanita).



Gambar 1.7: Univariate analysis native vs income

Keterangan dari distribusi native dapat dicermati dari tabel berikut.

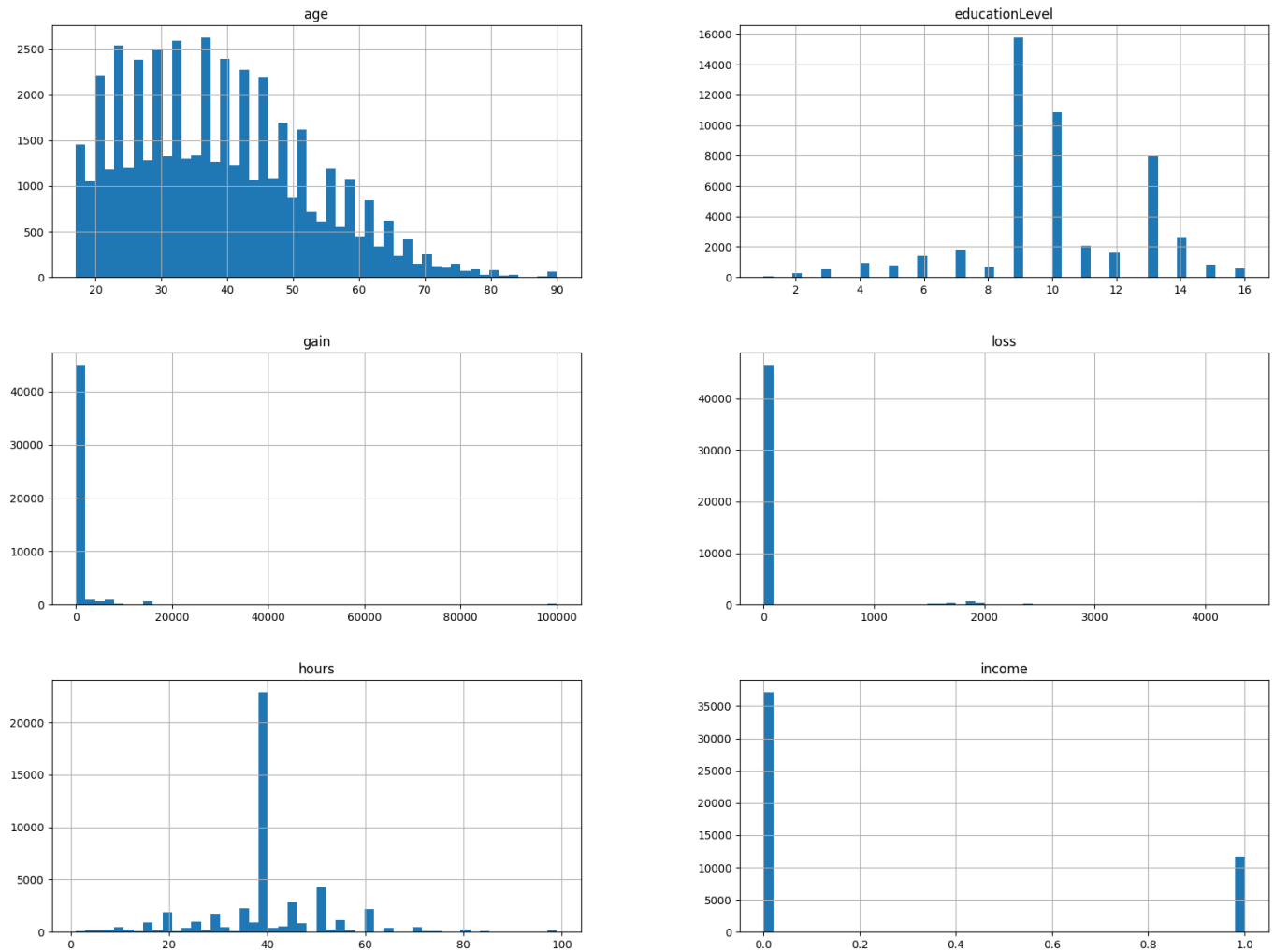
*nilai yang ditampilkan hanya 15 teratas, karena unique value yang terlalu banyak.

Tabel 1.7: Distribusi native

native	sample count	percentage
USA	43832	89.7
Mexico	951	1.9
?	857	1.8
Philippines	295	0.6
Germany	206	0.4
PuertoRic	184	0.4
Canada	182	0.4

native	sample count	percentage
El-Salvador	155	0.3
India	151	0.3
Cuba	138	0.3
England	127	0.3
China	122	0.2
South	115	0.2
Jamaica	106	0.2
Italy	105	0.2

Pada tabel 1.7 dapat dicermati bahwa modus jatuh pada kategori USA, dengan perbedaan persentase hingga 87,8% dengan kategori berikutnya.



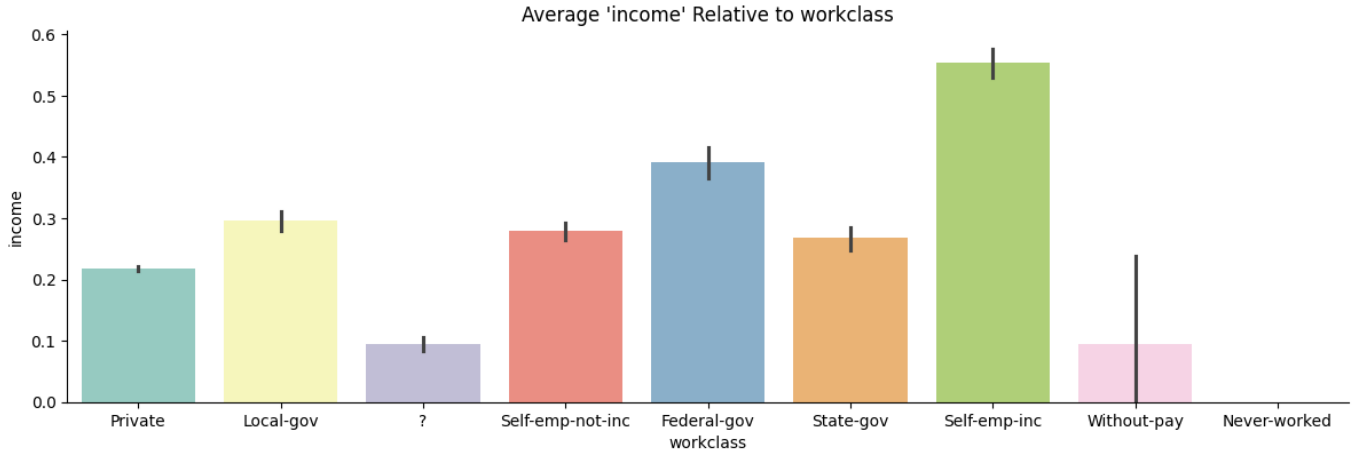
Gambar 1.8: Univariate analysis distribusi numerical features

Pada gambar 1.8, dapat dilihat bahwa distribusi numerikal mayoritas memiliki nilai modus dengan perbedaan yang sangat tinggi dibandingkan dengan nilai-nilai lainnya.

Analisis dan interpretasi hasil Univariate Analysis:

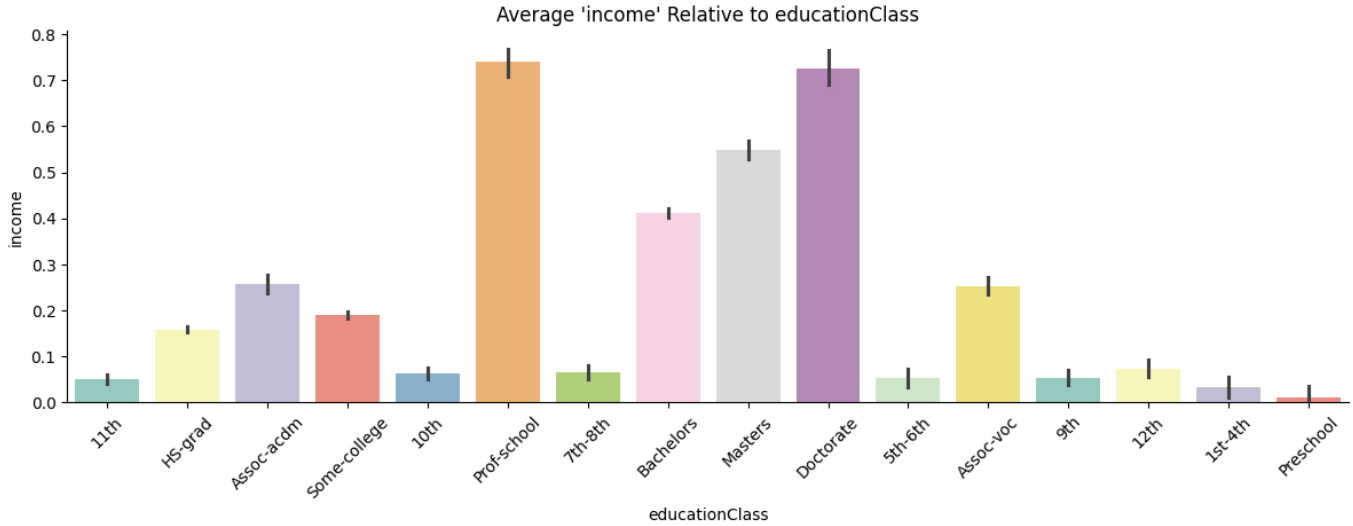
1. *Missing values* ditandai dengan '?'.
2. Mayoritas dari distribusi variabel sangat berat kepada modus, dengan perbedaan persentase yang sangat besar. Hal ini menyiratkan bahwa dataset ini dapat diatasi *missing values*-nya dengan metode statistika, seperti *Mode Imputation*.

Multivariate Analysis



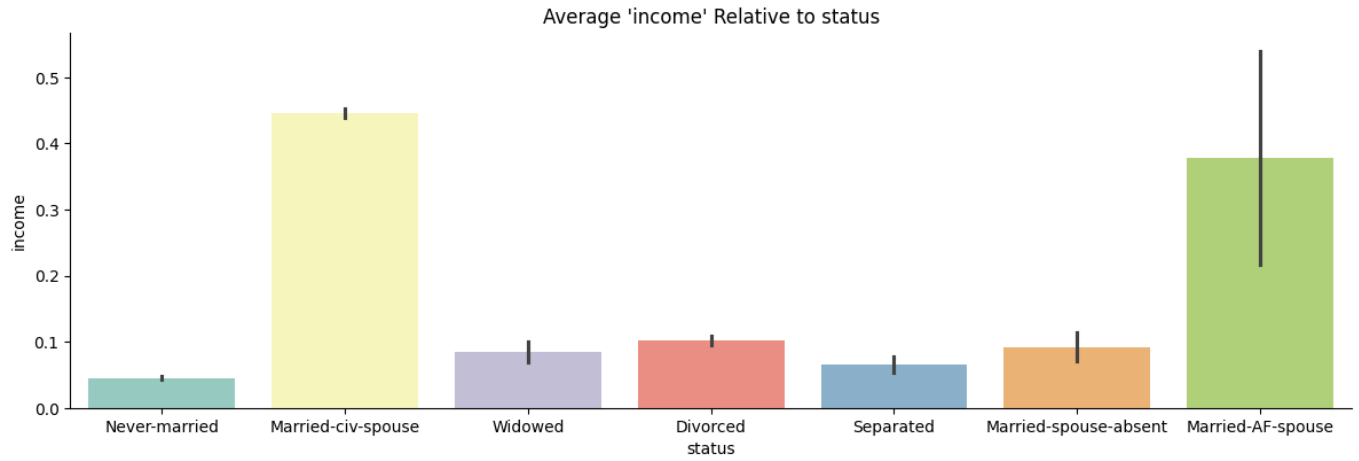
Gambar 2.0: Multivariate analysis workclass vs income

Pada workclass, dapat dilihat bahwa semua workclass memiliki income, kecuali tentunya never-worked. Rata-rata yang bekerja pada pemerintah (-gov) memiliki income yang lebih tinggi dibandingkan dengan workclass lain, namun self-emp-inc memiliki pendapatan paling tinggi.



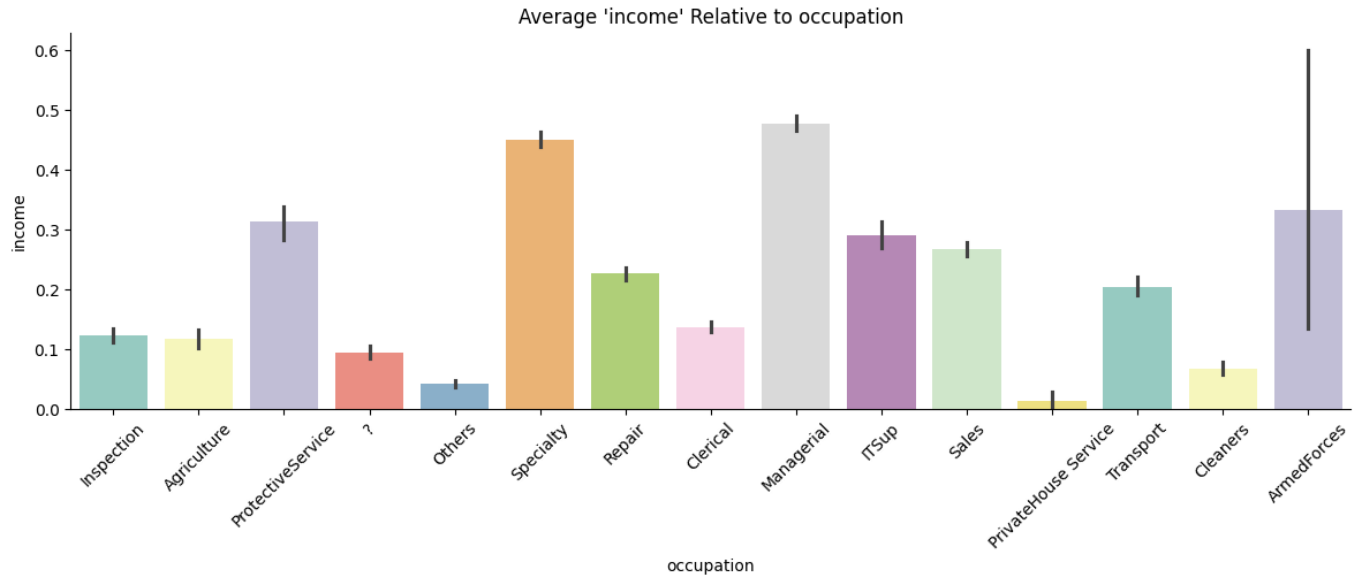
Gambar 2.1: Multivariate analysis educationClass vs income

Pada educationClass, dapat dilihat dengan jelas bahwa terdapat kesenjangan yang sangat tinggi mulai dari tingkat pendidikan S1 (bachelors), dan lebih tinggi lagi semakin tinggi gelar yang dimiliki (masters, doctorate, prof-school).



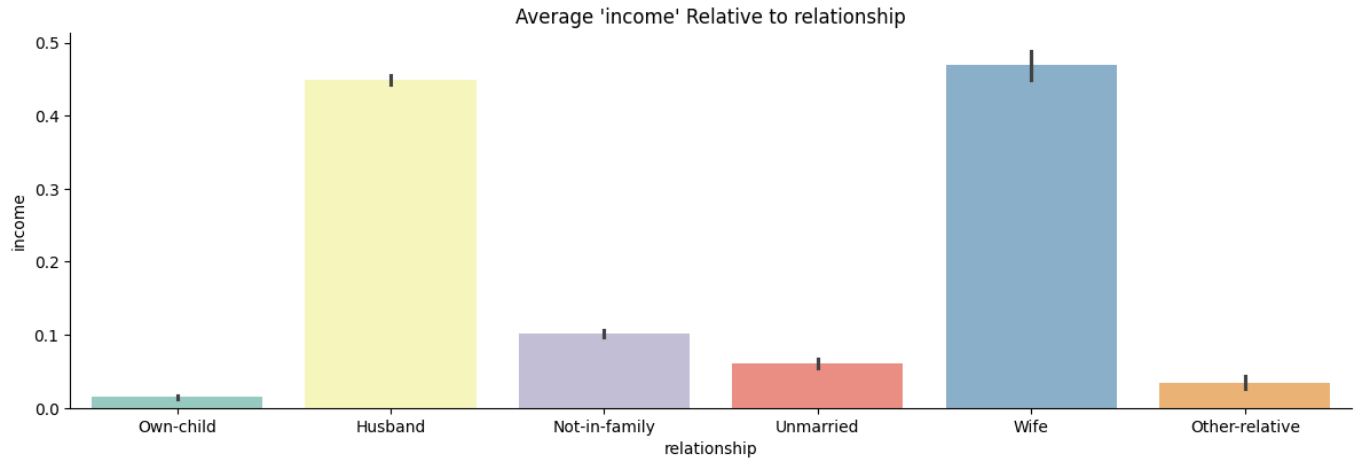
Gambar 2.2: Multivariate analysis status vs income

Pada status pernikahan, dapat dilihat pula tren bahwa data dengan status pernikahan stabil memiliki pendapatan yang lebih tinggi dibandingkan mereka yang tidak menikah, bercerai, berpisah, atau cerai mati.



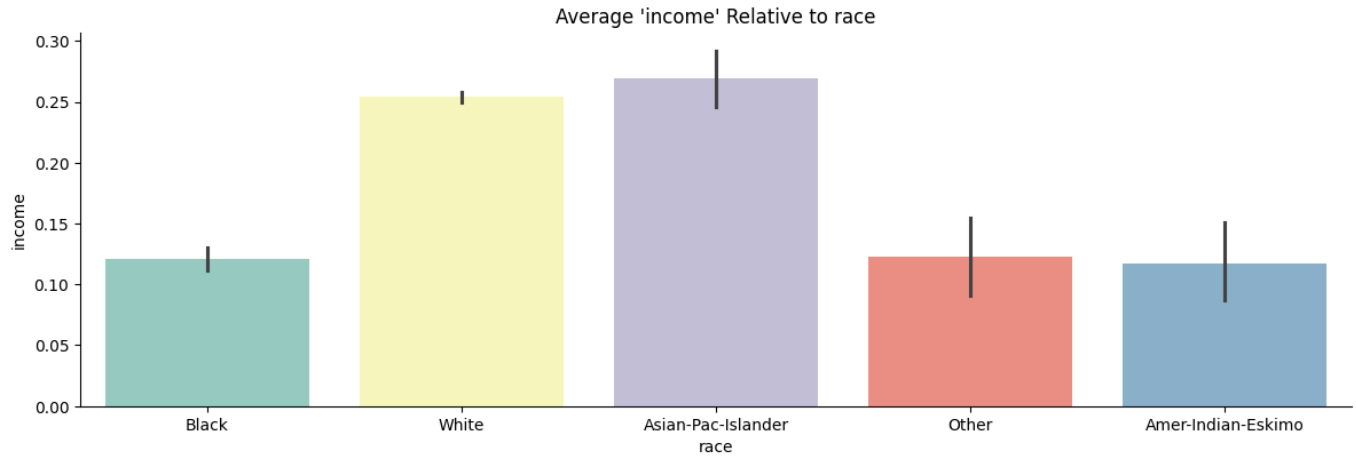
Gambar 2.3: Multivariate analysis occupation vs income

Pada occupation, tiga sektor kerja paling tinggi pendapatannya adalah Specialty (spesialis), manajerial (eksekutif), dan protective service (jasa keamanan). Selain itu, tidak terdapat tren yang secara jelas dapat di diamati.



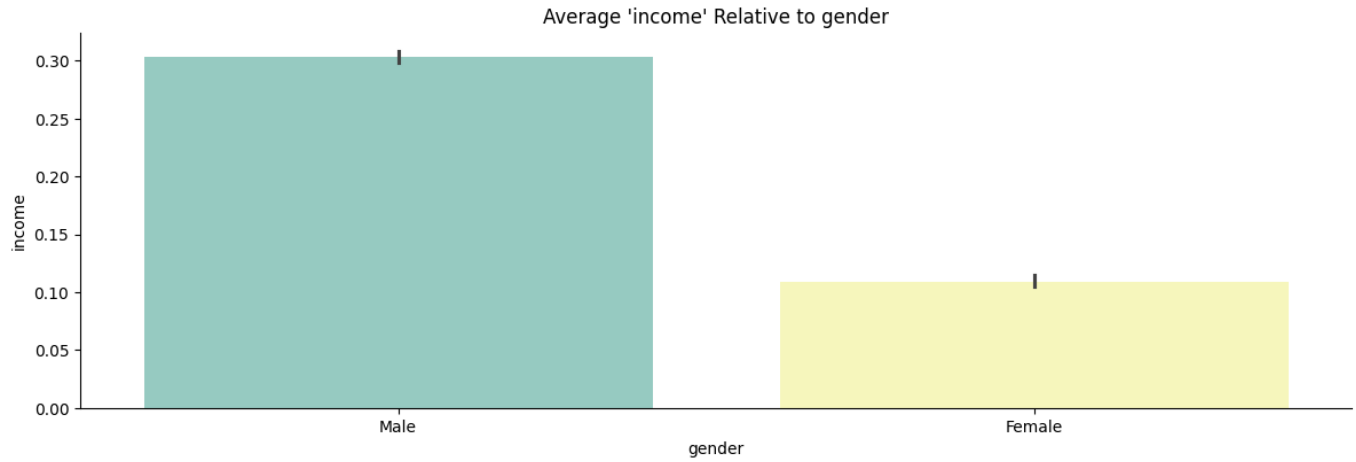
Gambar 2.4: Multivariate analysis relationship vs income

Pada relationship, dapat dilihat bahwa menguatkan hasil pengamatan status pernikahan, yang berstatus sebagai husband (suami) dan istri (istri) memiliki pendapatan yang tertinggi dari data-data lain.



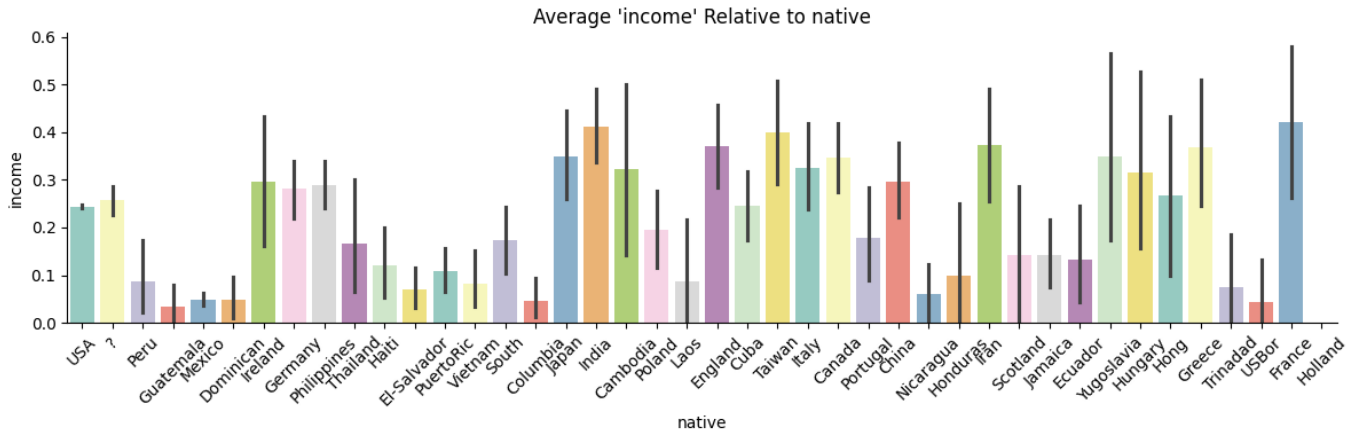
Gambar 2.5: Multivariate analysis race vs income

Pada race, dua ras dengan pendapatan tertinggi adalah White (putih) dan Asian-Pac (asia-pasifik)



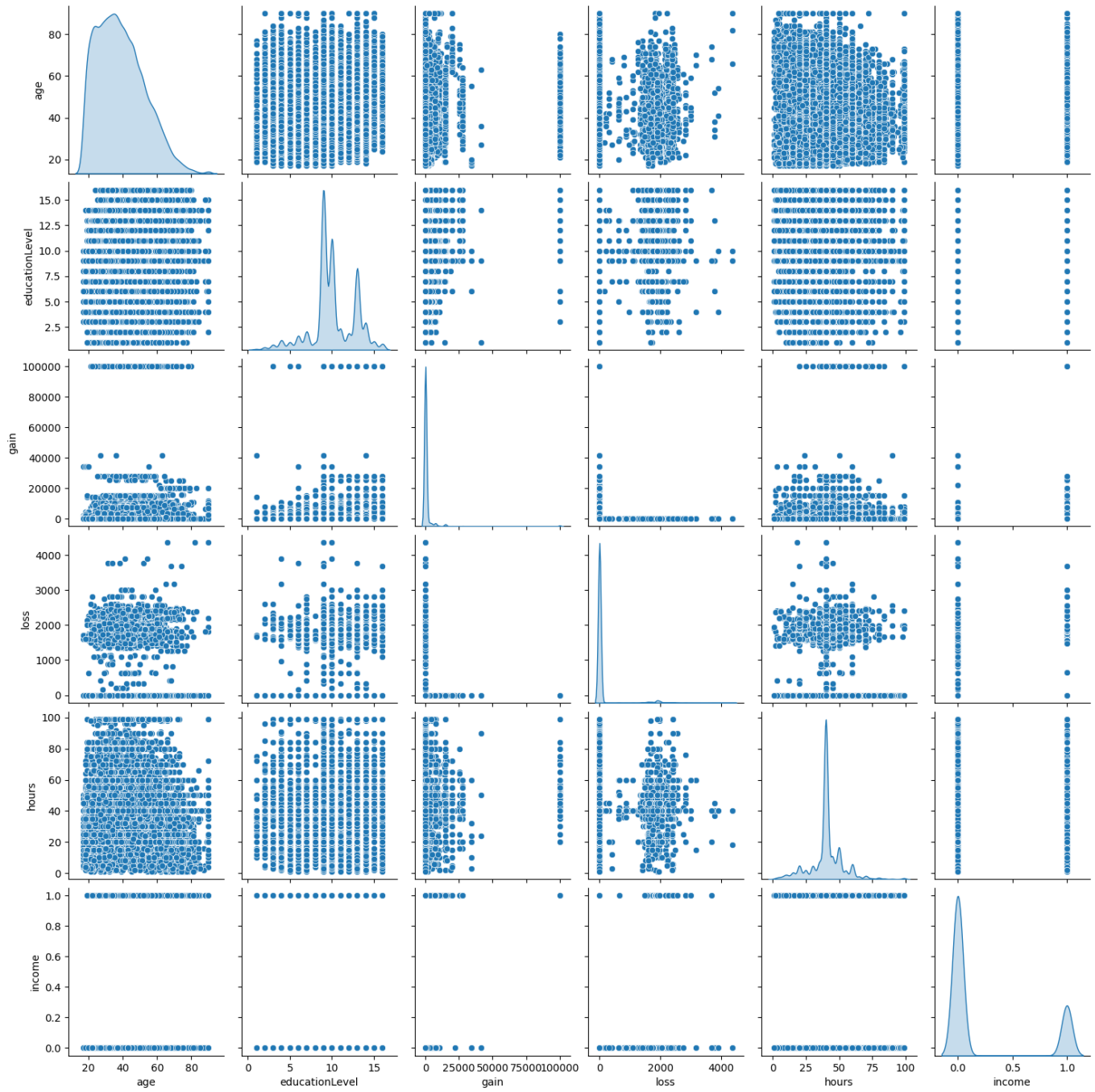
Gambar 2.6: Multivariate analysis gender vs income

Pada gender, dapat dilihat bahwa pendapatan laki-laki lebih tinggi dari pendapatan perempuan.



Gambar 2.7: Multivariate analysis native vs income

pada native (negara asal/buyut), dapat dilihat bahwa tren pendapatan kurang dapat diamati, artinya negara asal/buyut bukan faktor yang kuat yang dapat memengaruhi pendapatan.



Gambar 2.8: Multivariate analysis numerical-values vs income

Pada visualisasi fitur numerik terhadap income, dapat dilihat bahwa kurang terlihat pola dalam pengaruh fitur numerik. Hal ini menyiratkan bahwa nilai korelasi antar fitur numerik dan income tidak terlalu signifikan.

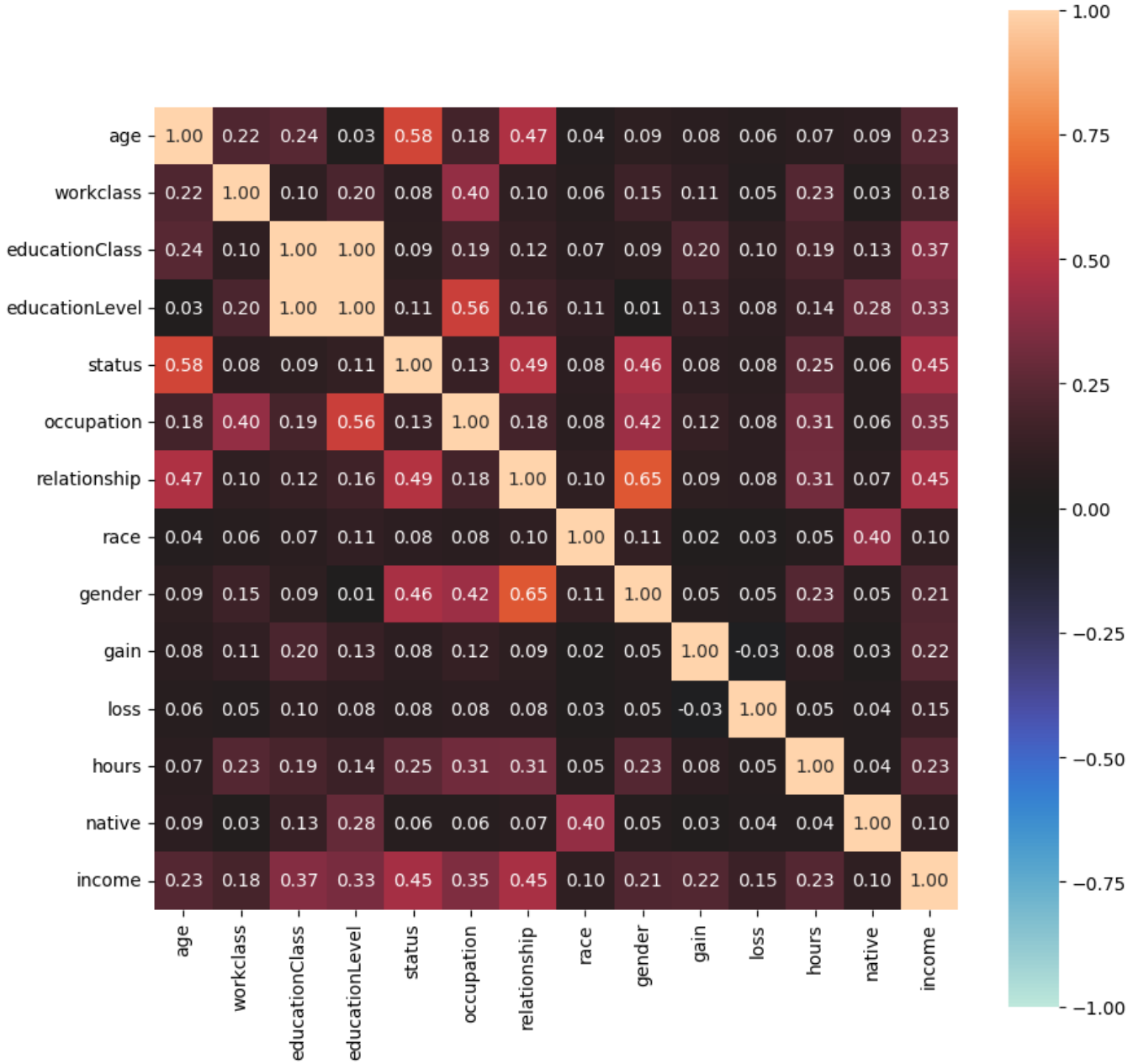
Analisis dan interpretasi hasil Multivariate Analysis:

1. Hasil dari Multivariate Analysis menunjukkan bahwa tren yang paling jelas, ada pada educationClass, status, dan relationship.

2. Pada gambar 2.1 yang memvisualisasikan analisis educationClass terhadap income, dapat dilihat bahwa terjadi lompatan yang tinggi dimulai dari pendidikan S1, dan income semaking tinggi seiring dengan semakin tingginya derajat pendidikan.

3. Pada gambar 2.2 dan 2.4, status dan relationship, terlihat bahwa sampel yang berada dalam hubungan pernikahan yang sehat akan memiliki income yang lebih tinggi dibandingkan status yang lain, dan ini

dikuatkan pada analisis terhadap relationship.



Gambar 3.0: Correlation matrix variabel-variabel

Hasil dari correlation matrix di atas menunjukkan bahwa fitur yang kuat seperti educationClass tidak terlalu jauh nilai korelasinya dengan nilai rendah, seperti capital-gain. Maka dari itu, tidak akan dilakukan drop kolom lagi.

Data Preparation

Secara berurutan, proses Data Preparation yang akan dilakukan adalah

1. Features Encoding
    - Features Encoding adalah pengubahan fitur-fitur kategorikal menjadi representasi numerik. Alasan diperlukan Features Encoding adalah sebagai berikut:
        - Meskipun terdapat model yang dapat mengolah data kategorikal, mayoritas dari model yang ada tetap membutuhkan input dalam bentuk numerik.

        - Lanjutan dari poin pertama, terkadang terdapat hubungan tingkatan (ordinalitas) yang ada dalam dataset. Merepresentasikan data kategorikal menjadi numerik dapat membantu mengawetkan hubungan ini, sehingga dimengerti secara implisit oleh model.
        - Model machine learning adalah model berbasis matematika dan statistika, sehingga pengolahan dalam bentuk numerik akan mempercepat proses training dan testing.
    - Teknik-teknik Features Encoding yang akan dilakukan pada proyek ini adalah:
        1. Ordinal Encoding
        2. One Hot Encoding

2. Train Test Split

    - Train Test Split adalah prosedur standar dalam melakukan Machine Learning, dimana dataset akan dibagi menjadi data latih (train) dan test. Hal ini dilakukan untuk memastikan bahwa model yang telah dilatih dapat menyelesaikan masalah pada data asli yang belum pernah ditemuinya.

3. KNN Imputation

    - KNN Imputation adalah salah satu proses yang akan diuji di proyek ini, dimana missing values akan diisi dengan melihat kemiripan fitur-fitur lain pada baris yang sama, dan menginput nilai baru. KNN Imputation dapat menyebabkan Data Leakage, sehingga perlu dianggap sebagai proses transformasi, dari pada pre-processing. Dikarenakan itulah KNN Imputation dilakukan setelah proses Train Test Split.

4. Standardization

    - Standardization juga merupakan prosedur standar dalam melakukan Machine Learning. Menurut Google Machine Learning Developers [5], standarisasi adalah metode untuk mengubah nilai-nilai numerik pada data ke skala yang sangat mirip, untuk meningkatkan performa dan stabilitas pada model saat train dan test.

Setelah penjelasan terkait proyek di atas, berikut adalah alur yang dilewati pada tahap Data Preparation ini.

1. Features Encoding. Input pada tahap ini adalah tiga DataFrame, dengan keterangan:

    - Satu DataFrame asli, masih terdapat missing values, dan
    - Dua DataFrame yang sama persis dimana sudah dilakukan proses Drop. Pada DataFrame asli dan satu DataFrame Drop, dilakukan Ordinal Encoding, dan One Hot Encoding pada satu DataFrame yang tersisa. Sehingga, output dari tahap ini adalah tiga DataFrame,
        - Dropped One Hot Encoded,
        - Dropped Ordinal Encoded, dan
        - Data Asli Ordinal Encoded.

2. Train Test Split Train Test Split dilakukan ke setiap DataFrame, memisahkan mereka menjadi x dan y.

3. KNN Imputation KNN Imputation hanya dilakukan ke DataFrame Data Asli Ordinal Encoded. KNN Imputation dilakukan setelah Train Test Split untuk mencegah terjadinya Data Leakage.

4. Standarisasi Pada tahap ini standarisasi dilakukan ke semua x_train dan x_test pada semua DataFrame, menyiapkan data tersebut untuk train dan test.

Model Development

Proses Model Development yang akan Penulis lakukan dapat dibagi menjadi beberapa tahap:

1. DataFrame initialization: menyiapkan dataframe untuk menyimpan hasil training model.
2. Pipeline Prep: menyiapkan pipeline untuk mempermudah proses training.
3. Model Training: melatih model.
4. Visualization

Pembuatan model dan training dilakukan di saat yang hampir bersamaan, dengan menggunakan metode Pipeline.

Model Explanation, pros and cons

Pada proyek ini, Penulis menggunakan empat algoritma, antara lain Random Forest, K-Nearest Neighbor, XGBoost, dan AdaBoost.

1. **Random Forest**

- Random Forest adalah Supervised Learning Based Algorithm, dan merupakan implementasi dari ensemble learning. Model Ensemble sendiri adalah sekelompok model yang bekerja sama untuk meningkatkan performa prediksi. Random Forest merupakan ensemble (kumpulan) dari banyak model *decision tree*, yang digabungkan dengan teknik *Bagging*, dimana dari masing-masing decision tree akan menghasilkan sebuah prediksi, *Bagging* pada kasus klasifikasi akan mengambil prediksi terbanyak pada seluruh pohon sebagai prediksi akhir. Hal ini sangat cocok untuk proyek ini karena sifat ini membuat Random Forest tidak rentan terhadap bias.
- Dalam proyek ini, parameter untuk algoritma Random Forest yang digunakan adalah:
    - n_estimator = 50 n_estimator merupakan parameter yang menentukan jumlah pohon (decision tree) yang akan digunakan pada algortima Random Forest. Penulis menggunakan 50 untuk percobaan dan akurasi yang didapatkan pun sudah cukup baik. Karena akurasi dari algoritma Random Forest bukan tujuan prioritas dari proyek, 50 pohon dirasa cukup.
    - max_depth = 16 max_depth merupakan parameter yang menentukan kedalaman pohon, lebih tepatnya seberapa banyak pohon dapat membelah untuk melakukan komputasi/pengamatan. Sama seperti parameter pertama, akurasi sudah baik, sehingga max_depth = 16 dirasa sudah cukup.
    - random_state = 55 random_state merupakan parameter untuk mengatur generator random untuk memastikan setiap jalannya proses training/testing konsisten.
    - n_jobs = -1 n_jobs merupakan parameter yang digunakan untuk mengatur berapa jumlah pekerjaan yang berjalan secara paralel. n_jobs = -1 berarti semua proses berjalan secara paralel.
- Dari penjelasan singkat terkait algoritma Random Forest di atas, beberapa kelebihan dan kekurangannya adalah sebagai berikut:
    - Kelebihan:
    1. Dapat digunakan untuk klasifikasi maupun regresi.
    2. Dapat digunakan pada data kategorikal maupun numerikal. Dapat bekerja tanpa scaling dan transformation sekalipun.

3. Dapat melakukan feature selection secara implisit, dan tahan terhadap outliers.
4. Dapat bekerja pada problem linier maupun non-linier, tahan pada bias, dan akurat.

- Kekurangan:

1. Mahal secara komputasi, terutama pada dataset besar.
2. Tidak fleksibel, tidak terlalu banyak yang dapat diatur sendiri. (Kurang efektif untuk hyperparameter tuning)

- **K-Nearest Neighbor** KNN bekerja dengan menggunakan kesamaan fitur untuk memprediksi nilai dari setiap data yang baru. Secara matetmatis, algortima KNN menghitung jarak (Euclidean) antara setiap poin data dan memilih klasifikasi berdasarkan mayoritas tetangga terdekat. Jika dilihat dari kompleksitas alrgortima, KNN dapat dikategorikan dalam algoritma yang lebih sederhana. KNN menjadi algoritma yang sering digunakan, termasuk pada proyek ini, karena sifatnya yang sederhana, dan tidak memiliki asumsi. Namun perlu diperhatikan pada curse of Dimensionality, yaitu tidak terlalu efektif pada dataset dengan fitur yang sangat banyak.

- Dalam proyek ini, parameter yang digunakan untuk algoritma KNN adalah:

  - n_neighbors = 10 n_neighbors adalah parameter untuk mengatur berapa jumlah tetangga yang akan dipertimbangkan untuk proses penghitungan jarak Euclidean. Penulis memilih 10 tetangga, dikarenakan terdapat beberapa fitur yang sangat lemah korelasinya, sehingga diharapkan bahwa fitur lemah tersebut tidak diprioritaskan dalam proses training.

- Dari penjelasan singkat terkait algoritma KNN di atas, beberapa kelebihan dan kekurangannya adalah sebagai berikut:

  - Kelebihan:

1. Sederhana, intuitif, dan mudah digunakan.
2. Sangat efektif pada problem multi-class.
3. Mudah dituning karena jarak dapat diatur antara Euclidean, Hamming, Manhattan, Minkowski, dst.
4. Tidak ada asumsi.

  - Kekurangan:

1. Curse of Dimensionality, lemah terhadap dataset dengan dimensi besar.
2. KNN bukan algoritma tercepat.
3. Scaling dan Transformasi wajib dilakukan.
4. Lemah terhadap outliers, missing values, dan imbalanced dataset.

- **XGBoost** eXtreme Gradient Boosting (XGBoost) adalah model berbasis ensemble learning dan boosting, dan merupakan turunan dari framework Gradient Boosting Decision Tree. Boosting sendiri merupakan proses yang membuat dan menggabungkan weak learner models secara iteratif hingga menghasilkan suatu strong learner model. Cara boosting bekerja adalah membangun model secara berurutan dengan fokus pada data yang salah sebelumnya. Dalam kasus XGBoost, atau Gradient Boosting pada umumnya, iterasi pembuatan dan penggabungan model dilakukan berdasarkan gradien error (gradient of error) atau yang sering disebut Gradient Loss Function. Secara fundametnal, perbedaan XGBoost dan AdaBoost adalah loss function yang digunakan.

- Dari penjelasan singkat terkait algoritma XGBoost di atas, beberapa kelebihan dan kekurangannya adalah sebagai berikut:

    - Kelebihan:
    1. Cepat.
    2. Sangat tahan terhadap outliers
    3. Fleksibel, mampu beradaptasi pada variasi data yang tinggi.
    4. Built-in regularisasi.
    5. Akurasi tinggi.
    - Kekurangan:
    1. Mahal secara komputasi.
    2. Kompleks.
    3. Mirip seperti Random Forest, XGBoost kurang bisa dilakukan hyperparameter tuning.

- **AdaBoost** Adaptive Boosting (AdaBoost) adalah Supervised Learning Based Algorithm yang mengimplementasikan ensemble learning dan boosting. Mirip dengan XGBoost, AdaBoost bekerja dengan membuat sejumlah weak learners berbasis decision tree, kemudian membuat model-model berikutnya dengan jawaban salah dari model sebelumnya. Berbeda dengan XGBoost, AdaBoost menggunakan Exponential Loss Function.

- Dalam proyek ini, parameter untuk algoritma AdaBoost yang digunakan adalah:

    - n_estimators = 50 Mirip dengan algoritma Random Forest, AdaBoost menerima parameter n_estimators untuk mengatur jumlah decision trees yang akan dihasilkan. Penulis menggunakan 50 sebagai percobaan, dan hasil yang didapatkan sudah cukup baik.
    - random_state = 123 random_state merupakan parameter untuk mengatur generator random untuk memastikan setiap jalannya proses training/testing konsisten.

- Dari penjelasan singkat terkait algoritma AdaBoost di atas, beberapa kelebihan dan kekurangannya adalah sebagai berikut:

    - Kelebihan:
    1. Potensi tinggi, meningkatkan performa model secara signifikan.
    2. Murah secara komputasi.
    3. Fleksibel, dapat digunakan di berbagai kasus.
    4. Dapat digunakan bersama model lain.
    - Kekurangan:
    1. Varians tinggi.
    2. Kurang efektif dalam problem linier.
    3. Lebih rentan terhadap outliers.

Evaluation

Dalam proyek ini, beberapa metrik evaluasi yang digunakan adalah sebagai berikut. Sebelum memasuki penjelasan metrik lebih lanjut, perlu dipahami bahwa:

- TN = True Negative, data negatif yang diprediksi negatif (benar)
- TP = True Positive, data positif yang diprediksi negatif (benar)
- FN = False Negative, data negatif yang diprediksi positif (salah)
- FP = False positive, data positif yang diprediksi negatif (salah)

Pertama, Accuracy, yang dapat dihitung dengan rumus: $$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$

Accuracy merepresentasikan angka data yang benar di prediksi dibagi total jumlah data. Idealnya, akurasi memberikan ide seberapa baik model dapat memprediksi data, namun kekurangan dari metrik ini adalah kurang adilnya metrik jika dataset yang digunakan *unbalanced*.

Berikutnya adalah precision, yang dapat dihitung dengan rumus:

$$Precision = \frac{TP}{FP + TP}$$

Precision adalah rasio prediksi benar positif (TP) dari total prediksi positif. Semakin tinggi presisi, artinya semakin sedikit jumlah prediksi positif salah (FP).

Kemudian terdapat metrik Recall, seperti berikut.

$$Recall = \frac{TP}{TP + FN}$$

Recall menghitung nilai dari betulnya prediksi positif dari jumlah aktual positif. Semakin tinggi nilai recall berarti semakin sedikit False Negatives (FN).

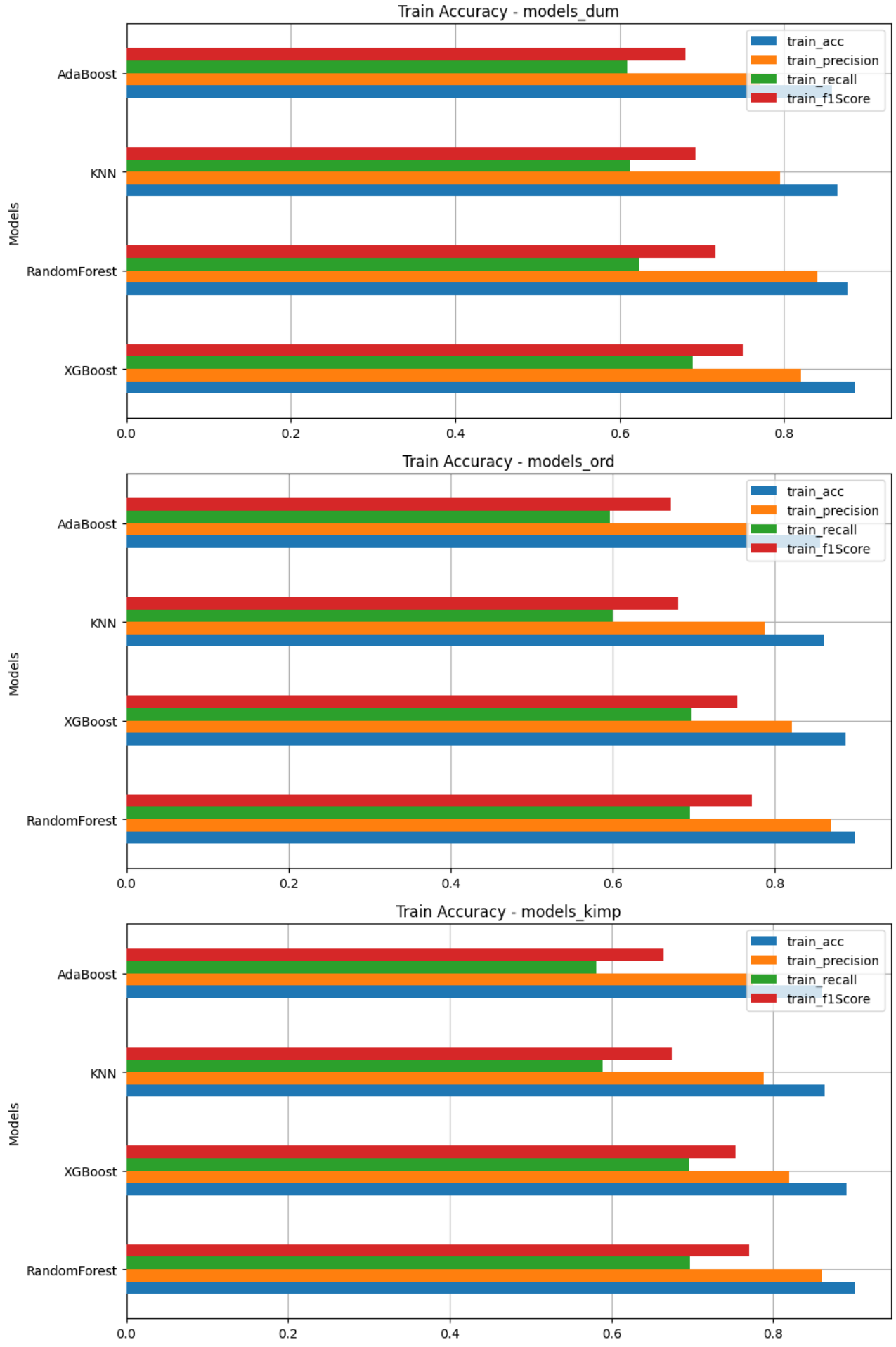
Setelah menghitung metrik Accuracy, Precision, dan Recall, kita bisa mencari nilai F1 Score dengan rumus berikut.

$$F1 Score = 2* \frac{Precision * Recall}{Precision + Recall}$$

F1 Score adalah sebuah nilai harmonis yang menggunkan presisi dan Recall, artinya nilai F1 Score yang tinggi memiliki Precision dan Recall yang tinggi.

Model Training Results

- Visualization



Gambar 4.0: Hasil model training

Table 2.0 Evaluasi training Accuracy, Precision, Recall, dan F1 Score data Dropped One Hot Encoded

Model	Accuracy	Precision	Recall	F1-Score
KNN	0.864862	0.794974	0.612125	0.691669
RandomForest	0.877515	0.840669	0.623536	0.716003
XGBoost	0.88582	0.821095	0.689026	0.749285
AdaBoost	0.858154	0.770313	0.608653	0.680007

Table 2.1 Evaluasi training Accuracy, Precision, Recall, dan F1 Score data Dropped Ordinal Encoded

Model	Accuracy	Precision	Recall	F1-Score
KNN	0.860758	0.787511	0.599424	0.680714
RandomForest	0.89845	0.868751	0.69488	0.772148
XGBoost	0.887442	0.8215	0.696864	0.754067
AdaBoost	0.855402	0.767718	0.596547	0.671394

Table 2.2 Evaluasi training Accuracy, Precision, Recall, dan F1 Score data KNN Imputed Ordinal Encoded

Model	Accuracy	Precision	Recall	F1-Score
KNN	0.86414	0.787736	0.589262	0.674195
RandomForest	0.900721	0.859779	0.697597	0.770243
XGBoost	0.891053	0.82002	0.696071	0.752979
AdaBoost	0.859909	0.775178	0.581347	0.664414

1. Dari hasil training, dua model terbaik adalah Random Forest dan XGBoost. Hal ini dibuktikan bukan hanya dengan training accuracy yang sangat tinggi, melainkan F1 Score yang sangat tinggi juga. F1 Score yang tinggi merupakan indikasi bahwa model tersebut jarang melakukan kesalahan dalam prediksi nilai benar maupun salah. Penjelasan F1 Score akan dijelaskan lebih dalam pada tahap Evaluation.

2. Dapat diamati juga bahwa dari dataset drop One Hot Encoded dan Ordinal Encoded, terdapat perubahan yang cukup signifikan (peningkatan hingga dua persen) pada training accuracy. Hal ini menunjukkan pentingnya mengawetkan ordinalitas pada dataset (jika ada), yang dalam hal ini ada terutama pada fitur educationClass.

3. Pengujian prediksi terakhir akan menggunakan test data dari test_ord, dikarenakan dataset ini mengimplementasikan Ordinal Encoding dan memiliki tingkat integritas yang lebih tinggi dari dataset imputasi. Mengingat bahwa jumlah dataset ini melimpah dari awal, metode Drop tetap menjadi metode Data Handling terbaik karena pengawetan integritasnya.

Testing dan Prediksi Akhir

Melihat hasil dari hasil training diatas, terutama dari test_dum dan test_ord, dapat dilihat bahwa terjadi peningkatan akurasi dari dua model dengan akurasi tertinggi, yaitu Random Forest dan XGBoost. Hal ini membenarkan bahwa penting untuk menyimpan fitur ordinality (jika ada) yang dalam hal ini ada pada dataset, terutama educationClass.

Secara konsisten, model Random Forest dan XGBoost menghasilkan akurasi dan F1 Score yang tinggi, sebuah indikasi dari model yang unggul.

Pengujian prediksi terakhir akan menggunakan test data dari test_ord dikarenakan dataset ini mengimplementasikan metode Drop dan Ordinal Encoding sehingga dataset ini menjaga sifat ordinalitasnya dan memiliki tingkat integritas yang lebih tinggi dari dataset imputasi. Mengingat bahwa jumlah dataset ini melimpah dari awal, metode Drop tetap menjadi metode Data Handling terbaik karena pengawetan integritasnya.

Metode Testing akan sama dengan metode Training, menggunakan Pipeline untuk meningkatkan efisiensi Testing.

Hasil Testing

Table 3.0 Evaluasi test Accuracy, Precision, Recall, dan F1 Score data Dropped One Hot Encoded

Model	Accuracy	Precision	Recall	F1-Score
KNN	0.858943	0.770925	0.619469	0.686948
RandomForest	0.928366	0.930556	0.770796	0.843175
XGBoost	0.940084	0.909438	0.844248	0.875631
AdaBoost	0.86845	0.772126	0.671681	0.71841

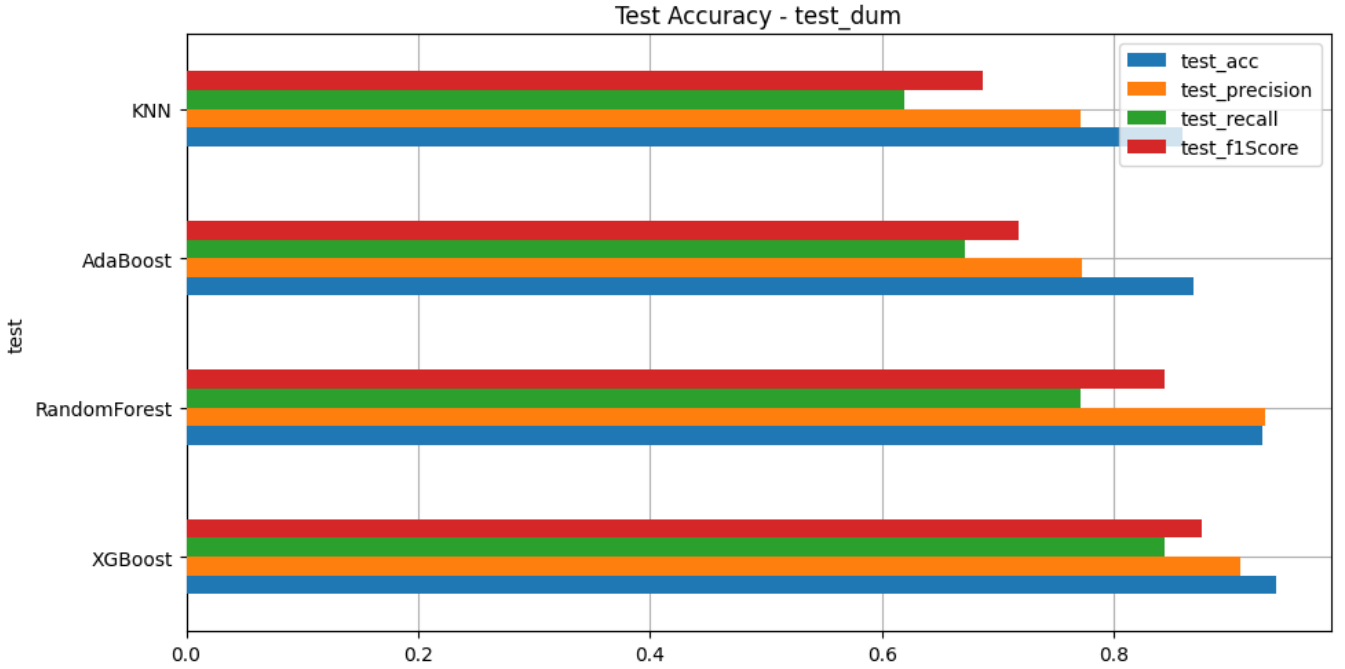
Table 3.1 Evaluasi test Accuracy, Precision, Recall, dan F1 Score data Dropped Ordinal Encoded

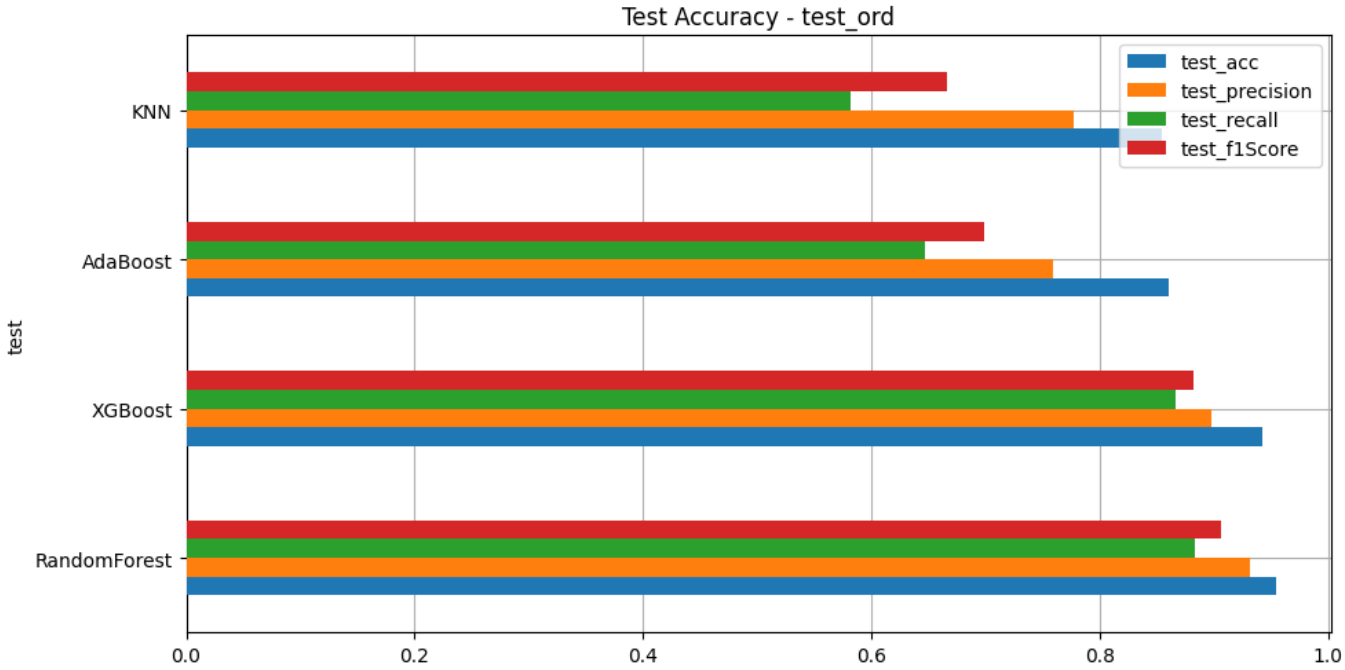
Model	Accuracy	Precision	Recall	F1-Score
KNN	0.853858	0.77686	0.582301	0.665655
RandomForest	0.954455	0.93097	0.883186	0.906449
XGBoost	0.941853	0.897342	0.866372	0.881585
AdaBoost	0.860491	0.759086	0.646903	0.698519

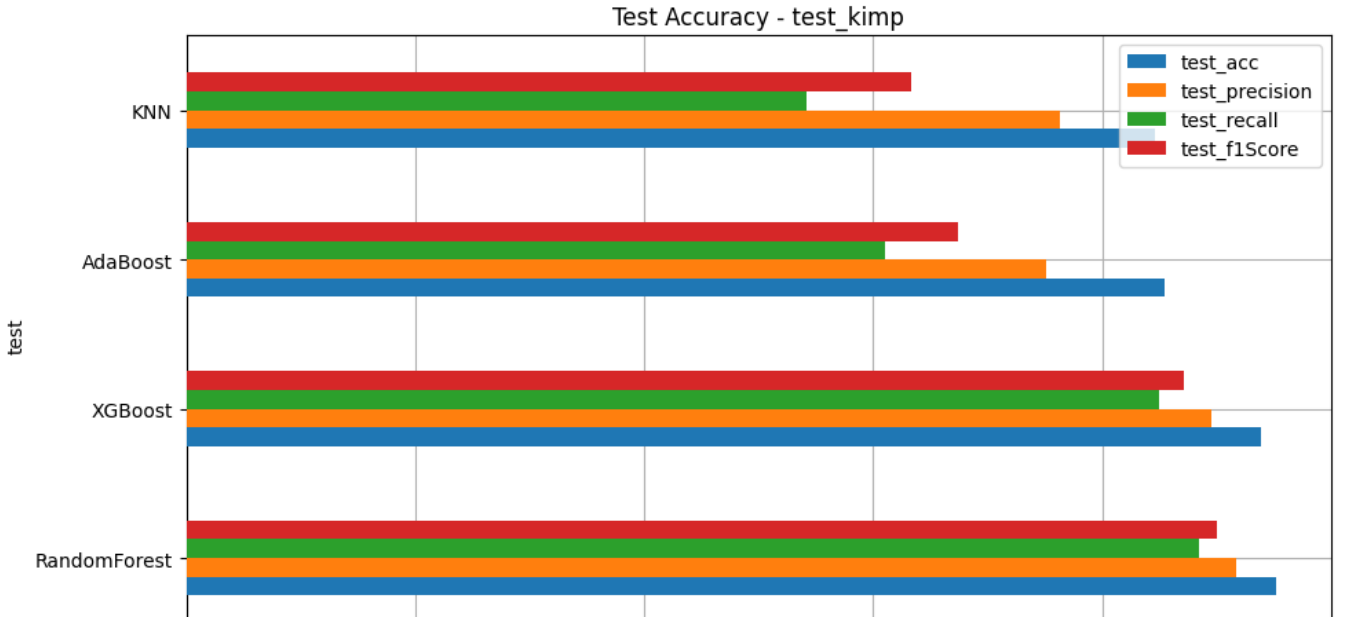
Table 3.2 Evaluasi training Accuracy, Precision, Recall, dan F1 Score data KNN Imputed Ordinal Encoded

Model	Accuracy	Precision	Recall	F1-Score
KNN	0.845855	0.762911	0.541216	0.63322
RandomForest	0.951894	0.917098	0.884263	0.900382
XGBoost	0.938588	0.895522	0.849292	0.871795
AdaBoost	0.854452	0.751025	0.610325	0.673404

Ringkasan dan visualisasi dari metode-metode data handling, dan akurasinya dapat dilihat sebagai berikut.

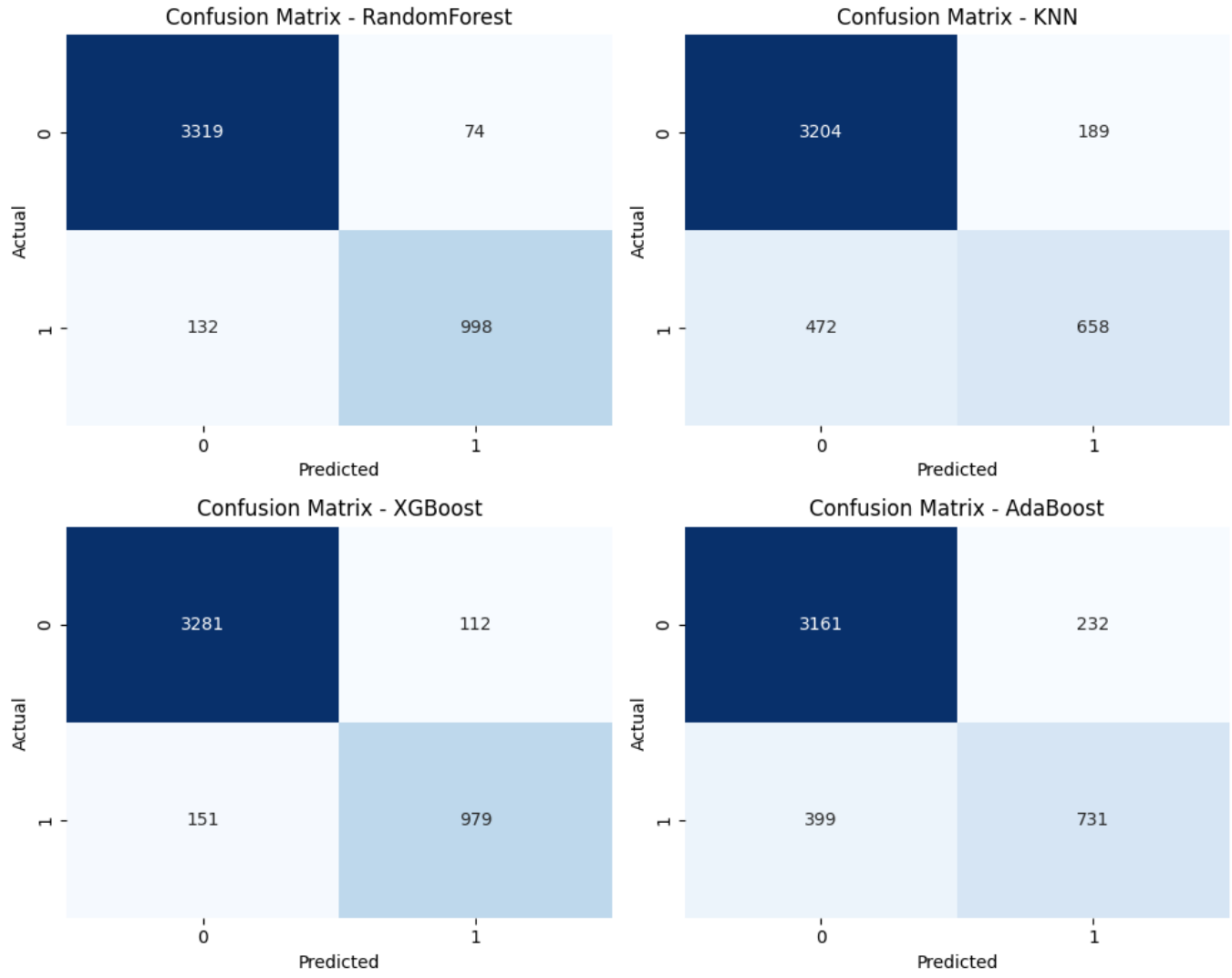






Gambar 5.0: Hasil testing

Hasil tes yang akan ditinjau, dan digunakan untuk prediksi akhir, adalah dari test data dengan Dropped Encoded. Hal ini dikarenakan metode Drop adalah metode terbaik dalam menjaga integritas data, dimana hasilnya dapat dilihat dari visualisasi Confusion Matrix berikut.



Gambar 5.1: Confusion matrix testing

Hasil dan Kesimpulan Proyek

Berdasarkan dari hasil Data Understanding, Data Preparation, Model Development, dan Evaluation, kita dapat menyimpulkan beberapa hal berikut.

1. Menjawab Problem Statement 1: Fitur-fitur yang paling berpengaruh terhadap income pada problem ini adalah educationClass, status, dan relationship.
2. Menjawab Probelm Statement 2: **Ya, income dapat diprediksi,** dan berdasarkan nilai *Accuracy, Recall, Precision, dan F1 Score*, model Random Forest adalah model dengan performa terbaik, mencapai akurasi hingga 95,4%, presisi hingga 93%, recall 88,3%, dan F1Score 90,6%. Hal ini merupakan salah satu bentuk kelebihan dari Random Forest yaitu sifatnya yang melakukan *feature selection* secara implisit. Ditambah lagi sifat yang diwariskan dari *ensemble learning*, yaitu tahan pada bias dan overfitting.

3. Terdapat pengaruh yang dapat diamati dalam bentuk peningkatan performa model jika pengawetan ordinalitas (jika ada dalam dataset) dilakukan.
4. *KNN Imputation* memiliki dampak yang kurang signifikan jika dibandingkan dengan poin nomor 3, namun performa tetap meningkat.

Berdasarkan hasil-hasil penemuan di atas, dapat disimpulkan bahwa proyek ini berhasil dan berjalan sesuai dengan keinginan Penulis, yaitu menjawab Problem Statements dan mencapai Predictive Modelling Goals yang dirumuskan, serta berhasil mengobservasi perbedaan ordinalitas dan pengaruh KNN Imputation terhadap dataset.

Referensi

[Dataset] Becker, Barry and Kohavi ,Ronny. (1996). Adult. UCI Machine Learning Repository. https://doi.org/10.24432/C5XW20.

[1] Badan Pusat Statistika Indonesia. (2023). Diakses dari https://www.bps.go.id/id/pressrelease/2023/11/06/2002/tingkat-pengangguran-terbuka--tpt--sebesar-5-32-persen-dan-rata-rata-upah-buruh-sebesar-3-18-juta-rupiah-per-bulan.html .

[2] R. Pramudjasi, Juliansyah, D. Lestari. (2019). Effect of population and education and wages on unemployment in paser regency. Journal of Faculty of Economics and Business Universitas Mulawarman. Diakses dari https://journal.feb.unmul.ac.id/index.php/KINERJA/article/download/5284/472.

[3] S.A.E. Sari, F.W.Pangestuty. (2020). Analisis Pengaruh jumlah Penduduk, Tingkat Pendidikan, dan Produk Domestik Regional Bruto Terhadap Tingkat Pengangguran Terbuka di Provinsi Jawa Timur Tahun 2017-2020. Journal of Developement Econoic and Social Studies. Diakses dari https://jdess.ub.ac.id/index.php/jdess/article/download/78/57/373.

[4] B. Jepchumba. (2020). Getting started with using Visual Machine Learning Tools for building your Machine Learning Models. Microsoft Community Hub. Diakses dari https://techcommunity.microsoft.com/t5/educator-developer-blog/getting-started-with-using-visual-machine-learning-tools-for/ba-p/3578397.

[5] Google Machine Learning Developers. Normalization. Diakses dari https://developers.google.com/machine-learning/data-prep/transform/normalization#:~:text=The%20goal%20of%20normalization%20is,training%20stability%20of%20the%20model.

Daftar Pustaka

[1] P. Schmitt, J. Mandel, M. Guedj. (2015). A Comparison of Six Methods for Missing Data Imputation. Journal of Biometrics and Biostatics. DOI: 10.472/2155-6180.1000224. Diakses dari https://www.hilarispublisher.com/open-access/a-comparison-of-six-methods-for-missing-data-imputation-2155-6180-1000224.pdf.

[2] Kleindessner, M., Awasthi, P., & Morgenstern, J. (2019). Fair k-Center Clustering for Data Summarization. International Conference on Machine Learning. Diakses dari https://www.semanticscholar.org/paper/Fair-k-Center-Clustering-for-Data-Summarization-Kleindessner-Awasthi/9c26bbf34bdab544a000038d628a8fb232d60cb6.

[3] Goutte, C., & Gaussier, E. (2005). A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. Advances in Information Retrieval, 345–359. Diakses dari doi:10.1007/978-3-540-31865-1_25.

[4] J. Brownlee. (2020). Machine Learing Mastery. Diakses dari https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/ .

[5] Dicoding Academy. Diakses dari https://www.dicoding.com/ .