

Laporan Final Project
Studi Kasus Promosi Karyawan Menggunakan MCDM Weighted Product



DOSEN PENGAMPU:

Ir. Nanang Yudi Setiawan, ST., M.Kom.

Sistem Pendukung Keputusan - C

DISUSUN OLEH:

Nandana Rifqi Irfansyah

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2023

Studi Kasus Promosi Karyawan Menggunakan MCDM Weighted Product

Multi Criteria Decision Making (CADM)

Multi Criteria Decision Making (MCDM) adalah suatu metode atau pendekatan yang digunakan untuk mengatasi keputusan kompleks yang melibatkan beberapa kriteria atau faktor. Dalam konteks ini, keputusan yang diambil tidak hanya dipengaruhi oleh satu kriteria, tetapi oleh sejumlah kriteria yang beragam. Menurut Mulliner, Malys, dan Maliene (2016), Multi Criteria Decision Making adalah seperangkat metode yang berhubungan dengan evaluasi serangkaian alternatif yang banyak, sering bertentangan, dan berbagai kriteria. Tujuan dari Multi Criteria Decision Making adalah untuk memberikan pilihan, peringkat, deskripsi, klasifikasi, pengelompokan, dan untuk mengurutkan alternatif dari yang paling disukai hingga opsi yang paling tidak disukai. Terdapat tiga tahap yang diikuti oleh semua metode Multi Criteria Decision Making yaitu :

1. Menentukan kriteria dan alternatif yang relevan
2. Melampirkan ukuran numerik untuk kepentingan relatif dari kriteria dan dampak terhadap alternatif dari kriteria tersebut
3. Memproses nilai numerik untuk menentukan peringkat dari masing-masing alternatif.

MCDM memiliki dua kategori yakni Multiple Objective Decision Making (MODM) dan Multiple Attribute Decision Making (MADM). Dalam dua kriteria tersebut, MCDM memiliki banyak metode yang telah dikembangkan hingga saat ini. Metode yang sering digunakan diantaranya adalah Analytic Hierarchy Process (AHP), Simple Additive Weighting Method (SAW), Weighted Product Model (WPM), Fuzzy Inference, dan masih banyak lagi.

Metode Weighted Product

Weighted Product merupakan salah satu pendekatan dari hasil evolusi konsep dalam Multi Criteria Decision Making (MCDM) yang termasuk ke dalam kategori Multiple Attribute Decision Making (MADM), dimana weighted product melibatkan pemilihan alternatif berdasarkan sekumpulan atribut atau kriteria pada keputusan. Metode Weighted Product (WP) menggunakan struktur matematis sehingga dapat mengevaluasi dengan pemahaman yang lebih mendalam terhadap setiap alternatif. Metode WP menggunakan perkalian dengan menghubungkan peringkat atribut (kriteria), dimana peringkat setiap atribut dikalikan dengan eksponen yang berasal dari bobot atribut yang sesuai. Terdapat langkah-langkah yang harus dilakukan dalam melaksanakan WP, yaitu :

1. Menentukan tingkat prioritas bobot untuk setiap atribut atau kriteria
2. Membuat tabel bobot kriteria yang akan dipilih

3. Menghitung Vektor S_i , dimana setiap data (X_{ij}) akan dikalikan tetapi sebelumnya dilakukan pemangkatan dengan bobot dari kriterianya. Dengan rumus :

$$S_i = \prod_{j=1}^n X_{ij}^{kW_j}$$

4. Menghitung vektor V_i dengan cara membagi hasil masing - masing vektor S_i

$$\text{dengan jumlah seluruh } S_i. \text{ Dengan rumus : } V_i = \frac{S_i}{\sum_{j=1}^m S_i}$$

Studi Kasus

Studi kasus berfokus pada industri teknologi, khususnya dalam manajemen karyawan. Seiring dengan pertumbuhan perusahaan di bidang teknologi, manajemen karyawan menjadi aspek krusial untuk menjaga keberlanjutan operasional dan inovasi. Dengan jumlah karyawan yang bervariasi, dari yang berpengalaman hingga yang baru bergabung, perusahaan perlu mengidentifikasi karyawan yang memiliki potensi untuk dipromosikan atau mendapatkan tindakan lebih lanjut. Tujuan utama dari studi kasus ini adalah memberikan landasan metodologis, dan Weighted Product dipilih sebagai metode evaluasi yang dapat menyaring karyawan berdasarkan kriteria-kriteria tertentu.

Dalam konteks ini, metode Weighted Product digunakan untuk menilai dan membandingkan karyawan-karyawan berdasarkan sejumlah atribut yang relevan dengan kriteria promosi. Penggunaan metode ini memungkinkan perusahaan untuk memberikan bobot yang sesuai dengan tingkat signifikansi masing-masing kriteria, sehingga pengambilan keputusan promosi dapat dilakukan dengan lebih objektif. Studi kasus ini mencerminkan tantangan umum dalam industri teknologi, di mana keberhasilan perusahaan tidak hanya ditentukan oleh teknologi yang digunakan, tetapi juga oleh keunggulan sumber daya manusia yang dimilikinya.

Kebutuhan Sistem dan Data

Kebutuhan Sistem:

1. Sistem Manajemen Karyawan: Dibutuhkan sistem manajemen karyawan yang dapat memfasilitasi pengelolaan data karyawan secara efisien. Sistem ini harus mencakup informasi seperti ID karyawan, riwayat pelatihan, usia, penilaian tahun sebelumnya, lama bekerja, pencapaian KPI, penghargaan yang pernah diterima, dan skor rata-rata pelatihan.
2. Metode Evaluasi Karyawan: Perlu adanya metode evaluasi karyawan untuk menentukan kelayakan promosi. Metode ini harus mencakup atribut-atribut yang relevan seperti jumlah pelatihan, usia, penilaian tahun sebelumnya, dan faktor-faktor kunci lainnya yang dianggap penting oleh perusahaan.

3. Weighted Product Model: Implementasi sistem Weighted Product Model yang dapat mengolah data karyawan berdasarkan bobot kriteria tertentu. Sistem ini harus dapat menghitung nilai produk, vektor SI, vektor Vi, serta memberikan peringkat karyawan berdasarkan hasil analisis Weighted Product.

Data yang Dibutuhkan:

1. Data Karyawan: Informasi lengkap tentang setiap karyawan termasuk employee ID, jumlah pelatihan, usia, penilaian tahun sebelumnya, lama bekerja, KPIs met lebih dari 80, penghargaan yang pernah diterima, dan skor rata-rata pelatihan.
2. Bobot Kriteria: Kriteria-kriteria yang diperlukan untuk mengevaluasi karyawan perlu memiliki bobot yang sesuai. Bobot ini menunjukkan tingkat kepentingan setiap kriteria dalam proses pengambilan keputusan.
3. Metode Penormalan Data: Diperlukan metode atau fungsi penormalan data untuk memastikan bahwa semua atribut atau kriteria memiliki skala yang seragam dan dapat data kinerja karyawan dapat diolah.
4. Hasil Evaluasi Karyawan: Setelah proses evaluasi menggunakan Weighted Product Model, data hasil evaluasi seperti vektor Vi dan peringkat karyawan perlu disimpan untuk memberikan pemahaman yang jelas tentang rekomendasi promosi.

Tujuan dan Manfaat

Tujuan :

1. Menghilangkan atau mengurangi tingkat subjektivitas dalam penilaian karyawan dengan menggunakan kriteria yang terukur dan terdefinisi dengan jelas.
2. Mempercepat dan meningkatkan efisiensi dalam proses penentuan promosi karyawan dengan otomatisasi menggunakan sistem pendukung keputusan.
3. Memastikan keputusan promosi karyawan sesuai dengan strategi dan tujuan jangka panjang organisasi.
4. Meningkatkan akurasi penilaian karyawan dengan menggunakan metode MCDM untuk memberikan bobot pada setiap kriteria, yang dapat mencerminkan pentingnya masing-masing faktor.
5. Mengurangi kemungkinan adanya bias atau pandangan subjektif dari para pengambil keputusan dengan menggunakan model matematis yang obyektif.

Manfaat :

1. Menghasilkan keputusan yang lebih terukur dan dapat diukur, karena setiap kriteria memiliki bobot yang jelas dan dapat dihitung.
2. Meningkatkan kepuasan karyawan karena keputusan promosi dapat dijelaskan dengan lebih transparan.
3. Memastikan bahwa karyawan yang dipromosikan memiliki keterampilan dan kompetensi yang sesuai dengan kebutuhan pekerjaan, sehingga dapat meningkatkan produktivitas di tingkat organisasi.

4. Meningkatkan efisiensi dalam manajemen sumber daya manusia dengan memanfaatkan teknologi dan model matematis.
5. Menyediakan pendekatan yang lebih adil dan setara dalam pengambilan keputusan promosi, mengurangi potensi ketidaksetaraan dan ketidakadilan.
6. Organisasi dapat membangun tim yang kompeten dan berkualitas tinggi, meningkatkan daya saing di pasar.
7. Menciptakan rekam jejak yang dapat diakses dan dilacak, membantu organisasi dalam melakukan audit dan evaluasi terhadap keputusan promosi yang diambil.

Sumber Data

Uncleaned_employees_final_dataset (1).csv (1.14 MB) 📄 🔄 >

Detail Compact Column 10 of 13 columns

employee_id	department	region	education	gender	recruitment_chan...	no_of_trainings	age	previous_year_rat...
3	Sales & Marketing	region_2	Bachelors	m	other	1	20	1
8724	Technology	region_26	Bachelors	m	sourcing	1	24	
74438	HR	region_4	Bachelors	f	other	1	31	3
72255	Sales & Marketing	region_13	Bachelors	m	other	1	31	1
38562	Procurement	region_2	Bachelors	f	other	3	31	2
64486	Finance	region_29	Bachelors	m	sourcing	1	30	4
46232	Procurement	region_7	Bachelors	m	sourcing	1	36	3
54542	Finance	region_2	Bachelors	m	other	1	33	5
67269	Analytics	region_22	Bachelors	m	sourcing	2	36	3
66174	Technology	region_7	Masters & above	m	other	1	51	4
76383	Technology	region_22	Bachelors	m	sourcing	1	29	5
68245	Sales & Marketing	region_16	Bachelors	m	sourcing	2	48	5
42639	Sales & Marketing	region_17	Masters & above	m	sourcing	1	48	3
38963	Sales & Marketing	region_4	Masters & above	f	other	1	34	3
54855	Analytics	region_24	Bachelors	m	other	1	37	3
42996	Operations	region_11	Bachelors	m	sourcing	1	38	5
12737	Sales & Marketing	region_7	Bachelors	m	sourcing	1	31	4
27561	Operations	region_27	Bachelors	f	sourcing	1	26	5
26622	Sales & Marketing	region_17	Bachelors	m	sourcing	1	40	5
31582	Procurement	region_7	Bachelors	f	other	1	49	3
29793	Procurement	region_27	Bachelors	m	other	1	27	2

Sumber data yang digunakan dalam proyek ini diambil dari Kaggle berikut : [Sumber Data Kaggle](#). Dari sumber data yang diambil, terdapat atribut-atribut pada kolom, yakni :

- **Employee ID**, Tanda identifikasi unik setiap karyawan
- **Departement**, menunjukkan Departemen Karyawan
- **Region**, Wilayah Geografis tempat karyawan bekerja
- **Education**, Tingkat Pendidikan karyawan
- **Gender**, Jenis Kelamin Karyawan
- **Recruitment Channel**, Jalur Karyawan direkrut di perusahaan
- **Number of Training**, Jumlah pelatihan yang diikuti karyawan
- **Age**, Usia karyawan
- **Previous Year Rating**, Penilaian Kinerja karyawan tahun sebelumnya

- **KPIs** met more than 80, Indikator pencapaian karyawan berhasil mencapai target kinerja yang telah ditetapkan atau tidak
- **Length of Service**, Jumlah tahun karyawan bekerja di perusahaan
- **Awards Won**, Jumlah penghargaan yang diterima karyawan
- **Average Training Score**, rata-rata nilai pelatihan yang diikuti karyawan

Proses Analisis

Proses analisis pertama yang dilakukan adalah melakukan seleksi atribut pada sumber data. Dimana tidak semua atribut yang terdapat pada sumber data tidak diperlukan dalam menentukan keputusan. Dari 12 atribut pada sumber data, diambil 7 diantaranya, yakni Previous Year Rating, Training Score, Length of Service, Awards, No of Training, Age, dan Key Performance Indikator (KPI) dengan alasan bahwa atribut-atribut tersebut terkait langsung dengan kinerja dan kontribusi positif dari karyawan terhadap perusahaan. Atribut-atribut yang dipakai merupakan kriteria final yang akan dipilih dalam pemeringkatan untuk promosi karyawan.

Setelah kriteria yang akan dipakai telah ditentukan, selanjutnya adalah dilakukan perangkingan serta pemberian bobot pada masing-masing kriteria tersebut. Dari analisis yang dilakukan, diperoleh perangkingan sebagai berikut :

1. Previous Year Rating dengan bobot 0.20
2. Training Score dengan bobot 0.18
3. Length of dengan bobot Service 0.18
4. Awards dengan bobot 0.14
5. No of Trainings dengan bobot 0.1
6. Age dengan bobot 0.1
7. KPI dengan bobot 0.1

Setelah masing-masing kriteria telah dirangking dan diberi bobot, selanjutnya adalah membuat **Tabel Bobot Kriteria ID Karyawan** yang akan dipilih untuk Promosi Jabatan dengan 7 Kriteria dan 10 Kandidat Karyawan. 10 Kandidat Karyawan ini didapatkan dari 10 Data Karyawan dengan semua kriteria memiliki nilai setidaknya > 0 (Clean Data) sehingga dapat dilakukan perhitungan dan Bersumber dari Dataset Kaggle tentang Employee Performance :

Alternatif No ID Karyawan	Alternatif	Kriteria						
		No_of_training	Age	Previous_year_rating	Length_of_service	KPIs_met_more_than_80	awards_won	avg_training_score
38562	A1	3	31	2	9	1	1	65

64486	A2	1	30	4	7	1	1	61
46232	A3	1	36	3	2	1	1	68
54542	A4	1	33	5	3	2	1	57
67269	A5	2	36	3	3	1	1	85
66174	A6	1	51	4	11	1	1	75
76303	A7	1	29	5	2	2	2	76
31162	A8	1	33	5	7	2	2	56
52945	A9	1	35	5	7	2	2	46
30703	A10	1	41	3	10	1	2	97

Perhitungan Manual Weighted Product

Untuk melakukan perbandingan Apakah Hasil Pemeringkatan yang dilakukan oleh Model yang dibangun pada Tools Google Collab menggunakan Bahasa Python itu valid dan benar, maka kelompok kami juga melakukan perhitungan Rumus Weighted Product secara manual.

1. Menghitung Vektor Si

Data dikalikan tetapi sebelumnya dilakukan pemangkatan dengan bobot dari masing-masing kriteria.

- $A1. (3^{0.1}) \times (31^{0.1}) \times (2^{0.2}) \times (9^{0.18}) \times (1^{0.1}) \times (1^{0.14}) \times (65^{0.18}) = 5.69039416.$
- $A2. (1^{0.1}) \times (30^{0.1}) \times (4^{0.2}) \times (7^{0.18}) \times (1^{0.1}) \times (1^{0.14}) \times (61^{0.18}) = 5.515706407$
- $A3. (1^{0.1}) \times (36^{0.1}) \times (3^{0.2}) \times (2^{0.18}) \times (1^{0.1}) \times (1^{0.14}) \times (68^{0.18}) = 4.316100062$
- $A4. (1^{0.1}) \times (33^{0.1}) \times (5^{0.2}) \times (3^{0.18}) \times (2^{0.1}) \times (1^{0.14}) \times (57^{0.18}) = 5.292826943$
- $A5. (2^{0.1}) \times (36^{0.1}) \times (3^{0.2}) \times (3^{0.18}) \times (1^{0.1}) \times (1^{0.14}) \times (85^{0.18}) = 5.180059319$
- $A6. (1^{0.1}) \times (51^{0.1}) \times (4^{0.2}) \times (11^{0.18}) \times (1^{0.1}) \times (1^{0.14}) \times (75^{0.18}) = 6.54833255$
- $A7. (1^{0.1}) * (29^{0.1}) * (5^{0.2}) * (2^{0.18}) * (2^{0.1}) * (2^{0.14}) * (76^{0.18}) = 5.636543272$
- $A8. (1^{0.1}) * (33^{0.1}) * (5^{0.2}) * (7^{0.18}) * (2^{0.1}) * (2^{0.14}) * (56^{0.18}) = 6.771487574$
- $A9. (1^{0.1}) * (35^{0.1}) * (5^{0.2}) * (7^{0.18}) * (2^{0.1}) * (2^{0.14}) * (46^{0.18}) = 6.574489879$
- $A10. (1^{0.1}) * (41^{0.1}) * (3^{0.2}) * (10^{0.18}) * (1^{0.1}) * (2^{0.14}) * (97^{0.18}) = 6.86223755$

2. Menghitung Vektor Vi

Menghitung vektor Vi dengan cara membagi hasil masing-masing vektor Si dengan jumlah seluruh Si .

- $A1 = 5.69039416 / 58.38817772 = 0.097458$
- $A2 = 5.515706407 / 58.38817772 = 0.094466$
- $A3 = 4.316100062 / 58.38817772 = 0.073921$
- $A4 = 5.292826943 / 58.38817772 = 0.090649$
- $A5 = 5.18005932 / 58.38817772 = 0.088718$
- $A6 = 6.548332552 / 58.38817772 = 0.112152$
- $A7 = 5.636543272 / 58.38817772 = 0.096536$
- $A8 = 6.771487574 / 58.38817772 = 0.115974$
- $A9 = 6.574489879 / 58.38817772 = 0.112600$
- $A10 = 6.862237554 / 58.38817772 = 0.117528$

Hasil : $A10 > A18 > A9 > A6 > A1 > A7 > A2 > A4 > A5 > A3$

Excel Version :

Dataset Performance Karyawan .XLSX

File Edit View Insert Format Data Tools Help

Menu

100%

123

Arial

10

B

I

A

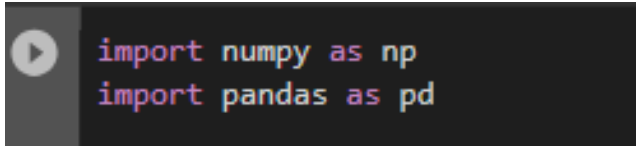
K30

	A	B	C	D	E	F	G	H	I	J
1	kode kandidat	employee_id	no_of_trainings	age	previous_year_r	length_of_servic	KPIs_met_more	awards_won	avg_training_score	
2		bobot	0.1	0.1	0.2	0.18	0.1	0.14	0.18	
3	A1	38562	3	31	2	9	1	1	65	
4	A2	64486	1	30	4	7	1	1	61	
5	A3	46232	1	36	3	2	1	1	68	
6	A4	54542	1	33	5	3	2	1	57	
7	A5	67269	2	36	3	3	1	1	85	
8	A6	66174	1	51	4	11	1	1	75	
9	A7	76303	1	29	5	2	2	2	76	
10	A8	31162	1	33	5	7	2	2	56	
11	A9	52945	1	35	5	7	2	2	46	
12	A10	30703	1	41	3	10	1	2	97	
13										
14		Weighted Product Excel Version								
15	Kandidat	Vektor Si	Vektor Vi		Pemeringkatan	Kandidat	Skor WP			
16	A1	5.690394159	0.09745798518		1	A10	0.117528			
17	A2	5.515706407	0.09446615089		2	A8	0.115974			
18	A3	4.316100062	0.07392078722		3	A9	0.1126			
19	A4	5.292826943	0.09064894897		4	A6	0.112152			
20	A5	5.180059319	0.08871760554		5	A1	0.097458			
21	A6	6.548332552	0.1121516856		6	A7	0.09653			
22	A7	5.636543272	0.09653569425		7	A2	0.0944661			
23	A8	6.771487574	0.115973607		8	A4	0.0906489			
24	A9	6.574489879	0.1125996758		9	A5	0.088718			
25	A10	6.862237554	0.1175278596		10	A3	0.073921			
26	Total Vektor Si	58.38817772			Hasil : A10 > A18 > A9 > A6 > A1 > A7 > A2 > A4 > A5 > A3					

Tools Pengembangan Menggunakan Python

Screenshot Code in Google Colab :

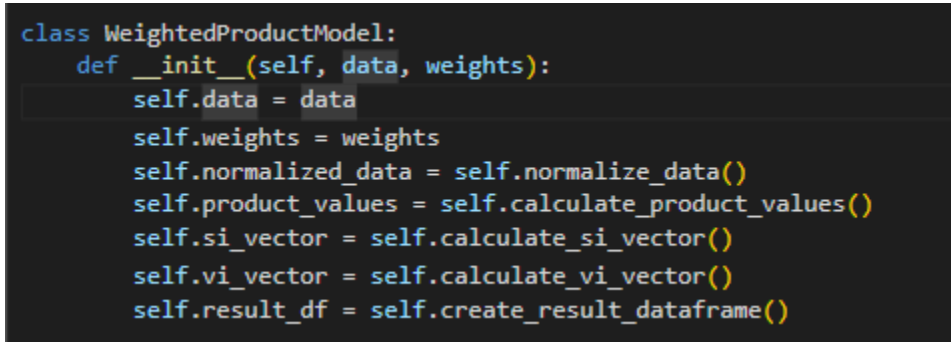
1. Import Library dan Class WeightedProduct



```
import numpy as np
import pandas as pd
```

Pada sel ini, kami mengimpor dua pustaka yang diperlukan untuk analisis data, yaitu NumPy untuk operasi numerik dan Pandas untuk manipulasi data dalam bentuk DataFrame.

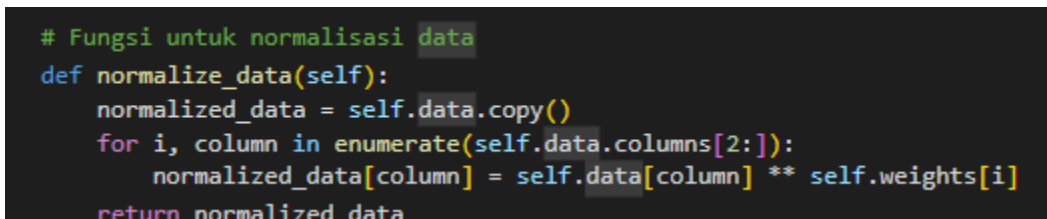
2. Class WeightedProductModel



```
class WeightedProductModel:
    def __init__(self, data, weights):
        self.data = data
        self.weights = weights
        self.normalized_data = self.normalize_data()
        self.product_values = self.calculate_product_values()
        self.si_vector = self.calculate_si_vector()
        self.vi_vector = self.calculate_vi_vector()
        self.result_df = self.create_result_dataframe()
```

Selanjutnya, kami mendefinisikan kelas `WeightedProductModel` yang akan digunakan untuk menerapkan metode Weighted Product Model pada data yang diberikan. Inisialisasi kelas ini dilakukan dengan mengambil data dan bobot sebagai input, dan menghitung beberapa nilai yang diperlukan, termasuk vektor VI dan hasil DataFrame.

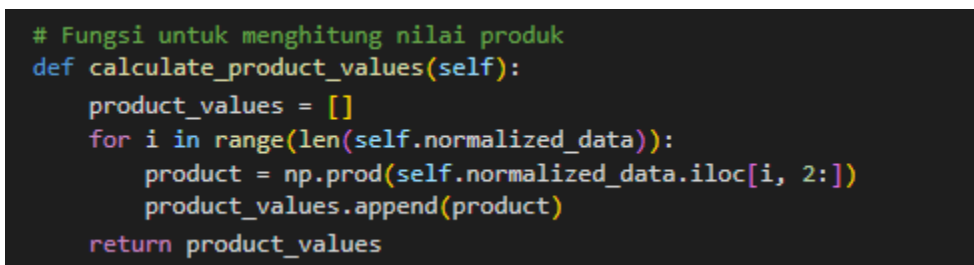
3. Fungsi `normalize_data`



```
# Fungsi untuk normalisasi data
def normalize_data(self):
    normalized_data = self.data.copy()
    for i, column in enumerate(self.data.columns[2:]):
        normalized_data[column] = self.data[column] ** self.weights[i]
    return normalized_data
```

Selanjutnya kami membuat Fungsi `normalize_data` yang digunakan untuk normalisasi data. Setiap kolom di luar dua kolom pertama diangkat ke bobot yang sesuai.

4. Fungsi `calculate_product_values`



```
# Fungsi untuk menghitung nilai produk
def calculate_product_values(self):
    product_values = []
    for i in range(len(self.normalized_data)):
        product = np.prod(self.normalized_data.iloc[i, 2:])
        product_values.append(product)
    return product_values
```

Kemudian kami juga membuat Fungsi `calculate_product_values` yang digunakan untuk menghitung nilai produk dari setiap baris dalam data yang telah dinormalisasi.

5. Fungsi calculate_si_vector, calculate_vi_vector & Fungsi Dataframe Hasil

```
# Fungsi untuk menghitung vektor SI
def calculate_si_vector(self):
    total_product = np.sum(self.product_values)
    si_vector = [value / total_product for value in self.product_values]
    return si_vector

# Fungsi untuk menghitung vektor Vi
def calculate_vi_vector(self):
    vi_vector = [value / np.sum(self.si_vector) for value in self.si_vector]
    return vi_vector

# Fungsi untuk membuat DataFrame hasil
def create_result_dataframe(self):
    result_df = pd.DataFrame({
        'kode_kandidat': self.data['kode_kandidat'],
        'vector_vi': self.vi_vector,
        'ranking': np.arange(1, len(self.vi_vector) + 1)
    })
    return result_df
```

Berikutnya kami membuat Fungsi calculate_si_vector untuk menghitung vektor SI dengan membagi setiap nilai produk dengan total produk dari semua alternatif. Kemudian Fungsi calculate_vi_vector yang digunakan untuk menghitung vektor VI dengan membagi setiap nilai SI dengan total SI. Serta Fungsi create_result_dataframe membuat DataFrame hasil yang berisi kolom-kolom 'kode_kandidat', 'vector_vi', dan 'ranking'.

6. Fungsi rank_alternatives

```
def compare_alternatives(self):
    comparisons = []
    for i in range(len(self.vi_vector)):
        for j in range(i + 1, len(self.vi_vector)):
            ratio = self.vi_vector[i] / self.vi_vector[j]
            comparisons.append((self.data.iloc[i, 0], self.data.iloc[j, 0], ratio))
    return comparisons

# Fungsi untuk memberikan peringkat pada alternatif
def rank_alternatives(self):
    sorted_indices = np.argsort(self.vi_vector)[::-1]
    ranked_alternatives = [self.data.iloc[i, 0] for i in sorted_indices]
    return ranked_alternatives
```

Kemudian terdapat Fungsi rank_alternatives yang berfungsi untuk memberikan peringkat pada setiap alternatif berdasarkan nilai vektor Vi.

7. Data

```
# Data
data = {
    'kode_kandidat': ['A1', 'A2', 'A3', 'A4', 'A5', 'A6', 'A7', 'A8', 'A9', 'A10'],
    'employee_id': [38562, 64486, 46232, 54542, 67269, 66174, 76303, 31162, 52945, 30703],
    'no_of_trainings': [3, 1, 1, 1, 2, 1, 1, 1, 1, 1],
    'age': [31, 30, 36, 33, 36, 51, 29, 33, 35, 41],
    'previous_year_rating': [2, 4, 3, 5, 3, 4, 5, 5, 5, 3],
    'length_of_service': [9, 7, 2, 3, 3, 11, 2, 7, 7, 10],
    'KPIs_met_more_than_80': [1, 1, 1, 2, 1, 1, 2, 2, 2, 1],
    'awards_won': [1, 1, 1, 1, 1, 1, 2, 2, 2, 2],
    'avg_training_score': [65, 61, 68, 57, 85, 75, 76, 56, 46, 97]
}

# Bobot
weights = [0.1, 0.1, 0.2, 0.18, 0.1, 0.14, 0.18]

# DataFrame
df = pd.DataFrame(data)

# Instansiasi model
wpm = WeightedProductModel(df, weights)
```

Pada sel ini, kami menentukan data yang akan digunakan untuk analisis dengan Weighted Product Model. Berikutnya, kami mendefinisikan bobot untuk masing-masing fitur dalam analisis. Selanjutnya, kami membuat DataFrame dari data yang telah ditentukan sebelumnya. Instansiasi Model dan Menampilkan Hasil

8. Instansiasi Model dan Hasil

Pada langkah ini, kami membuat instance dari kelas Weighted Product Model dengan menggunakan Data Frame dan bobot yang telah disiapkan sebelumnya. Dan dilanjutkan menampilkan hasil dari analisis Weighted Product Model, yang meliputi DataFrame hasil peringkat alternatif.

```
# Menampilkan hasil
print(wpm.result_df)

# Rank alternatives
ranked_alternatives = wpm.rank_alternatives()
print("\nRanking of Alternatives:")
print(" > ".join(ranked_alternatives))
```

	kode_kandidat	vector_vi	ranking
0	A1	0.097458	1
1	A2	0.094466	2
2	A3	0.073921	3
3	A4	0.090649	4
4	A5	0.088718	5
5	A6	0.112152	6
6	A7	0.096536	7
7	A8	0.115974	8
8	A9	0.112600	9
9	A10	0.117528	10

Ranking of Alternatives:
A10 > A8 > A9 > A6 > A1 > A7 > A2 > A4 > A5 > A3

Dan hasil yang didapatkan bahwa Kode Kandidat A10 adalah Kandidat dengan Rangkaing No 1 (Teringgi) dari Seluruh Kandidat yang ada untuk Promosi Karyawan

Code Python - Weighted Product Model

```
import numpy as np
import pandas as pd

class WeightedProductModel:
    def __init__(self, data, weights):
        self.data = data
        self.weights = weights
        self.normalized_data = self.normalize_data()
        self.product_values = self.calculate_product_values()
        self.si_vector = self.calculate_si_vector()
        self.vi_vector = self.calculate_vi_vector()
        self.result_df = self.create_result_dataframe()

    # Fungsi untuk normalisasi data
    def normalize_data(self):
        normalized_data = self.data.copy()
        for i, column in enumerate(self.data.columns[2:]):
            normalized_data[column] = self.data[column] ** self.weights[i]
        return normalized_data
```

```

# Fungsi untuk menghitung nilai produk
def calculate_product_values(self):
    product_values = []
    for i in range(len(self.normalized_data)):
        product = np.prod(self.normalized_data.iloc[i, 2:])
        product_values.append(product)
    return product_values

# Fungsi untuk menghitung vektor SI
def calculate_si_vector(self):
    total_product = np.sum(self.product_values)
    si_vector = [value / total_product for value in self.product_values]
    return si_vector

# Fungsi untuk menghitung vektor Vi
def calculate_vi_vector(self):
    vi_vector = [value / np.sum(self.si_vector) for value in self.si_vector]
    return vi_vector

# Fungsi untuk membuat DataFrame hasil
def create_result_dataframe(self):
    result_df = pd.DataFrame({
        'kode_kandidat': self.data['kode_kandidat'],
        'vector_vi': self.vi_vector,
        'ranking': np.arange(1, len(self.vi_vector) + 1)
    })
    return result_df

def compare_alternatives(self):
    comparisons = []
    for i in range(len(self.vi_vector)):
        for j in range(i + 1, len(self.vi_vector)):
            ratio = self.vi_vector[i] / self.vi_vector[j]
            comparisons.append((self.data.iloc[i, 0], self.data.iloc[j, 0], ratio))
    return comparisons

# Fungsi untuk memberikan peringkat pada alternatif
def rank_alternatives(self):
    sorted_indices = np.argsort(self.vi_vector)[::-1]
    ranked_alternatives = [self.data.iloc[i, 0] for i in sorted_indices]
    return ranked_alternatives

# Data
data = {
    'kode_kandidat': ['A1', 'A2', 'A3', 'A4', 'A5', 'A6', 'A7', 'A8', 'A9', 'A10'],
    'employee_id': [38562, 64486, 46232, 54542, 67269, 66174, 76303, 31162, 52945, 30703],

```

```
'no_of_trainings': [3, 1, 1, 1, 2, 1, 1, 1, 1, 1],  
'age': [31, 30, 36, 33, 36, 51, 29, 33, 35, 41],  
'previous_year_rating': [2, 4, 3, 5, 3, 4, 5, 5, 5, 3],  
'length_of_service': [9, 7, 2, 3, 3, 11, 2, 7, 7, 10],  
'KPIs_met_more_than_80': [1, 1, 1, 2, 1, 1, 2, 2, 2, 1],  
'awards_won': [1, 1, 1, 1, 1, 1, 2, 2, 2, 2],  
'avg_training_score': [65, 61, 68, 57, 85, 75, 76, 56, 46, 97]  
}
```

```
# Bobot
```

```
weights = [0.1, 0.1, 0.2, 0.18, 0.1, 0.14, 0.18]
```

```
# DataFrame
```

```
df = pd.DataFrame(data)
```

```
# Instansiasi model
```

```
wpm = WeightedProductModel(df, weights)
```

```
# Menampilkan hasil
```

```
print(wpm.result_df)
```

```
# Rank alternatives
```

```
ranked_alternatives = wpm.rank_alternatives()
```

```
print("\nRanking of Alternatives:")
```

```
print("> ".join(ranked_alternatives))
```

[Tautan Google Colab_Weighted Product_Employee Promotion](#)

Hasil Analisis Weighted Product



Hasil analisis Weighted Product pada dataset karyawan menunjukkan bahwa Kandidat A10 dengan Employee_ID 30703 mendominasi peringkat tertinggi, memposisikan dirinya sebagai kandidat utama untuk mendapatkan promosi di perusahaan. Employee_ID 30703 memiliki nilai Vector VI tertinggi, yaitu sebesar 0.117528, menunjukkan kontribusi yang signifikan terhadap seluruh kriteria yang dievaluasi. Berlanjut ke peringkat kedua, Kandidat A8 dengan Employee_ID 31162 meraih posisi tersebut dengan nilai Vector VI sebesar 0.115974. Karyawan ini menunjukkan kinerja yang baik dan layak untuk dipertimbangkan dalam keputusan promosi. Selanjutnya, Kandidat A9 dengan Employee_ID 52945 berada di peringkat ketiga dengan nilai Vector VI 0.112600, menunjukkan bahwa karyawan ini memiliki potensi yang signifikan untuk mendapatkan promosi.

Kandidat A6 (Employee_ID 66174) berada di peringkat keempat dengan Vector VI 0.112152, menunjukkan kinerja yang solid, diikuti oleh Kandidat A1 (Employee_ID 38562) di peringkat kelima dengan Vector VI 0.097458. Kandidat A7 (Employee_ID 76303) dan A2 (Employee_ID 64486) menduduki peringkat keenam dan ketujuh masing-masing, sementara Kandidat A4 (Employee_ID 54542) dan A5 (Employee_ID 67269) berada di peringkat kedelapan dan kesembilan. Peringkat terendah ditempati oleh Kandidat A3 (Employee_ID 46232) dengan Vector VI 0.073921. Meskipun mendapatkan peringkat terendah, namun hasil analisis ini memberikan informasi berharga bagi perusahaan untuk melakukan pembenahan atau pengembangan karyawan sesuai dengan kriteria yang ditetapkan.

Dengan demikian, hasil pemeringkatan ini memberikan pandangan hierarki tentang karyawan yang layak dipertimbangkan untuk mendapatkan promosi. Implementasi Weighted

Product secara efektif memberikan pemahaman mendalam tentang kontribusi masing-masing karyawan terhadap tujuan promosi, memungkinkan perusahaan mengambil keputusan untuk memberikan Promosi Karyawan seperti Kenaikan Jabatan atau Kenaikan Gaji terhadap kandidat Karyawan yang tepat.

Kesimpulan

Berdasarkan analisis menggunakan metode Weighted Product pada studi kasus promosi karyawan, dapat disimpulkan bahwa metode ini memberikan ranking prioritas terhadap kandidat-kandidat yang paling sesuai dengan kriteria yang telah ditetapkan. Kelebihan dari metode Weighted Product terletak pada kemampuannya memberikan bobot yang berbeda pada setiap atribut, sehingga dapat disesuaikan dengan kepentingan relatif dari masing-masing kriteria. Namun, perlu diingat bahwa hasil analisis sangat tergantung pada pengaturan bobot yang dipilih dan metode ini bisa sensitif terhadap skala data. Oleh karena itu, pemilihan bobot dan interpretasi hasil perlu dilakukan dengan hati-hati untuk memastikan keakuratan dan keandalan rekomendasi promosi karyawan.

Referensi

Decision support system for new employee recruitment using weighted product method,
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7892459>

Simulation of the Weighted Product Method in the Tri Pramana-Based Formative-Summative Evaluation Application, <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9990611>

Implementation of Weighted Product Method as Fuzzy Multi-Criteria Decision Analysis (FMADM) in Vendor Selection,
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10010305>

Dataset Employee's Performance for HR Analytics from Kaggle

<https://www.kaggle.com/datasets/sanjanchaudhari/employees-performance-for-hr-analytics>