



USER INTERVIEW BINA PERTIWI



AGENDA

Introduction

Latihan 1

Latihan 2

Latihan 3

Latihan 4

Latihan 5



INTRODUCTION

**JOSUA PANE –
DATA SCIENTIST ASSOCIATE**

LATIHAN 1

Anda memiliki dataset mengenai penjualan tiket di sebuah bioskop sepanjang tahun. Dataset ini memiliki beberapa kolom berikut:

- tanggal_film (Tanggal film ditayangkan)
- judul_film (Nama film)
- durasi_film (Durasi film dalam menit)
- kapasitas_auditorium (Kapasitas tempat duduk auditorium)
- tiket_terjual (Jumlah tiket film yang terjual)
- harga_tiket (Harga tiket)

Namun, dataset ini memiliki beberapa masalah: beberapa nilai hilang atau tidak sesuai, serta adanya duplikasi. Anda harus membersihkan dataset ini agar dapat digunakan untuk analisis lebih lanjut.

SOLUSI LATIHAN 1

```
import pandas as pd
import numpy as np

df = pd.read_csv('data1.csv')

#handling nilai yang hilang / dan tanggal yang tidak sesuai
num_cols = ['durasi_film', 'kapasitas_auditorium',
            'tiket_terjual', 'harga_tiket']

for col in num_cols:
    df[col] = df[col].fillna(df[col].median())

df['judul_film'] = df['judul_film'].fillna('Unknown')

df['tanggal_film'] = pd.to_datetime(df['tanggal_film'])

#drop duplicate value
df = df.drop_duplicates()

df.to_csv('data1_new.csv', index=False)
```

LATIHAN 2

Anda diberi dataset yang mencakup informasi seputar perjalanan pelanggan selama satu bulan, seperti:

- tanggal_waktu (Tanggal dan waktu mulai perjalanan)
- jarak (Jarak perjalanan dalam kilometer)
- durasi (Durasi perjalanan dalam menit)
- harga (Harga perjalanan)
- driver_rating (Rating yang diberikan oleh pelanggan kepada pengemudi)
- customer_rating (Rating yang diberikan oleh pengemudi kepada pelanggan)

Anda diminta untuk membuat visualisasi yang efektif dari data tersebut untuk mengeksplorasi pola dan hubungan antara fitur-fiturnya.

SOLUSI LATIHAN 2

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

# asumsi data sudah di-import, cleaning, dan sudah di lakukan EDA
# mulai tahap membuat visualisasi
# buat heatmap correlation

plt.figure(figsize=(12, 10))

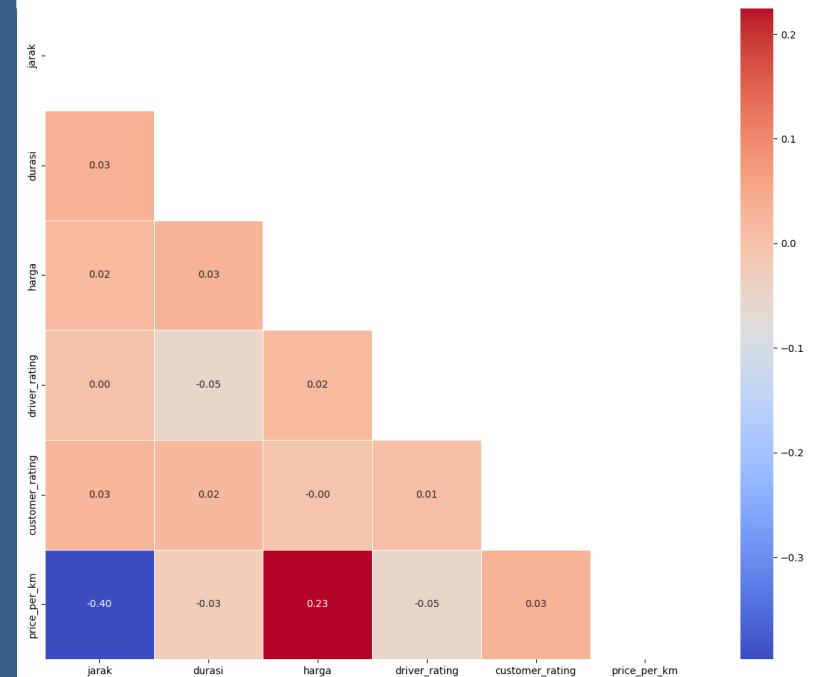
numerical_cols = ['jarak', 'durasi', 'harga', 'driver_rating',
                  'customer_rating', 'price_per_km']

correlation_matrix = df[numerical_cols].corr()

mask = np.triu(np.ones_like(correlation_matrix, dtype=bool))

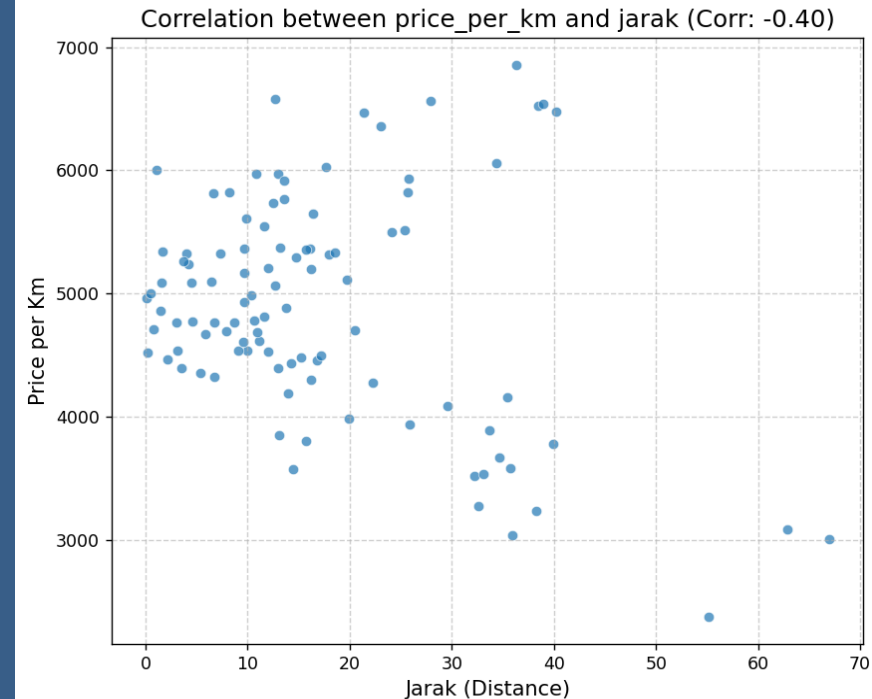
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
            linewidths=0.5)

plt.title('Correlation Matrix', fontsize=16)
```



SOLUSI LATIHAN 2

```
# contoh visualisasi scatter plot
plt.subplot(1, 2, 1)
sns.scatterplot(x='jarak', y='harga_per_km',
data=data, alpha=0.7)
plt.title('Correlation harga_per_km dan jarak')
plt.xlabel('Jarak', fontsize=12)
plt.ylabel('Harga per km', fontsize=12)
plt.grid(True, alpha=0.6)
```



LATIHAN 3

Anda bekerja di divisi Human Resources sebuah perusahaan dan diberi dataset yang mencakup informasi tentang karyawan, termasuk:

- umur (Umur karyawan)
- jenis_kelamin (Jenis kelamin karyawan)
- pendidikan (Tingkat pendidikan karyawan)
- lama_bekerja (Lama bekerja di perusahaan dalam tahun)
- gaji (Gaji karyawan)

Tugas Anda adalah menggunakan statistik deskriptif dan inferensial untuk menganalisis dataset ini dan memahami pola dan hubungan antara fitur-fiturnya.

SOLUSI LATIHAN 3

```
#asumsi data, library yang dibutuhkan sudah di-import, cleaning,  
dan sudah di eksplorasi
```

```
# Statistik Deskriptif
```

```
print("\nStatistik Deskriptif Umum:")  
print(df.describe(include='all'))
```

```
# Statistik Deskriptif untuk Setiap Fitur
```

```
print("\nStatistik Umur:")  
print(df['umur'].describe())
```

```
print("\nStatistik Gaji:")  
print(df['gaji'].describe())
```

```
print("\nDistribusi Jenis Kelamin:")  
print(df['jenis_kelamin'].value_counts())
```

```
print("\nDistribusi Pendidikan:")  
print(df['pendidikan'].value_counts())
```

SOLUSI LATIHAN 3

```
# Korelasi antar variabel numerik
```

```
print("Korelasi antar fitur numerik:")  
print(df[['umur', 'lama_bekerja', 'gaji']].corr())  
  
sns.heatmap(df[['umur', 'lama_bekerja', 'gaji']].corr())  
plt.title('Korelasi antara Umur, Lama Bekerja, dan Gaji')  
plt.show()
```

```
# contoh statistic inferensial  
# rata-rata gaji berdasarkan pendidikan
```

```
pendidikan_gaji = df.groupby('pendidikan')['gaji'].mean()  
print("\nRata-rata Gaji berdasarkan Pendidikan:")  
print(pendidikan_gaji)
```

```
sns.boxplot(x='pendidikan', y='gaji', data='data3')  
plt.title('Gaji berdasarkan Pendidikan')  
plt.show()
```

LATIHAN 4

Anda diberi dataset mengenai pelanggan sebuah perusahaan kartu kredit. Dataset ini mencakup informasi seperti:

- pelanggan_id (ID unik untuk setiap pelanggan)
- usia (Usia pelanggan)
- jenis_kelamin (Jenis kelamin pelanggan)
- pendapatan (Pendapatan tahunan pelanggan)
- jml_kartu_kredit (Jumlah kartu kredit yang dimiliki pelanggan)
- pengeluaran_bulanan (Pengeluaran bulanan rata-rata pelanggan)

Anda diminta untuk membuat model machine learning yang akan memprediksi apakah seorang pelanggan akan tertarik pada penawaran kartu kredit baru. Anda harus menyiapkan lembar kerja yang meliputi proses pengembangan model, termasuk pemilihan fitur, pelatihan, validasi, dan evaluasi.

SOLUSI LATIHAN 4

```
#asumsi data, library yang dibutuhkan sudah di-import, cleaning,  
dan sudah di eksplorasi, dan sudah didapatkan data calon  
pelanggan kartu kreditt potensial
```

```
# pemilihan fitur
```

```
# set dulu variabel target y
```

```
df['tertarik'] = (  
    (df['pendapatan'] > 80000000) &  
    (df['jml_kartu_kredit'] < 3) &  
    (df['pengeluaran_bulanan'] > 5000000)
```

```
).astype(int)
```

```
features = ['usia', 'pendapatan', 'jml_kartu_kredit',  
            'pengeluaran_bulanan']
```

```
X = df[features]
```

```
y = df['tertarik']
```

```
scaler = StandardScaler()
```

```
X_scaled = scaler.fit_transform(X)
```

SOLUSI LATIHAN 4

```
# data split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y,
test_size=0.2, stratify=y, random_state=42)

# data train
model = RandomForestClassifier(random_state=42, n_estimators=100)
model.fit(X_train, y_train)

#validasi dan evaluasi
y_pred = model.predict(X_test)
y_proba = model.predict_proba(X_test)[:1]

print("Evaluasi", classification_report(y_test, y_pred))
print("Confusion Matrix", confusion_matrix(y_test, y_pred))
```

LATIHAN 5

Kami telah menyediakan contoh data produk di Sephora. Dari data ini, kami ingin tahu wawasan apa yang bisa Anda dapatkan.

Beberapa bagian dari wawasan harus terkait dengan model pembelajaran mesin dan visualisasi data.

Data: <https://binapertiwi.link/filesdatascientisttest>

SOLUSI LATIHAN 5

1. Tahapan Awal Data Examination

Jumlah fitur data 'train' = 16

Data columns (total 17 columns):				
#	Column	Non-Null Count		Dtype
0	id	8000	non-null	int64
1	brand	8000	non-null	object
2	category	8000	non-null	object
3	name	8000	non-null	object
4	size	8000	non-null	object
5	rating	8000	non-null	float64
6	number_of_reviews	8000	non-null	int64
7	love	8000	non-null	int64
8	price	8000	non-null	float64
9	value_price	8000	non-null	float64
10	URL	8000	non-null	object
11	MarketingFlags	8000	non-null	bool
12	options	8000	non-null	object
13	details	8000	non-null	object
14	how_to_use	8000	non-null	object
15	ingredients	8000	non-null	object
16	exclusive	8000	non-null	int64

Jumlah fitur data 'test' = 20

Data columns (total 21 columns):				
#	Column	Non-Null Count		Dtype
0	id	1164	non-null	int64
1	brand	1164	non-null	object
2	category	1164	non-null	object
3	name	1164	non-null	object
4	size	1164	non-null	object
5	rating	1164	non-null	float64
6	number_of_reviews	1164	non-null	int64
7	love	1164	non-null	int64
8	price	1164	non-null	float64
9	value_price	1164	non-null	float64
10	URL	1164	non-null	object
11	MarketingFlags	1164	non-null	bool
12	MarketingFlags_content	1164	non-null	object
13	options	1164	non-null	object
14	details	1164	non-null	object
15	how_to_use	1164	non-null	object
16	ingredients	1164	non-null	object
17	online_only	1164	non-null	int64
18	exclusive	1164	non-null	int64
19	limited_edition	1164	non-null	int64
20	limited_time_offer	1164	non-null	int64

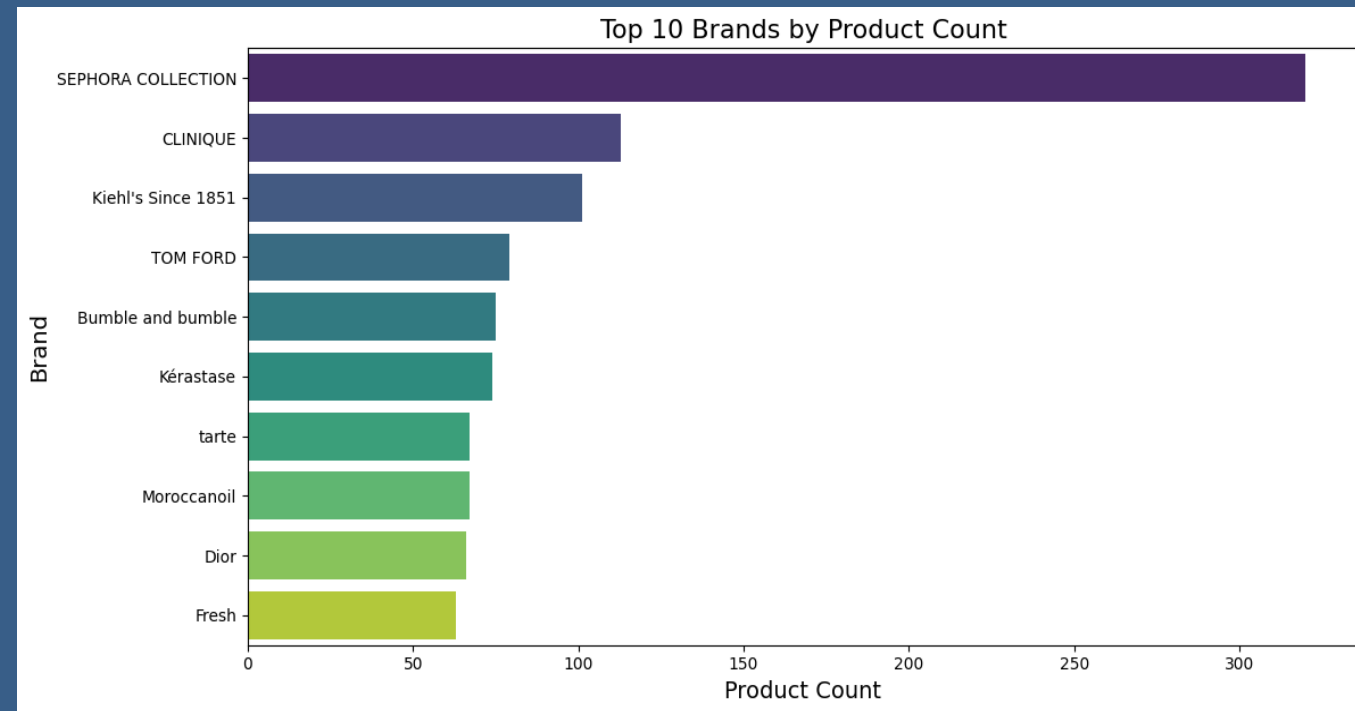
SOLUSI LATIHAN 5

Visualisasi Data

1. Show Top 10 Brands by Product Count with BarPlot

```
# Visualisasi 1

if "brand" not in df.columns:
    raise ValueError("Kolom 'Brand' tidak ditemukan dalam dataset.")
brand_counts = df['brand'].value_counts()
top_10_brands = brand_counts.head(10)
print("Top 10 Brands by Product Count:\n",
      top_10_brands)
plt.figure(figsize=(12, 6))
sns.barplot(x=top_10_brands.values,
            y=top_10_brands.index, palette="viridis")
plt.xlabel("Product Count", fontsize=14)
plt.ylabel("Brand", fontsize=14)
plt.title("Top 10 Brands by Product Count",
          fontsize=16)
plt.tight_layout()
plt.show()
```



Brand milik Sephora sendiri, yaitu "SEPHORA COLLECTION" mendominasi. Disusul dengan produk dari brand "CLINIQUE" dan "Kiehl's Since 1851"

SOLUSI LATIHAN 5

Visualisasi Data

2. Price vs. Rating Correlation (Scatter Plot)

```
# Visualisasi 2
if "rating" in df.columns:
    df["rating"] =
pd.to_numeric(df["rating"], errors='coerce')
if "review_count" in df.columns:
    df["review_count"] =
pd.to_numeric(df["review_count"],
errors='coerce')
else:
    df["review_count"] = 1
df.dropna(subset=["price", "rating"],
inplace=True)
plt.figure(figsize=(12, 8))
size_factor = 200
sizes = (df["review_count"] /
df["review_count"].max()) * size_factor + 10
plt.scatter(df["price"], df["rating"],
s=sizes, alpha=0.6, color="skyblue",
edgecolor="grey", label="Produk")
plt.xlabel("Price ($)", fontsize=14)
plt.ylabel("Rating", fontsize=14)
plt.title("Price vs. Rating Correlation",
fontsize=16)
plt.legend()
plt.grid(True, linestyle='--', alpha=0.5)
plt.tight_layout()
plt.show()
```



Dari graf di atas dapat dilihat bahwa tidak ada korelasi linear antara 'Harga' dan 'Rating'. Contoh: Ada produk dengan harga tinggi (misalnya HERMÈS perfumes dengan harga \$190) yang mendapat rating tinggi (sekitar 4.5-5), distribusi titik pada graf menunjukkan bahwa harga bukanlah faktor penentu utama dalam menentukan rating. Produk dengan rentang harga mulai dari yang rendah hingga menengah juga mendapatkan rating tinggi. Hal ini menunjukkan bahwa customer menilai produk berdasarkan kualitas atau persepsi merek, bukan semata-mata harga.

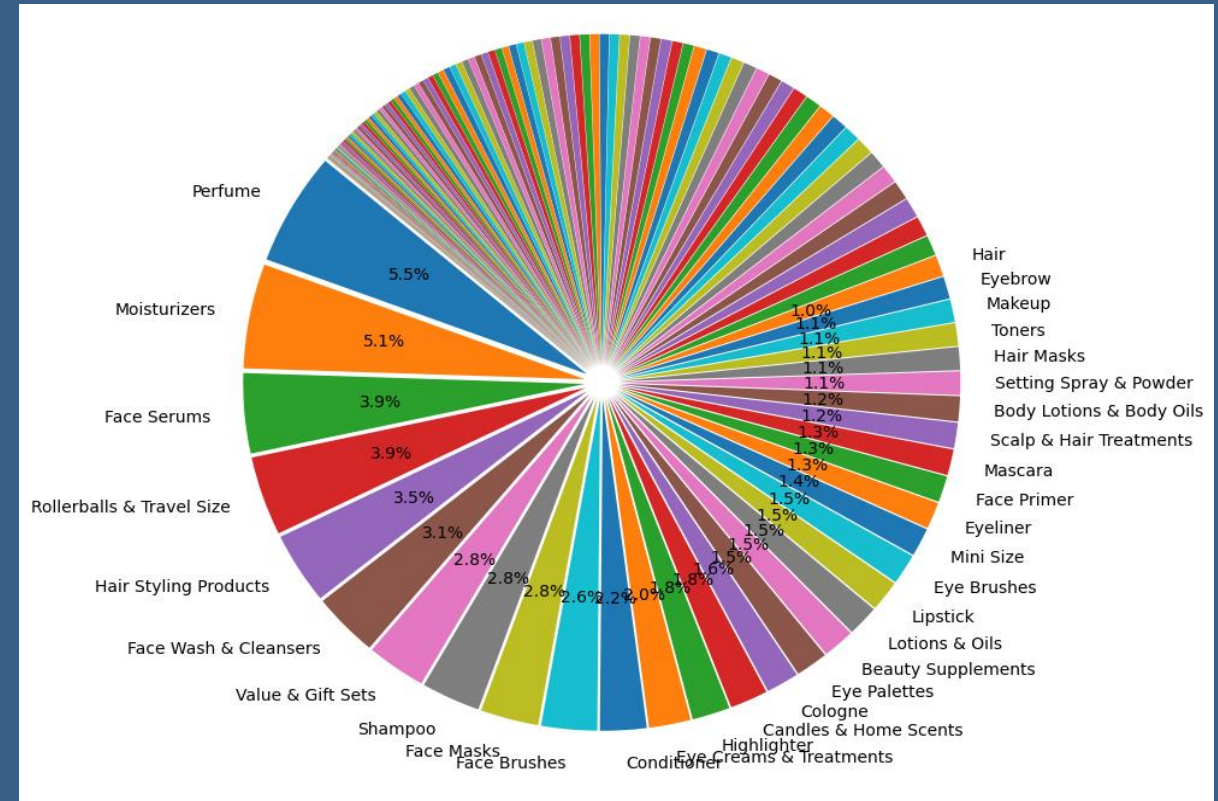
SOLUSI LATIHAN 5

Visualisasi Data

3. Price vs. Rating Correlation (Scatter Plot)

```
# Visualisasi 3

if "category" not in df.columns:
    raise ValueError("Kolom 'category' tidak ditemukan dalam dataset.")
category_counts = df['category'].value_counts()
total = category_counts.sum()
new_labels = [cat if (count / total * 100) >= 1 else ''
               for cat, count in zip(category_counts.index, category_counts)]
plt.figure(figsize=(10, 10))
plt.pie(
    category_counts,
    labels=new_labels,
    autopct=custom_autopct,
    explode=[0.05]*len(category_counts)
)
plt.title("Category Distribution",
          fontsize=16)
plt.tight_layout()
plt.show()
```



Dari graf diatas dapat dilihat bahwa produk dengan kategori terbanyak yaitu "Perfume" dengan 5.5% dari total produk, dan diikuti oleh "Moisturizers" dengan 5.1%. Insight dari pie chart ini juga memungkinkan tim marketing atau board direction melihat dengan jelas kategori mana yang layak mendapatkan perhatian lebih besar dalam hal ini : produk wewangian (Parfum, Cologne) dan perawatan kulit (misalnya, Sabun Cuci Muka, Pelembab).

SOLUSI LATIHAN 5

Implementasi Machine Learning 1 (Recommendation System)

1. Sistem Rekomendasi dengan Content-based Filtering

Sistem akan menyarankan produk yang menyerupai inputan user berdasarkan fiturnya, seperti kategori, bahan(ingredient), dan merek.

```
required_cols = ['name', 'category', 'ingredients', 'brand']
for col in required_cols:
    if col not in df.columns:
        raise ValueError(f"Kolom '{col}' tidak ditemukan dalam dataset.")

for col in required_cols:
    df[col] = df[col].fillna('')

def create_soup(x):
    return f"{x['category']} {x['ingredients']} {x['brand']}"

df['soup'] = df.apply(create_soup, axis=1)
tfidf = TfidfVectorizer(stop_words='english')
tfidf_matrix = tfidf.fit_transform(df['soup'])
cosine_sim = cosine_similarity(tfidf_matrix, tfidf_matrix)
indices = pd.Series(df.index, index=df['name']).drop_duplicates()

def get_recommendations(name, cosine_sim=cosine_sim, top_n=10):
    if name not in indices:
        raise ValueError(f"Produk '{name}' tidak ditemukan dalam dataset.")
    idx = indices[name]
    sim_scores = list(enumerate(cosine_sim[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:top_n+1]
    product_indices = [i[0] for i in sim_scores]
    return df[['name', 'category', 'ingredients', 'brand']].iloc[product_indices]

recommended_products = get_recommendations("Omega 3 Dietary Supplement", cosine_sim, top_n=10)
print("Rekomendasi untuk 'Omega 3 Dietary Supplement':")
print(recommended_products)
```

Gambar. Konfigurasi Model Recommendation System

SOLUSI LATIHAN 5

Implementasi Machine Learning

Hasil Sistem Rekomendasi, sebagai contoh disini kita mencoba untuk mencari produk yang menyerupai produk "Omega 3 Dietary Supplement". Berikut hasil rekomendasinya:

```
recommended_products = get_recommendations("Omega 3 Dietary Supplement", cosine_sim, top_n=10)
print("Rekomendasi untuk 'Omega 3 Dietary Supplement':")
print(recommended_products)
```

Rekomendasi untuk 'Omega 3 Dietary Supplement':

	name	category \
1898	Double Mousse Gentle Cleansing Foam	Face Wash & Cleansers
5018	SUPERMASK - The Peeling Mask	Face Masks
5464	TROPICALCLEANSE™ Daily Exfoliating Cleanser	Face Wash & Cleansers
5339	Sugar Wonder Drops Lip Primer	Lip Balms & Treatments
7571	PRO Strength Lactic Pore Treatment	Face Serums
692	Power Peel Multi-Acid Resurfacing Pads	Moisturizer & Treatments
1683	Glycolic Exfoliating & Resurfacing Wipes	Moisturizer & Treatments
4739	Firming Sleeping Cream	Night Creams
3071	Vitamin C Glow Face Mask	Mini Size
2847	Hyaluronic Marine Oil-Free Moisture Cushion	Moisturizers

Dari graf diatas dapat dilihat top 10 produk yang memiliki kemiripan dengan produk "Omega 3 Dietary Supplement". Produk dengan nilai kemiripan tertinggi yaitu proudk "Double Mousse Gentle Cleansing Foam" pada category "Face Wash & Cleanser"

SOLUSI LATIHAN 5

Implementasi Machine Learning 2 (Natural Language Processing)

2. Sistem Klasifikasi Kategori Produk dengan TF-IDF dan Logistic Regression

Digunakan dengan mengelompokkan produk berdasarkan fitur 'name' dan 'details' menggunakan TF-IDF word embedding dan Logistic Regression model. Model klasifikasi berhasil mencapai nilai akurasi **82%**. Hal ini memungkinkan otomatisasi pengelompokan produk, meningkatkan akurasi pencarian, serta membantu dalam sistem rekomendasi berbasis kategori. Dengan model ini, produk dapat dikategorikan dengan otomatis, mendukung sistem rekomendasi yang sebelumnya sudah dilakukan, serta mempermudah analisis tren pasar dan preferensi customer

```
import numpy as np
from sklearn.utils.class_weight import compute_class_weight

class_weights = compute_class_weight('balanced', classes=np.unique(train_df['category_encoded']), y=train_df['category_encoded'])
weights_dict = dict(enumerate(class_weights))

model = LogisticRegression(max_iter=500, class_weight=weights_dict)
model.fit(X_train, train_df['category_encoded'])
```

LogisticRegression

LogisticRegression(class_weight={0: np.float64(56.33802816901409),
1: np.float64(28.169014084507044),
2: np.float64(4.333694474539545),
3: np.float64(1.5649452269170578),
4: np.float64(2.965159377316531),
5: np.float64(5.633802816901408),
6: np.float64(6.259780907668231),
7: np.float64(1.1046672190002762),
8: np.float64(11.267605633802816),
9: np.float64(0.6123698714...
20: np.float64(1.3101867016049786),
21: np.float64(2.353531136369635)

```
model.fit(X_train, train_df['category_encoded'])
```

Gambar. Konfigurasi Model Regresi Logistik

SOLUSI LATIHAN 5

Implementasi Machine Learning 2 (Natural Language Processing)

Hasil Sistem Klasifikasi Kategori Produk dengan TF-IDF dan Logistic Regression

Product Name	Actual Category	Predicted Category
Brow Zings Pro Palette	Eye Palettes	Makeup Palettes
Full Frontal Volume Lift Curl Mascara	Mascara	Mascara
Clementine California Cologne Absolue Pure Perfume	Perfume Gift Sets	Perfume Gift Sets
Orange Sanguine Cologne Absolue Pure Perfume	Perfume Gift Sets	Perfume
Pacific Lime Cologne Absolue Pure Perfume Leather	Perfume Gift Sets	Perfume Gift Sets
Vanille Insensée Cologne Absolue Pure Perfume	Perfume Gift Sets	Perfume Gift Sets
Satin Luxe Classic Cream Lipstick	Lipstick	Lipstick
Little One Eyeshadow Palette	Eye Palettes	Eye Palettes
Bois de Balincourt Votive Set	Perfume Gift Sets	Candles & Home Scents
Prism AHA BHA Exfoliating Glow Serum	Face Serums	Facial Peels

Dari graf diatas dapat dilihat contoh 10 produk yang diklasifikasi dari data 'test.csv'. Dari ke 10 produk tampak 6 produk berhasil diprediksi kategorinya dengan tepat, 2 produk diprediksi mendekati dengan kategori sebenarnya, dan 2 produk gagal diprediksi (kategorinya jauh meleset). Hal ini menunjukkan bahwa performa model sudah cukup baik, dengan tingkat akurasi pelatihan sebesar 82%



THANK YOU

Josua Pane