

Nama : Elena Ghini Rachman

NIM : 2502055204

Group8

GSLC Assignment Week 9 Data Mining and Visualization - DTSC6005001

Laporan EDA (Laporan Eksplorasi Data)

Shipping E-Commerce

(Python language)

Data di ambil dari Kaggle dengan link sebagai berikut.

<https://www.kaggle.com/code/hamzamanssor/shipping-e-commerce-ml-models>

Link Google collab:

<https://colab.research.google.com/drive/1jmvfbKNweXrqods-BCh2thAVVCBEuyRO?usp=sharing>

Link Github: https://github.com/elenaghini/Shipping_e-commerce

Harap lakukan tugas berikut untuk tugas GSLC Anda:

1. Pilih 1 variabel dependen dari kumpulan data pilihan Anda untuk AOL, dan satu set variabel independen. Hitung korelasi antara variabel dependen dan variabel independen, analisis apa yang tersirat dari nilai-nilai tersebut. Pastikan menggunakan metode korelasi yang benar sesuai dengan jenis variabel yang dibandingkan!

Jawaban

Data excel csv bernama **shipping_ecommerce** memiliki 10 variabel dan salah satunya memiliki variabel dependen yaitu **customer_rating**. Selanjutnya 9 variabel indenpenden yaitu variabel **Customer_care_calls**, **Prior_purchases**, **Discount_offered**, **Weight_in_gms**, **Class**, **Warehouse_block**, **Mode_of_Shipment**, **Product_importance**, dan **Gender**.

Berikut implementasi menggunakan bahasa Python:

```
#Load the required libraries
import pandas as pd
import numpy as np
import scipy.stats as stats
import statsmodels.api as sm # Import statsmodels.api
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix
from sklearn import linear_model
from sklearn.preprocessing import PolynomialFeatures
from statsmodels.formula.api import ols
from scipy import stats
```

```
#Install and import required libraries for Random Forest
!pip install scikit-learn
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score, accuracy_score
# import data ke data frame pandas
from google.colab import drive
drive.mount('/content/drive')
path = "/content/drive/MyDrive/testing/shipping_ecommerce.csv"
df = pd.read_csv(path)
```

view 5 first list data untuk memastikan data sudah terbaca dengan baik atau tidak

```
df.head()
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

	Customer_care_calls	Customer_rating	Prior_purchases	Discount_offered	Weight_in_gms	Warehouse_block	Mode_of_Shipment	Product_importance	Gender	Class
0	5	4	2	10	5395	A	Ship	medium	M	1
1	4	3	2	6	5867	F	Ship	medium	F	0
2	3	4	2	2	5957	D	Ship	medium	M	0
3	3	1	2	27	2551	D	Ship	medium	M	1
4	7	5	4	9	1329	B	Ship	medium	M	1

Define independent variables and dependent variable

```
IndependentVars = ["Customer_care_calls", "Discount_offered",
"Prior_purchases", "Weight_in_gms", "Class"]
depentVar = "Customer_rating"
```

These lines define the independent variables as a list of column names (IndependentVars) and the dependent variable as a single column name (depentVar).

Loop through all independent variables and calculate Pearson correlation with the dependent variable

```
for IndependentVar in IndependentVars:
    cor_value = df[[depentVar,
IndependentVar]].corr(method='pearson')[depentVar][IndependentVar]
    # Print result to console
    print("Correlation between %s and %s \t: %f" % (depentVar,
IndependentVar, cor_value))
```

This loop iterates over each independent variable, calculates the Pearson correlation coefficient between it and the dependent variable, and prints the correlation value.

Mencari korelasi cara kedua

```
correlation1 = df["Customer_rating"].corr(df["Customer_care_calls"],
method="pearson")
correlation2 = df["Customer_rating"].corr(df["Discount_offered"],
method="pearson")
```

```
correlation3 = df["Customer_rating"].corr(df["Prior_purchases"],
method="pearson")
correlation4 = df["Customer_rating"].corr(df["Weight_in_gms"],
method="pearson")
correlation5 = df["Customer_rating"].corr(df["Class"], method="pearson")

print("Correlation between Customer_rating and Customer_care_calls:",
correlation1)
print("Correlation between Customer_rating and Discount_offered:",
correlation2)
print("Correlation between Customer_rating and Prior_purchases:",
correlation3)
print("Correlation between Customer_rating and Weight_in_gms:", correlation4)
print("Correlation between Customer_rating and Class:", correlation5)
```

Output:

```
Correlation between Customer_rating and Customer_care_calls      : 0.012270
Correlation between Customer_rating and Discount_offered         : -0.003103
Correlation between Customer_rating and Prior_purchases          : 0.013161
Correlation between Customer_rating and Weight_in_gms           : -0.001960
Correlation between Customer_rating and Class                    : 0.013199
```

Metode korelasi Pearson untuk mengukur hubungan linier antara variabel-variabel yang berbeda. Berikut adalah interpretasi nilai-nilai korelasi yang Anda peroleh:

1. Korelasi antara Customer_rating dan Customer_care_calls
Nilai korelasi sebesar 0.01226955 menunjukkan bahwa ada hubungan yang sangat lemah dan hampir tidak signifikan antara rating pelanggan dan jumlah panggilan layanan pelanggan. Korelasi positif menunjukkan bahwa adanya peningkatan rating pelanggan cenderung berkorelasi dengan peningkatan jumlah panggilan layanan pelanggan, tetapi hubungannya sangat rendah.
2. Korelasi antara Customer_rating dan Discount_offered
Nilai korelasi sebesar -0.003103001 menunjukkan bahwa tidak ada hubungan yang signifikan antara rating pelanggan dan diskon yang ditawarkan. Nilai korelasi yang mendekati nol menunjukkan bahwa tidak ada hubungan linier yang jelas antara kedua variabel ini.
3. Korelasi antara Customer_rating dan Prior_purchases
Nilai korelasi sebesar 0.0131613 menunjukkan bahwa ada hubungan yang sangat lemah dan hampir tidak signifikan antara rating pelanggan dan jumlah pembelian sebelumnya. Korelasi positif menunjukkan bahwa adanya peningkatan rating pelanggan cenderung berkorelasi dengan peningkatan jumlah pembelian sebelumnya, tetapi hubungannya sangat rendah.
4. Korelasi antara Customer_rating dan Weight_in_gms
Nilai korelasi sebesar -0.001959518 menunjukkan bahwa tidak ada hubungan yang signifikan antara rating pelanggan dan berat barang dalam gram. Nilai korelasi yang mendekati nol menunjukkan bahwa tidak ada hubungan linier yang jelas antara kedua variabel ini.
5. Korelasi antara Customer_rating dan Class

Nilai korelasi sebesar 0.01319878 menunjukkan bahwa ada hubungan yang sangat lemah dan hampir tidak signifikan antara rating pelanggan dan kelas barang. Korelasi positif menunjukkan bahwa adanya peningkatan rating pelanggan cenderung berkorelasi dengan peningkatan kelas barang, tetapi hubungannya sangat rendah.

Berdasarkan hasil analisis korelasi yang telah dilakukan, kesimpulan yang dapat diambil adalah sebagai berikut:

Tidak ada hubungan yang signifikan antara rating pelanggan dengan jumlah panggilan layanan pelanggan, diskon yang ditawarkan, berat barang dalam gram, dan kelas barang. Korelasi antara rating pelanggan dan variabel-variabel tersebut sangat rendah atau mendekati nol, menunjukkan bahwa hubungan linier antara kedua variabel tersebut hampir tidak ada.

Terdapat hubungan yang sangat lemah dan hampir tidak signifikan antara rating pelanggan dengan jumlah pembelian sebelumnya. Korelasi positif yang sangat rendah menunjukkan bahwa ada kecenderungan peningkatan rating pelanggan yang berkorelasi dengan peningkatan jumlah pembelian sebelumnya, tetapi hubungannya sangat lemah.

2. Mengapa kita perlu melakukan uji statistik? Jika memungkinkan pada kumpulan data Anda, coba lakukan pengujian ini dan analisis apa artinya. Berikut adalah daftar atau artikel yang dapat membantu Anda mempelajari lebih lanjut tentang mereka.

<https://medium.com/@anushka.da3/types-of-statistical-tests-b8ceb90e13b3>

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6639881/>

Jawaban

Kita perlu melakukan uji statistik untuk mendapatkan pemahaman yang lebih baik tentang hubungan antara variabel independen (Customer_care_calls, Prior_purchases, Discount_offered, Weight_in_gms, Class, Warehouse_block, Mode_of_Shipment, Product_importance, dan Gender) dan variabel dependen (customer_rating) dalam dataset shipping_ecommerce. Uji statistik membantu kita dalam menguji hipotesis dan mengevaluasi signifikansi statistik dari hubungan tersebut.

Untuk melakukan pengujian, kita dapat menggunakan berbagai metode statistik, tergantung pada jenis data dan tujuan analisis yang ingin dicapai. Berikut ini beberapa contoh pengujian yang dapat dilakukan:

1. Analisis Korelasi

Kita dapat menggunakan uji korelasi (misalnya, korelasi Pearson) untuk melihat apakah ada hubungan linier antara variabel independen dan variabel dependen. Jika terdapat korelasi yang kuat dan signifikan, maka variabel independen memiliki pengaruh terhadap variabel dependen.

2. Analisis Regresi

Kita dapat melakukan analisis regresi (misalnya, regresi linier) untuk mengukur sejauh mana variabel independen mempengaruhi variabel dependen. Hal ini dapat membantu kita dalam membangun model prediksi untuk customer_rating berdasarkan variabel independen yang ada.

```
# Linear regression model
lm_model = sm.OLS(df['Customer_rating'], sm.add_constant(df[['Prior_purchases', 'Discount_offered'])).fit()
print(lm_model.summary())
```

OLS Regression Results

```
=====
Dep. Variable:      Customer_rating      R-squared:                0.000
Model:              OLS                  Adj. R-squared:           -0.000
Method:             Least Squares        F-statistic:             0.9749
Date:               Fri, 19 May 2023      Prob (F-statistic):      0.377
Time:               06:17:02              Log-Likelihood:          -19411.
No. Observations:   10998                AIC:                     3.883e+04
Df Residuals:       10995                BIC:                     3.885e+04
Df Model:           2
Covariance Type:    nonrobust
=====
```

	coef	std err	t	P> t	[0.025	0.975]
const	2.9498	0.037	79.726	0.000	2.877	3.022
Prior_purchases	0.0121	0.009	1.358	0.175	-0.005	0.029
Discount_offered	-0.0002	0.001	-0.212	0.832	-0.002	0.001

```
=====
Omnibus:            35327.953      Durbin-Watson:           1.985
Prob(Omnibus):      0.000          Jarque-Bera (JB):        768.693
Skew:               0.005          Prob(JB):                1.20e-167
Kurtosis:           1.705          Cond. No.                59.3
=====
```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Dalam model ini, nilai p-value untuk kedua koefisien Prior_purchases (0.175) dan Discount_offered (0.832) lebih besar dari tingkat signifikansi yang umum digunakan (misalnya 0.05), sehingga tidak ada bukti yang cukup untuk menolak hipotesis nol, yang berarti bahwa kedua variabel independen tidak memiliki pengaruh yang signifikan terhadap Customer_rating. Model regresi linear ini memiliki kemampuan yang sangat rendah dalam menjelaskan variasi dalam Customer_rating dan tidak cocok untuk digunakan dalam memprediksi atau menjelaskan Customer_rating berdasarkan variabel independen yang digunakan.

3. Analisis ANOVA

ANOVA adalah metode statistik yang digunakan untuk membandingkan mean (rata-rata) antara tiga atau lebih kelompok independen. Dalam dataset ini, jika kita ingin membandingkan mean customer_rating antara beberapa kelompok berdasarkan variabel independen seperti Warehouse_block, kita dapat menggunakan ANOVA. Hal ini akan membantu kita mengetahui apakah ada perbedaan signifikan dalam customer_rating di antara kelompok-kelompok tersebut.

```
# ANOVA
anova_model = ols('Customer_rating ~ Warehouse_block', data=df).fit()
print(anova_model.summary())
```

OLS Regression Results

Dep. Variable:	Customer_rating	R-squared:	0.000			
Model:	OLS	Adj. R-squared:	-0.000			
Method:	Least Squares	F-statistic:	0.4766			
Date:	Sat, 20 May 2023	Prob (F-statistic):	0.753			
Time:	03:50:24	Log-Likelihood:	-19411.			
No. Observations:	10998	AIC:	3.883e+04			
Df Residuals:	10993	BIC:	3.887e+04			
Df Model:	4					
Covariance Type:	nonrobust					

	coef	std err	t	P> t	[0.025	0.975]
Intercept	2.9569	0.033	89.519	0.000	2.892	3.022
Warehouse_block[T.B]	0.0235	0.047	0.503	0.615	-0.068	0.115
Warehouse_block[T.C]	0.0371	0.047	0.795	0.427	-0.054	0.129
Warehouse_block[T.D]	0.0611	0.047	1.309	0.191	-0.030	0.153
Warehouse_block[T.F]	0.0398	0.040	0.985	0.325	-0.039	0.119

Omnibus:	34956.789	Durbin-Watson:	1.985
Prob(Omnibus):	0.000	Jarque-Bera (JB):	768.164
Skew:	0.005	Prob(JB):	1.57e-167
Kurtosis:	1.705	Cond. No.	6.44

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Berikut adalah hasil analisis ANOVA yang telah dijalankan:

- Warehouse_block
Derajat kebebasan (df) antara kelompok Warehouse_block adalah 4.
Jumlah kuadrat antara kelompok (Sum Sq) adalah 4.
Mean square (Mean Sq) adalah 0.9526.
Nilai F yang diperoleh adalah 0.477.
Nilai p-value (Pr(>F)) yang diperoleh adalah 0.753.
- Residuals
Derajat kebebasan (df) untuk residu adalah 10993.
Jumlah kuadrat residu (Sum Sq) adalah 21972.
Mean square residu (Mean Sq) adalah 1.9987.

Kesimpulannya hasil analisis ANOVA menunjukkan bahwa tidak ada perbedaan yang signifikan dalam rata-rata Customer_rating antara kelompok Warehouse_block. Nilai p-value yang tinggi (0.753) menunjukkan tidak adanya bukti yang cukup untuk menolak hipotesis nol, yang berarti tidak ada perbedaan yang signifikan dalam rata-rata Customer_rating antara kelompok Warehouse_block. Analisis ANOVA menunjukkan bahwa variabel Warehouse_block tidak memiliki pengaruh yang signifikan terhadap Customer_rating. Tidak ada perbedaan yang signifikan dalam rata-rata Customer_rating antara kelompok Warehouse_block.

- Chi-Square
Uji chi-square digunakan untuk menguji hubungan antara dua variabel kategorikal. Dalam data set ini, kita dapat menggunakan uji chi-square untuk menguji apakah ada hubungan antara variabel independen (misalnya, Warehouse_block, Mode_of_Shipment, Product_importance) dan variabel dependen (customer_rating). Misalnya, kita dapat menguji apakah ada hubungan antara Product_importance dan Mode_of_Shipment dan yang lainnya, contoh implementasinya sebagai berikut.

```

0 d # Chi-square test
contingency_table = pd.crosstab(df['Mode_of_Shipment'], df['Product_importance'])
result = stats.chi2_contingency(contingency_table)
print("\nMode of Shipment ~ Product Importance:")
print("Chi-square statistic:", result[0])
print("P-value:", result[1])

contingency_table1 = pd.crosstab(df['Warehouse_block'], df['Mode_of_Shipment'])
result1 = stats.chi2_contingency(contingency_table1)
print("\nWarehouse Block ~ Mode of Shipment:")
print("Chi-square statistic:", result1[0])
print("P-value:", result1[1])

contingency_table2 = pd.crosstab(df['Warehouse_block'], df['Product_importance'])
result2 = stats.chi2_contingency(contingency_table2)
print("\nWarehouse Block ~ Product Importance:")
print("Chi-square statistic:", result2[0])
print("P-value:", result2[1])

contingency_table3 = pd.crosstab(df['Gender'], df['Product_importance'])
result3 = stats.chi2_contingency(contingency_table3)
print("\nGender ~ Product Importance:")
print("Chi-square statistic:", result3[0])
print("P-value:", result3[1])

```

Mode of Shipment ~ Product Importance:
 Chi-square statistic: 2.2138917567882053
 P-value: 0.696486805133734

Warehouse Block ~ Mode of Shipment:
 Chi-square statistic: 0.027478221360348262
 P-value: 0.999999998531578

Warehouse Block ~ Product Importance:
 Chi-square statistic: 7.999960531829085
 P-value: 0.4334739757616384

Gender ~ Product Importance:
 Chi-square statistic: 1.1182809304120136
 P-value: 0.5717002489789514

Dalam analisis Chi-square yang kita lakukan, kita menguji hubungan antara beberapa variabel kategorikal dalam dataset Anda. Berikut adalah hasil analisis Chi-square yang Anda jalankan:

1. Chi-square test antara Warehouse_block dan Mode_of_Shipment

Statistik Chi-square (X-squared) yang diperoleh adalah 0.027478.

Derajat kebebasan (df) adalah 8.

Nilai p-value yang diperoleh adalah 1.

Kesimpulannya tidak ada hubungan yang signifikan antara variabel Warehouse_block dan Mode_of_Shipment. Nilai p-value yang tinggi (1) menunjukkan tidak adanya bukti yang cukup untuk menolak hipotesis nol, yang berarti tidak ada hubungan yang signifikan antara kedua variabel tersebut.

2. Chi-square test antara Mode_of_Shipment dan Product_importance

Statistik Chi-square (X-squared) yang diperoleh adalah 2.2139.

Derajat kebebasan (df) adalah 4.

Nilai p-value yang diperoleh adalah 0.6965.

Kesimpulannya tidak ada hubungan yang signifikan antara variabel Mode_of_Shipment dan Product_importance. Nilai p-value yang tinggi (0.6965) menunjukkan tidak adanya bukti yang cukup untuk menolak hipotesis nol, yang berarti tidak ada hubungan yang signifikan antara kedua variabel tersebut.

3. Chi-square test antara Gender dan Product_importance

Statistik Chi-square (X-squared) yang diperoleh adalah 1.1183.

Derajat kebebasan (df) adalah 2.

Nilai p-value yang diperoleh adalah 0.5717.

Kesimpulannya tidak ada hubungan yang signifikan antara variabel Gender dan Product_importance.

Nilai p-value yang tinggi (0.5717) menunjukkan tidak adanya bukti yang cukup untuk menolak hipotesis nol, yang berarti tidak ada hubungan yang signifikan antara kedua variabel tersebut.

4. Chi-square test antara Warehouse_block dan Product_importance

Statistik Chi-square (X-squared) yang diperoleh adalah 8.

Derajat kebebasan (df) adalah 8.

Nilai p-value yang diperoleh adalah 0.4335.

Kesimpulannya tidak ada hubungan yang signifikan antara variabel Warehouse_block dan Product_importance. Nilai p-value yang tinggi (0.4335) menunjukkan tidak adanya bukti yang cukup untuk menolak hipotesis nol, yang berarti tidak ada hubungan yang signifikan antara kedua variabel tersebut.

Dalam kesimpulannya, analisis Chi-square menunjukkan bahwa tidak ada hubungan yang signifikan antara variabel-variabel yang diuji dalam dataset Anda. Nilai p-value yang tinggi menunjukkan tidak adanya bukti yang cukup untuk menolak hipotesis nol, yang berarti tidak ada hubungan yang signifikan antara variabel-variabel tersebut.

Setelah melakukan pengujian, kita dapat melihat hasilnya untuk mengambil kesimpulan. Jika terdapat hubungan yang signifikan antara variabel independen dan variabel dependen, maka variabel independen tersebut memiliki pengaruh terhadap customer_rating dalam konteks dataset shipping_ecommerce. Namun, jika tidak ada hubungan yang signifikan, maka variabel independen mungkin tidak memiliki pengaruh yang kuat terhadap customer_rating atau terdapat faktor-faktor lain yang perlu dipertimbangkan.

5. T-test

Metode t-test digunakan untuk membandingkan mean (rata-rata) dari dua kelompok yang independen. Dalam dataset ini, kita dapat menggunakan t-test untuk membandingkan mean customer_rating antara dua kelompok yang berbeda. Misalnya, kita dapat membandingkan mean customer_rating antara kelompok berbeda berdasarkan variabel Gender. Hal ini akan membantu kita mengetahui apakah ada perbedaan signifikan dalam customer_rating antara kelompok gender tertentu.

```
# T-test cara kedua bisa dicari dengan sebagai berikut
male_ratings = df.loc[df['Gender'] == 'M', 'Customer_rating']
female_ratings = df.loc[df['Gender'] == 'F', 'Customer_rating']
t_stat_gender, p_value_gender = stats.ttest_ind(male_ratings, female_ratings)
print("T-test results for Customer_rating between males and females:")
print("T-statistic:", t_stat_gender)
print("P-value:", p_value_gender)

class1_ratings = df.loc[df['Class'] == 0, 'Customer_rating']
class2_ratings = df.loc[df['Class'] == 1, 'Customer_rating']
t_stat_class, p_value_class = stats.ttest_ind(class1_ratings, class2_ratings)
print("\nT-test results for Customer_rating between class 0 and class 1:")
print("T-statistic:", t_stat_class)
print("P-value:", p_value_class)
```

```
T-test results for Customer_rating between males and females:
T-statistic: 0.29774346481792036
P-value: 0.7659045919177931
```

```
T-test results for Customer_rating between class 0 and class 1:
T-statistic: -1.3841684557345875
P-value: 0.16633497048509163
```

1. Dua sampel untuk membandingkan rata-rata dari dua kelompok data yang berbeda, yaitu male_ratings dan female_ratings.

Hasil pengujian menunjukkan nilai t-statistik sebesar 0.29776 dengan derajat kebebasan (df) sebesar 10995. Nilai p-value yang dihasilkan adalah 0.7659.

Pada tingkat signifikansi 0.05, jika p-value lebih kecil dari 0.05, maka kita dapat menolak hipotesis nol (null hypothesis) bahwa tidak ada perbedaan yang signifikan antara rata-rata kedua kelompok. Namun, dalam kasus ini, p-value (0.7659) lebih besar dari 0.05, sehingga tidak ada cukup bukti statistik untuk menolak hipotesis nol. Oleh karena itu, dapat dikatakan bahwa tidak ada perbedaan yang signifikan antara rata-rata male_ratings dan female_ratings.

Selain itu, interval kepercayaan 95% untuk perbedaan rata-rata antara kedua kelompok adalah dari -0.04481952 hingga 0.06087470. Hal ini menunjukkan rentang perkiraan di mana perbedaan sebenarnya antara rata-rata kedua kelompok mungkin berada.

Estimasi rata-rata untuk male_ratings adalah 2.994499, sementara estimasi rata-rata untuk female_ratings adalah 2.986472.

2. Dua sampel untuk membandingkan rata-rata dari dua kelompok data yang berbeda, yaitu class1 dan class2.

Hasil pengujian menunjukkan nilai t-statistik sebesar -1.3838 dengan derajat kebebasan (df) sebesar 9505.5. Nilai p-value yang dihasilkan adalah 0.1665.

Pada tingkat signifikansi 0.05, jika p-value lebih kecil dari 0.05, maka kita dapat menolak hipotesis nol (null hypothesis) bahwa tidak ada perbedaan yang signifikan antara rata-rata kedua kelompok. Namun, dalam kasus ini, p-value (0.1665) lebih besar dari 0.05, sehingga tidak ada cukup bukti statistik untuk menolak hipotesis nol. Oleh karena itu, dapat dikatakan bahwa tidak ada perbedaan yang signifikan antara rata-rata class1 dan class2.

Selain itu, interval kepercayaan 95% untuk perbedaan rata-rata antara kedua kelompok adalah dari -0.09191156 hingga 0.01584445. Hal ini menunjukkan rentang perkiraan di mana perbedaan sebenarnya antara rata-rata kedua kelompok mungkin berada.

Estimasi rata-rata untuk class1 adalah 2.967756, sementara estimasi rata-rata untuk class2 adalah 3.005790.

Uji statistic lain dengan summary statistics:

```
#Summary statistics
summary = df.describe()
print(summary)
```

	Customer_care_calls	Customer_rating	Prior_purchases
count	10998.000000	10998.000000	10998.000000
mean	4.054555	2.990453	3.567558
std	1.141497	1.413635	1.522924
min	2.000000	1.000000	2.000000
25%	3.000000	2.000000	3.000000
50%	4.000000	3.000000	3.000000
75%	5.000000	4.000000	4.000000
max	7.000000	5.000000	10.000000

	Discount_offered	Weight_in_gms	Class
count	10998.000000	10998.000000	10998.000000
mean	13.373704	3633.873522	0.596745
std	16.206183	1635.382636	0.490573
min	1.000000	1001.000000	0.000000
25%	4.000000	1839.250000	0.000000
50%	7.000000	4149.000000	1.000000
75%	10.000000	5049.750000	1.000000
max	65.000000	7846.000000	1.000000

3. Latih model pembelajaran mesin prediktif di R untuk memprediksi variabel dependen pilihan Anda! Ingat, harap minta semua anggota grup Anda mencoba model prediksi yang berbeda (mis. Satu menggunakan

Random Forest, yang lain menggunakan Support Vector Machine, sementara yang lain menggunakan Multi-layer Perceptron). Evaluasi model dan jelaskan hasilnya!

Harap kirimkan kode R Anda (penurunan harga R atau R, keduanya dapat diterima. Pastikan untuk menulis beberapa dokumentasi untuk menjelaskan apa yang dilakukan setiap baris kode) dan laporan EDA dalam format .pdf (Anda dapat menggunakan kata, powerpoint, canva, dll.)! Anda disarankan untuk mengunggah kode ke Github dan memasukkan tautannya ke dalam laporan, sehingga Anda dapat menerbitkannya nanti sebagai bagian dari portofolio Anda selama Anda menjadi siswa. Jangan ragu untuk menghubungi saya jika Anda memiliki pertanyaan. Terima kasih dan GBU semuanya.

Catatan tambahan: JANGAN COPY DAN PASTE LAPORAN DARI TEMAN ANDA! Tulis versi Anda sendiri! Plagiarisme tidak akan ditoleransi!

Jawaban

Menggunakan Random Forest

Berikut code Python disertai penjelasannya:

```
# Preprocess the data
df['Customer_rating'] = df['Customer_rating'].astype('category')
df['Customer_care_calls'] = pd.to_numeric(df['Customer_care_calls'])
df['Prior_purchases'] = pd.to_numeric(df['Prior_purchases'])
df['Discount_offered'] = pd.to_numeric(df['Discount_offered'])
df['Weight_in_gms'] = pd.to_numeric(df['Weight_in_gms'])
```

Bagian ini melakukan pra-pemrosesan data dengan mengonversi kolom Customer_rating menjadi tipe data kategori (category) dan mengonversi kolom numerik (Customer_care_calls, Prior_purchases, Discount_offered, Weight_in_gms) menjadi tipe data numerik menggunakan fungsi pd.to_numeric().

```
# Encode categorical variables
cat_cols = ['Warehouse_block', 'Mode_of_Shipment', 'Product_importance',
'Gender', 'Class']
label_encoders = {}
for col in cat_cols:
    label_encoders[col] = LabelEncoder()
    df[col] = label_encoders[col].fit_transform(df[col])
```

Bagian ini melakukan encoding pada variabel kategorikal dalam DataFrame menggunakan label encoding. Kolom kategorikal yang ditentukan dalam cat_cols diiterasi, dan objek LabelEncoder dibuat dan dilatih pada setiap kolom. Kemudian, kolom yang sesuai dalam DataFrame diubah menggunakan encoder yang dilatih.

```
# Split the data into training and testing sets
X = df.drop('Customer_rating', axis=1)
y = df['Customer_rating']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1,
random_state=123)
```

Bagian ini membagi data yang telah diproses menjadi variabel independen (X) dan variabel dependen (y). Fungsi `train_test_split` dari `scikit-learn` digunakan untuk secara acak membagi data menjadi set pelatihan dan pengujian. Ukuran set pengujian diatur menjadi 0,1 (10% dari data), dan `random_state` diatur sebagai 123 untuk tujuan reproducibility.

```
# Train the random forest model
model = RandomForestClassifier(random_state=123)
model.fit(X_train, y_train)
```

Bagian ini menginisialisasi model klasifikasi random forest dengan `random_state` 123 dan melatih model pada data pelatihan (`X_train` dan `y_train`) menggunakan metode `fit`.

```
# Make predictions on the test data
predictions = model.predict(X_test)
```

Bagian ini menggunakan model random forest yang telah dilatih untuk melakukan prediksi pada data pengujian (`X_test`) menggunakan metode `predict`.

```
# Calculate accuracy
accuracy = accuracy_score(y_test, predictions)
print("Accuracy:", accuracy)
```

```
output:
Accuracy: 0.19454545454545455
```

Bagian ini menghitung akurasi dari prediksi model dengan membandingkan label prediksi (`predictions`) dengan label aktual (`y_test`) menggunakan fungsi `accuracy_score` dari `scikit-learn`. Skor akurasi kemudian dicetak ke konsol.

Kita melakukan evaluasi model menggunakan metrik akurasi. Akurasi mengukur seberapa banyak prediksi model yang cocok dengan label aktual pada data pengujian.

Hasil evaluasi model menunjukkan akurasi sebesar 0.1945, atau sekitar 19.45%. Ini berarti model yang telah dilatih dengan menggunakan random forest classifier mampu memprediksi dengan benar sekitar 19.45% dari data pada set pengujian.

Namun, akurasi yang relatif rendah ini menunjukkan bahwa model yang digunakan mungkin tidak sesuai dengan data atau fitur-fitur yang ada. Mungkin ada faktor-faktor lain yang tidak diperhitungkan oleh model, atau mungkin perlu dilakukan penyesuaian lebih lanjut pada pemrosesan data atau pemilihan model.

Ada beberapa **faktor** yang dapat menyebabkan akurasi rendah dan beberapa cara untuk meningkatkannya. Berikut adalah beberapa faktor yang mungkin mempengaruhi akurasi rendah dan beberapa solusi yang dapat diterapkan sebagai berikut:

1. Kurangnya jumlah data pelatihan
Jumlah data pelatihan yang terbatas dapat mengakibatkan model memiliki keterbatasan dalam mempelajari pola yang ada dalam data. Solusinya adalah mencoba mendapatkan lebih banyak data pelatihan jika memungkinkan. Jika tidak, teknik augmentasi data seperti flipping, rotasi, atau zoom dapat diterapkan untuk membuat variasi baru dari data yang ada.

2. Ketidakseimbangan kelas

Jika jumlah sampel dalam setiap kelas tidak seimbang, model cenderung memiliki kecenderungan untuk memprediksi lebih banyak sampel ke kelas mayoritas. Hal ini dapat menyebabkan akurasi yang rendah karena performa model yang buruk dalam mengidentifikasi sampel pada kelas minoritas. Solusinya adalah menggunakan teknik penanganan ketidakseimbangan kelas seperti oversampling (duplikasi data minoritas) atau undersampling (mengurangi jumlah data mayoritas).

3. Parameter yang tidak dioptimalkan

Hyperparameter pada model RandomForest mungkin tidak diatur dengan optimal. Penting untuk menjalankan eksperimen dengan berbagai nilai hyperparameter untuk menemukan konfigurasi yang lebih baik. Beberapa hyperparameter yang relevan untuk RandomForest antara lain jumlah pohon (`n_estimators`), kedalaman pohon (`max_depth`), dan jumlah fitur yang dipertimbangkan saat mencari split (`max_features`).

4. Fitur yang tidak relevan atau kurang informatif

Fitur yang tidak memiliki hubungan yang kuat dengan variabel target atau fitur yang memiliki banyak nilai yang hilang atau tidak relevan dapat mengurangi kinerja model. Penting untuk melakukan analisis fitur dan mempertimbangkan untuk menghilangkan atau mengganti fitur yang tidak memberikan kontribusi yang signifikan terhadap prediksi.

Kurangnya validasi silang: Dalam kasus ini, tidak ada informasi yang diberikan tentang metode validasi yang digunakan. Validasi silang seperti validasi silang lipat-k adalah penting untuk mengevaluasi model secara obyektif dan mengurangi kemungkinan overfitting. Pastikan untuk menggunakan metode validasi yang tepat dan mempertimbangkan validasi silang untuk mendapatkan estimasi kinerja yang lebih akurat.