Rakamin Academy

Homework
Data Pre-Processing

Final Project - Stage 2

By: ec-Team (Kelompok 9)

1 . Data Cleansing

Lakukan pembersihan data, sesuai yang diajarkan di kelas, seperti:

A. Handle missing values
B. Handle duplicated data
C. Handle outliers
D. Feature transformation
E. Feature encoding
F. Handle class imbalance

Di laporan homework, tuliskan apa saja yang telah dilakukan dan metode yang digunakan.

* Tetap tuliskan jika memang ada tidak yang perlu di-handle (contoh: “Tidak perlu feature encoding karena semua feature sudah numerical” atau “Outlier tidak di-handle karena akan fokus menggunakan model yang robust terhadap outlier”).

2. Feature Engineering

Cek feature yang ada sekarang, lalu lakukan:

A. Feature selection (membuang feature yang kurang relevan atau redundan)

B. Feature extraction (membuat feature baru dari feature yang sudah ada)

C. Tuliskan minimal 4 feature tambahan (selain yang sudah tersedia di dataset) yang mungkin akan sangat membantu membuat performansi model semakin bagus (ini hanya ide saja, untuk menguji kreativitas teman-teman, tidak perlu benar-benar dicari datanya dan tidak perlu diimplementasikan)

* Untuk 2A & 2B, tetap tuliskan jika memang tidak bisa dilakukan (contoh: “Semua feature digunakan untuk modelling (tidak ada yang dihapus), karena semua feature relevan”)

Rakamin Academy

Preprocessing Data

1. Features extraction
2. Features selection
3. Handle missing values
4. Handle duplicates data
5. Handle outliers
6. Feature Transformation
7. Feature Scaling
8. Feature Encoding
9. Handle Class Imbalance

Feature Extraction (membuat feature baru dari feature yang sudah ada)

Nama Feature	Deskripsi Feature
Total_visit_duration	Jumlah durasi dari feature Administrative_Duration, Informational_Duration dan ProductRelated_Duration
Total_pageviews	Jumlah halaman yang dikunjungi dari feature Administrative, Informational, dan ProductRelated

Feature Extraction ini tidak digunakan untuk preprocessing dan modelling karena dengan feature yang ada sudah cukup relevan

Rekomendasi Feature Tambahan (selain yang sudah tersedia di dataset)

- UserID / Invoice sebagai identifier
- Date-time untuk memprediksi waktu yang tepat untuk campaign notification
- Gender untuk mengoptimalisasi rekomendasi produk
- Tanggal lahir (DD/MM/YYYY) untuk optimalisasi dan campaign spesial diskon pada hari kelahiran
- Tanggal registrasi & Riwayat pembelian untuk menghitung customer lifetime value

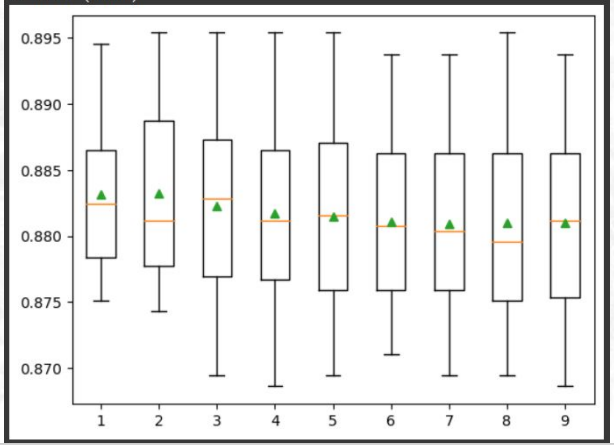
Feature Selection - Numerical Features : ANOVA

```python
# define the evaluation method
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=42)
# define the pipeline to evaluate
model = LogisticRegression(solver='liblinear')
fs = SelectKBest(score_func=f_classif)
pipeline = Pipeline(steps=[('anova',fs), ('lr', model)])
# define the grid
grid = dict()
grid['anova__k'] = [i+1 for i in range(X.shape[1])]
# define the grid search
search = GridSearchCV(pipeline, grid, scoring='accuracy', n_jobs=-1, cv=cv)
# perform the search
results = search.fit(X, y)
# summarize best
print('Best Mean Accuracy: %.3f' % results.best_score_)
print('Best Config: %s' % results.best_params_)
```

```
Best Mean Accuracy: 0.883
Best Config: {'anova__k': 2}
```

```python
# evaluate a give model using cross-validation
def evaluate_model(model, X, y):
 cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=42)
 scores = cross_val_score(model, X, y, scoring='accuracy', cv=cv, n_jobs=-1, error_score='raise')
 return scores

# define number of features to evaluate
num_features = [i+1 for i in range(X.shape[1])]
# enumerate each number of features
results = list()
for k in num_features:
 # create pipeline
 model = LogisticRegression(solver='liblinear')
 fs = SelectKBest(score_func=f_classif, k=k)
 pipeline = Pipeline(steps=[('anova',fs), ('lr', model)])
 # evaluate the model
 scores = evaluate_model(pipeline, X, y)
 results.append(scores)
 # summarize the results
 print('>%d %.3f (%.3f)' % (k, mean(scores), std(scores)))
# plot model performance for comparison
pyplot.boxplot(results, labels=num_features, showmeans=True)
pyplot.show()
```

```
>1 0.883 (0.006)
>2 0.883 (0.007)
>3 0.882 (0.006)
>4 0.882 (0.007)
>5 0.881 (0.006)
>6 0.881 (0.006)
>7 0.881 (0.006)
>8 0.881 (0.006)
>9 0.881 (0.006)
```



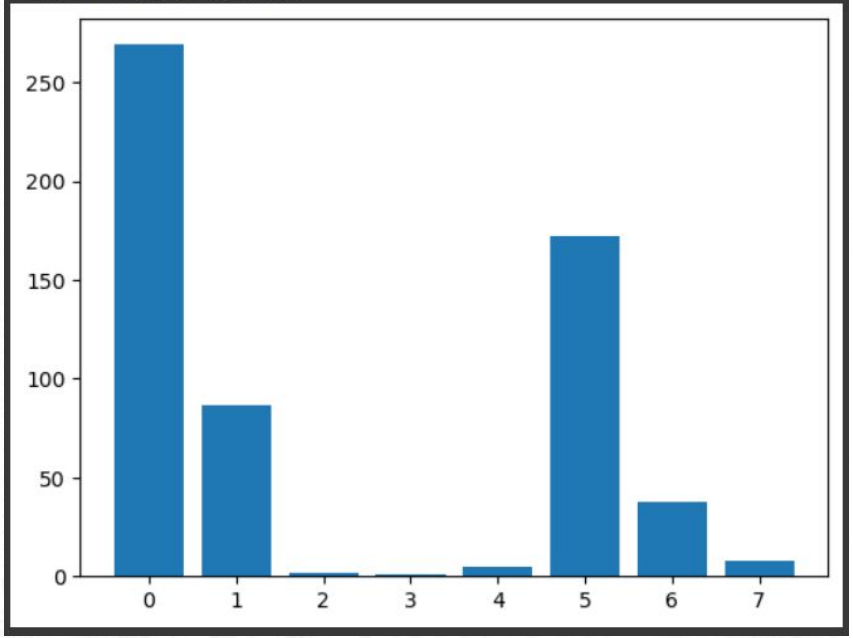
Berdasarkan hasil ANOVA seharusnya ada 2 numerical feature yang terbaik untuk dipilih dalam modelling. Namun, berdasarkan visualisasi disamping feature 1, 2, 4, 6, 7 dan 8 (Admisnistrative, Administrative_Duration, Informational_Duration, ProductRelated_Duration, BounceRates dan ExitRates) merupakan yang terbaik karena scorenya melewati garis. Jika melewati garis tandanya feature tersebut berkorelasi dengan target.

Feature Selection - Categorical Features : Chi-square

Rakamin Academy

```python
# split into input (X) and output (y) variables
X = df[['SpecialDay', 'Month', 'OperatingSystems', 'Browser', 'Region', 'TrafficType', 'VisitorType', 'Weekend']].values
y = df['Revenue'].values

# format all fields as string
X = X.astype(str)

# prepare train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=2, random_state=42)

# prepare input data
oe = OrdinalEncoder()
oe.fit(X_train)
X_train_enc = oe.transform(X_train)
X_test_enc = oe.transform(X_test)

# prepare output data
le = LabelEncoder()
le.fit(y_train)
y_train_enc = le.transform(y_train)
y_test_enc = le.transform(y_test)

# feature selection
fs = SelectKBest(score_func=chi2, k='all')
fs.fit(X_train_enc, y_train_enc)
X_train_fs = fs.transform(X_train_enc)
X_test_fs = fs.transform(X_test_enc)

# get feature names
feature_names = ['SpecialDay', 'Month', 'OperatingSystems', 'Browser', 'Region', 'TrafficType', 'VisitorType', 'Weekend']

# print feature names and their scores
for i in range(len(fs.scores_)):
    print('Feature %s: %f' % (feature_names[i], fs.scores_[i]))
```

```
Feature SpecialDay: 269.089947
Feature Month: 86.221173
Feature OperatingSystems: 1.962391
Feature Browser: 0.691710
Feature Region: 4.455475
Feature TrafficType: 172.424443
Feature VisitorType: 37.529341
Feature Weekend: 8.146599
```



Berdasarkan hasil Chi-square, feature Special Day (9), TrafficType (14), Month (10), dan VisitorType (15) memiliki korelasi kuat dengan target.

Data Cleansing

Handle missing values

Tidak perlu dilakukan *handle missing values*, karena dalam dataset ini tidak ada nilai null pada setiap featurenya.

```
df.isna().sum()
```

```
Administrative             0
Administrative_Duration    0
Informational              0
Informational_Duration     0
ProductRelated             0
ProductRelated_Duration    0
BounceRates                0
ExitRates                  0
PageValues                 0
SpecialDay                 0
Month                      0
OperatingSystems           0
Browser                    0
Region                     0
TrafficType                0
VisitorType                0
Weekend                    0
Revenue                    0
dtype: int64
```

Handle duplicates data

```
df.duplicated().sum()
```

```
125
```

Ditemukan adanya 125 row yang duplikat. Meskipun tidak ada feature identifier, dataset ini merupakan data real-time sehingga data yang duplikat tersebut perlu didrop. ec-Team berasumsi data duplikat tersebut terjadi karena adanya error saat user berkunjung.

Jumlah data setelah drop data duplikat adalah 12,205.

Data Cleansing

Rakamin Academy

Handle outliers

Untuk mengidentifikasi dan menghapus outlier, Ec-team menggunakan interquartile range(IQR). Tim kami memutuskan untuk memotong hanya data yang terjauh yang mana terletak antara percentile ke-2 dan ke-98. Jika dipotong terlalu banyak beresiko menghilangkan informasi yang penting.

Handling Outlier

In [68]:
```
# Hitung nilai Q1, Q3, dan IQR dari data train
Q1_train = df_copy.quantile(0.02)
Q3_train = df_copy.quantile(0.98)
IQR_train = Q3_train - Q1_train

# Lakukan feature scaling pada data train menggunakan nilai Q1, Q3, dan IQR dari data tra
df_copy = (df_copy - Q1_train) / IQR_train
```

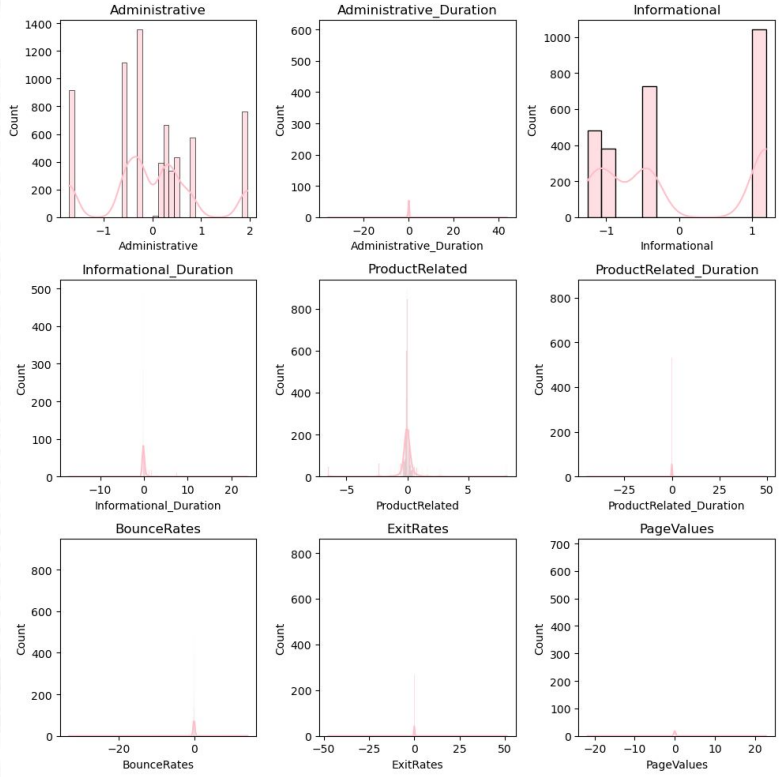
Data Cleansing

Feature transformation

ec-Team menggunakan metode Yeo-Johnson untuk mengubah distribusi data dari feature numerical mendekati normal

```
# Inisialisasi objek StandardScaler dan terapkan pada numerical_data

# melakukan scaling pada data continuous
continuous_cols = ['Administrative', 'Administrative_Duration', 'Informational', 'Informat
# Membuat objek PowerTransformer
pt = PowerTransformer(method='yeo-johnson')

#Transformasi kolom-kolom dalam daftar continuous_cols
df_copy[continuous_cols] = pt.fit_transform(df_copy[continuous_cols])
#df_copy[continuous_cols] = np.sqrt(df_copy[continuous_cols])
#df_copy[continuous_cols] = np.log(df_copy[continuous_cols])
#df_copy[continuous_cols] = np.reciprocal(df_copy[continuous_cols])
```

```
# Membuat ukuran plot
plt.figure(figsize=(10,10))

# Iterasi pada setiap kolom dalam daftar continuous_cols
for i, col in enumerate(continuous_cols):
    plt.subplot(3, 3, i+1)
    sns.histplot(df_copy[col], kde=True, color='pink')
    plt.title(col)

# Menampilkan plot
plt.tight_layout()
plt.show()
```



Data Cleansing

Rakamin Academy

Feature scaling

ec-Team melakukan scaling dengan metode StandardScaler untuk menormalisasikan data agar performa model lebih optimal

	Month_Aug	Month_Sep	Month_Oct	Month_Nov	Month_Dec	VisitorType_New_Visitor	VisitorType_Other	VisitorType_Returning_Visitor	Weekend_False	Weekend_True
count	12205.000000	12205.000000	12205.000000	12205.000000	12205.000000	12205.000000	12205.000000	12205.000000	12205.000000	12205.000000
mean	0.035477	0.036706	0.044982	0.244326	0.139779	0.138714	0.006637	0.854650	0.765752	0.234248
std	0.184990	0.188047	0.207272	0.429705	0.346772	0.345662	0.081198	0.352468	0.423545	0.423545
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	1.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	1.000000	0.000000
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	1.000000	0.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Data Cleansing

Feature encoding

- Label encoding dilakukan untuk mengubah value dari feature Revenue menjadi numeric
- Hot encoding dilakukan untuk mengubah value dari feature VisitorType, Weekend, dan Month menjadi numeric

```
# One-hot encoding pada kolom Month, VisitorType, dan Weekend
df_copy = pd.get_dummies(df_copy, columns=['Month', 'VisitorType', 'Weekend'])

# Label encoding pada kolom Revenue
le = LabelEncoder()
df_copy['Revenue'] = le.fit_transform(df_copy['Revenue'])

# Menampilkan hasil encoding
print(df_copy.head())
print(df_copy['Revenue'].unique())
```

```
     Administrative  Administrative_Duration  Informational  \
0                 0                      0.0              0
117               0                      0.0              0
118               0                      0.0              0
119               0                      0.0              0
120               0                      0.0              0

     Informational_Duration  ProductRelated  ProductRelated_Duration  \
0                       0.0               1                      0.0
117                     0.0              11                    577.0
118                     0.0               8                     95.0
119                     0.0               6                    301.5
120                     0.0               6                    243.0

     BounceRates  ExitRates  PageValues  SpecialDay  ...  Month_Aug  \
0       0.200000   0.200000         0.0         0.0  ...          0
117     0.018182   0.027273         0.0         0.2  ...          0
118     0.000000   0.075000         0.0         0.0  ...          0
119     0.000000   0.111111         0.0         0.0  ...          0
120     0.000000   0.033333         0.0         0.0  ...          0

     Month_Sep  Month_Oct  Month_Nov  Month_Dec  VisitorType_New_Visitor  \
0            0          0          0          0                        0
117          0          0          0          0                        0
118          0          0          0          0                        0
119          0          0          0          0                        0
120          0          0          0          0                        0

     VisitorType_Other  VisitorType_Returning_Visitor  Weekend_False  \
0                    0                              1              1
117                  0                              1              1
118                  0                              1              1
119                  0                              1              1
120                  0                              1              1

     Weekend_True
0               0
117             0
118             0
119             0
120             0
```

Data Cleansing

Rakamin Academy

Handle Class Imbalance

ec-Team menggunakan metode SMOTE untuk mengatasi imbalance data.

```
from imblearn.over_sampling import SMOTE

# Separate features and target
X = df_copy.iloc[:,:-1]
y = df_copy.iloc[:, -1]

# Apply SMOTE to training data
oversample = SMOTE()
X_train_resampled, y_train_resampled = oversample.fit_resample(X_train, y_train)
```

3. Git

Upload project teman-teman di sebuah repository git. Berkolaborasilah di Git jika ada perubahan version dari waktu ke waktu.

A. Buat Repository Git
B. Upload file notebook atau file pengerjaan lainnya pada repository tersebut

Untuk file README, dapat merupakan summary dari proses data preproses yang telah dilakukan. Boleh menggunakan repositori yang sama atau membuat baru.

Link Git ec-Team:
https://github.com/EC-Teams/Final-Project-Online-Shopping-Intention