

HOMework

WEEK 15

Final Project - Stage 2

Kelompok 2 - Synergies:

1. Mellia Anggreani
2. Burhanuddin Yusuf Robbani
3. David Melanius Nai
4. Alfath Arrahman
5. Moch Agung Laksono
6. Dzul Wulan Ningtyas
7. Zaima Syarifa Asshafa



All Default Features from Dataset

```
df_bank.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
#   Column             Non-Null Count  Dtype
---  -
0   RowNumber          10000 non-null  int64
1   CustomerId         10000 non-null  int64
2   Surname            10000 non-null  object
3   CreditScore        10000 non-null  int64
4   Geography          10000 non-null  object
5   Gender             10000 non-null  object
6   Age               10000 non-null  int64
7   Tenure            10000 non-null  int64
8   Balance           10000 non-null  float64
9   NumOfProducts     10000 non-null  int64
10  HasCrCard         10000 non-null  int64
11  IsActiveMember    10000 non-null  int64
12  EstimatedSalary   10000 non-null  float64
13  Exited            10000 non-null  int64
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

Diperoleh dataset dengan isi **10K baris** dan **14 kolom** dengan **kolom Exited sebagai variabel target**. Terlihat semua atribut sudah memiliki tipe data yang sesuai dan isi datanya juga sudah sesuai.

Variabel target yang digunakan untuk machine learning adalah **kolom exited** dan sisanya adalah variabel fitur. Metode ML yang digunakan adalah **tipe supervised learning** sebab labelnya sudah disediakan dan karena bentuknya merupakan nilai binary atau kategorikal, maka metode ML yang digunakan adalah **klasifikasi**.

Lampiran Keterangan Atribut

Atribut	Keterangan
CustomerId	Nomor akun
Surname	Nama belakang nasabah
CreditScore	Nilai kredit
Geography	Negara tempat tinggal nasabah
Gender	Jenis kelamin
Age	Usia nasabah
Tenure	Lamanya menjadi nasabah
Balance	Saldo rekening
NumOfProducts	Jumlah produk yang dibeli nasabah melalui bank
HasCrCard	Kepemilikan kartu kredit
isActiveMember	Keaktifan nasabah
EstimatedSalary	Gaji nasabah
Exited	Keputusan nasabah churn atau tidak

Removing Irrelevant Features (Part 1)

```
df_bank = df_bank.drop(columns = ['RowNumber'])
```

CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

Melakukan **penghapusan fitur row number** untuk mengecek apakah ada data duplikat nantinya.

Removing Duplicates

```
df_bank.duplicated().any()
```

```
False
```

Setelah dilakukan penghapusan fitur row number sebelumnya, **tidak ditemukan adanya data duplikat** pada dataset yang akan kami gunakan.

Removing Irrelevant Features (Part 2)

```
df_bank = df_bank.drop(columns = ['CustomerId', 'Surname'])
```

CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
619	France	Female	42	2	0.00	1	1	1	101348.88	1
608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
502	France	Female	42	8	159660.80	3	1	0	113931.57	1
699	France	Female	39	1	0.00	2	0	0	93826.63	0
850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

Setelah dilakukan pengecekan ada atau tidaknya duplikat pada dataset, kami lakukan **penghapusan fitur customer id dan surname**, karena kedua fitur tersebut **tidak memberikan informasi** yang penting untuk digunakan sebagai model klasifikasi.

Handling Missing Value

```
df_bank.isnull().any()
```

```
CreditScore      False
Geography        False
Gender           False
Age             False
Tenure          False
Balance         False
NumOfProducts   False
HasCrCard       False
IsActiveMember  False
EstimatedSalary False
Exited          False
dtype: bool
```

```
# checking if there is any irrelevant values in categorical features
```

```
print(df_bank.Geography.value_counts())
print(df_bank.Gender.value_counts())
```

```
Geography
France      5014
Germany     2509
Spain       2477
Name: count, dtype: int64

Gender
Male        5457
Female      4543
Name: count, dtype: int64
```

Setelah dilakukan penghapusan fitur sebelumnya, kami lakukan pengecekan apakah fitur yang tersedia memiliki nilai kosong di dalamnya dan setelah dicek **tidak ditemukan adanya nilai kosong** dan **semua nilai pada fitur kategorikal juga relevan** terhadap nama kolomnya.

Feature Encoding

```
cats_updated = ['Geography', 'Gender']

for col in cats_updated:
    print(f'value counts of column {col}')
    print(df_bank[col].value_counts())
    print('---'*10, '\n')
```

```
value counts of column Geography
Geography
France    5014
Germany   2509
Spain     2477
Name: count, dtype: int64
-----
```

```
value counts of column Gender
Gender
1      5457
0      4543
Name: count, dtype: int64
-----
```

Pada dataset kami, terdapat **2 fitur kategorikal yang perlu dikonversi** menjadi numerikal yaitu **fitur geografi dan fitur gender**. Masing-masing fitur tersebut kami tangani dengan pendekatan yang berbeda, **fitur gender kami gunakan label encoding** dikarenakan fitur tersebut hanya memiliki 2 nilai saja (Male & Female) sedangkan pada **fitur geografi kami gunakan one-hot encoding** dikarenakan fitur tersebut memiliki lebih dari 2 nilai (Prancis, Jerman, dan Spanyol) serta fitur tersebut bukan bersifat ordinal (data yang mempunyai tingkatan level).

Feature Encoding

```
# convert gender feature from categorical into numerical by using Label encoding

mapping_gender = {
    'Female' : 0,
    'Male' : 1
}

df_bank['Gender'] = df_bank['Gender'].map(mapping_gender)
```

```
# convert Geography feature from categorical into numerical by using one-hot encoding

from sklearn.preprocessing import OneHotEncoder

## Converting type of columns to category

df_bank['Geography'] = df_bank['Geography'].astype('category')

## Assigning numerical values and storing it in another columns

df_bank['Geo_new'] = df_bank['Geography'].cat.codes

## Create an instance of One-hot-encoder

enc = OneHotEncoder()

## Passing encoded columns

enc_data = pd.DataFrame(enc.fit_transform(
    df_bank[['Geo_new']]).toarray())

## Merge with main

df_bank = df_bank.join(enc_data)

## rename the column

df_bank = df_bank.rename(columns={0 : "is_France", 1 : "is_Germany", 2 : "is_Spain"})

## drop irrelevant column

df_bank = df_bank.drop(columns = ['Geography', 'Geo_new'])

## show the result

df_bank.head(1)
```

	CreditScore	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	is_France	is_Germany	is_Spain
0	619	0	42	2	0.0	1	1	1	101348.88	1	1.0	0.0	0.0

Setelah dilakukan one-hot encoding pada fitur geografi, kami perlu **cek kembali apakah salah satu hasil fitur tersebut memiliki pengaruh signifikan terhadap target serta apakah ada indikasi multikolinearitas** antar fiturnya. Kami gunakan **metode test statistik chi2 dan nilai VIF (Variance Inflation Factor)** untuk digunakan sebagai parameter apakah fitur tersebut bersifat multikolinearitas.

Feature Encoding

```
# Checking the significant of new feature to the target by using chi2 statistic test (categorical vs categorical)
X = df_bank.drop(columns = ['CreditScore', 'Age', 'Tenure', 'NumOfProducts', 'Balance', 'EstimatedSalary', 'Exited'])
y = df_bank['Exited']
print(X.columns)
chi2(X,y)
```

```
array([7.01557451e-13, 6.98496209e-01, 1.56803624e-27, 1.25300579e-13,
       5.81457176e-51, 4.92250487e-06]))
```

Setelah dilakukan uji statistik chi2, **hanya fitur 'HasCrCard' yang tidak memiliki pengaruh signifikan** ($p > 0.05$) terhadap target, sedangkan **hasil ketiga fitur dari proses encoding semuanya memiliki pengaruh signifikan** ($p < 0.05$) terhadap target.

```
### Checking multicollinearity by VIF

from statsmodels.stats.outliers_influence import variance_inflation_factor

vif_select = df_bank
vif_data = pd.DataFrame()
vif_data["feature"] = vif_select.columns

# calculating VIF for each feature
vif_data["VIF"] = [variance_inflation_factor(vif_select.values, i)
                  for i in range(len(vif_select.columns))]

print(vif_data)
```

	feature	VIF
0	CreditScore	1.001643
1	Gender	1.013210
2	Age	1.110478
3	Tenure	1.002156
4	Balance	1.339246
5	NumOfProducts	1.123001
6	HasCrCard	1.001617
7	IsActiveMember	1.046623
8	EstimatedSalary	1.001048
9	Exited	1.177569
10	is_France	41.366649
11	is_Germany	22.804330
12	is_Spain	21.087015

Setelah dilakukan pengecekan nilai VIF, ternyata **hasil ketiga fitur dari proses encoding semuanya memiliki nilai VIF > 5** yang artinya ada **indikasi multikolinearitas**, lakukan penggabungan fitur is_Germany dengan is_Spain menjadi not_France **apabila hasil model evaluasinya mengalami overfitting**.

Feature Selection

(Jika model overfit)

Apabila hasil model evaluasi mengalami overfit, maka akan kami lakukan **feature selection dengan 2 cara**, secara otomatis menggunakan **library SelectKBest** atau **manual dengan menghapus fitur yang tidak berpengaruh signifikan dengan uji chi2 serta fitur yang memiliki nilai VIF tinggi di atas 5**.

Manual - Feature Selection

```
# backing plan for feature selection if the default model is overfit
df_bank2 = df_bank.copy()
df_bank2 = df_bank2.drop(columns = ['HasCrCard', 'Tenure', 'EstimatedSalary', 'is_Spain'])
```

Automatic - Using SelectKBest for Feature Selection if the model is overfit

```
X = df_bank[['CreditScore', 'Age', 'Gender', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard', 'IsActiveMember', 'EstimatedSalary', 'is_France', 'is_Germany', 'is_Spain']]
y = df_bank['Exited']
from sklearn.feature_selection import SelectKBest, mutual_info_classif
X_new = SelectKBest(mutual_info_classif, k=10).fit(X, y)
X_new
```

```
SelectKBest
SelectKBest(score_func=<function mutual_info_classif at 0x000001F731C3FCE0>)
```

```
X_new.get_feature_names_out()
```

```
array(['CreditScore', 'Age', 'Gender', 'Tenure', 'Balance',
       'NumOfProducts', 'IsActiveMember', 'is_France', 'is_Germany',
       'is_Spain'], dtype=object)
```


Feature Engineering + Handling Outlier

(Jika model underfit)

```
## create a copy
df_bank_new = df_bank.copy()

## grouping credit score
bins = [300, 629, 689, 719, 850]
labels = ['Bad Credit', 'Fair Credit', 'Good Credit', 'Excellent Credit']

df_bank_new['CreditScore_Category'] = pd.cut(df_bank_new['CreditScore'], bins=bins, labels=labels, right=False)

## grouping age
bins = [12, 29, 44, 59, 78]
labels = ['Gen Z', 'Millennials', 'Gen X', 'Gen Boomer']

df_bank_new['Age_Category'] = pd.cut(df_bank_new['Age'], bins=bins, labels=labels, right=False)

## grouping numofproduct
df_bank_new['Products_Category'] = np.where(df_bank_new['NumOfProducts'] == 1, 'Single Product', 'Multi Products')
```

Kami memutuskan akan menggunakan hasil dari fitur engineering di atas **apabila hasil model evaluasi mengalami underfitting**. Beberapa fitur engineering yang kami lakukan adalah melakukan **metode binning** dengan cara **mengelompokkan data menjadi kelompok tertentu**, fitur yang diolah adalah **fitur yang memiliki nilai outlier** seperti fitur **skor kredit, usia, dan jumlah produk**.

Handling Outliers

(Untuk data awal modeling)

```
# Split the data into training and testing with the proportion of 70:30

X = df_bank.drop(columns=['Exited'])
y = df_bank[['Exited']]

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

```
# Removing outliers using Z-Score

from scipy import stats

print(f'Jumlah baris sebelum memfilter outlier {len(data_train)}')

for col in ['CreditScore', 'Age']:
    zscore = np.abs(stats.zscore(data_train[col]))
    filtered_entries = (zscore < 3)

data_train = data_train[filtered_entries]

print(f'Jumlah baris setelah memfilter outlier: {len(data_train)}')

Jumlah baris sebelum memfilter outlier 7000
Jumlah baris setelah memfilter outlier: 6906
```

Kami memutuskan untuk menghapus nilai outlier terhadap fitur tertentu dengan **metode Z-score**. Data yang kami gunakan untuk menghapus outlier adalah data training agar tidak terjadi data leaking terhadap data testing. Diperoleh data training setelah dilakukan penghapusan outlier sebanyak 6906 baris (berkurang 1.3%).

Feature Transformation (Scaling)

```
# Standardization

from sklearn.preprocessing import StandardScaler
ss = StandardScaler()

numerical_features = X.columns.to_list()
for n in numerical_features:
    scaler = ss.fit(X_train[[n]])
    X_train[n] = scaler.transform(X_train[[n]])
    X_test[n] = scaler.transform(X_test[[n]])
```

Kami memutuskan untuk melakukan scaling data training dan data testing dengan **metode standarisasi** agar semua fitur yang ada memiliki bentuk distribusi mendekati normal dan jarak nilai min-max antar feature tidak terlalu jauh.

Handling Imbalance Data

(Jika hasil F1 score mengalami overfitting)

```
# using undersampling for majority class with the proportion feature target is 60:40

from imblearn import under_sampling
X_under, y_under = under_sampling.RandomUnderSampler(random_state = 42, sampling_strategy = 0.667).fit_resample(X_train, y_train)


y_under.value_counts()

Exited
0    2169
1    1447
Name: count, dtype: int64
```

Kami memutuskan untuk menggunakan **metode undersampling** untuk handle data yang imbalance dengan **proporsi 70:30**. Namun penggunaan data tersebut kami gunakan sebagai **langkah alternatif terakhir karena pada metrik model evaluasi yang akan kami gunakan adalah metrik F1 Skor yang lebih robust terhadap data imbalance**.

GITHUB REPOSITORY

[Link GitHub](#)


Bank_Churn_Prediction
Public

Watch 1
Fork 0
Star 0

main
1 Branch
0 Tags

Add file
Code



dzulwulann
update readme bussnines recommendation
0bf7a34 · 10 minutes ago
23 Commits

image	update readme bussnines recommendation	10 minutes ago
README.md	update readme bussnines recommendation	10 minutes ago
Stage 0 - Synergies.pdf	add file stage 0	2 days ago

README

Churn Prediction for Bank Customer

Creating a Machine Learning model to predict Customer who has the potential to churn.

Stage 0

Problem Statement

Perusahaan Rakamin Bank Center (RBC) tidak memiliki model Machine Learning (ML) untuk memprediksi nasabah mana yang akan churn. Dari data historikal yang ada, diperoleh jumlah nasabah churn sebesar **20,37%** dari keseluruhan data. Mengacu pada laman [uxpressia.com](#) tentang "*How to Approach Customer Churn Measurements in Banking*", toleransi nasabah churn maksimal sebesar **10%**. Sementara itu, jumlah nasabah churn pada data yang kita miliki melebihi batas toleransi tersebut. Dengan model ML yang dibuat, diharapkan menjadi acuan bagi tim bisnis untuk mengambil langkah strategi mengatasi nasabah yang terdeteksi churn.

Goals

Membuat model Machine Learning dengan tingkat akurasi > **70%** dan tingkat presisi > **0** untuk membantu bank Rakamin Bank Center (RBC) dalam memprediksi nasabah yang akan churn dan membantu tim bisnis dalam menentukan strategi terhadap nasabah yang akan churn

Objectives

- Mengidentifikasi variabel yang memiliki relevansi dengan keputusan nasabah untuk berhenti berlangganan
- Mempersiapkan data historikal yang digunakan untuk model Machine Learning
- Membangun model prediktif untuk mengklasifikasikan nasabah yang berpotensi churn
- Melakukan optimasi model sehingga mendapatkan hasil yang terbaik

About

Predicting churn for bank customer

[Readme](#)
[Activity](#)
0 stars
1 watching
0 forks
[Report repository](#)


Releases


No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Contributors 2


dzulwulann
Dzul Wulan Ningtyas


YusufRo
Burhanuddin Yusuf Robbani