

Navigating Bank Churn: A Comparative Analysis of Decision Tree and SVM Models Using CRISP-DM Methodology

Dwi Yulianto¹, Marcello Roy², Nicholas Soesilo³

¹ Department of Informatics, Universitas Multimedia Nusantara, Tangerang, Banten, Indonesia,

² Department of Informatics, Universitas Multimedia Nusantara, Tangerang, Banten, Indonesia,

³ Department of Informatics, Universitas Multimedia Nusantara, Tangerang, Banten, Indonesia

¹ dwi.yulianto@student.umn.ac.id, ² marcello.roy@student.umn.ac.id, ³ nicholas.soesilo@student.umn.ac.id

Abstract—In the development of the banking industry which continues to grow rapidly, it is important for banks to overcome the level of customer attrition which is usually called bank churn. Because, there are several factors that influence this so they need to be addressed. In this research, machine learning classification techniques are used to create models with a focus on two algorithms, namely Decision Tree and Support Vector Machine. This research uses CRISP-DM as a methodology which consists of 6 stages. The analysis results show that both provide good performance, with Decision Tree achieving an accuracy of around 93.13%, slightly higher compared to 93.08% for SVM. Decision Trees are proven to be more effective in identifying customers who have actually made the switch with high accuracy.

Keywords—Machine Learning, Classification, Decision Tree, Support Vector Machine

I. INTRODUCTION

Dalam perkembangan industri perbankan yang terus tumbuh pesat, mengatasi tingkat atrisi pelanggan, yang umumnya disebut sebagai bank churn, menjadi tantangan yang sangat penting. Sebab, sudah menjadi keharusan bagi pihak bank untuk menjaga loyalitas nasabahnya atas layanan yang diberikannya karena pelanggan dapat berpindah ke kompetitor lain karena berbagai alasan, seperti layanan keuangan yang lebih baik, tingkat suku bunga yang rendah dan lainnya [1]. Di dalam ranah pengambilan keputusan yang berbasis data, machine learning muncul sebagai alat yang sangat berguna untuk menemukan pola-pola rumit dan memprediksi perilaku pelanggan yang pernah berkaitan dengan perusahaan. Penelitian ini bertujuan untuk mengeksplorasi mengenai machine learning, dengan fokus pada penerapan algoritma Decision Tree dan Support Vector Machine (SVM). Dengan menerapkan metodologi CRISP-DM, studi kami bertujuan untuk melakukan analisis perbandingan menyeluruh untuk memahami faktor-faktor yang mempengaruhi bank churn.

Berbagai jenis algoritma machine learning seperti supervised, unsupervised, semi-supervised, dan reinforcement learning hadir dalam bidang data yang kompleks ini. Selain itu, deep learning, yang merupakan bagian dari cakupan lebih luas dari metode machine learning, dapat menganalisis data secara cerdas dalam skala besar [2]. Supervised learning merupakan fokus dalam penelitian ini melibatkan pelatihan algoritma pada data yang berlabel, memungkinkan mesin untuk belajar dan memprediksi hasil berdasarkan data masukan [3]. Sebaliknya, unsupervised learning menangani dataset tanpa

label, mengekstrak struktur bawaan dari data masukan [3]. Ranah reinforcement learning memperkenalkan algoritma yang mampu menilai apakah hasil kesalahan merupakan reward atau pelanggaran [4].

Dalam konteks supervised learning, fokus penelitian ini terletak pada mengatasi permasalahan klasifikasi yang sering dihadapi dalam dunia perbankan. Sasarannya adalah menemukan algoritma terbaik untuk mengenali pola-pola yang mencerminkan vektor, mendekati perilaku pelanggan melalui analisis riwayat data [5]. Ilustrasi visual dibawah ini digunakan untuk menjelaskan proses supervised learning dengan cara yang mudah dipahami.

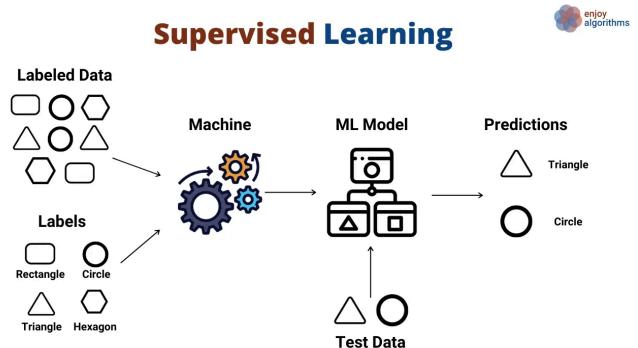


Fig. 1. Supervised Learning Process [5]

Selanjutnya, dalam mengklasifikasikan supervised learning, algoritma dapat dibagi menjadi kelompok regresi dan klasifikasi. Algoritma regresi efektif ketika memprediksi nilai kontinu riil, seperti perkiraan saldo rekening pelanggan, sementara algoritma klasifikasi mampu membedakan kategori, seperti memprediksi apakah seorang pelanggan cenderung churn atau tidak [6]. Contoh algoritma klasifikasi meliputi Support Vector Machine (SVM), Decision Tree, dan lainnya [6].

Di era digital ini, integrasi machine learning ke dalam sektor perbankan menawarkan perubahan paradigma, memfasilitasi strategi proaktif untuk retensi pelanggan. Saat kita menyelami studi ini, tujuan kami adalah melakukan analisis perbandingan yang cermat terhadap algoritma klasifikasi, khususnya Decision Tree dan SVM.

Tidak hanya itu, peran algoritma klasifikasi seperti Decision Tree dan SVM menjadi penting dalam menganalisis fenomena churn di dunia perbankan. Decision Tree menghasilkan model yang dapat dijelaskan secara visual, menggambarkan keputusan dengan jelas, dan membantu mengidentifikasi faktor-faktor yang berpengaruh terhadap kecenderungan pelanggan untuk beralih. Sementara itu, SVM, memiliki kelebihan dalam menangani data yang kompleks dan memiliki dimensi tinggi dijadikan sebagai solusi untuk menganalisis perilaku pelanggan dan memproyeksikan kemungkinan adanya churn.

Penekanan pada interpretasi visual dari Decision Tree dan kemampuan SVM dalam mengeksplorasi kompleksitas data memberikan gambaran yang lebih mudah dipahami mengenai fenomena churn di sektor perbankan. Diharapkan hasil penelitian ini tidak hanya memberikan wawasan yang mendalam tetapi juga mempermudah pemahaman bagi para pemangku kepentingan perbankan, membantu mereka mengambil langkah-langkah yang lebih efektif dalam menjaga loyalitas nasabah mereka.

Penelitian ini akan menggunakan Metodologi CRISP-DM (Cross-Industry Standard Process for Data Mining) karena merupakan standar industri yang umum digunakan dalam melakukan analisis data. CRISP-DM memiliki langkah yang sistematis dan terbukti efektif dalam data mining dari awal hingga akhir. Dengan menggunakan pendekatan ini, diharapkan penelitian ini dapat memberikan pemahaman yang lebih baik terkait fenomena churn di sektor perbankan.

II. LITERATURE REVIEW

A. CRISP-DM

CRISP-DM, atau Cross-Industry Standard Process for Data Mining merupakan salah satu kerangka kerja utama dalam industri data mining [7]. Framework ini dikembangkan untuk memberikan panduan terstruktur dalam menjalankan proyek data mining, dalam CRISP-DM terdapat enam tahap utama yang mencakup business understanding, data understanding, data preparation, modelling, evaluation, dan deployment [7]. Keenam tahap tersebut bisa divisualisasikan pada gambar dibawah ini,

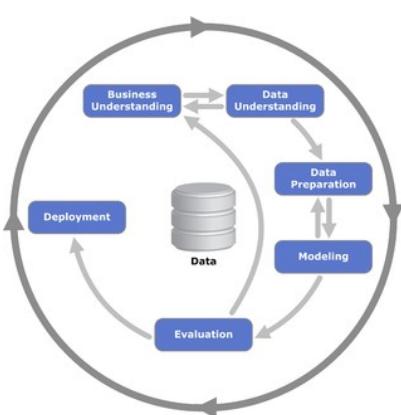


Fig. 1. Supervised Learning Process [8]

Meskipun telah ada beberapa pengembangan baru dalam dunia data mining, CRISP-DM tetap menjadi standar yang diakui dan umum diterapkan di berbagai sektor.

Untuk mengetahui sumber daya yang dibutuhkan dan dapat diakses, lingkungan bisnis harus dievaluasi. Salah satu aspek terpenting dari fase ini adalah memilih tujuan data mining. Pertama, jenis data mining (seperti klasifikasi) dan kriteria keberhasilan data mining harus dijelaskan (seperti presisi). Pembuatan rencana proyek diperlukan.“.

Berdasarkan penjelasan di atas, pada fase ini yang terpenting adalah membuat tujuan data mining, membuat rencana bagaimana mencapai tujuan tersebut, memilih jenis data mining, dan hasil yang diharapkan dari proyek tersebut.

Fase kedua adalah fase Pemahaman Data. Pada fase ini, sangat penting untuk mengumpulkan data dari sumber data, mengeksplorasi dan mendeskripsikannya, serta menilai kualitas data [9]. Agar data lebih mudah dipahami, panduan pengguna menguraikan deskripsi data menggunakan analisis statistik dan membuat atribut serta penyusunannya .

Tahap ketiga adalah Persiapan Data. Menetapkan kriteria inklusi dan eksklusi harus menjadi langkah pertama dalam pemilihan data [9]. Pembersihan data dapat digunakan untuk mengatasi kualitas data yang buruk [10]. Karakteristik yang diturunkan harus dibangun berdasarkan model yang digunakan (ditentukan pada tahap pertama) [9]. Pendekatan yang berbeda dapat dilakukan untuk setiap langkah ini dan bergantung pada model [11].

Fase keempat adalah Pemodelan. Dalam tahap pemodelan, kita akan mengembangkan kasus uji, model, dan pendekatan pemodelan semuanya merupakan bagian dari langkah pemodelan data. Anda dapat menggunakan teknik penambangan data apa pun. Parameter tertentu harus ditetapkan untuk membangun model. Adalah tepat untuk membandingkan model dengan kriteria evaluasi dan memilih yang terbaik untuk penilaian [12].

Kelima adalah tahap Evaluasi. Temuan yang ditemukan pada fase ini akan dibandingkan dengan tujuan perusahaan yang telah ditetapkan. Akibatnya, perlu untuk menentukan tindakan tambahan dan mengevaluasi hasilnya .

Fase terakhir adalah Deployment. Tahap penerapan terdiri dari perencanaan penerapan, pemantauan, dan pemeliharaan [13]. Hasil akhirnya dapat berupa laporan akhir atau komponen perangkat lunak [13].

B. Machine Learning

Machine Learning (ML) telah menjadi bidang transformasional dalam ranah kecerdasan buatan, menunjukkan kemampuan yang luar biasa untuk memungkinkan sistem belajar dan meningkat dari pengalaman tanpa pemrograman eksplisit [14]. Salah satu paradigma dasar dalam ML adalah supervised learning, di mana algoritma dilatih dengan dataset berlabel untuk membuat prediksi atau keputusan [15]. Efektivitas algoritma ML tergambar dengan jelas dalam berbagai domain, seperti pengenalan gambar, pemrosesan bahasa alami, dan diagnostik kesehatan.

C. Decision Tree

Decision Tree adalah sebuah pohon yang setiap simpulnya mewakili suatu atribut, setiap tautan mewakili

suatu aturan, dan setiap daun mewakili hasilnya (nilai kategorikal atau kontinu) [16]. Sangat mudah untuk mengumpulkan data dan menghasilkan beberapa kesimpulan yang mendalam karena pohon keputusan sangat mirip dengan cara berpikir manusia.

Decision sendiri terdiri dari sembilan jenis yang berbeda. Jenis pohon keputusan adalah Iterative Dichotomies 3 (ID3), Successor of ID3 (C4.5), Classification and Regression Tree (CART), CHI-Squared Automatic Detector (CHAID), Multivariate Adaptive Regression Splines (MARS), Generalized, Unbiased, Interaction Detection Estimation (GUIDE), Conditional Inference Trees (CTREE), Classification Rule dengan Unbiased Interaction, Selection and Estimation (CRUISE), dan Quick, Unbiased, and Efficient Statistical Tree (QUEST). Penelitian ini akan menggunakan pohon keputusan C4.5 karena metode yang akan digunakan pada python adalah entropy.

Entropi berarti mengukur ketidakmurnian atau keacakan suatu dataset dan nilai entropi selalu berada di antara 0 dan 1 [17]. Nilai entropi dapat dikategorikan baik jika nilainya mendekati 0. Rumus Entropi dapat dilihat pada persamaan 1 dibawah ini.

$$\text{Entropy}(S) = \sum_{i=1}^c P_i \log_2 P_i$$

Equation 1. Decision Tree

Penjelasan Persamaan satu adalah:

- *P* adalah nilai nomor sampel subset [17].
- *I* adalah nilai atribut ke-n [17].

D. Support Vector Machine (SVM)

Support vector machine atau disebut juga SVM merupakan salah satu algoritma klasifikasi pembelajaran mesin yang diawasi. Vapnik, Noser, dan Guyon menerbitkan model SVM pertama pada tahun 1992, dan SVM telah berkembang secara signifikan sejak saat itu [18].

Rumus SVM dapat dilihat pada Persamaan 2 di bawah ini [18].

$$g(x) = w^T x + b$$

Equation 2. SVM Formula

Dari rumus diatas berikut penjelasannya.

- *x* adalah vektor masukan [18].
- *w* adalah parameter bobot [18].
- *b* adalah biasnya [18].

E. Penelitian Terdahulu

Dalam studi “Prediksi Customer Churn Bank dengan Metode Machine Learning” oleh He Zhu, fokusnya adalah pada prediksi customer churn di sektor perbankan menggunakan teknik machine learning. Studi ini menggunakan data dari Bank ABC untuk menganalisis faktor-faktor penentu churn pelanggan dan membandingkan efektivitas algoritma regresi logistik dan hutan acak. He Zhu

menemukan bahwa *random forest* memiliki kinerja yang lebih baik daripada regresi logistik, yang selaras dengan literatur yang ada. Penelitian ini menyoroti usia pelanggan sebagai faktor paling berpengaruh dalam memprediksi churn, sementara kepemilikan kartu kredit memiliki korelasi paling lemah. Studi ini memberikan wawasan berharga dalam memahami dan memprediksi perilaku nasabah di industri perbankan, serta menyoroti potensi metode pembelajaran mesin untuk meningkatkan analisis keuangan [19].

Penelitian lain yang relevan adalah studi berjudul “A Hybrid Machine Learning Model for Bank Customer Churn Prediction”. Studi ini mengembangkan model hibrida *machine learning* untuk memprediksi churn pelanggan bank. Model ini menggabungkan single learners dan ensemble learners untuk meningkatkan kinerja prediktif. Setelah menyesuaikan parameter, model SVM dan Adaboost dipilih dan digabungkan menjadi model hibrida. Model SVM-Adaboost menunjukkan efisiensi sebesar 87%, memberikan keputusan yang terinformasi untuk strategi retensi pelanggan [20].

III. METHODOLOGY

Berdasarkan literatur review sebelumnya, penelitian ini menggunakan Framework CRISP-DM dalam melakukan data mining dan menggunakan bahasa pemrograman python dalam mengaplikasikan semua algoritma yang akan digunakan.

Penelitian ini dimulai dengan tahap pertama dari CRISP-DM, yaitu business understanding.

A. Business Understanding

Implementasi machine learning dengan teknik CRISP-DM ini berhubungan langsung dengan dataset bank. Tujuan dari data mining adalah untuk mendapatkan prediksi akurat apakah nasabah akan churn atau bertahan berdasarkan data historis yang ada pada suatu bank. Jenis data mining yang akan digunakan pada penelitian mengenai bank customer churn analysis ini adalah data mining klasifikasi, lebih tepatnya membandingkan algoritma Decision Tree dan Support Vector Machine. Hal ini akan bermanfaat bagi bisnis karena bank dapat mengambil tindakan pencegahan atau memberi penawaran khusus untuk menjaga loyalitas nasabah.

B. Data Understanding

Data yang digunakan dalam penelitian ini adalah data sekunder. Dataset mengenai data credit bank ini peroleh dari platform kaggle sebagai platform open-source penyedia dataset. Sumber data bisa ditemukan dalam lampiran pada artikel penelitian ini. Dataset tersebut mengandung 20071 rows and 21 columns pada sheet pertama yang berisi mengenai data nasabah serta 306876 rows and 8 columns pada sheet kedua yang berisi mengenai data transaksi nasabah ketika di bank. Kolom yang terdapat di dataset setelah dilakukan pemilihan kolom yang berkaitan dengan tujuan penelitian ini adalah sebagai berikut ;

TABLE I. TABEL DAFTAR KOLOM

Name of Column	Means
CLIENTNUM	Nomor identifikasi unik untuk setiap pemegang kartu.
Date	Tanggal tertentu terkait dengan data yang direkam.
Type	Jenis transaksi atau jenis data yang direpresentasikan.
Target Revenue	Pendapatan yang ditargetkan terkait dengan kartu kredit.
Attrition_Flag	Status apakah pemegang kartu telah berhenti menggunakan kartu kredit atau tidak.
Customer_Age	Usia pemegang kartu.
Gender	Jenis kelamin pemegang kartu.
Dependent_count	Jumlah orang yang bergantung pada pemegang kartu.
Education_Level	Tingkat pendidikan dari pemegang kartu.
Marital_Status	Status perkawinan pemegang kartu.
Income_Category	Kategori pendapatan dari pemegang kartu.
Months_on_book	Jumlah bulan kartu telah berada dalam kepemilikan pemegang kartu.
Total_Relationship_Count	Jumlah hubungan yang dimiliki pemegang kartu dengan institusi keuangan.
Months_Inactive_12_mon	Jumlah bulan tidak aktif dalam setahun.
Contacts_Count_12_mon	Jumlah kontak yang dilakukan dalam setahun.
Credit Limit	Batas kredit yang diberikan.
Total_Revolving_Balance	Total saldo yang berputar (hutang yang belum dibayar) pada kartu kredit.
Avg_Open_To_Buy	Rata-rata jumlah kredit yang masih tersedia untuk digunakan.
Total_Trans_Ct	Jumlah total transaksi yang dilakukan.
Avg_Utilization_Ratio	Rasio penggunaan rata-rata dari batas kredit yang diberikan.
Quarter	Kuartal waktu di mana data dikumpulkan atau direkam.
Date_Leave	Tanggal saat Customer Churn

Setelah mengetahui daftar kolom dan maksud dari setiap kolom diatas , kita perlu untuk melakukan beberapa import library python yang mendukung proses analisis ini.

```
import pandas as pd
import numpy as np
import matplotlib as plt
import seaborn as sns
from sklearn.inspection import permutation_importance
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import cross_val_score, confusion_matrix, classification_report
from sklearn.metrics import accuracy_score, precision_recall_fscore_support
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, export_text, plot_tree
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, export_graphviz
import graphviz
from sklearn.cluster import KMeans
from sklearn import metrics
from sklearn.preprocessing import LabelEncoder
```

Dataset yang sudah didapat dari sumber bisa di import ke Python dengan menggunakan bantuan library Python. Data yang telah di import disimpan menjadi variable di Python. Untuk dataset mengenai Nasabah diberi nama “credit_info” dan data mengenai transaksi nasabah diberi nama “credit_finace”.

credit_info											
CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Months_on_book
0	712672083	Existing Customer	65	F	0	High School	Married	Less than \$40K	Blue		
1	713049933	Existing Customer	47	F	1	Graduate	Married	40K–60K	Silver		
2	713049933	Existing Customer	48	F	1	Graduate	Married	40K–60K	Silver		
3	713135883	Existing Customer	65	F	0	College	Married	Less than \$40K	Blue		
4	713135883	Existing Customer	64	F	0	College	Married	Less than \$40K	Blue		
...
20066	900203139	Existing Customer	24	F	0	College	Single	Less than \$40K	Silver		
20067	900203141	Existing Customer	24	F	1	College	Divorced	Less than \$40K	Silver		
20068	900203144	Existing Customer	24	M	2	College	Married	60K–80K	Blue		
20069	900203145	Existing Customer	24	M	0	College	Single	40K–60K	Blue		
20070	900203146	Existing Customer	24	M	1	College	Single	Less than \$40K	Blue		

20071 rows x 21 columns

Fig. 3. Dataset pertama “credit_info”

Inilah bagian penting dari fase Pemahaman Data. Pada fase ini, kita perlu mendapatkan wawasan sebanyak mungkin dari kumpulan data. Untuk melakukannya, pertama-tama kita perlu menghapus kolom yang tidak perlu yang ada di dataset dan kemudian mendapatkan informasi tentang tipe data apa yang terdapat dalam dataset tersebut, mendapatkan bentuk dari dataset tersebut, apa statistik dasar dari data tersebut, bentuk dari dataset tersebut. dataset, dan yang terpenting adalah visualisasi dari dataset tersebut sehingga kita dapat memiliki gambaran besar dari dataset tersebut. Berikut proses Analisis Data Eksplorasi pada Fase Pemahaman Data CRISP-DM dalam penelitian ini. Prosesnya dilakukan dengan Python.

Credit_Info

```
credit_info = pd.read_excel('credit_2018_2019.xlsx', sheet_name='info_all')
credit_finance = pd.read_excel('credit_2018_2019.xlsx', sheet_name='Finance_all')
```

Fig. 3. Import dataset

Pada gambar diatas dilakukan import data dari file excel yang bernama “credit_2018_2019.xlsx” dan disesuaikan dengan halaman pada excel tersebut.

credit_info.describe()

	CLIENTNUM	Customer_Age	Dependent_count	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mon	Contacts_Count_12_mon	Credit_Limit
count	2.007100e+04	20071.000000	20071.000000	20071.000000	20071.000000	20071.000000	20071.000000	20071.00
mean	7.510799e+08	46.331872	1.954810	41.039510	3.816701	2.665388	2.449106	8637.12
std	5.472039e+07	8.389822	1.201685	10.392045	1.553397	1.612679	1.104183	9084.35
min	7.080821e+08	24.000000	0.000000	13.000000	1.000000	0.000000	0.000000	1400.0C
25%	7.134850e+08	41.000000	1.000000	35.000000	3.000000	1.600000	2.000000	2548.5C
50%	7.188436e+08	47.000000	2.000000	41.000000	4.000000	3.000000	2.000000	4532.0C
75%	7.811404e+08	52.000000	3.000000	48.000000	5.000000	3.000000	3.000000	11062.0C
max	9.002031e+08	65.000000	5.000000	68.000000	6.000000	6.000000	6.000000	35000.0C

Fig. 4. Basic statistics of the credit_info

Disini dilakukan untuk mengecek deskripsi data yang dimana merupakan statistik dasar dari suatu data frame yang diberi nama “credit_info”. Terlihat bahwa terdapat beberapa komponen seperti Count, Mean, Std, Minimal nilai, Kuartil 1, Kuartil 2, Kuartil 3, dan maksimal nilai.

```

credit_info.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20071 entries, 0 to 20070
Data columns (total 21 columns):
 #   Column           Non-Null Count Dtype  
--- 
 0   CLIENTNUM        20071 non-null  int64  
 1   Attrition_Flag   20071 non-null  object  
 2   Customer_Age     20071 non-null  int64  
 3   Gender            20071 non-null  object  
 4   Dependent_count   20071 non-null  int64  
 5   Education_Level  20071 non-null  object  
 6   Marital_Status    20071 non-null  object  
 7   Income_Category   20071 non-null  object  
 8   Card_Category     20071 non-null  object  
 9   Months_on_book   20071 non-null  int64  
 10  Total_Relationship_Count 20071 non-null  int64  
 11  Months_Inactive_12_mon 20071 non-null  int64  
 12  Contacts_Count_12_mon 20071 non-null  int64  
 13  Credit_Limit      20071 non-null  float64 
 14  Total_Revolving_Bal 20071 non-null  int64  
 15  Avg_Open_To_Buy   20071 non-null  float64 
 16  Total_Trans_Ct    20071 non-null  int64  
 17  Avg_Utilization_Ratio 20071 non-null  float64 
 18  Quarter           20071 non-null  object  
 19  Year              20071 non-null  int64  
 20  Date_Leave        20071 non-null  object  
dtypes: float64(3), int64(10), object(8)
memory usage: 3.2+ MB

```

Fig. 5. Data Information of the credit_info

Pada gambar 5 diatas dilakukan pengecekan informasi mengenai data frame credit_info per kolomnya. Hasilnya adalah ditampilkan berupa berapa banyak datanya dan juga tipe datanya.

```

missing_values = credit_info.isnull().sum()
percentage_missing = (missing_values / len(credit_info)) * 100
missing_data = pd.DataFrame({'Missing Values': missing_values, 'Percentage': percentage_missing})
print(missing_data)

Missing Values Percentage
CLIENTNUM          0       0.0
Attrition_Flag     0       0.0
Customer_Age       0       0.0
Gender             0       0.0
Dependent_count    0       0.0
Education_Level    0       0.0
Marital_Status     0       0.0
Income_Category    0       0.0
Card_Category      0       0.0
Months_on_book     0       0.0
Total_Relationship_Count 0       0.0
Months_Inactive_12_mon 0       0.0
Contacts_Count_12_mon 0       0.0
Credit_Limit       0       0.0
Total_Revolving_Bal 0       0.0
Avg_Open_To_Buy   0       0.0
Total_Trans_Ct    0       0.0
Avg_Utilization_Ratio 0       0.0
Quarter           0       0.0
Year              0       0.0
Date_Leave        0       0.0

```

Fig. 6. Check Missing Value of the credit_info

Pada gambar 6 menunjukkan hasil dari pengecekan missing value pada dataframe credit_info. Terlihat bahwa data mentah yang berasal dari sumber data sudah bersih dari missing value. Hal itu menggambarkan bahwa data yang berasal dari sumber ini merupakan data yang berkualitas. Oleh karena itu dalam kasus ini tidak dilakukan penghapusan missing value.

```

unique_counts = credit_info.nunique()
unique_counts

CLIENTNUM           11571
Attrition_Flag      2
Customer_Age         42
Gender              2
Dependent_count     6
Education_Level      6
Marital_Status       3
Income_Category      5
Card_Category        4
Months_on_book       56
Total_Relationship_Count 6
Months_Inactive_12_mon 7
Contacts_Count_12_mon 7
Credit_Limit          6219
Total_Revolving_Bal  2232
Avg_Open_To_Buy       6814
Total_Trans_Ct         126
Avg_Utilization_Ratio 965
Quarter              5
Year                  2
Date_Leave             10
dtype: int64

```

Fig. 7. Check Unique Value of the credit_info

Pada gambar diatas dilakukan teknik untuk melakukan cek nilai unik pada dataframe. Hal ini ditujukan untuk mengetahui apakah ada data duplikat pada CLIENTNUM dan ternyata ditemukan plagiat. Sebab saat dataframe dibaca jumlah row adalah 20071 dan saat di cek nilai unik menunjukkan 11571. Hal ini karena beberapa nasabah memiliki beberapa record di bank tersebut.

```

sns.countplot(x='Attrition_Flag', data=credit_info)
plt.title('Distribution of Attrition_Flag')
plt.show()

```

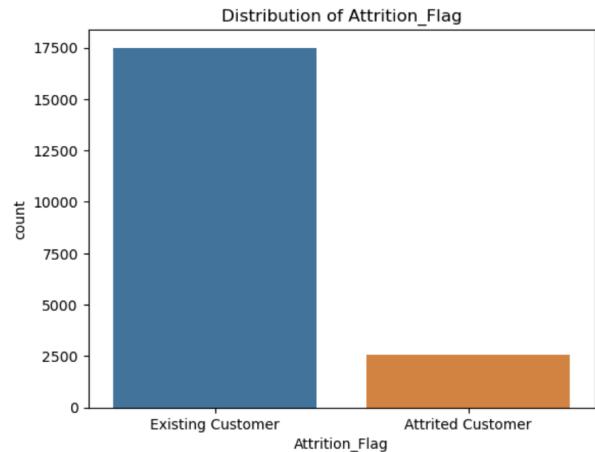


Fig. 7. Distribusi Status Nasabah

Gambar 7 diatas menunjukkan distribusi menurut status nasabah di bank tersebut. Visualisasi diatas menunjukkan bahwa sekitar 17500 an nasabah masih bertahan dan sekitar 2500 an nasabah sudah tidak lagi berhubungan dengan produk bank. Hal ini menunjukkan bahwa persentase pelanggan yang churn relatif kecil namun perlu dikurangi.

```

churn_data = credit_info[credit_info['Attrition_Flag'] == 'Attrited Customer']
# Membuat bar chart untuk distribusi Card_Catagory
plt.figure(figsize=(8, 6))
ax = sns.countplot(x='Card_Catagory', data=churn_data, palette='viridis')

# Menambahkan keterangan jumlah per kategori kartu
for p in ax.patches:
    ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2., p.get_height()), ha='center', va='center', xytext=(0, 10), textcoords='offset points')

plt.title('Distribution of Card Category')
plt.xlabel('Card Category')
plt.ylabel('Count')
plt.show()

```

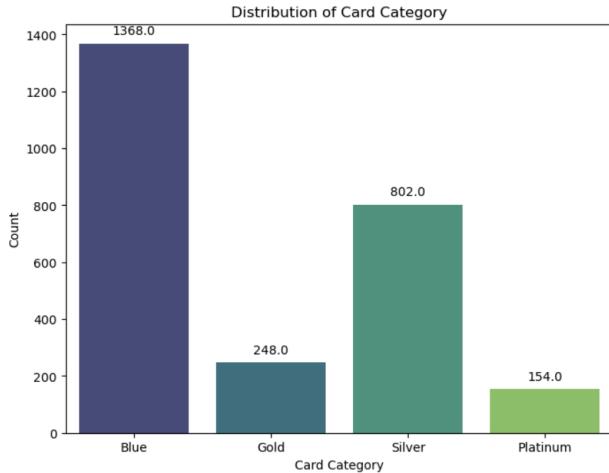


Fig. 8. Distribusi Kategori Kartu Nasabah

Gambar 8 diatas menunjukkan distribusi persebaran kartu nasabah yang terdapat di bank. Visualisasi menunjukkan bahwa nasabah dominan memegang kartu blue dan yang paling sedikit adalah nasabah yang menggunakan kartu platinum

```

# Menghitung total transaksi per jenis Card_Catagory
total_transactions = credit_info.groupby('Card_Catagory')['Total_Trans_Ct'].sum().reset_index()

# Membuat bar chart untuk total transaksi per jenis Card_Catagory
plt.figure(figsize=(10, 6))
ax = sns.barplot(x='Card_Catagory', y='Total_Trans_Ct', data=total_transactions, palette='coolwarm')

# Menambahkan label keterangan jumlah di atas setiap batang
for p in ax.patches:
    ax.annotate(f'{int(p.get_height())}', (p.get_x() + p.get_width() / 2., p.get_height()), ha='center', va='center', xytext=(0, 10), textcoords='offset points')

plt.title('Total Transactions per Card Category')
plt.xlabel('Card Category')
plt.ylabel('Total Transactions')
plt.show()

```

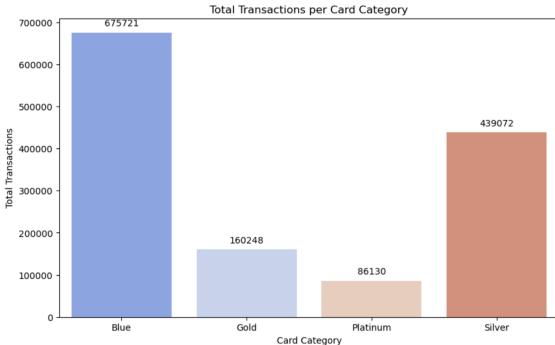


Fig. 9. Total Transaksi per Kartu Nasabah

Pada gambar 9 ditunjukkan distribusi total transaksi yang berdasarkan pada kartu yang dimiliki oleh nasabah. Dari gambar tersebut diketahui bahwa total transaksi tertinggi adalah kartu dengan kategori blue sedangkan total transaksi yang terendah adalah kartu dengan kategori platinum.

```

plt.figure(figsize=(10, 6))
sns.countplot(x='Income_Catagory', data=credit_info, palette='muted')
plt.title('Distribution of Income Category')
plt.xlabel('Income Category')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()

```

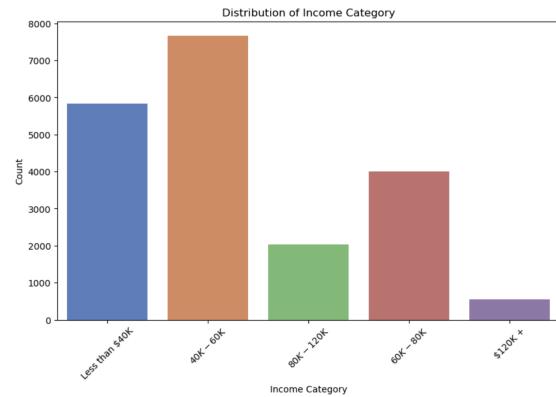


Fig. 10. Distribusi Kategori Pendapatan Nasabah

Pada gambar 10 ditunjukkan distribusi kategori pendapatan nasabah yang ada di bank. Dominan dari nasabah masuk di kategori pendapatan \$40K - 60K dan yang minoritas adalah nasabah dengan pendapatan lebih dari \$120K. Terlihat bahwa data ini sudah ter-binning dari sumber datanya oleh karena itu pada proses data preprocessing nanti tidak dilakukan binning pada kategori income nasabah.

```

plt.figure(figsize=(10, 8))
sns.boxplot(x='Attrition_Flag', y='Customer_Age', data=credit_info, hue='Income_Catagory', palette='pastel')
plt.title('Comparison of Customer Income Distribution by Churn')
plt.xlabel('Churn')
plt.ylabel('Customer Age')
plt.show()

```

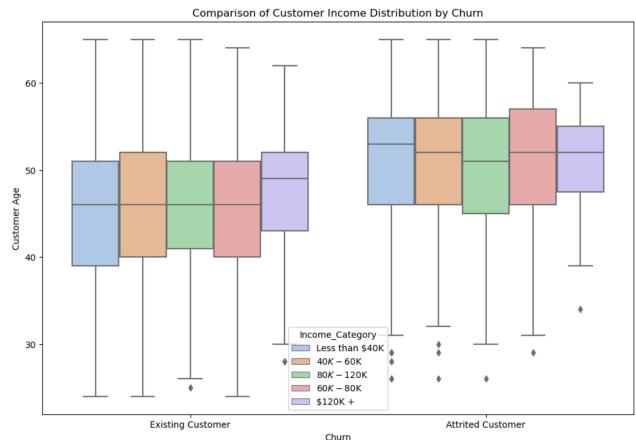


Fig. 11. Perbandingan Kategori Pendapatan Nasabah Berdasarkan Status dan Usia

Gambar 11 menunjukkan bahwa rata-rata nasabah yang masih bertahan pada layanan bank adalah dengan usia dibawah 50 karena mungkin pada usia ini seseorang masih berada di fase produktif. Sedangkan gambar juga menunjukkan bahwa nasabah yang churn kebanyakan adalah diatas 50 yang kemungkinan sudah tidak produktif.

```
# Filter data untuk pelanggan yang churn
churn_data = credit_info[credit_info['Attrition_Flag'] == 'Attrited Customer']

# Bar chart untuk Marital Status
plt.figure(figsize=(14, 7))
plt.subplot(1, 2, 1)
sns.countplot('Marital_Status', data=credit_info, hue='Attrition_Flag', palette='pastel')
plt.title('Marital Status Distribution')

# Bar chart untuk Education Level
plt.subplot(1, 2, 2)
sns.countplot('Education_Level', data=credit_info, hue='Attrition_Flag', palette='pastel')
plt.title('Education Level Distribution')

plt.tight_layout()
plt.show()
```

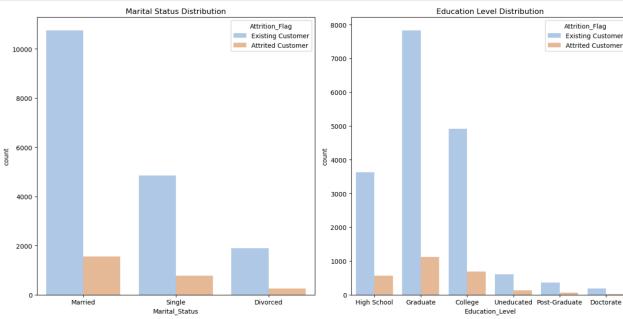


Fig. 12. Distribusi Nasabah Berdasarkan Status Pernikahan dan Tingkat Pendidikan

Pada gambar 12 ditunjukkan bahwa yang paling banyak bertahan adalah nasabah yang sudah menikah selain itu nasabah yang paling banyak bertahan di bank adalah nasabah dengan tingkat pendidikan lulus.

```
churn_data = credit_info[credit_info['Attrition_Flag'] == 'Attrited Customer']
existing_data = credit_info[credit_info['Attrition_Flag'] == 'Existing Customer']

# Pie chart untuk Gender Distribution pada pelanggan yang Churn
plt.figure(figsize=(2, 6))
plt.subplot(1, 2, 1)
churn_gender_distribution = churn_data['Gender'].value_counts()
plt.pie(churn_gender_distribution, labels=churn_gender_distribution.index,
        autopct='%1.1f%%', colors=['lightcoral', 'skyblue'])
plt.title('Churned Customers - Gender Distribution')

# Pie chart untuk Gender Distribution pada pelanggan yang Existing
plt.subplot(1, 2, 2)
existing_gender_distribution = existing_data['Gender'].value_counts()
plt.pie(existing_gender_distribution, labels=existing_gender_distribution.index,
        autopct='%1.1f%%', colors=['lightcoral', 'skyblue'])
plt.title('Existing Customers - Gender Distribution')

plt.show()
```



Fig. 13. Komposisi Nasabah Berdasarkan Gender

Berdasarkan gambar 13 terlihat bahwa nasabah yang bertahan maupun yang sudah meninggalkan layanan bank adalah dengan jenis kelamin perempuan.

Credit_Finance

credit_finance							
CLIENTNUM	Attrition_Flag	Year	Quarter	Date	Type	Trans_Amount	Revenue
0	708082083	Existing Customer	2018	Q1	Q1,2018	Shop	120.1200
1	708083283	Attrited Customer	2018	Q1	Q1,2018	Shop	135.5550
2	708084558	Attrited Customer	2018	Q1	Q1,2018	Shop	396.2700
3	708085458	Existing Customer	2018	Q1	Q1,2018	Shop	122.9550
4	708086958	Existing Customer	2018	Q1	Q1,2018	Shop	134.6400
...
306871	721164483	Existing Customer	2019	Q4	Q4,2019	Cash	56.5440
306872	708085133	Existing Customer	2019	Q4	Q4,2019	Cash	238.9592
306873	900202780	Existing Customer	2019	Q4	Q4,2019	Cash	13.6496
306874	779770683	Existing Customer	2019	Q4	Q4,2019	Cash	60.1008
306875	709632483	Existing Customer	2019	Q4	Q4,2019	Cash	2.0672

306876 rows × 8 columns

Fig. 14. Dataframe Credit_Finance

Pada gambar 14 ditunjukkan isi dari data frame Credi_Finance yang dimana merupakan dataframe yang mengandung record data mengenai transaksi nasabah di bank.

credit_finance.describe()				
	CLIENTNUM	Year	Trans_Amount	Revenue
count	3.068760e+05	306876.000000	306876.000000	306876.000000
mean	7.516599e+08	2018.499381	239.960680	6.815908
std	5.533922e+07	0.500000	336.611644	8.825891
min	7.080821e+08	2018.000000	0.000000	0.000000
25%	7.135329e+08	2018.000000	50.490000	1.688865
50%	7.189092e+08	2018.000000	125.467500	3.663747
75%	7.833819e+08	2019.000000	282.398525	8.652000
max	9.002031e+08	2019.000000	5373.900000	150.202500

Fig. 15. Statistik Dasar Dataframe Credit_Finance

Pada gambar 15 ditunjukkan statistik dasar mengenai data frame Credit_Finance yang dimana terdiri atas berapa banyak baris, rata-rata, std, nilai minimal, kuartal 1, kuartal 2, kuartal 3 dan nilai maksimal.

credit_finance.info()				
	Column	Non-Null Count	Dtype	
0	CLIENTNUM	306876	non-null	int64
1	Attrition_Flag	306876	non-null	object
2	Year	306876	non-null	int64
3	Quarter	306876	non-null	object
4	Date	306876	non-null	object
5	Type	306876	non-null	object
6	Trans_Amount	306876	non-null	float64
7	Revenue	306876	non-null	float64
	dtypes:	float64(2), int64(2), object(4)		
	memory usage:	18.7+ MB		

Fig. 16.Informasi Dataframe Credit_Finance

Pada gambar 16 ditunjukkan bahwa informasi mengenai DataFrame tersebut yang dimana terdiri dari jumlah baris dan juga tipe data per kolomnya.

```

missing_values = credit_finance.isnull().sum()
percentage_missing = (missing_values / len(credit_finance)) * 100
missing_data = pd.DataFrame({'Missing Values': missing_values, 'Percentage': percentage_missing})
print(missing_data)

      Missing Values Percentage
CLIENTNUM          0       0.0
Attrition_Flag     0       0.0
Year               0       0.0
Quarter            0       0.0
Date               0       0.0
Type               0       0.0
Trans_Amount        0       0.0
Revenue            0       0.0

```

Fig. 17. Check Missing Value Dataframe Credit_Finance

Pada gambar 17 diatas dilakukan check terhadap missing value dan hasil menunjukkan bahwa tidak ada satupun missing value di dataframe ini, yang menunjukkan bahwa data ini sudah bagus sehingga tidak diperlukan penghapusan missing value.

```

#cek nilai unik di dataframe
unique_counts = credit_finance.nunique()
print(unique_counts)

CLIENTNUM      11571
Attrition_Flag 2
Year           2
Quarter         4
Date            8
Type            4
Trans_Amount    125931
Revenue         133015
dtype: int64

```

Fig. 18. Check Unique Value Dataframe Credit_Finance

Pada gambar ditunjukkan banyaknya nilai unik di setiap kolom dataframe. Hasil menunjukkan bahwa CLIENTNUM memiliki 11571 unique value namun pada data record banyaknya baris kolomnya adalah 306876 hal ini menunjukkan adanya transaksi yang melebihi satu kali. Sehingga nanti diperlukan memfilter data menggunakan group by.

```

# scatter plot untuk korelasi antara Pendapatan dan Jumlah Transaksi
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Revenue', y='Trans_Amount', data=credit_finance, palette='coolwarm')
plt.title('Correlation between Revenue and Trans Amount')
plt.xlabel('Revenue')
plt.ylabel('Trans_Amount')
plt.legend(title='Attrition Flag', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()

/usr/folders/lv/kv66r7m0b2z1j7vt1zb0d400000gp/T/ipykernel_2458/2036832622.py:2: UserWarning: Ignoring 'palette' because no 'hue' variable has been assigned.
  sns.scatterplot(x='Revenue', y='Trans_Amount', data=credit_finance, palette='coolwarm')
No artists with labels found to put in legend. Note that artists whose label start with an underscore are ignored when legend() is called with no argument.

```

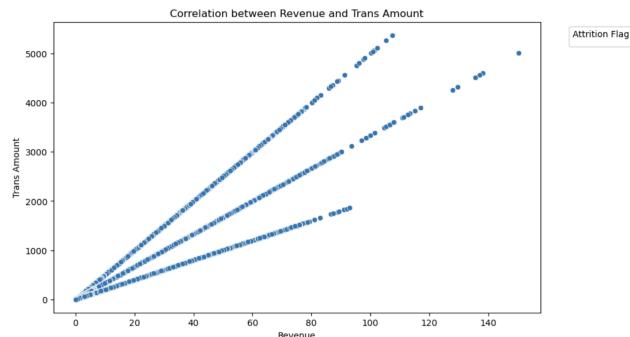


Fig. 19. Korelasi Antara Trans_Amount dengan Revenue

Pada gambar 19 diatas ditunjukkan visualisasi korelasi antara jumlah transaksi nasabah dengan keuntungan yang didapatkan oleh bank. Dari gambar tersebut bisa dilihat adanya korelasi positif dan terbagi menjadi 3 yang kemungkinan ini tergantung dengan tipe kartu ataupun tipe pembelian.

```

plt.figure(figsize=(12, 6))
sns.barplot(x='Type', y='Trans_Amount', data=credit_finance, ci=None)
plt.title('Transaction Amount by Transaction Type')
plt.xlabel('Transaction Type')
plt.ylabel('Amount')
plt.show()

```

/var/folders/lv/kv66r7m0b2z1j7vt1zb0d400000gp/T/ipykernel_2458/2036832622.py:2: FutureWarning:

The 'ci' parameter is deprecated. Use 'errorbar=None' for the same effect.

sns.barplot(x='Type', y='Trans_Amount', data=credit_finance, ci=None)

Transaction Amount by Transaction Type

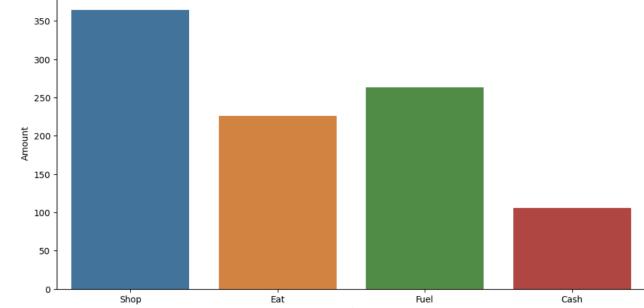


Fig. 20. Distribusi Transaksi Berdasarkan Tipe Pembelian

Gambar 20 menunjukkan distribusi transaksi nasabah yang dikategorikan berdasarkan tipe pembelian dari transaksi itu sendiri. Transaksi dengan jumlah terbanyak adalah transaksi berjenis shop.

C. Data Preparation

Disini akan dilakukan data preparation untuk mempersiapkan data yang baik agar supaya modeling yang dihasilkan nantinya bagus dan akurat

```

credit_info['Customer_Age'] = pd.to_numeric(credit_info['Customer_Age'], errors='coerce')
credit_info_sorted = credit_info.sort_values(by='Customer_Age', ascending=False)
latest_records = credit_info_sorted.drop_duplicates(subset='CLIENTNUM', keep='first')

```

Fig. 21. Penghapusan Data Duplikat

Karena terdapat duplikat di CLIENTNUM yang disebabkan adanya record data dengan CLIENTNUM sama namun dengan tahun yang berbeda mengingat adanya beberapa record dengan tahun yang beda di data maka akan dihapus duplikat dan diambil tahun terakhir record yang ada berdasarkan Customer_Age karena akan menunjukkan data terakhir dicatat.

```

# Group by CLIENTNUM and calculate the sum for specific columns
sum_trans_amount = credit_info.groupby('CLIENTNUM')[['Months_Inactive_12_mon', 'Contacts_Count_12_mon',
                                                       'Total_Revolving_Bal', 'Total_Trans_Ct']].sum().reset_index()
# Group by CLIENTNUM and calculate the mean for specific columns
average_values = credit_info.groupby('CLIENTNUM')[['Avg_Open_To_Buy', 'Avg_Utilization_Ratio']].mean().reset_index()

# Merge the dataframes
latest_records = pd.merge(latest_records, sum_trans_amount, on='CLIENTNUM', how='left')
latest_records = pd.merge(latest_records, average_values, on='CLIENTNUM', how='left')

```

Fig. 22. Penghapusan Data Duplikat dengan Agregasi kolom

Dikarenakan adanya pada data transaksi juga terdapat data CLIENTNUM yang duplikat karena beberapa transaksi dilakukan satu customer maka dilakukan group by berdasarkan record data terakhir. Dalam proses ini tidak dilakukan menghilangkan dengan dihapus melainkan akan dilakukan group by pada kolom yang bersifat agregasi.

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Months_on_b
0	712672083	Existing Customer	65	F	0	High School	Married	Less than \$40K	Blue	
1	77021858	Existing Customer	65	M	1	Graduate	Married	40K–60K	Blue	
2	780689733	Existing Customer	65	M	0	College	Single	40K–60K	Silver	
3	778357458	Existing Customer	65	F	3	Graduate	Married	Less than \$40K	Blue	
4	778247358	Existing Customer	65	M	1	Graduate	Single	40K–60K	Silver	
...
11565	900203138	Existing Customer	24	F	2	College	Married	Less than \$40K	Silver	
11567	900203137	Existing Customer	24	F	0	Graduate	Single	60K–80K	Silver	
11568	900203136	Existing Customer	24	F	0	College	Single	60K–80K	Silver	
11569	900201723	Existing Customer	24	F	0	High School	Single	40K–60K	Silver	
11570	900203146	Existing Customer	24	M	1	College	Single	Less than \$40K	Blue	

11571 rows x 27 columns

Fig. 23. Hasil Penghapusan Duplikat

Pada gambar 23 ditunjukkan data terbaru setelah dilakukannya penghapusan data duplikat yang dimana awalnya berjumlah 20071 menjadi 11571 sesuai nilai uniknya.

```
# Karena adanya penambahan kolumn akibat agregasi maka kolumn lama yang ada iniisial x dibelakangnya akan dihapus
columntodrop = ['Months_Inactive_12_mon_x', 'Contacts_Count_12_mon_x', 'Total_Revolving_Bal_x',
                 'Avg_Open_To_Buy_x', 'Total_Trans_Ct_x', 'Avg_Utilization_Ratio_x']

# mengubah nama DataFrame supaya relevan dan sesuai yang sudah difilter
credit_info = latest_records.drop(columns=columntodrop)
```

Fig. 24. Penghapusan Kolom

karena setelah dilakukannya merge data left maka terdapat beberapa kolom yang duplikat dan tidak diperlukan maka dihapus disini.

```
# melakukan rename terhadap kolumn yang tadinya hasil merger supaya menjadi relevan
rename_columns = {'Months_Inactive_12_mon_y': 'Months_Inactive',
                  'Contacts_Count_12_mon_y': 'Contacts_Count',
                  'Total_Revolving_Bal_y': 'Total_Revolving_Bal',
                  'Total_Trans_Ct_y': 'Total_Trans_Ct',
                  'Avg_Open_To_Buy_y': 'Avg_Open_To_Buy',
                  'Avg_Utilization_Ratio_y': 'Avg_Utilization_Ratio'}
```

Fig. 25. Melakukan Rename Kolom

karena setelah data di merge maka ada beberapa kolom yang duplikat dan menambahkan seperti index column_y maka disini perlu dilakukan rename supaya data lebih baik untuk di proses.

```
filtered_finance = credit_finance.groupby('CLIENTNUM', as_index=False).agg({'Trans_Amount': 'sum',
                                                                           'Revenue': 'sum'})

# setelah sesuai maka digabungkanlah DataFrame 'cust_credit' dan 'grouped_finance' berdasarkan 'CLIENTNUM'
data_credit = pd.merge(credit_info, filtered_finance, on='CLIENTNUM', how='left')
```

Fig. 26. Melakukan Filtering Data

pada data credit_finance juga terdapat data duplikat pada CLIENTNUM sehingga perlu di group by karena satu nasabah pernah melakukan beberapa kali transaksi. Lalu data frame pertama dan kedua di merge berdasarkan CLIENTNUM karena sudah tidak ada duplikat.

```
#ubah tipe data
categorical_columns = ['Gender', 'Education_Level', 'Marital_Status', 'Income_Category', 'Card_Category']

for col in categorical_columns:
    data_credit[col] = data_credit[col].astype('category')
data_credit['CLIENTNUM'] = data_credit['CLIENTNUM'].astype(str)
```

Fig. 27. Mengubah Data Menjadi Kategorikal

Pada proses ini dilakukan pengubahan data yang bersifat kategorikal menjadi kategorikal.

```
# Memisahkan tahun dan kuartal
data_credit[['Quarter', 'Year']] = data_credit['Date_Leave'].str.split(',', expand=True)

# Mengonversi kuartal menjadi nilai numerik (hilangkan 'none' dengan 0)
data_credit['Quarter'] = data_credit['Quarter'].apply(lambda x: 0 if x == 'none' else int(x[1:]))

# Penyesuaian untuk menghitung awal dari setiap kuartal
data_credit['Month_Start'] = data_credit.apply(lambda x: (x['Quarter'] - 1) * 3 + 1 if x['Quarter'] > 0 else None, axis=1)

# Ubah 'Year_Quarter' menjadi datetime dengan format 'YYYY-MM-DD'
data_credit['Year_Quarter'] = pd.to_datetime(data_credit.apply(lambda x: f'{x["Year"]}-{x["Quarter"]}-01').format(int(x['Month_Start'])), errors='ignore')

# Hapus kolom yang tidak diperlukan
data_credit = data_credit.drop(columns=['Quarter', 'Year', 'Month_Start'])
```

Fig. 28. Melakukan Formatting Date

Pada proses mengubah format Quarter dan Year menjadi Date_Leave karena supaya bisa dipakai untuk melakukan trend analysis mengenai customer churn.

```
data_credit['Year_Quarter'].unique()
array([        'NaT', '2018-10-01T00:00:00.000000000', '2018-01-01T00:00:00.000000000',
       '2018-04-01T00:00:00.000000000', '2019-07-01T00:00:00.000000000',
       '2019-04-01T00:00:00.000000000', '2019-01-01T00:00:00.000000000',
       '2019-10-01T00:00:00.000000000'], dtype='datetime64[ns]')
```

Fig. 29. Hasil Formatting

Setelah dilakukan proses formatting sebelumnya, maka data menjadi datetime dan menampilkan data sesuai kuartal dimana kuartal 1 adalah bulan 1, kuartal 2 adalah bulan 4, kuartal 3 adalah bulan 7 dan quartal 4 adalah bulan 10.

```
# Visualisasi untuk customer yang churn per quarter
churned_per_year_quarter = data_credit[data_credit['Attrition_Flag'] == 'Attrited Customer'].groupby(['Year_Quarter'])

fig, ax = plt.subplots(figsize=(12, 8))
churned_per_year_quarter.plot(marker='o', linestyle='-', linewidth=2, label='Churned Customers')
ax.set_title('Number of Churned Customers per Quarter Within Each Year')
ax.set_xlabel('Year Quarter')
ax.set_ylabel('Number of Churned Customers')
ax.legend()
ax.grid(True)
plt.show()

# Filter pelanggan yang churned
churned_customers = data_credit[data_credit['Attrition_Flag'] == 'Attrited Customer']

# Visualisasi: Pie chart - Distribution of churned customers across quarters for each year
unique_years = churned_customers['Year_Quarter'].dt.year.unique()

fig, axes = plt.subplots(rows=1, ncols=len(unique_years), figsize=(16, 8))
for i, year in enumerate(unique_years):
    churned_per_year_quarter = churned_customers[churned_customers['Year_Quarter'].dt.year == year].groupby(['Year'])
    axes[i].pie(churned_per_year_quarter, labels=churned_per_year_quarter.index, autopct='%1.1f%%', colors=['red', 'blue'])
    axes[i].set_title(f'Distribution of Churned Customers - {year}')
plt.show()
```

Fig. 30. Membuat Visualisasi Trendline dan Pie Chart

Pada gambar 30 dilakukan proses pembuatan visualisasi trendline dan pie chart untuk mengetahui trend nasabah yang churn dari waktu ke waktu serta komposisi pelanggan yang churn.

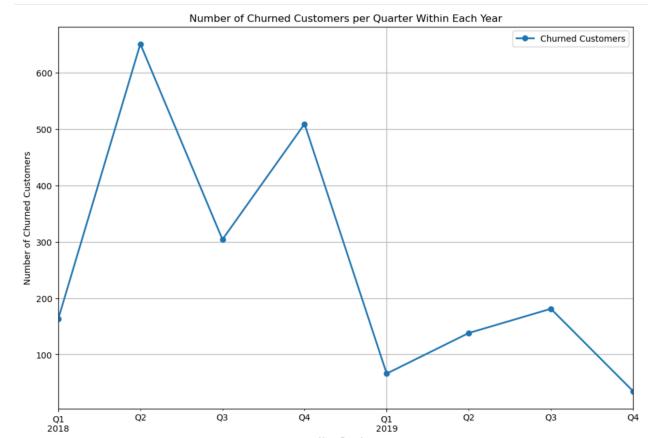


Fig. 31. Trendline Nasabah yang Meninggalkan Layanan Bank

Gambar 31 menunjukkan bahwa tingkat churn tertinggi adalah di Q2 pada 2018 dan terus berkurang sampai di Q4 2019.

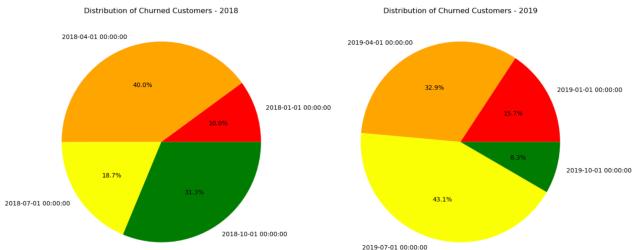


Fig. 32. Pie Chart Komposisi Churn Berdasarkan Waktu

Gambar 32 menunjukkan komposisi nasabah yang churn pada quartal masing-masing, terlihat pada 2018 tingkat churn paling tinggi adalah di Quartal 2 dan pada 2019 pada Quartal 3

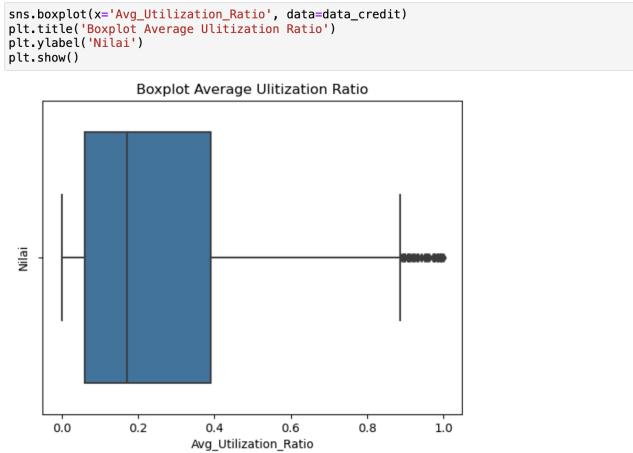


Fig. 33. Boxplot Average Utilization Ratio

Pada gambar 33 terlihat bahwa boxplot untuk Average Utilization Ratio dari nasabah dan ternyata masih terdapat outlier. Oleh karena itu perlu dilakukan penghapusan outlier pada kolom-kolom numerik.

```
# Menentukan kolom numerik yang akan dihapus outlier-nya
numeric_columns = ['Customer_Age', 'Dependent_count', 'Months_on_book', 'Total_Relationship_Count',
                   'Credit_Limit', 'Months_Inactive', 'Contacts_Count', 'Total_Revolving_Bal', 'Total_Trans_Ct',
                   'Avg_Open_To_Buy', 'Avg_Utilization_Ratio', 'Trans_Amount', 'Revenue']

# Menghitung IQR
Q1 = data_credit[numeric_columns].quantile(0.25)
Q3 = data_credit[numeric_columns].quantile(0.75)
IQR = Q3 - Q1

# deklarasi lower dan upper bound
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Proses membersihkan data outlier
data_credit_cleaned = data_credit.copy()
for column in numeric_columns:
    data_credit_cleaned = data_credit_cleaned[(data_credit_cleaned[column] >= lower_bound[column]) &
                                              (data_credit_cleaned[column] <= upper_bound[column])]

data_credit = data_credit_cleaned
```

Fig. 34. Penghapusan Outlier

Gambar 34 menunjukkan proses penghapusan outlier dengan metode IQR (Interquartile Range) yang akan membuat data lebih bagus dan valid.

```
sns.boxplot(x='Avg_Utilization_Ratio', data=data_credit)
plt.title('Boxplot Average Utilization Ratio')
plt.ylabel('Nilai')
plt.show()
```

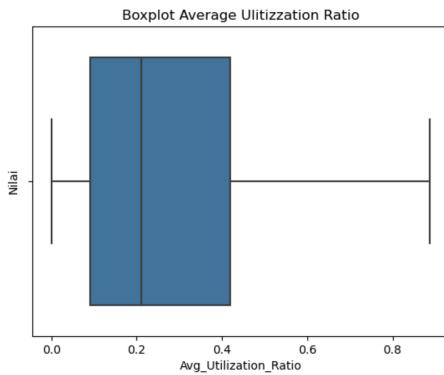


Fig. 35. Cek Outlier dengan Boxplot

Setelah data dibersihkan dari outlier maka pada gambar 35 ditunjukkan hasil nya bahwa data sudah terbebas dari outlier.

```
sns.boxplot(x='Total_Trans_Ct', data=data_credit)
plt.title('Boxplot Total Transaksi Customer')
plt.ylabel('Nilai')
plt.show()
```

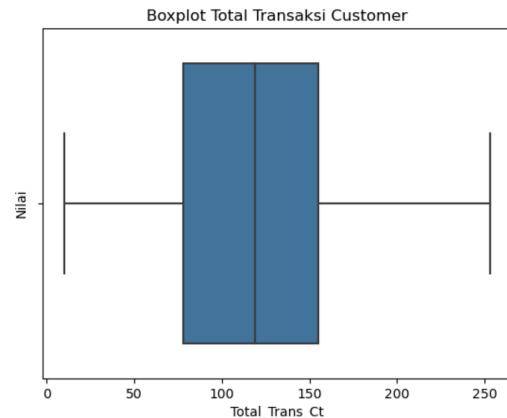


Fig. 36. Cek Outlier dengan Boxplot

Untuk memastikan datanya sudah tidak ada outlier, maka dilakukan percobaan pada satu kolom lainnya dan ternyata pada gambar 36 ditunjukkan bahwa data sudah terbebas dari outlier.

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Months_on_b
0	712672083	Existing Customer	65	F	0	High School	Married	Less than \$40K	Blue	
1	770721858	Existing Customer	65	M	1	Graduate	Married	40K–60K	Blue	
2	780689733	Existing Customer	65	M	0	College	Single	40K–60K	Silver	
3	778357458	Existing Customer	65	F	3	Graduate	Married	Less than \$40K	Blue	
4	778247358	Existing Customer	65	M	1	Graduate	Single	40K–60K	Silver	
...
11566	900203138	Existing Customer	24	F	2	College	Married	Less than \$40K	Silver	
11567	900203137	Existing Customer	24	F	0	Graduate	Single	60K–80K	Silver	
11568	900203136	Existing Customer	24	F	0	College	Single	60K–80K	Silver	
11569	900201723	Existing Customer	24	F	0	High School	Single	40K–60K	Silver	
11570	900203146	Existing Customer	24	M	1	College	Single	Less than \$40K	Blue	

10038 rows x 22 columns

Fig. 37. Data Setelah Penghapusan Outlier

Pada gambar 37 ditampilkan data credit setelah outlier dihapus dan terlihat bahwa data telah berkurang. Oleh karena itu data terbaru ini merupakan data tanpa outlier.

```

# Membutuhkan encoding manual untuk setiap kolom
attrition_flag_mapping = {'Attrited Customer': 1, 'Existing Customer': 0}
gender_mapping = {'F': 0, 'M': 1}
education_level_mapping = {'High School': 0, 'Graduate': 1, 'College': 2, 'Uneducated': 3, 'Post-Graduate': 4, 'Doctorate': 5}
marital_status_mapping = {'Divorced': 0, 'Married': 1, 'Single': 2}
income_category_mapping = {'Less than $40K': 0, '$40K - $60K': 1, '$60K - $80K': 2, '$80K - $120K': 3, '$120K +': 4}
card_category_mapping = {'Blue': 0, 'Silver': 1, 'Gold': 2, 'Platinum': 3}

# Proses encoding memerlukan petak yang udah ditentukan
data_credit['Attrition_Flag'] = data_credit['Attrition_Flag'].map(attrition_flag_mapping)
data_credit['Gender'] = data_credit['Gender'].map(gender_mapping)
data_credit['Education_Level'] = data_credit['Education_Level'].map(education_level_mapping)
data_credit['Marital_Status'] = data_credit['Marital_Status'].map(marital_status_mapping)
data_credit['Income_Category'] = data_credit['Income_Category'].map(income_category_mapping)
data_credit['Card_Category'] = data_credit['Card_Category'].map(card_category_mapping)

# cek hasil
for column in ['Attrition_Flag', 'Gender', 'Education_Level', 'Marital_Status', 'Income_Category', 'Card_Category']:
    unique_values = data_credit[column].unique()
    print(f"Encoded values in column {column}: {unique_values}")

```

Fig. 38. Melakukan Encoding pada Data Kategorikal

Pada gambar 38 dilakukan encoding yang bertujuan agar data tersebut bisa diproses oleh mesin saat digunakan pada tahap modeling

	data_credit																			
	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Months_on_b	Total_Relationship_Count	Credit_Limit	Months_Inactive	Contacts_Count	Total_Revolving_Bal	Total_Trans_Ct	Avg_Open_To_Buy	Avg_Utilization_Ratio	Trans_Amount	Revenue
0	712672083	0	65	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
1	770721658	0	65	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	
2	780689733	0	65	1	0	2	2	1	1	0	0	0	0	0	0	0	0	0	0	
3	778357468	0	65	0	3	1	1	0	0	0	0	0	0	0	0	0	0	0	0	
4	778247358	0	65	1	1	1	2	1	1	0	0	0	0	0	0	0	0	0	0	
...	
11566	900203138	0	24	0	2	2	1	0	1	0	0	0	0	0	0	0	0	0	0	
11567	900203137	0	24	0	0	1	2	2	2	1	0	0	0	0	0	0	0	0	0	
11568	900203136	0	24	0	0	2	2	2	2	1	0	0	0	0	0	0	0	0	0	
11569	900201723	0	24	0	0	0	2	1	1	1	0	0	0	0	0	0	0	0	0	
11570	900203146	0	24	1	1	2	2	0	0	0	0	0	0	0	0	0	0	0	0	

Fig. 39. Cek Hasil Encoding

Pada gambar 39 dilakukan cek hasil encoding yang sudah dilakukan dan terlihat bahwa data sudah ter-encoding dan siap untuk digunakan di proses selanjutnya.

Sebelum melakukan Modeling maka akan dilakukan terlebih dahulu cek korelasi antar kolomnya supaya bisa mengetahui variabel mana yang memiliki pengaruh besar terhadap status nasabah meninggalkan bank.

```

# Memilih variabel-variabel
selected_variables = ['CLIENTNUM', 'Attrition_Flag', 'Customer_Age', 'Gender',
                      'Dependent_count', 'Education_Level', 'Marital_Status',
                      'Income_Category', 'Card_Category', 'Months_on_book',
                      'Total_Relationship_Count', 'Credit_Limit', 'Date_Leave',
                      'Months_Inactive', 'Contacts_Count', 'Total_Revolving_Bal',
                      'Total_Trans_Ct', 'Avg_Open_To_Buy', 'Avg_Utilization_Ratio',
                      'Trans_Amount', 'Revenue', 'Quarter']

# Subset dataset menggunakan variabel-variabel utama
selected_data = data_credit[selected_variables]

# Exclude non-numeric columns from correlation analysis
non_numeric_columns = selected_data.select_dtypes(include=['object'])
correlation_matrix = numeric_columns.corr()

# Membuat visual korelasi heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title('Correlation Heatmap')
plt.show()

```

Fig. 40. Membuat Correlation Heatmap

Pada gambar 40 dilakukan proses untuk membuat visualisasi korelasi heatmap yang bertujuan untuk mengetahui mana variabel yang memiliki korelasi yang kuat pada variabel target sehingga akan dipilih sebagai variabel fitur nantinya saat proses Machine Learning.

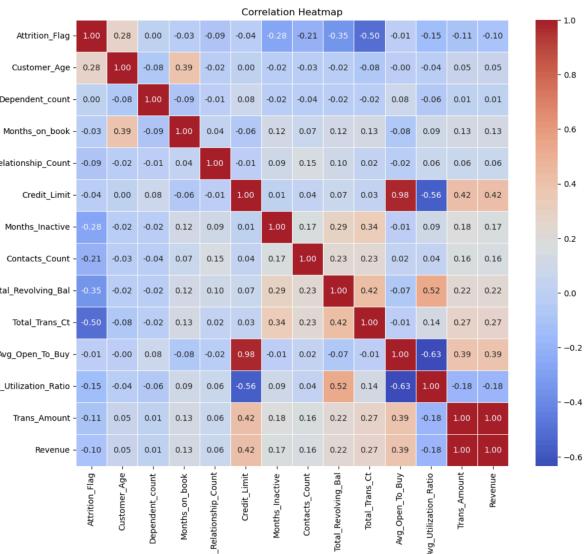


Fig. 41. Korelasi Heatmap Antar Variabel

Pada gambar 41 ditunjukkan hasil visualisasi korelasi antar variabel dan karena pada proses modeling akan berfokus pada penentuan customer churn maka bisa difokuskan pada variabel yang memiliki korelasi terkuat dengan variabel Attrition_Flag. Hasil menunjukkan bahwa variabel yang memiliki korelasi yang kuat adalah Total_Trans_Ct, Total_Revolving_Bal, Customer_Age, Avg_Utilization_Ratio dan Months_Inactive.

D. Modeling

Modeling merupakan fase CRISPM-DM yang dimana akan dilakukan training kepada data yang ada sehingga memperoleh model terbaik untuk memprediksi nasabah akan churn atau tidak.

Feature Selection dengan Random Forest

Pada tahap pemodelan, pemilihan fitur sangat penting untuk meningkatkan akurasi hasil yang diberikan oleh model. Dalam penelitian ini, dilakukan pemilihan fitur menggunakan algoritma Random Forest untuk mengidentifikasi variabel-variabel yang paling signifikan yang akan dijadikan fitur dalam pembuatan model. Metode Random Forest dipilih karena kekokohnya dan kemampuannya menangani dataset besar sambil secara efektif merangking pentingnya setiap variabel. Dengan memilih fitur-fitur yang paling relevan, hal ini ditujukan untuk mengoptimalkan kinerja model dan memastikan bahwa model memberikan hasil yang akurat dan dapat diandalkan dijadikan fitur untuk pembuatan model.

```

# Pilih variabel fitur dan target
features = ['Customer_Age', 'Gender', 'Dependent_count', 'Education_Level', 'Marital_Status',
            'Income_Category', 'Card_Category', 'Months_on_book', 'Total_Relationship_Count',
            'Credit_Limit', 'Months_Inactive', 'Contacts_Count', 'Total_Revolving_Bal',
            'Total_Trans_Ct', 'Avg_Open_To_Buy', 'Avg_Utilization_Ratio', 'Trans_Amount', 'Revenue']

target = 'Attrition_Flag'

```

Fig. 42. Mendeklarasi variabel fitur dan target

Pada gambar 42, variabel-variabel fitur dan target dideklarasikan untuk model. Fitur-fitur yang relevan dipilih untuk digunakan dalam pemodelan, sementara variabel

target 'Attrition_Flag' ditetapkan sebagai variabel yang akan diprediksi.

```
X = data_credit[features]
y = data_credit[target]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Fig. 43. Membagi data menjadi data train dan test

Gambar diatas menunjukkan proses pembagian data menjadi data latih (train) dan data uji (test) dengan proporsi 80:20. Langkah ini penting untuk memastikan bahwa model dapat dievaluasi secara efektif pada data yang tidak dilihat selama pelatihan.

```
model_rf = RandomForestClassifier(n_estimators=100, random_state=42)
model_rf.fit(X_train, y_train)

feature_importances = model_rf.feature_importances_
feature_importance_df = pd.DataFrame({'Feature': features, 'Importance': feature_importances})
feature_importance_df.sort_values(by='Importance', ascending=False)
```

Fig. 44. Melakukan pemilihan fitur dengan *RandomForestClassifier*

Pada gambar diatas modell dilatih pada data latih, dan pentingnya setiap fitur dihitung untuk mengidentifikasi fitur-fitur yang paling signifikan dalam memprediksi variabel target.

```
plt.figure(figsize=(10, 6))
plt.barh(feature_importance_df['Feature'], feature_importance_df['Importance'])
plt.xlabel('Importance')
plt.title('Feature Importance in Random Forest')
plt.gca().invert_yaxis()
plt.show()
```

Fig. 45. Memvisualisasikan Hasil Feature Importance

Code diatas digunakan untuk menampilkan visualisasi hasil pentingnya fitur menggunakan plot batang horizontal. Visualisasi ini menunjukkan tingkat pentingnya masing-masing fitur dalam model Random Forest, memungkinkan kita untuk melihat fitur mana yang paling berpengaruh dalam prediksi model.

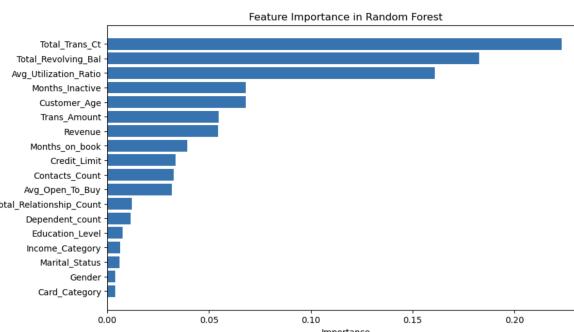


Fig. 45. Hasil Visualisasi Feature Importance

Pada gambar 45 ditunjukkan hasil visualisasi feature importance. Hasil menunjukkan bahwa variabel yang memiliki importance kuat dan di penelitian ini dipilih sebanyak 5 fitur yang memiliki importance tertinggi terhadap variabel target, yaitu Total_Trans_Ct, Total_Revolving_Bal, Customer_Age, Avg_Utilization_Ratio dan Months_Inactive.

Modeling dengan Decision Tree

Pada modeling menggunakan decision tree akan dilakukan beberapa langkah untuk mendapatkan hasil model yang baik.

```
# Pilih fitur yang akan digunakan untuk prediksi
features = ['Total_Trans_Ct', 'Total_Revolving_Bal', 'Customer_Age', 'Avg_Utilization_Ratio', 'Months_Inactive']

# Pilih target (variabel yang ingin diprediksi)
target = 'Attrition_Flag'
```

Fig. 46. Memilih Fitur dan Target

Pada Gambar 46, ditunjukkan proses pemilihan variabel fitur dan target yang akan digunakan dalam pelatihan data.

```
# Pisahkan data menjadi fitur (X) dan target (y)
X = data_credit[features]
y = data_credit[target]

# Pisahkan data menjadi set pelatihan dan pengujian
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Fig. 47. Membagi Data menjadi Train dan Test

Gambar 47 menunjukkan proses pembagian data menjadi set pelatihan (train) dan set pengujian (test) yang akan digunakan dalam modeling dengan decision tree.

```
# Normalisasi data
scaler = StandardScaler()
X_train_normalized = scaler.fit_transform(X_train)
X_test_normalized = scaler.transform(X_test)
```

Fig. 48. Normalisasi Data

Gambar 48 menampilkan proses normalisasi data yang akan digunakan untuk modeling. Normalisasi dilakukan untuk memastikan bahwa semua fitur berada pada skala yang sama, sehingga meningkatkan kinerja model.

```
# Random sampling
X_train_resampled, y_train_resampled = resample(X_train_normalized, y_train, random_state=42)

# Membuat model Decision Tree
model = DecisionTreeClassifier(max_depth=4, random_state=42)
model.fit(X_train_resampled, y_train_resampled)
```

Fig. 49. Menerapkan Teknik Random Sampling

Teknik random sampling digunakan karena adanya ketidakseimbangan data, dimana lebih banyak nasabah yang masih bertahan dibandingkan yang churn. Teknik ini membantu dalam mengatasi masalah ketidakseimbangan data dan memungkinkan pembuatan model decision tree yang lebih akurat.

```
# Visualisasi Decision Tree
plt.figure(figsize=(20, 10))
plot_tree(model, feature_names=features, class_names=[str(cls) for cls in model.classes_], filled=True,
rounded=True, fontsize=10)
plt.savefig('decision_tree.png')
plt.show()

# Melakukan prediksi pada set pengujian
y_pred = model.predict(X_test_normalized)
```

Fig. 50. Membuat Visualisasi Decision Tree

Gambar 50 menunjukkan proses pembuatan visualisasi decision tree. Visualisasi ini digunakan untuk menganalisis struktur pohon keputusa yang terbentuk dan melakukan prediksi pada set pengujian.

Modeling dengan Support Vector Machine (SVM)

```
# Pilih variabel fitur dan target
features = ['Total_Trans_Ct', 'Total_Revolving_Bal', 'Customer_Age', 'Avg_Utilization_Ratio', 'Months_Inactive']
target = 'Attrition_Flag'
```

Fig. 51. Pemilihan Variabel Fitur dan Target

Pada Gambar 51, dilakukan pemilihan variabel fitur yang memiliki korelasi tertinggi dengan variabel target untuk menghindari overfitting.

```
# Pisahkan variabel independen (X) dan dependen (y)
X = data_credit[features]
y = data_credit[target]

# Pisahkan data menjadi set pelatihan dan pengujian
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Fig. 52. Membagi Data menjadi Train dan Test

Gambar 52 menunjukkan proses pembagian data menjadi data pelatihan (train) dan data pengujian (test), dengan ukuran set pengujian sebesar 20%.

```
# Lakukan oversampling karena data tidak seimbang
oversampler = RandomOverSampler(random_state=42)
X_train_resampled, y_train_resampled = oversampler.fit_resample(X_train, y_train)
```

Fig. 53. Melakukan Random Sampling

Pada Gambar 53, diterapkan oversampling terhadap data untuk menangani ketidakseimbangan data sebelum melakukan pemodelan.

```
# Normalisasi skala data
scaler = StandardScaler()
X_train_resampled = scaler.fit_transform(X_train_resampled)
X_test = scaler.transform(X_test)
```

Fig. 54. Melakukan Normalisasi

Gambar 54 menampilkan proses normalisasi data menggunakan StandardScaler(), sehingga memastikan data yang digunakan akurat dalam membangun model SVM.

```
# Support Vector Machine (SVM)
svm_model = SVC()
svm_model.fit(X_train_resampled, y_train_resampled)

# Prediksi pada set pengujian
y_pred_svm = svm_model.predict(X_test)
```

Fig. 55. Membuat Model SVM

Pada Gambar 55, terlihat proses pembuatan model SVM dan melakukan prediksi pada set pengujian untuk evaluasi kinerja model.

E. Evaluation

Pada fase ini, kami akan membandingkan keakuratan kedua algoritma yang digunakan dalam proyek ini. Evaluasi dipisahkan menjadi dua kategori berbeda berdasarkan algoritma yang digunakan.

Decision Tree

Ini merupakan hasil akurasi yang telah kami lakukan menggunakan algoritma decision tree.

```
# Evaluasi kinerja model
accuracy_DT = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

# Tampilkan hasil
print("Accuracy: {:.4f}\n")
print("Confusion Matrix:\n{}\n")
print("Classification Report:\n{}\n")

# Menghitung confusion matrix untuk Decision Tree
conf_matrix_dt = confusion_matrix(y_test, y_pred)

# Membuat heatmap untuk Decision Tree
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_dt, annot=True, fmt='d', cmap='Blues', cbar=False,
            xticklabels=['No Attrition', 'Attrition'],
            yticklabels=['No Attrition', 'Attrition'])
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix - Decision Tree')
plt.show()
```

Fig. 56. Pembuatan Confusion Matrix Untuk Melakukan Evaluasi

Pada gambar 56 ditunjukkan proses untuk membuat confusion matrix yang berguna untuk mengevaluasi hasil model machine learning.

Support Vector Machine (SVM)

Berikut hasil akurasi prediksi algoritma SVM menggunakan Python.

```
# Evaluasi model SVM
print("\nSupport Vector Machine (SVM) setelah oversampling:")
print("Accuracy: {:.4f}, accuracy_score(y_test, y_pred_svm)")
print("Classification Report:\n{}\n", classification_report(y_test, y_pred_svm))
```

Fig. 57. Pembuatan Accuracy Score dan Classification Report

```
# Menghitung confusion matrix
conf_matrix_svm = confusion_matrix(y_test, y_pred_svm)

# Membuat heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_svm, annot=True, fmt='d', cmap='Blues', cbar=False,
            xticklabels=['No Attrition', 'Attrition'],
            yticklabels=['No Attrition', 'Attrition'])
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix - Support Vector Machine (SVM)')
plt.show()
```

Fig. 58. Pembuatan Confusion Matrix

F. Deployment

Setelah tahap evaluasi, di mana temuan model dievaluasi secara menyeluruh, maka dilanjutkan ke tahap implementasi. Pada tahap ini, model deteksi penipuan yang telah dikembangkan dan divalidasi akan diterapkan dalam lingkungan operasional bank melalui platform berbasis web. Untuk mendukung implementasi ini, digunakan suatu framework Flask, yang memungkinkan pembangunan dan pengembangan aplikasi web yang ringan dan efisien. Dengan menggunakan Flask, model deteksi penipuan dapat diintegrasikan secara baik ke dalam infrastruktur bank yang ada, memberikan kemampuan deteksi penipuan secara real-time dan meningkatkan keamanan serta keandalan sistem transaksi bank.

IV. RESULT AND ANALYSIS

Decision Tree

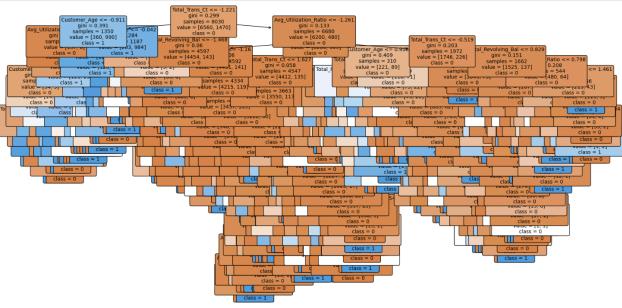


Fig. 59. Hasil Decision Tree dengan Max_Depth = none

Terlihat bahwa decision tree dengan tidak mengatur max_depth menghasilkan visualisasi yang sulit dipahami. Oleh karena itu, di penelitian ini digunakan max_depth bernilai 4.

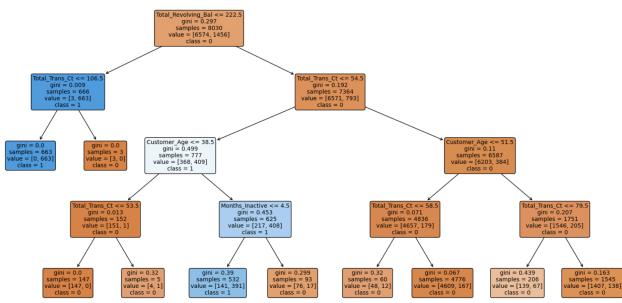


Fig. 60. Visualisasi Decision Tree dengan Max_Depth = 4

Terlihat saat menggunakan max_depth nya 4 maka visualisasi menunjukkan hasil yang lebih mudah dimengerti. Pada visualisasi tersebut ditunjukkan bahwa hal yang paling berpengaruh terhadap churn adalah hutang nasabah yang belum dibayar atas kartu kreditnya yang dimana kurang dari 222.5 maka akan lanjut ke faktor variabel selanjutnya dan begitupun selanjutnya.

Accuracy: 0.9313					
Confusion Matrix:					
[[1625 33] [105 245]]					
Classification Report:					
	precision	recall	f1-score	support	
0	0.94	0.98	0.96	1658	
1	0.88	0.70	0.78	350	
accuracy			0.93	2008	
macro avg	0.91	0.84	0.87	2008	
weighted avg	0.93	0.93	0.93	2008	

Fig. 61. Hasil Confusion Matrix Decision Tree

Model Decision Tree yang telah dibuat untuk memprediksi churn pelanggan pada studi kasus bank telah memberikan hasil evaluasi yang cukup baik. Dalam kasus ini, akurasi model adalah 0.9313, atau 93.13%. Ini menunjukkan bahwa sekitar 93% dari prediksi model benar secara keseluruhan.

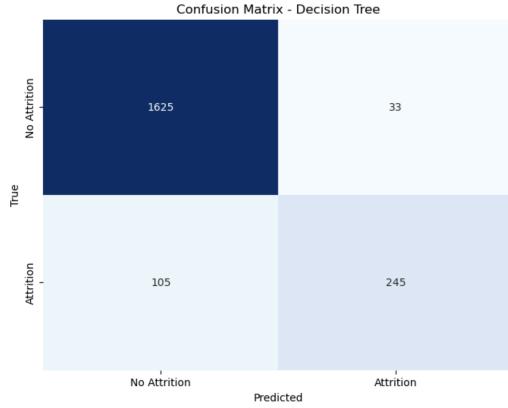


Fig. 62. Heatmap Confusion Matrix Decision Tree

Model menunjukkan hasil yang baik karena model mampu menghasilkan :

- True Positive (TP): 245 (Churn yang diprediksi benar)
- True Negative (TN): 1625 (Tidak Churn yang diprediksi benar)
- False Positive (FP): 33 (Tidak Churn yang diprediksi Churn)
- False Negative (FN): 105 (Churn yang diprediksi tidak Churn)

Secara keseluruhan model decision tree menunjukkan bahwa model memiliki kinerja yang baik dalam mengidentifikasi kelas 0 (tidak Churn), tetapi memiliki recall yang lebih rendah untuk kelas 1 (Churn), yang dapat menjadi area untuk diperhatikan dan ditingkatkan.

Support Vector Machine (SVM)

Support Vector Machine (SVM) setelah oversampling:					
Accuracy: 0.9307768924302788					
Classification Report:					
	precision	recall	f1-score	support	
0	0.95	0.96	0.96	1658	
1	0.82	0.77	0.80	350	
accuracy			0.93	2008	
macro avg	0.89	0.87	0.88	2008	
weighted avg	0.93	0.93	0.93	2008	

Fig. 63. Hasil Confusion Matrix SVM

Model SVM yang telah dibuat untuk memprediksi churn pelanggan pada studi kasus bank telah memberikan hasil evaluasi yang cukup baik. Dalam kasus ini, akurasi model adalah 0.9307, atau 93%. Ini menunjukkan bahwa sekitar 93% dari prediksi model benar secara keseluruhan.

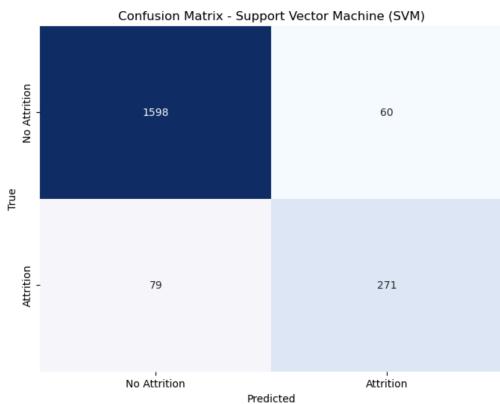


Fig. 64. Heatmap Confusion Matrix SVM

Model menunjukkan hasil yang baik karena model mampu menghasilkan :

- True Positive (TP): 271 (Churn yang diprediksi benar)
- True Negative (TN): 1598 (Tidak Churn yang diprediksi benar)
- False Positive (FP): 60 (Tidak Churn yang diprediksi Churn)
- False Negative (FN): 79 (Churn yang diprediksi tidak Churn)

Model SVM setelah proses oversampling memberikan hasil yang baik dengan akurasi yang tinggi. Keseimbangan antara presisi dan recall cukup baik untuk kelas 0 (tidak Churn), sementara untuk kelas 1 (Churn), recall (sensitivitas) sedikit lebih rendah, yang bisa menjadi area yang perlu diperhatikan tergantung pada kebutuhan bisnis spesifik. menunjukkan bahwa model memiliki kinerja yang baik dalam memprediksi kategori pelanggan yang Churn dan tidak Churn.

Berdasarkan hasil evaluasi, model Decision Tree dan SVM yang telah dilakukan menunjukkan kinerja yang baik dalam mengatasi masalah klasifikasi churn pelanggan. Decision Tree memiliki akurasi sedikit lebih tinggi sebesar 93.13% dibandingkan dengan 93.08% dari SVM. Namun, SVM memiliki True Positive (TP) dan True Negative (TN) yang lebih tinggi, menunjukkan kemampuannya yang lebih baik dalam mengidentifikasi pelanggan yang sebenarnya Churn dan tidak Churn. Decision Tree memiliki presisi yang lebih tinggi untuk kelas Churn, sementara SVM memiliki recall yang lebih tinggi. Pilihan antara keduanya dapat tergantung pada preferensi dan kebutuhan spesifik tugas klasifikasi. Jika fokus pada mengidentifikasi secara akurat pelanggan yang benar-benar melakukan Churn, maka SVM menjadi pilihan yang lebih baik dengan recall yang lebih tinggi. SVM menunjukkan kemampuan yang lebih baik dalam menghasilkan True Positive (TP) dan True Negative (TN) yang tinggi, menandakan keunggulan dalam mengklasifikasikan dengan akurat baik pelanggan yang Churn maupun yang tidak Churn. Dengan fokus pada akurasi yang lebih tinggi dan recall yang baik untuk kelas Churn, SVM dapat memberikan hasil yang lebih dapat diandalkan dalam mengidentifikasi pelanggan yang memerlukan perhatian lebih lanjut terkait potensi untuk berpindah.

Setelah mengetahui hasil evaluasi model Decision Tree dan SVM yang sudah dilakukan dalam penelitian ini, terdapat beberapa peluang yang dapat dioptimalkan untuk meningkatkan performa prediksi churn pelanggan. Pertama, penyempurnaan hyperparameter pada kedua model dapat dilakukan dengan menggunakan teknik penyetelan seperti Grid Search atau Random Search untuk menemukan kombinasi hyperparameter yang optimal. Selanjutnya, rekayasa fitur dapat dieksplorasi dengan mengevaluasi dan mengubah fitur-fitur yang digunakan atau mengenali fitur-fitur baru yang mungkin meningkatkan daya prediksi model. Apabila dataset relatif kecil, augmentasi data dapat diterapkan untuk meningkatkan keragaman dan jumlah sampel. Penanganan ketidakseimbangan kelas dapat diperkuat dengan mempertimbangkan teknik undersampling, SMOTE, atau menggunakan bobot kelas.

Deployment

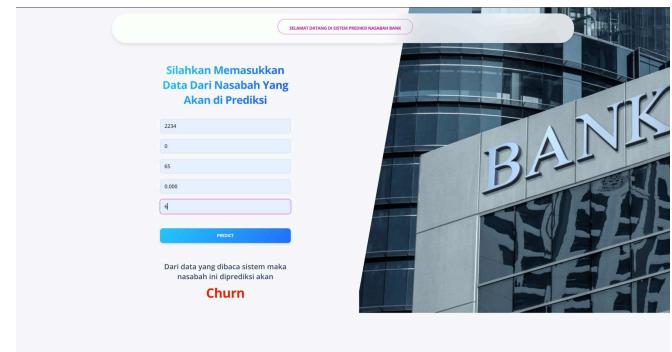


Fig. 65. Interface situs web untuk deployment

Gambar diatas merupakan tampilan website yang dimana terdapat beberapa bagian untuk dilakukan implementasi model.

- *Total Transaction Customer*: Input teks untuk memasukkan jumlah total transaksi yang dilakukan oleh nasabah.
- *Total Revolving Balance*: Input teks untuk memasukkan saldo berulang total dari nasabah.
- *Customer Age*: Input teks untuk memasukkan usia nasabah.
- *Avg Utilization Ratio*: Input teks untuk memasukkan rata-rata rasio pemanfaatan kredit nasabah.
- *Months Inactive*: Input teks untuk memasukkan jumlah bulan nasabah tidak aktif.

Setelah mengisi data-data di atas, Anda dapat mengklik tombol "Predict". Tombol ini akan mengirim data yang dimasukkan ke model prediksi untuk diproses. Hasil prediksi akan ditampilkan dalam bentuk teks yang menunjukkan apakah nasabah diprediksi akan berhenti (attrition) atau tidak, berdasarkan data yang dianalisis oleh sistem.

V. CONCLUSION

Hasil pada pemodelan antara Decision Tree dan SVM untuk mengidentifikasi pelanggan yang potensial untuk churn menunjukkan bahwa keduanya berhasil memberikan performa yang baik, dengan Decision Tree sedikit lebih unggul dalam hal akurasi sekitar 93.13%, dibandingkan dengan 93.08% pada SVM. Namun, jika dilihat lebih dalam, bisa dilihat bahwa keduanya memiliki kelebihan masing-masing model. Decision Tree lebih baik dalam mengidentifikasi pelanggan yang benar-benar pindah dengan akurasi tinggi, sementara SVM memiliki kecenderungan yang lebih besar untuk mengidentifikasi pelanggan yang sebenarnya melakukan pindah (True Positives) dengan lebih baik daripada Decision Tree. Untuk meningkatkan kualitas prediksi, maka dapat dilakukan dengan cara menyempurnakan parameter-parameter model, seperti mengoptimalkan hyperparameter. Selain itu, mengeksplorasi fitur-fitur baru atau penyesuaian pada fitur yang sudah digunakan juga dapat memberikan hasil yang positif. Dalam pengembangannya juga dapat mempertimbangkan teknik augmentasi data untuk menangani ketidakseimbangan dalam dataset. Selain itu, dapat dilakukan juga implementasi cross validation dan interpretasi model akan membantu memahami bagaimana model membuat sebuah keputusan dan di mana kita dapat meningkatkan keakuratannya.

REFERENCES

- [1] I. Kaur and J. Kaur, "Customer Churn Analysis and Prediction in Banking Industry using Machine Learning," 2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC), Nov. 2020, doi: <https://doi.org/10.1109/pdgc50313.2020.9315761>.
- [2] I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," SN Computer Science, vol. 2, no. 3, pp. 1–21, Mar. 2021, doi: <https://doi.org/10.1007/s42979-021-00592-x>.
- [3] M. Alloghani, D. Al-Jumeily, J. Mustafina, A. Hussain, and A. J. Aljaaf, "A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science," Unsupervised and Semi-Supervised Learning, pp. 3–21, Sep. 2019, doi: https://doi.org/10.1007/978-3-030-22475-2_1.
- [4] K. T. Jensen, "An introduction to reinforcement learning for neuroscience," arXiv.org, Nov. 13, 2023. <https://arxiv.org/abs/2311.07315> (accessed Dec. 22, 2023).
- [5] M. Kozan, "Supervised and Unsupervised Learning (an Intuitive Approach)", Medium, Sep. 01, 2021. <https://medium.com/@metehankozan/supervised-and-unsupervised-learning-an-intuitive-approach-cd8f8f64b644>
- [6] P. Thanh Noi and M. Kappas, "Comparison of Random Forest, k-Nearest Neighbor, and Support Vector Machine Classifiers for Land Cover Classification Using Sentinel-2 Imagery," Sensors (Basel, Switzerland), vol. 18, no. 1, p. 18, 2017, doi: <https://doi.org/10.3390/s18010018>.
- [7] S. Peker and Ö. Kart, "Transactional data-based customer segmentation applying CRISP-DM methodology: A systematic review," Journal of Data, Information and Management, Jan. 2023, doi: <https://doi.org/10.1007/s42488-023-00085-x>.
- [8] "CRISP DM Sebagai Salah Satu Standard untuk Menghasilkan Data Driven Decision Making yang Berkualitas," www.djkn.kemenkeu.go.id. <https://www.djkn.kemenkeu.go.id/artikel/baca/15134/CRISP-DM-Sebagai-Salah-Satu-Standard-untuk-Menghasilkan-Data-Driven-Decison-Making-yang-Berkualitas.html>
- [9] C. Schröer, F. Kruse, and J. M. Gómez, "A Systematic Literature Review on Applying CRISP-DM Process Model," Procedia Computer Science, vol. 181, no. 1, pp. 526–534, 2021, Available: <https://www.sciencedirect.com/science/article/pii/S187705921002416>
- [10] "Apa itu Pembersihan Data? - Penjelasan Pembersihan Data - AWS," Amazon Web Services, Inc. <https://aws.amazon.com/id/what-is/data-cleansing/>
- [11] S. Laoyan, "Semua yang Anda Perlu Ketahui tentang Manajemen Proyek Waterfall [2024] • Asana," Asana, Apr. 26, 2024. [Online]. Available: <https://asana.com/id/resources/waterfall-project-management-methodology>
- [12] J. I. Myung, Y. Tang, and M. A. Pitt, "Chapter 11 Evaluation and comparison of computational models," in Methods in enzymology on CD-ROM/Methods in enzymology, 2009, pp. 287–304. doi: [10.1016/s0076-6879\(08\)03811-1](https://doi.org/10.1016/s0076-6879(08)03811-1).
- [13] "Apa itu SDLC? - Penjelasan tentang Siklus Hidup Pengembangan Perangkat Lunak - AWS," Amazon Web Services, Inc. <https://aws.amazon.com/id/what-is/sdlc/>
- [14] "AI vs Machine Learning - Perbedaan Antara Kecerdasan Buatan dan ML - AWS," Amazon Web Services, Inc. <https://aws.amazon.com/id/compare/the-difference-between-artificial-intelligence-and-machine-learning/>
- [15] Salva and Salva, "Machine Learning (ML): Pengertian, algoritma, dan potensi," Indonesian Cloud - 100% Lokal | Multi Cloud Provider Indonesia | Berbasis cloud computing Indonesian Cloud. Kebutuhan teknologi untuk bisnis Anda mulai dari (IaaS), Cyber Security hingga solusi bisnis (SaaS), Oct. 12, 2023. <https://indonesiancloud.com/machine-learning-ml-pengertian-algoritma-dan-potensi-di-masa-depan/>
- [16] H. H. P. P. Prajapati, "Study and analysis of decision tree based classification algorithms," ijcseonline.org, Oct. 01, 2018. https://www.ijcseonline.org/full_paper_view.php?paper_id=2984
- [17] "Entropy in Machine Learning - JavatPoint," www.javatpoint.com. <https://www.javatpoint.com/entropy-in-machine-learning>
- [18] K. H. E. P. Putra, "Support Vector Machine Algorithm," School of Information Systems, Feb. 14, 2022. <https://sis.binus.ac.id/2022/02/14/support-vector-machine-algorithm/>
- [19] H. Zhu, "Bank Customer Churn Prediction with Machine Learning Methods," Advances in Economics, Management and Political Sciences, vol. 69, no. 1, pp. 23–29, Jan. 2024, doi: [10.54254/2754-1169/69/20230773](https://doi.org/10.54254/2754-1169/69/20230773).
- [20] S. Dutta, P. Bose, S. K. Bandyopadhyay, and M. Janarthanan, "A hybrid machine learning model for bank customer churn prediction," International Journal of Engineering Trends and Technology, vol. 70, no. 6, pp. 13–23, Jun. 2022, doi: [10.14445/22315381/ijett-v70i6p202](https://doi.org/10.14445/22315381/ijett-v70i6p202).