# Modul Praktikum
# Data Mining


## Tim Penyusun:

Dr. Rakhmat Arianto, S.ST., M.Kom

Ir. Rudy Ariyanto, ST., M.Cs

Prof. Dr. Eng. Rosa Andrie Asmara, ST., MT

**Jurusan Teknologi Informasi**

**Sistem Informasi Bisnis**

**Politeknik Negeri Malang**

**Februari 2025**

# DAFTAR ISI

# JOBSHEET 11

# Decision Tree

## Pendahuluan

Modul ini menjelaskan penerapan algoritma Decision Tree dengan menggunakan studi kasus harga mobil bekas yang dilengkapi dengan tahapan yang bisa diambil kesimpulannya

## Tujuan Praktikum

Setelah menyelesaikan praktikum ini, mahasiswa mampu:

- Mahasiswa dapat memahami dan mengidentifikasi parameter-parameter penting dalam Decision Tree.
- Mahasiswa dapat menggunakan scikit-learn sebagai tools untuk membuat model klasifikasi.
- Mahasiswa mampu melakukan pengujian dan evaluasi model menggunakan confusion matrix dan akurasi.
- Mahasiswa mampu mengoptimasi model melalui pemilihan parameter yang tepat (hyperparameter tuning).

## Peralatan yang dibutuhkan

Beberapa peralatan yang dibutuhkan dalam menyelesaikan praktikum ini adalah:

- Aplikasi Microsoft Excel
- Google Colab
- Google Drive
- Koneksi Internet
- Browser Web

## Praktikum

### Praktikum Simple Linear Regression

Studi Kasus yang digunakan dalam praktikum ini adalah data pada tabel sebagai berikut:

| ID | Umur | Penghasilan | Status Menikah | Memiliki Rumah | Membeli Produk |
|---|---|---|---|---|---|
| **1** | <30 | Rendah | Tidak | Tidak | Tidak |
| **2** | <30 | Rendah | Tidak | Ya | Tidak |
| **3** | 30-40 | Sedang | Tidak | Tidak | Ya |
| **4** | >40 | Tinggi | Ya | Tidak | Ya |
| **5** | >40 | Rendah | Ya | Ya | Tidak |
| **6** | >40 | Tinggi | Tidak | Ya | Ya |

| | | | | | |
|---|---|---|---|---|---|
| **7** | 30-40 | Sedang | Ya | Tidak | Ya |
| **8** | <30 | Rendah | Ya | Tidak | Tidak |
| **9** | <30 | Tinggi | Tidak | Ya | Ya |

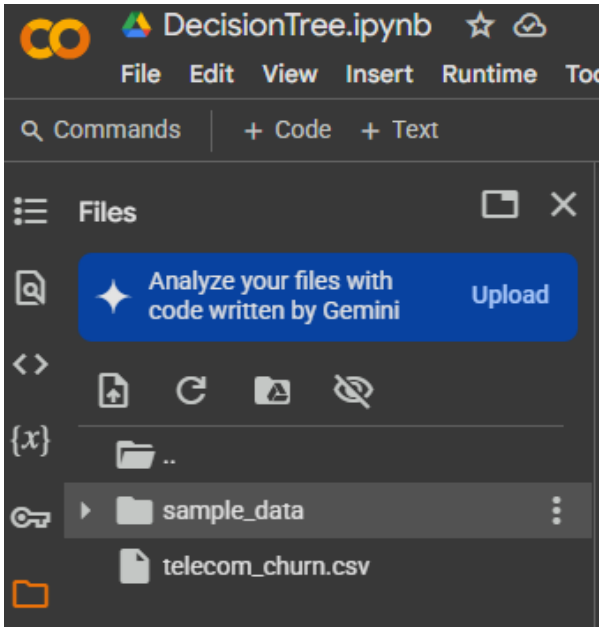
Lakukan praktikum sesuai tahapan berikut:

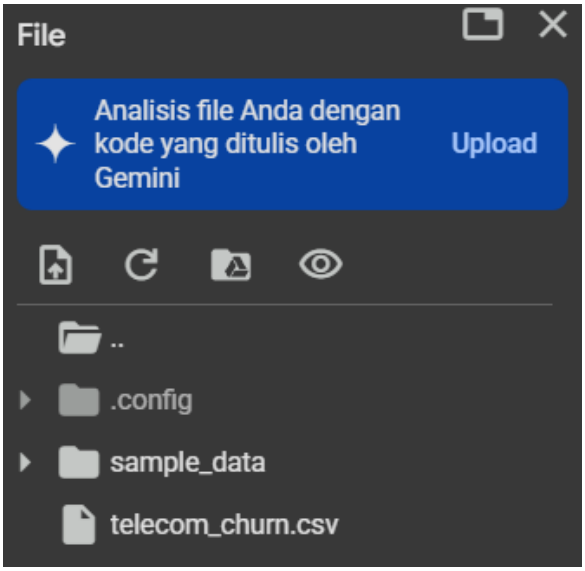
a. Buka aplikasi web browser

b. Buka Google Colabs dan berikan nama file “DecisionTree.ipynb”

c. Download dan gunakan dataset telecom_churn.csv sebagai bahan praktikum

d. Unggah file csv pada google colabs sehingga muncul pada bagian Files




e. Lakukan impor library yang dibutuhkan:

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
import matplotlib.pyplot as plt
```

f. Lakukan load dataset

```
# Load dataset
df = pd.read_csv("telecom_churn.csv")
```

g. Lakukan pemisahan fitur yang tidak digunakan

```
# Drop kolom yang tidak relevan
df = df.drop(columns=["state", "area code", "phone number"])
```

h. Lakukan encoding untuk kolom jenis data kategori

```
# Label encoding kolom kategori
le = LabelEncoder()
df['international plan'] = le.fit_transform(df['international plan'])
df['voice mail plan'] = le.fit_transform(df['voice mail plan'])
df['churn'] = df['churn'].astype(int)  # Ubah boolean ke 0/1
```

i. Tentukan target dari dataset yang digunakan

```
# Split fitur dan target
X = df.drop("churn", axis=1)
y = df["churn"]
```

j. Lakukan split dataset untuk training dan testing

```
# Split data training dan testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

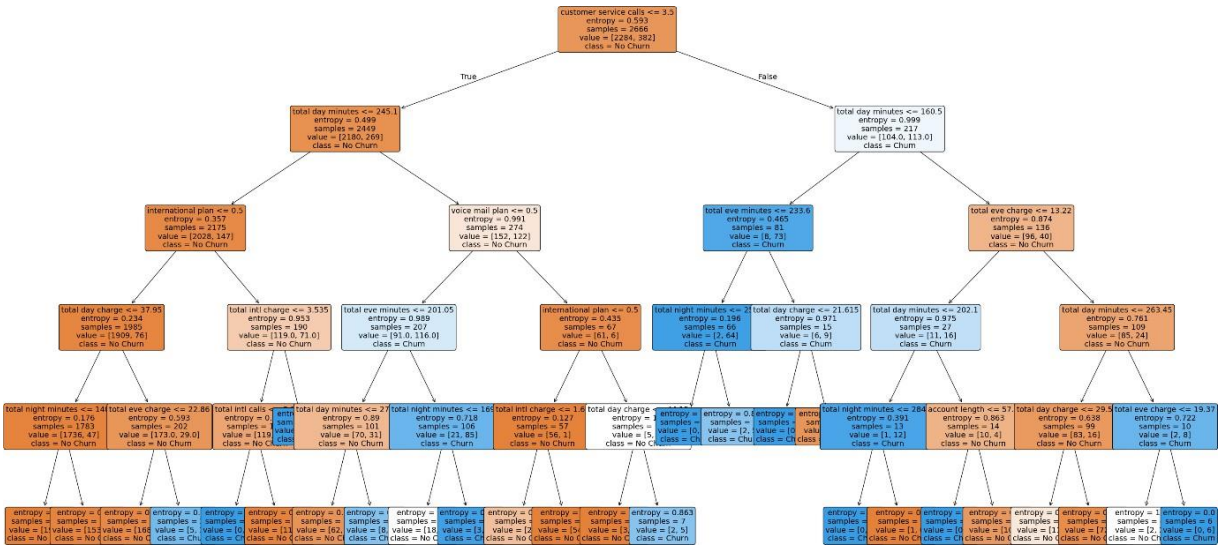
k. Membangun Model Decision Tree

```
model = DecisionTreeClassifier(criterion='entropy', max_depth=5, min_samples_split=10, random_state=42)
model.fit(X_train, y_train)
```

l. Menampilkan hasil decision tree

```
# Visualisasi tree dengan resolusi yang lebih tinggi dan pengaturan lainnya
plt.figure(figsize=(40,20)) # Increased figure size
plot_tree(model,
          feature_names=X.columns,
          class_names=["No Churn", "Churn"],
          filled=True,
          rounded=True, # Rounded boxes
          fontsize=14) # Increased font size
plt.show()
```

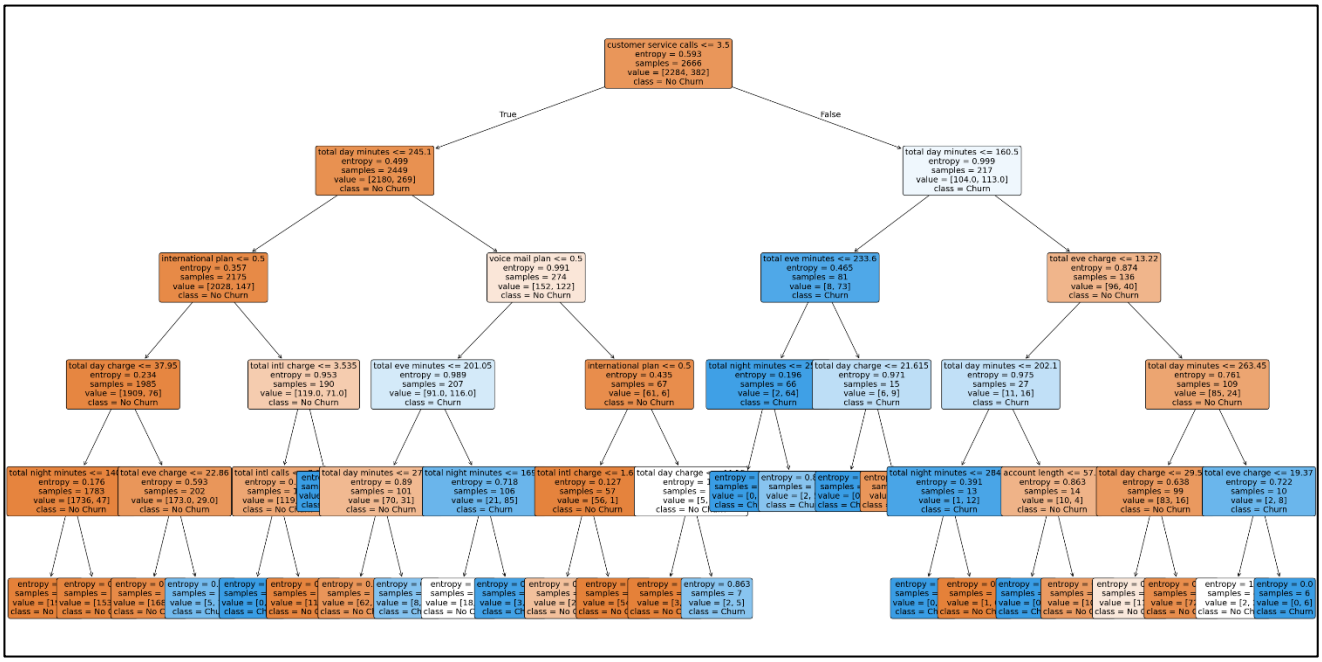
m. Luaran dari model Decision Tree yang terbentuk adalah sebagai berikut:



## Percobaan Saya

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
import matplotlib.pyplot as plt

# Load dataset
df = pd.read_csv("telecom_churn.csv")

# Drop kolom yang tidak relevan
df = df.drop(columns=["state", "area code", "phone number"])

# Label encoding kolom kategori
le = LabelEncoder()
df['international plan'] = le.fit_transform(df['international plan'])
df['voice mail plan'] = le.fit_transform(df['voice mail plan'])
df['churn'] = df['churn'].astype(int)  # Ubah boolean ke 0/1

# Split fitur dan target
x = df.drop("churn", axis=1)
y = df["churn"]

# Split data training dan testing
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

# Buat dan latih model
model = DecisionTreeClassifier(criterion='entropy', max_depth=5, min_samples_split=10, random_state=42)
model.fit(x_train, y_train)

# Visualisasi tree dengan resolusi yang lebih tinggi dan pengaturan lainnya
plt.figure(figsize=(40, 20))  # Increased figure size
plot_tree(model,
          feature_names=x.columns,  # diperbaiki dari X.columns ke x.columns
          class_names=["No Churn", "Churn"],
          filled=True,
          rounded=True,  # Rounded boxes
          fontsize=14)  # Increased font size
plt.show()
```

## Hasil Percobaan



n. Lakukan evaluasi dari hasil yang didapatkan dengan Algoritma Decision Tree

```
y_pred = model.predict(X_test)

print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
print("Accuracy Score:", accuracy_score(y_test, y_pred))
```

```
Confusion Matrix:
 [[553  13]
 [ 34  67]]
Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.98      0.96       566
           1       0.84      0.66      0.74       101

    accuracy                           0.93       667
   macro avg       0.89      0.82      0.85       667
weighted avg       0.93      0.93      0.93       667

Accuracy Score: 0.9295352323838081
```

o. Lakukan Optimasi Parameter (Tuning)

```
# Eksperimen parameter
optimized_model = DecisionTreeClassifier(criterion='gini', max_depth=6, min_samples_split=5)
optimized_model.fit(X_train, y_train)

# Evaluasi ulang
y_pred_opt = optimized_model.predict(X_test)
print("Akurasi Model Setelah Optimasi:", accuracy_score(y_test, y_pred_opt))
```

Loading...

```
Akurasi Model Setelah Optimasi: 0.9490254872563718
```

**Percobaan Saya**

```
# Prediksi dan evaluasi
y_pred = model.predict(x_test)
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
print("Accuracy Score:", accuracy_score(y_test, y_pred))

# Eksperimen parameter
optimized_model = DecisionTreeClassifier(criterion='gini', max_depth=6, min_samples_split=5)
optimized_model.fit(x_train, y_train)

# Evaluasi ulang
y_pred_opt = optimized_model.predict(x_test)
print("Akurasi Model Setelah Optimasi:", accuracy_score(y_test, y_pred_opt))
```

**Hasil Percobaan**

```
Confusion Matrix:
 [[553  13]
 [ 34  67]]
Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.98      0.96       566
           1       0.84      0.66      0.74       101

    accuracy                           0.93       667
   macro avg       0.89      0.82      0.85       667
weighted avg       0.93      0.93      0.93       667

Accuracy Score: 0.9295352323838081
Akurasi Model Setelah Optimasi: 0.9475262368815592
```

**LATIHAN**

1. Jelaskan secara mendetail setiap perintah dari code yang diketikkan dengan cara memberikan “comment” dari setiap baris code

```python
import pandas as pd  # Mengimpor library pandas untuk manipulasi data
from sklearn.preprocessing import LabelEncoder  # Mengimpor LabelEncoder untuk mengubah data kategorik ke numerik
from sklearn.model_selection import train_test_split  # Untuk membagi data menjadi data latih dan uji
from sklearn.tree import DecisionTreeClassifier, plot_tree  # Mengimpor Decision Tree dan fungsi visualisasinya
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score  # Untuk evaluasi performa model
import matplotlib.pyplot as plt  # Untuk membuat visualisasi grafik

# Load dataset
df = pd.read_csv("telecom_churn.csv")  # Membaca file CSV ke dalam DataFrame pandas

# Drop kolom yang tidak relevan
df = df.drop(columns=["state", "area code", "phone number"])  # Menghapus kolom yang tidak dibutuhkan untuk model

# Label encoding kolom kategori
le = LabelEncoder()  # Membuat objek LabelEncoder
df['international plan'] = le.fit_transform(df['international plan'])  # Mengubah kolom 'international plan' dari teks ke angka
df['voice mail plan'] = le.fit_transform(df['voice mail plan'])  # Mengubah kolom 'voice mail plan' dari teks ke angka
df['churn'] = df['churn'].astype(int)  # Mengubah nilai boolean True/False menjadi 1/0

# Split fitur dan target
x = df.drop("churn", axis=1)  # Memisahkan fitur (semua kolom kecuali target 'churn')
y = df["churn"]  # Menyimpan kolom target ('churn') dalam variabel y

# Split data training dan testing
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
# Membagi data menjadi 80% data latih dan 20% data uji secara acak (tetap karena random_state)

# Buat dan latih model
model = DecisionTreeClassifier(criterion='entropy', max_depth=5, min_samples_split=10, random_state=42)
# Membuat model decision tree dengan kriteria entropy, kedalaman maksimum 5, dan minimal 10 sampel untuk split
model.fit(x_train, y_train)  # Melatih model dengan data training

# Visualisasi tree dengan resolusi yang lebih tinggi dan pengaturan lainnya
plt.figure(figsize=(40, 20))  # Mengatur ukuran gambar visualisasi pohon
plot_tree(model,
          feature_names=x.columns,  # Menampilkan nama kolom fitur
          class_names=["No Churn", "Churn"],  # Label kelas target
          filled=True,  # Mengisi warna pada node pohon berdasarkan kelas
          rounded=True,  # Membuat kotak node berbentuk bulat
          fontsize=14)  # Mengatur ukuran font pada visualisasi
plt.show()  # Menampilkan visualisasi decision tree

# Prediksi dan evaluasi
y_pred = model.predict(x_test)  # Memprediksi hasil churn dari data uji
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))  # Menampilkan matriks kebingungan
print("Classification Report:\n", classification_report(y_test, y_pred))  # Menampilkan metrik evaluasi: precision, recall, f1-score
print("Accuracy Score:", accuracy_score(y_test, y_pred))  # Menampilkan akurasi model

# Eksperimen parameter
optimized_model = DecisionTreeClassifier(criterion='gini', max_depth=6, min_samples_split=5)
# Membuat model baru dengan parameter berbeda (kriteria gini, kedalaman 6, min split 5)
optimized_model.fit(x_train, y_train)  # Melatih model baru dengan data training

# Evaluasi ulang
y_pred_opt = optimized_model.predict(x_test)  # Memprediksi ulang dengan model yang dioptimasi
print("Akurasi Model Setelah Optimasi:", accuracy_score(y_test, y_pred_opt))  # Menampilkan akurasi model yang sudah dioptimasi
```

## TUGAS PRAKTIKUM

1. Gunakan fungsi code “feature_importances_” dari model dan jelaskan fitur apa saja yang paling berkontribusi pada prediksi churn!

```python
importances = model.feature_importances_
feature_names = x.columns
feature_importance_df = pd.DataFrame({'Feature': feature_names, 'Importance': importances})
feature_importance_df = feature_importance_df.sort_values(by='Importance', ascending=False)

print(feature_importance_df)
```

| | Feature | Importance |
|---|---|---|
| 4 | total day minutes | 0.289534 |
| 1 | international plan | 0.146160 |
| 16 | customer service calls | 0.145004 |
| 14 | total intl calls | 0.101929 |
| 15 | total intl charge | 0.068721 |
| 6 | total day charge | 0.057289 |
| 9 | total eve charge | 0.053798 |
| 7 | total eve minutes | 0.050591 |
| 10 | total night minutes | 0.043681 |
| 2 | voice mail plan | 0.038872 |
| 0 | account length | 0.004421 |
| 3 | number vmail messages | 0.000000 |
| 5 | total day calls | 0.000000 |
| 8 | total eve calls | 0.000000 |
| 11 | total night calls | 0.000000 |
| 12 | total night charge | 0.000000 |
| 13 | total intl minutes | 0.000000 |

**5 Fitur Teratas yang Paling Berkontribusi pada Prediksi Churn:**

1. **total day minutes — 0.2895**
   - **Kontribusi terbesar.**
   - Semakin banyak waktu panggilan pada siang hari, semakin tinggi potensi churn.
   - Bisa jadi karena pelanggan aktif menggunakan layanan di jam sibuk dan mengalami gangguan/kekecewaan.
2. **international plan — 0.1462**
   - Pelanggan yang berlangganan paket internasional memiliki perilaku berbeda terhadap churn.
   - Bisa jadi karena biaya tambahan atau ekspektasi layanan yang lebih tinggi.
3. **customer service calls — 0.1450**
   - Menunjukkan ketidakpuasan pelanggan.
   - Semakin sering pelanggan menelepon customer service, semakin tinggi risiko mereka untuk churn.
4. **total intl calls — 0.1019**
   - Banyaknya panggilan internasional juga berkontribusi, mungkin terkait dengan kebutuhan layanan yang lebih stabil atau terjangkau.
5. **total intl charge — 0.0687**
   - Besarnya biaya yang dikeluarkan untuk layanan internasional bisa menyebabkan ketidakpuasan jika tidak sesuai harapan.

2. Jelaskan Apa dampak dari pelanggan yang sering menghubungi customer service!

Frekuensi tinggi dalam menghubungi customer service adalah indikator kuat ketidakpuasan. Jika pelanggan terlalu sering menelepon customer service, ini dapat menandakan masalah atau keluhan yang belum terselesaikan. Dalam model, ini merupakan fitur penting (importance = 0.145004), artinya pelanggan yang sering menghubungi customer service lebih berisiko untuk churn.

3. Apakah penggunaan menit siang/malam mempengaruhi churn? Jelaskan!

   - Total day minutes sangat mempengaruhi churn (importance = 0.289534), artinya pelanggan yang menggunakan lebih banyak menit di siang hari kemungkinan memiliki interaksi intens dengan layanan, dan ketidakpuasan pada waktu puncak bisa menjadi alasan churn.

   - Total night minutes juga mempengaruhi tapi lebih kecil (importance = 0.043681), menunjukkan bahwa penggunaan malam tidak sekuat siang dalam memprediksi churn.

4. Bagaimana rekomendasi bisnis untuk mengurangi churn berdasarkan model?

   1. **Perhatikan pelanggan dengan penggunaan tinggi di siang hari**:
      - Menyediakan paket atau diskon khusus untuk penggunaan siang hari.
      - Memastikan jaringan dan kualitas layanan stabil saat jam sibuk.
   2. **Perbaiki pengalaman customer service**:
      - Memantau pelanggan yang sering menghubungi customer service.
      - Menindak lanjuti keluhan dengan cepat dan personalisasi solusi mereka.
   3. **Tinjau kembali pelanggan dengan international plan**:
      - Memberikan edukasi atau nilai tambah untuk pengguna paket ini agar mereka merasa dihargai.
      - Memantau kepuasan mereka lebih intens.
   4. **Gunakan notifikasi dini (early warning)**:
      - Membuat sistem yang mendeteksi pelanggan dengan pola risiko churn dan beri insentif seperti diskon, survey feedback, atau penawaran eksklusif.