DATA SCIENCE

IMPLEMENTASI STUDI KASUS RANDOM FOREST

Presentation by : Kelompok 5

Alfonsus William Hamonangan Sinaga (234311005)

Mohammad Fakhriza Maftukhin (234311018)

Muhammad Aulia Alfarouq (234311020)

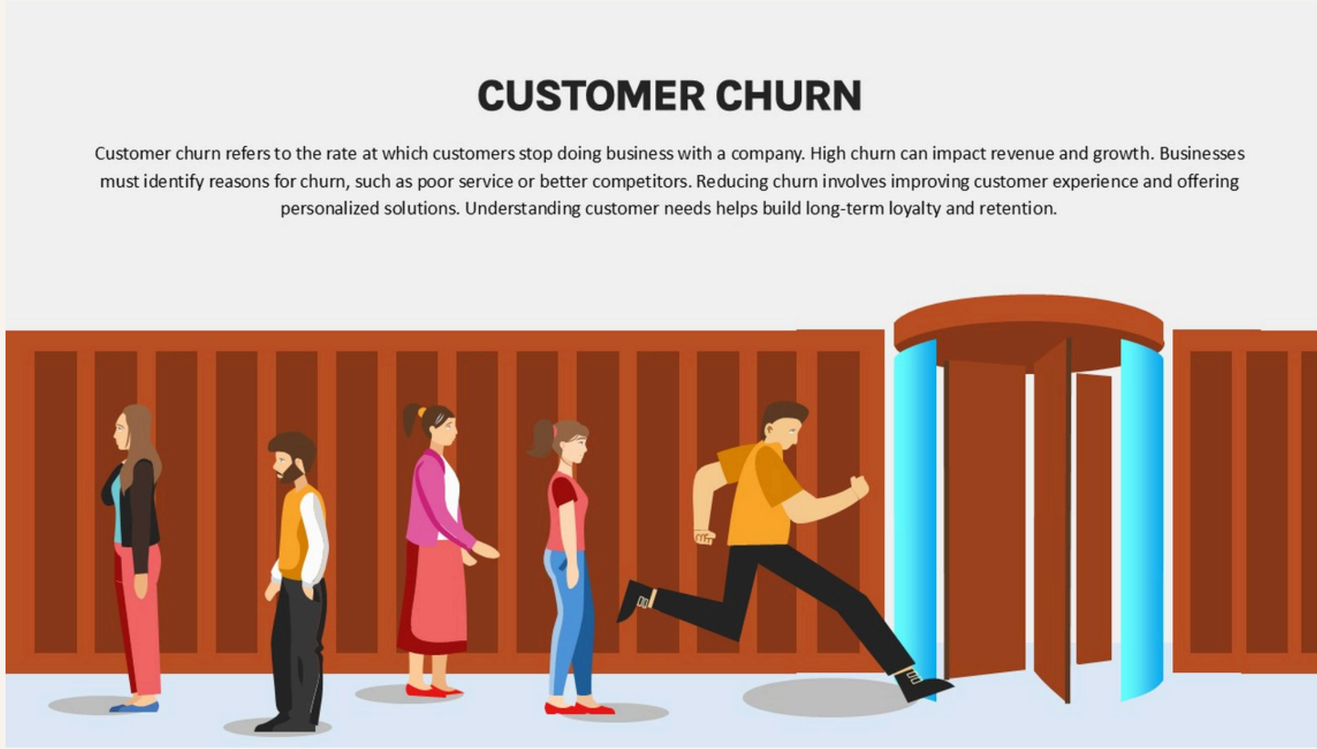
STUDI KASUS: LATAR BELAKANG

Skenario: Data `customer_churn_light.csv`

- MEMPREDIKSI PELANGGAN MANA YANG AKAN CHURN (BERHENTI) BERDASARKAN DATA PENGGUNAAN MEREKA.
- TARGET: `CHURN` (YES/NO)

- FITUR NUMERIK:
- `TENURE_MONTHS` (LAMA BERLANGGANAN)
- `MONTHLY_USAGE` (PENGGUNAAN BULANAN)
- `COMPLAINTS` (JUMLAH KOMPLAIN)
- `PAYMENT_LATE` (KETERLAMBATAN BAYAR)

- FITUR KATEGORIKAL:
- `PLAN_TYPE` (JENIS PAKET)



BAGAIMANA RANDOM FOREST BEKERJA

- Preprocessing Pipeline: Data `plan_type` (teks, mis: "Basic", "Premium") diubah menjadi kolom numerik (`plan_type_Basic`, `plan_type_Premium`) oleh OneHotEncoder secara otomatis.
- Bangun 200 Pohon: Model membangun 200 pohon. Setiap pohon dilatih pada sampel acak (subset) dari data pelanggan (`X_train`).
- Contoh Split Pohon: Sebuah pohon mungkin fokus pada fitur numerik, misal: `tenure_months < 6` DAN `complaints > 1` untuk memprediksi "Churn".
- Voting Akhir: Model melakukan voting pada data `X_test`. Jika 180 dari 200 pohon memprediksi "Churn", maka prediksi akhir adalah "Churn".

PENJELASAN KODE PROGRAM (PYTHON)

```
import pandas as pd
from sklearn.model_selection import train_test_split

# Muat data dari file CSV
df = pd.read_csv('customer_churn_light.csv')

# 1. Pisahkan Fitur (X) dan Target (y)
X = df.drop('churn', axis=1)
y = df['churn']

# 2. Bagi data menjadi set Latih dan Tes
# (80% latih, 20% tes)
# stratify=y memastikan proporsi
# 'churn' sama di data latih & tes
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2,
    random_state=42, stratify=y
)
```

Penjelasan

- `X` berisi semua kolom *kecuali* `churn` (fitur/pertanyaan).
- `y` hanya berisi kolom `churn` (target/jawaban).
- `train_test_split` adalah langkah krusial. Kita "menyembunyikan" 20% data (`X_test`, `y_test`) dari model. Model hanya boleh belajar (`fit`) pada `X_train` dan `y_train`.

KODE: 2. DEFINISI PREPROCESSING & MODEL

```
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.ensemble import RandomForestClassifier

# Tentukan kolom-kolom
categorical_cols = ['plan_type']
# (numeric_cols tidak perlu didefinisi
# jika kita pakai remainder='passthrough')

# 3. Buat Preprocessor
# OneHotEncoder mengubah data teks (plan_type)
# menjadi angka (0 atau 1)
preprocess = ColumnTransformer([
    ('cat', OneHotEncoder(handle_unknown='ignore'),
        categorical_cols)
], remainder='passthrough')

# 4. Tentukan Model
model = RandomForestClassifier(n_estimators=200,
                               random_state=42)
```

`ColumnTransformer` adalah "otak" preprocessing. kelas ini tahu bahwa harus menerapkan `OneHotEncoder` ke `categorical_cols` dan membiarkan (`passthrough`) sisa kolom lainnya.

`RandomForestClassifier` didefinisikan untuk menggunakan 200 pohon.

KODE: 3. TRAINING & EVALUASI PIPELINE

```
from sklearn.pipeline import Pipeline
from sklearn.metrics import accuracy_score

# 5. Buat Pipeline Utama
# Menggabungkan langkah (3) dan (4)
pipeline = Pipeline([
    ('prep', preprocess),
    ('rf', model)
])

# 6. Latih (Fit) Pipeline
# Pipeline secara otomatis menerapkan
# 'prep' SEBELUM melatih 'rf'
pipeline.fit(X_train, y_train)

# 7. Prediksi & Evaluasi
preds = pipeline.predict(X_test)
accuracy = accuracy_score(y_test, preds)

print("Accuracy:", accuracy)
```

`Pipeline` menggabungkan `preprocess` dan `model` menjadi satu objek.

`pipeline.fit(X_train, ...)`: Ini adalah perintah utamanya. Pipeline secara otomatis menerapkan preprocessing ke `X_train`, lalu melatih model RF pada data yang sudah bersih itu.

`pipeline.predict(X_test)`: Pipeline menerapkan preprocessing yang *sama* ke `X_test` lalu membuat prediksi.

`accuracy_score` memvalidasi hasil kita.

HASIL OUTPUT PROGRAM

CONSOLE OUTPUT

```
Accuracy: 1.0
   customer_id  tenure_months  monthly_usage  complaints  payment_late  \
5            6             15           12.5           0             1
1            2              3            2.1           1             1

  plan_type
5   Premium
1     Basic
['No' 'Yes']
5     No
1    Yes
Name: churn, dtype: object
```

Akurasi 1.0: Model memprediksi data uji dengan akurasi 100% sempurna pada sampel ini.

Data Sampel:

- Pelanggan ID 6 (Tenure 15 bulan, Usage 12.5)
- Pelanggan ID 2 (Tenure 3 bulan, Usage 2.1)

Hasil Prediksi:

- **ID 6 (No):** Diprediksi setia (tidak churn).
- **ID 2 (Yes):** Diprediksi akan berhenti (churn). Kemungkinan karena masa langganan pendek dan penggunaan rendah.

THANK YOU