# LUCKY FINGERS

**BANKING DATASET MARKETING**

# TEAM LUCKY FINGERS

**KYVLAN**

DATA SCIENTIST

**MUTIA**

DATA SCIENTIST

**HARYANTO**

DATA SCIENTIST

**SHAFLY**

DATA SCIENTIST

# Latar Belakang

- Deposito berjangka merupakan sumber pendapatan utama bank. Bank berencana melakukan kampanye pemasaran kepada pelanggan melalui telepon.
- Namun bank membutuhkan investasi besar untuk call center karena diharuskan merekrut banyak pekerja dalam melaksanakan kampanye tersebut.
- Oleh karena itu diperlukan identifikasi target customer yang berpotensi untuk dihubungi melalui telepon.

# PROBLEM STATEMENT


BANKING DATA SET

# IDENTIFIED PROBLEM

Dengan data Klien yang sangat banyak, manager marketing akan kesulitan untuk menentukan Klien seperti apa yang memiliki kemungkinan paling besar untuk membeli atau tertarik dengan produk yang ditawarkan.


# Penyelesaian Masalah

**Menganalisa Client client yang berpotensi untuk diberikan penawaran deposit**

# OBJECTIVE

GOAL

Memprediksi customer yang berpontensial untuk dihubungi penawaran deposito

**OBJECTIVE**

Membuat Model klasifikasi customer yang berpotensial dan Mengimplementasikan model

# Business Metric

**Business Metric :**
CVR

# Dataset yang digunakan

| | |
|---|---|
| 01 | AGE |
| 02 | JOB |
| 03 | MARITAL |
| 04 | EDUCATION |
| 05 | DEFAULT |
| 06 | HOUSING |
| 07 | LOAN |
| 08 | CONTACT |
| 09 | MONTH |
| 10 | DURATION |
| 11 | CAMPAIGN |
| 12 | POUTCOME |
| 13 | Y |

# Missing Value

## Dataset Train

**Pada dataset Train, tidak ada data yang hilang atau memiliki NaN value. Hal ini dapat dilihat pada data disamping.**

```
[ ]  data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41787 entries, 0 to 41786
Data columns (total 29 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   age                 41787 non-null  float64
 1   balance             41787 non-null  float64
 2   duration            41787 non-null  float64
 3   campaign            41787 non-null  float64
 4   previous            41787 non-null  int64
 5   y                   41787 non-null  int64
 6   job_admin.          41787 non-null  int64
 7   job_blue-collar     41787 non-null  int64
 8   job_entrepreneur    41787 non-null  int64
 9   job_housemaid       41787 non-null  int64
 10  job_management      41787 non-null  int64
 11  job_retired         41787 non-null  int64
 12  job_self-employed   41787 non-null  int64
 13  job_services        41787 non-null  int64
 14  job_student         41787 non-null  int64
 15  job_technician      41787 non-null  int64
 16  job_unemployed      41787 non-null  int64
 17  job_unknown         41787 non-null  int64
 18  default_no          41787 non-null  int64
 19  default_yes         41787 non-null  int64
 20  loan_no             41787 non-null  int64
 21  loan_yes            41787 non-null  int64
 22  contact_cellular    41787 non-null  int64
 23  contact_telephone   41787 non-null  int64
 24  contact_unknown     41787 non-null  int64
 25  poutcome_failure    41787 non-null  int64
 26  poutcome_other      41787 non-null  int64
 27  poutcome_success    41787 non-null  int64
 28  poutcome_unknown    41787 non-null  int64
dtypes: float64(4), int64(25)
memory usage: 9.2 MB
```

# Duplikasi Value

## Dataset Train

**Pada dataset Train, tidak ada data yang terduplikat.**
**Hal ini dapat dilihat pada data disamping.**

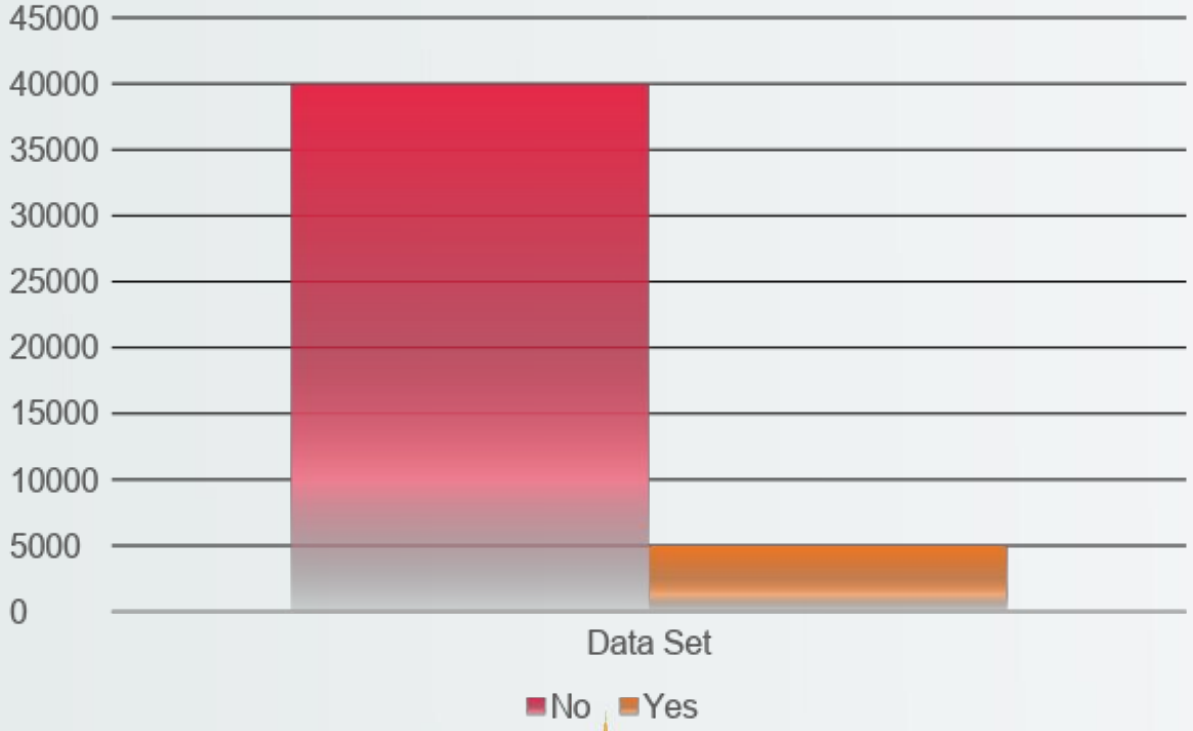
In [6]:
```
print('Jumlah data duplicate: ', train.duplicated().sum())
```

```
Jumlah data duplicate:  0
```

Terdapat 0 data yang terduplikasi, dilakukan dropping pada data duplikat.

In [7]:
```
train.drop_duplicates(inplace=True)
```

# IMBALANCE DATASET



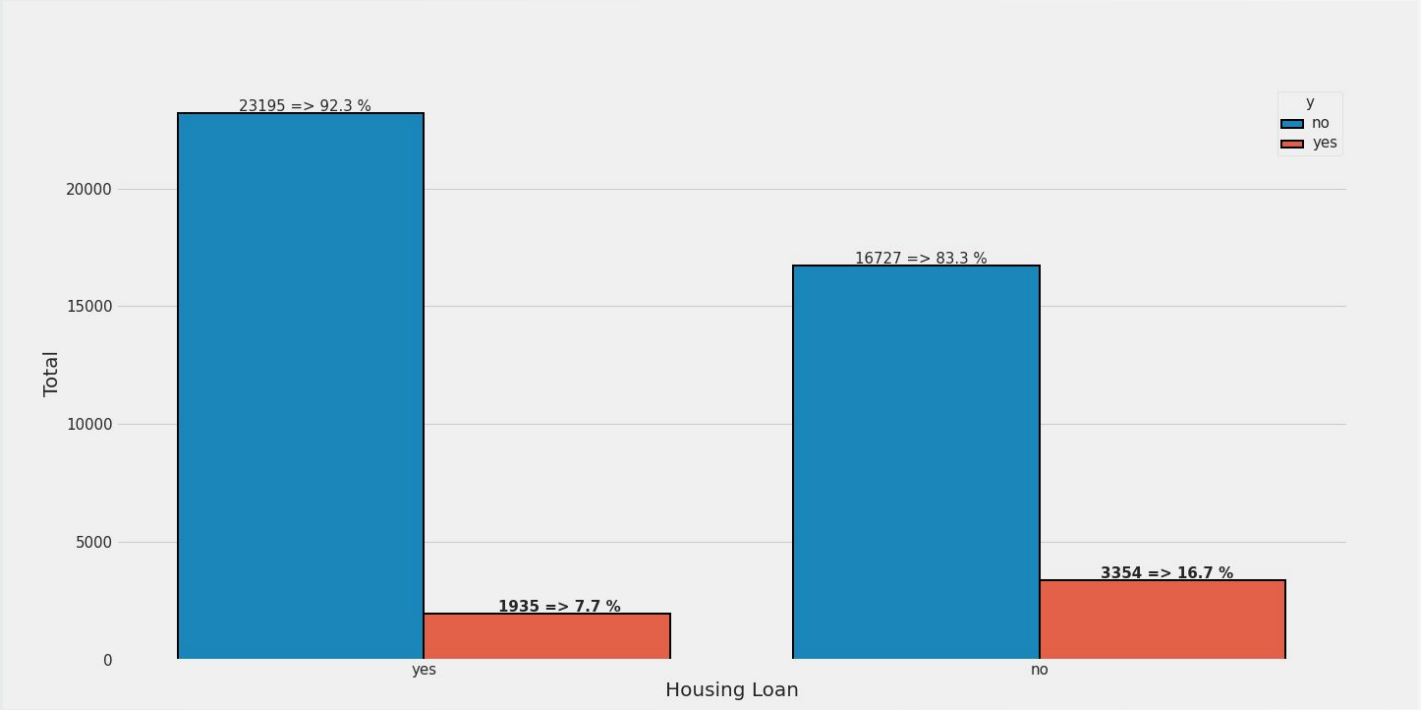
| | count | unique | top | freq |
|---|---|---|---|---|
| **job** | 45211 | 12 | blue-collar | 9732 |
| **marital** | 45211 | 3 | married | 27214 |
| **education** | 45211 | 4 | secondary | 23202 |
| **default** | 45211 | 2 | no | 44396 |
| **housing** | 45211 | 2 | yes | 25130 |
| **loan** | 45211 | 2 | no | 37967 |
| **contact** | 45211 | 3 | cellular | 29285 |
| **month** | 45211 | 12 | may | 13766 |
| **poutcome** | 45211 | 4 | unknown | 36959 |
| **y** | 45211 | 2 | no | 39922 |

# DATA CATEGORICAL

Untuk kolom dengan tipe data Category, didapatkan beberapa insight sebagai berikut:

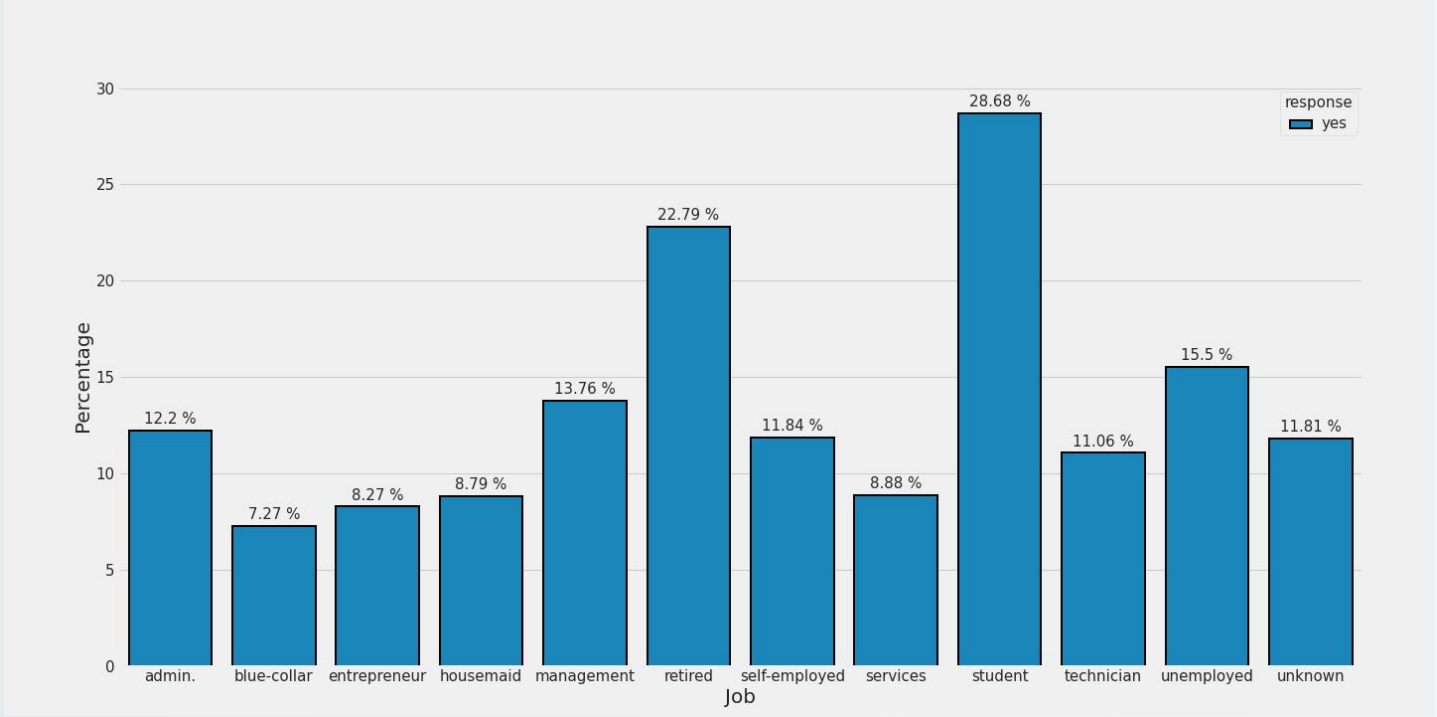
- **blue-collar** adalah jenis pekerjaan dengan distribusi terbanyak di dataset
- Data didominasi oleh klien dengan status perkawinan **married**
- **secondary** menjadi jenjang pendidikan terbanyak yang dimiliki oleh klien di data ini
- Sekitar 50% dari klien pernah mengajukan pinjaman untuk **housing**, namun sedikit yang mengajukan pinjaman untuk personal
- Kontak dengan klien sering dilakukan melalui **cellular**
- Rata-rata kontak dengan klien dilakukan pada bulan **may**

# Housing Loan



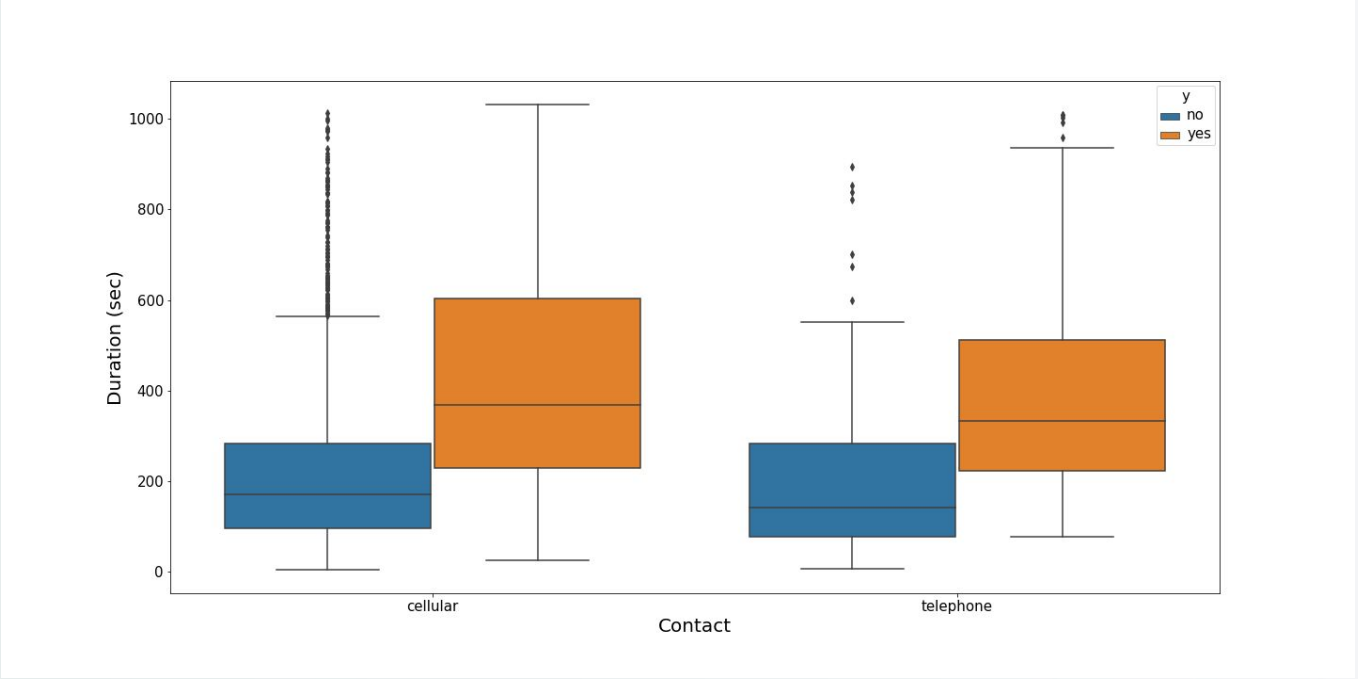

Customer yang tidak memiliki pinjaman rumah memiliki persentase untuk mengambil tawaran deposito lebih tinggi

# Job



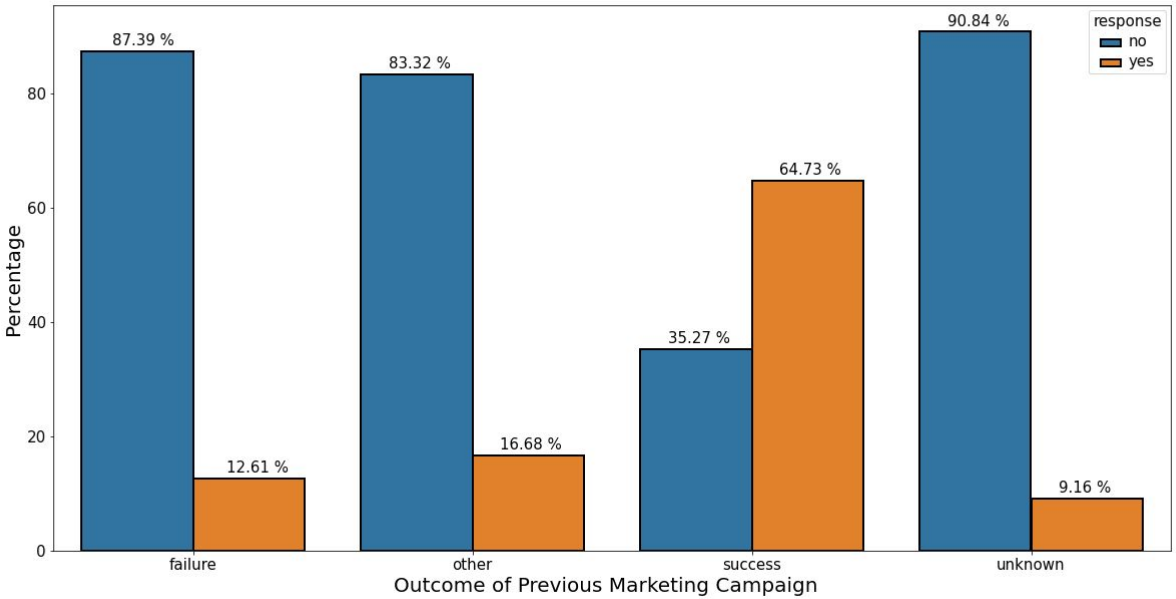

Customer dengan profesi sebagai student dan retired memiliki persentase lebih tinggi untuk mengambil tawaran deposito.

# Contact and Duration




Customer yang menolak tawaran deposito memiliki durasi yang lebih sedikit dengan nilai tengah 2 - 3 menit.
Customer yang mengambil tawaran deposito memiliki durasi dengan nilai tengah di kisaran 5 - 6 menit.

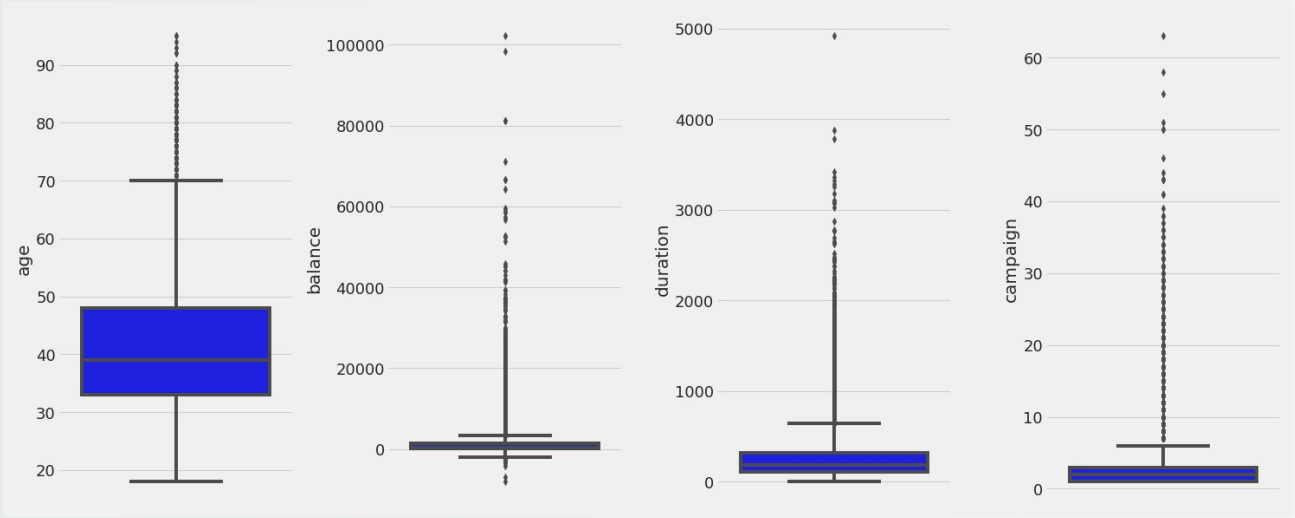
# Poutcome / Outcome of Previous Campaign




Customer yang mengambil penawaran pada kampanye marketing sebelumnya berpotensi untuk mengambil penawaran pada kampanye saat ini.

# Outlier

Ditemukan outlier pada beberapa kolom seperti age, balance, duration dan campaign




# Correlation

Tidak ditemukan korelasi kuat antar kolom/feature maupun feature dan target.

# Features Engineering

- Penghapusan nilai outlier yang terdapat pada hasil 3 kalinya standar deviasi pada feature duration.
- Pada feature job, education dan contact memiliki nilai unknown dan dilakukan feature engineering dengan mengisinya dengan nilai modus pada 3 feature tersebut.
- Undersampling dilakukan karena dari hasil cek imbalance data pada target tersebut tanpa feature previous, pdays, dan day.

# Model Building

- Data numerik di preprocessing scaling menggunakan MinMaxScaler.
- Data kategori di encode menggunakan one hot encoding.
- Pada model data train 80% dan data test 20%.
- Pada model dilakukan hyperparameter default, kemudian hyperparameter tuning.

| **Model** | **Recall** | **Precision** | **AUC** |
|---|---|---|---|
| KNN | 54% | 30% | 76% |
| SVM | 77% | 34% | 85% |

# Model SVM

•Dengan menggunakan parameter default model SVM mendapatkan akurasi Train sebesar 0.8 dan akurasi Test sebesar 0.78.

•Dilakukan hyperparameter tuning berupa Gridsearch CV dengan 3 Fold. Adapun parameter yang dioptimasi adalah:

•gamma : Rentang log -3 sampai 3, 0.001 adalah gamma terbaik
•C : Rentang log -3 sampai 3, 1000 adalah C terbaik
•Hasilnya, akurasi model meningkat pada Train data maupun Test data menjadi 0.84 pada Train data, dan 0.825 pada Test data.

# Mengapa SVM?

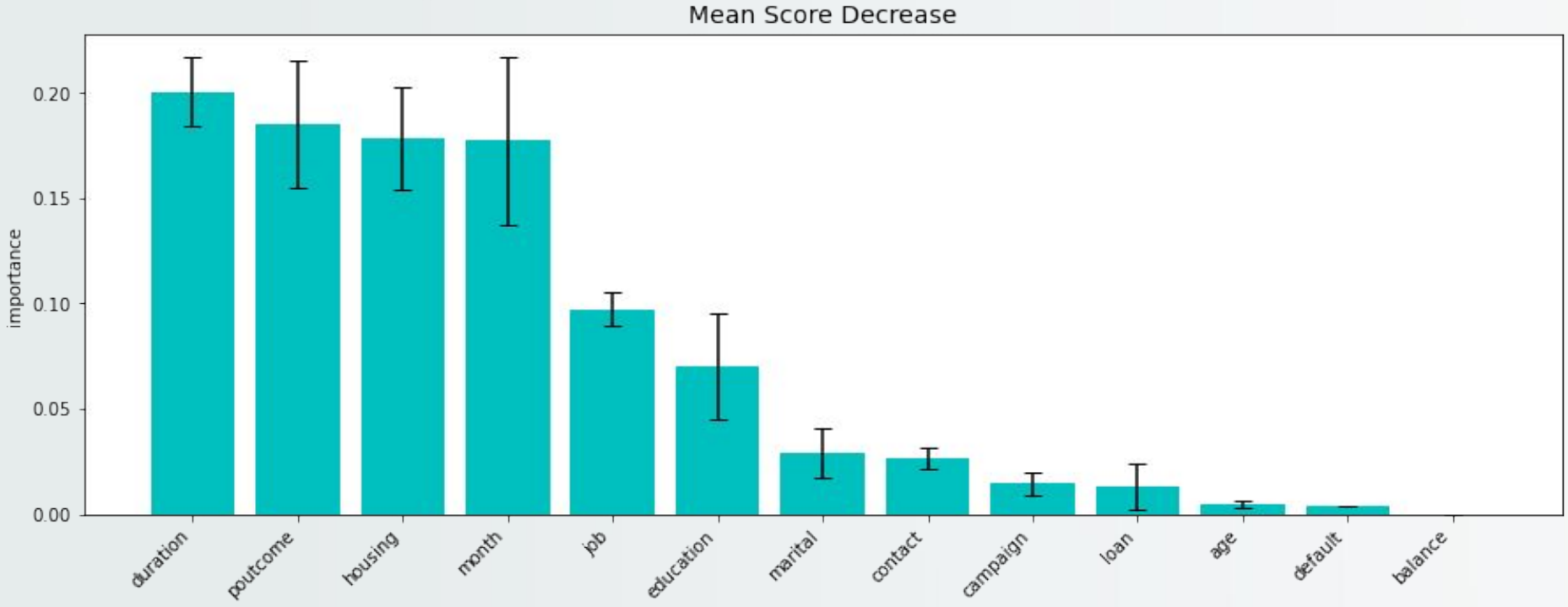
SVM merupakan model machine learning yang terbilang cukup bagus dari segi akurasi, meskipun waktu processing yang dibutuhkan lebih tinggi daripada KNN namun hasil akurasi lebih baik.

SVM juga merupakan modelling yang paling sering digunakan sebagai best practice dalam industri machine learning, karena lebih simple dan default serta tidak perlu banyak menggunakan parameter tuning lain yang memerlukan pembelajaran lebih lanjut.

# Model Evaluasi

•Hasil prediksi dari model menunjukkan cukup banyak data pada False Positive, di mana model memprediksi kelas yes tapi kenyataannya no. Hal ini dikarenakan dataset Test yang digunakan tidak terdistribusi secara normal, dimana data no mendominasi sekitar 4000 data, dan data yes hanya 521. Namun, efek False Positive ini berkurang pada hasil prediksi model SVM.

# Feature Importance




Terlihat dari plot feature importance, bahwa feature housing dan duration memegang peranan penting dalam pembuatan model SVM. Sedangkan feature contact dan month terlihat tidak terlalu penting dalam feature importance ini.

# Perbandingan Kedua Model

## CLASSIFICATION REPORT

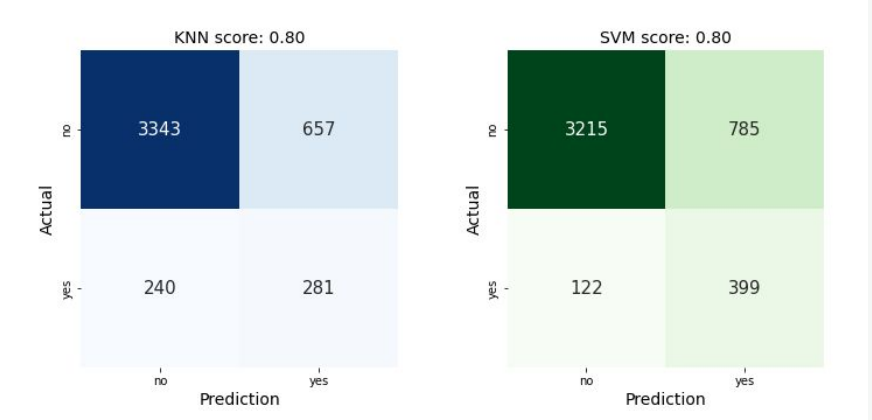
KNN Result

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| no | 0.93 | 0.84 | 0.88 | 4000 |
| yes | 0.30 | 0.54 | 0.39 | 521 |
| accuracy | | | 0.80 | 4521 |
| macro avg | 0.62 | 0.69 | 0.63 | 4521 |
| weighted avg | 0.86 | 0.80 | 0.82 | 4521 |

SVM Result

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| no | 0.96 | 0.80 | 0.88 | 4000 |
| yes | 0.34 | 0.77 | 0.47 | 521 |
| accuracy | | | 0.80 | 4521 |
| macro avg | 0.65 | 0.78 | 0.67 | 4521 |
| weighted avg | 0.89 | 0.80 | 0.83 | 4521 |

## CONFUSION MATRIX



# Simulasi Bisnis

1. Kumpulan semua dataset Klien
2. Terapkan model SVM yang sudah didapat
3. Hasilnya akan berupa prediksi klien mana yang akan membeli produk tersebut atau target market
4. Lalu jalankan marketing campaign yang lebih berfokus pada klien yang sudah ditarget ini

# Conversion Rate

Menggunakan dataset test (4521 data)

Sebelum menggunakan model:

521 / 4521 = 11.52%

Setelah menggunakan model:

399 / 1184 = 33.69%

# THANK YOU