

# Data Trailblazers

---

## Dokumen Laporan Final Project

Bank Marketing



# Data Trailblazers

## Kelompok 7

### Members of Team:

1. Moh. Alamsyah Adi K.
2. Dheanira Nabila M.
3. M Rizki Mardanu H.
4. Rachmansyah Putra H.
5. Farid Arrahman Juliano
6. Ryukazu Andara S
7. Abdurrahman Faras
8. Ari Anjari

# 1. Apa problem yang mau diselesaikan dari dataset tersebut?

Salah satu produk bank adalah deposito berjangka. Salah satu cara bank memasarkan produk tersebut kepada nasabahnya adalah melalui telepon. Metode ini membutuhkan waktu yang lama, namun dinilai paling efektif dibandingkan dengan metode pemasaran lain. Biaya investasi untuk melakukan penawaran melalui telepon tidaklah murah. Semakin besar budget yang dikeluarkan tidak selalu berbanding lurus dengan jumlah nasabah baru yang membuka deposito. Oleh sebab itu dibutuhkan model yang dapat memprediksi nasabah potensial dengan cepat agar tim marketing dapat melakukan penawaran deposito hanya kepada nasabah yang diprediksi akan membuka deposito sehingga akan meminimalisir biaya marketing.



## 2. Sebagai siapa kalian pada dataset tersebut?

DataTrailblazers bertanggung jawab untuk memberikan rekomendasi bisnis berdasarkan problem dan data yang tersedia.

- M Alamsyah Adi K : Data Scientist
- Dheanira Nabila M : Data Scientist
- Rachmansyah Putra H : Data Scientist
- Ryukazu Andara S : Data Analyst
- M Rizki Mardanu H : Data Scientist
- Abdurrahman Faras : Data Analyst
- Ari Anjari : Data Analyst
- Farid Arrahman Juliano : Data Analyst

Data Scientist bertugas melakukan data pre-processing yang meliputi mengecek null value, normality test, normalisasi data, dan feature engineering

Data Analyst bertugas melakukan modelling yang meliputi baseline/ benchmark, metric evaluation, hyper parameter tuning, pembuatan dashboard, dsb

### 3. Apa goal yang mau dicapai?

Biaya marketing berkurang karena bank hanya melakukan promosi kepada nasabah yang diprediksi akan membuka deposito.

### 4. Apa objective yang sesuai dengan goal tersebut?

- Seleksi awal nasabah yang potensial
- Meningkatkan conversion rate nasabah yang membuka deposito

### 5. Apa business metrics yang cocok untuk mengukur ketercapaian objective tersebut?

Biaya marketing.

A teal-tinted background image showing a person's hands typing on a laptop keyboard. A mug is visible on the desk next to the laptop. The image is partially obscured by a white area on the right side of the slide.

# **DESCRIPTIVE STATISTICS**

# Pengecekan Statistik Descriptive Data

```
# melihat informasi data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 17 columns):
#   Column      Non-Null Count  Dtype
---  -
0   age         45211 non-null  int64
1   job         45211 non-null  object
2   marital     45211 non-null  object
3   education   45211 non-null  object
4   default     45211 non-null  object
5   balance     45211 non-null  int64
6   housing     45211 non-null  object
7   loan        45211 non-null  object
8   contact     45211 non-null  object
9   day         45211 non-null  int64
10  month       45211 non-null  object
11  duration    45211 non-null  int64
12  campaign    45211 non-null  int64
13  pdays       45211 non-null  int64
14  previous    45211 non-null  int64
15  poutcome    45211 non-null  object
16  y           45211 non-null  object
dtypes: int64(7), object(10)
memory usage: 5.9+ MB
```

- **Tidak ada** kolom dengan tipe data kurang sesuai, atau nama kolom dan isinya kurang sesuai.
- **Tidak ada** kolom yang memiliki nilai kosong.



# Pengecekan Statistik Descriptive Data

```
df.describe()
```

	age	balance	day	duration	campaign	pdays	previous
count	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000
mean	40.936210	1362.272058	15.806419	258.163080	2.763841	40.197828	0.580323
std	10.618762	3044.765829	8.322476	257.527812	3.098021	100.128746	2.303441
min	18.000000	-8019.000000	1.000000	0.000000	1.000000	-1.000000	0.000000
25%	33.000000	72.000000	8.000000	103.000000	1.000000	-1.000000	0.000000
50%	39.000000	448.000000	16.000000	180.000000	2.000000	-1.000000	0.000000
75%	48.000000	1428.000000	21.000000	319.000000	3.000000	-1.000000	0.000000
max	95.000000	102127.000000	31.000000	4918.000000	63.000000	871.000000	275.000000

- Terdapat nilai mean dan median yang berbeda cukup signifikan, diantaranya kolom **balance**, **duration**, dan **pdays** sehingga terdapat dugaan adanya outliers.
- Terdapat nilai min dan median yang cukup aneh untuk kolom **pdays**, di mana min dan median bernilai -1. Setelah dilakukan pemeriksaan terhadap deskripsi kolom, untuk nasabah yang memiliki **pdays** bernilai -1 merupakan nasabah yang belum pernah dihubungi oleh call center terkait telemarketing pada campaign sebelumnya.
- Terdapat nilai min yang cukup aneh untuk kolom **balance**, yaitu nilai min yang merupakan bilangan negatif. Kolom **balance** merupakan saldo rata-rata tahunan di mana secara logika hal ini cukup aneh apabila seorang nasabah memiliki saldo rata-rata tahunan yang bernilai negatif. Perlu dipastikan lagi apakah ini merupakan kesalahan pada input data, atau terdapat beberapa nasabah yang memanfaatkan fasilitas kredit.

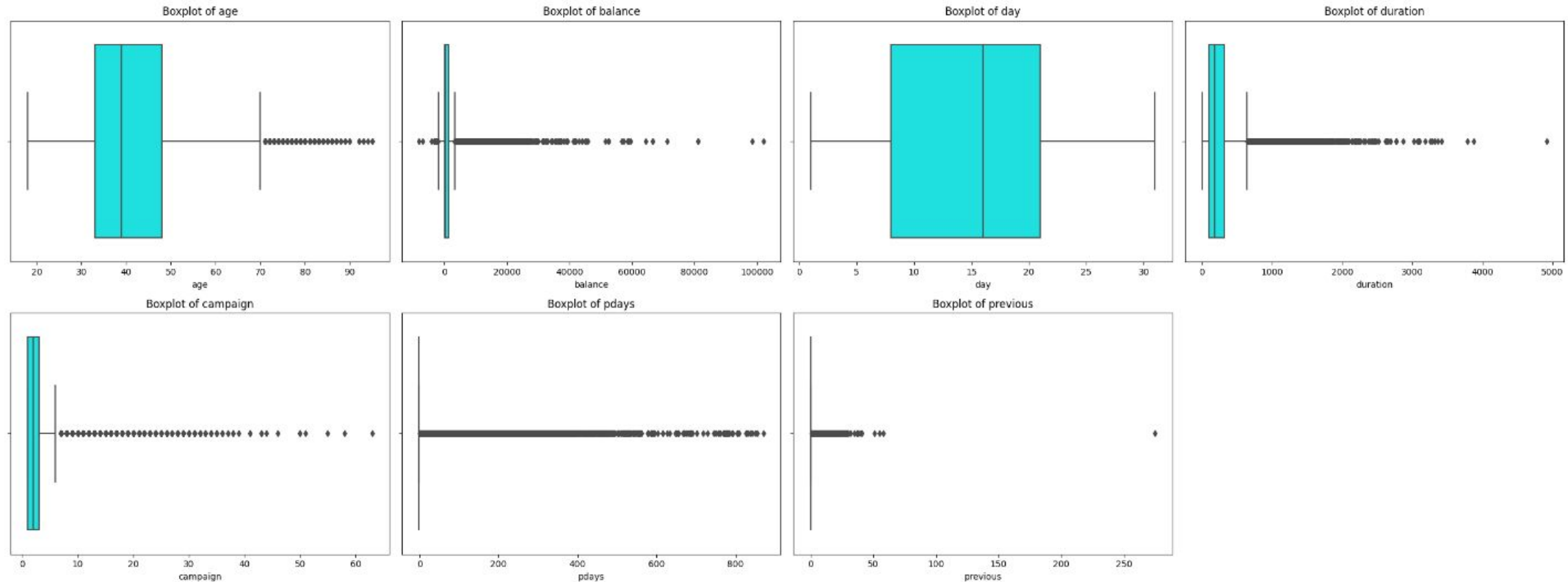


A teal-tinted background image showing a person's hands typing on a laptop keyboard. A mug is visible on the desk next to the laptop. The image is partially obscured by a white area on the right side of the slide.

# UNIVARIATE ANALYSIS

# Univariate Analysis Features

Boxplot for Numerical Features



# Univariate Analysis

Dari hasil visualisasi distribution plot dan boxplot, dapat diketahui bahwa:

- Hampir seluruh kolom numerical memiliki jenis distribusi positively skewed atau right-skewed, kecuali untuk distribusi untuk kolom **day** di mana kolom tersebut memiliki jenis distribusi multimodal.
- Terlihat bahwa kolom **balance**, **duration**, **campaign**, **pdays** dan **previous** memiliki persebaran data yang cukup luas.
- Terdapat banyak value yang berpotensi sebagai outliers pada seluruh kolom numerical, kecuali kolom **day** yang terlihat tidak memiliki outliers.
- Terdapat beberapa value yang berpotensi menjadi outliers ekstrim pada kolom **balance**, **duration** dan **previous**.
- Kolom **balance**, **duration**, **campaign** didominasi oleh nilai yang kecil. Variasi data yang banyak pada nilai tinggi membuat outlier menjadi semakin banyak.
- Kolom **pdays** memiliki nilai -1 (belum dihubungi sebelumnya) yang mendominasi yaitu sebanyak 36954 dari 45211 baris atau sekitar 81.7% data sehingga kotak pada boxplot menjadi berpusat pada nilai -1 dan nilai-nilai diatas -1 menjadi outlier dengan jumlah cukup banyak yaitu sekitar 18.3%.
- Kolom **previous** memiliki nilai 0 yang mendominasi yang sesuai dengan nilai -1 pada kolom **pdays** karena nilai ini memiliki korelasi yang kuat. Hal ini menyebabkan value selain 0 akan menjadi outlier. Namun ada satu nilai yang benar-benar sangat jauh yaitu 275 (outlier ini dapat dihapus karena hanya satu saja dan jaraknya sangat jauh).

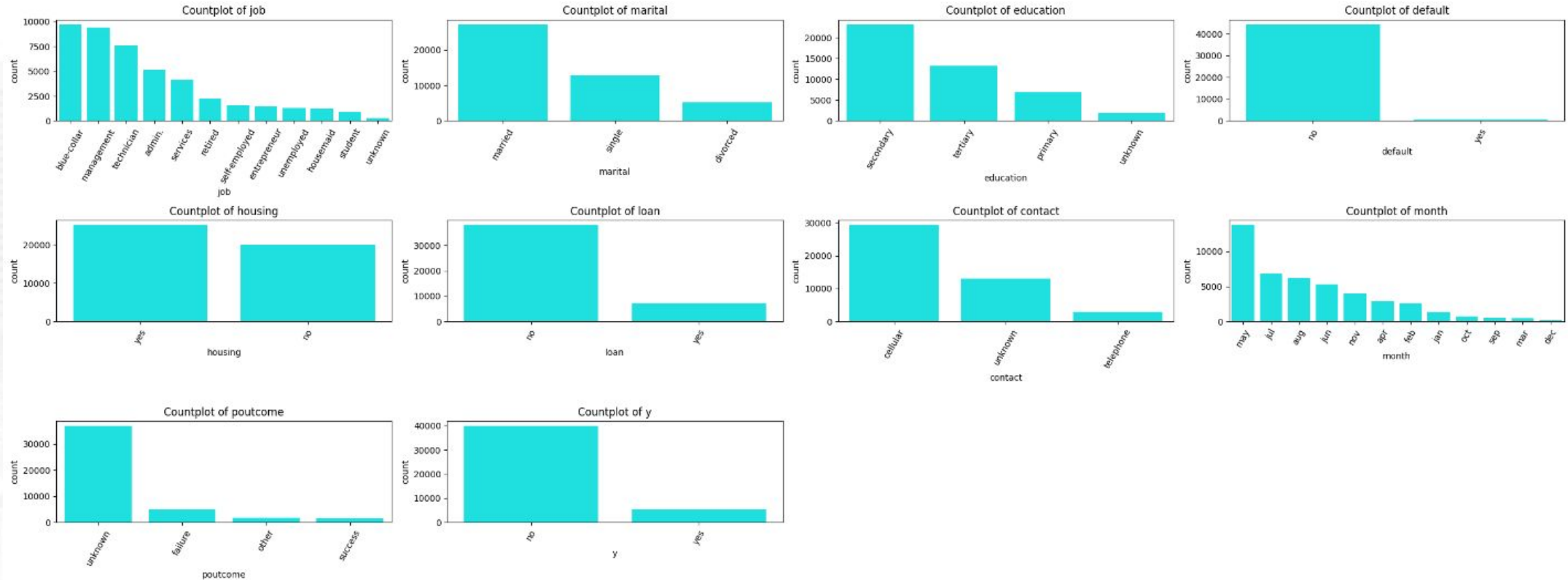




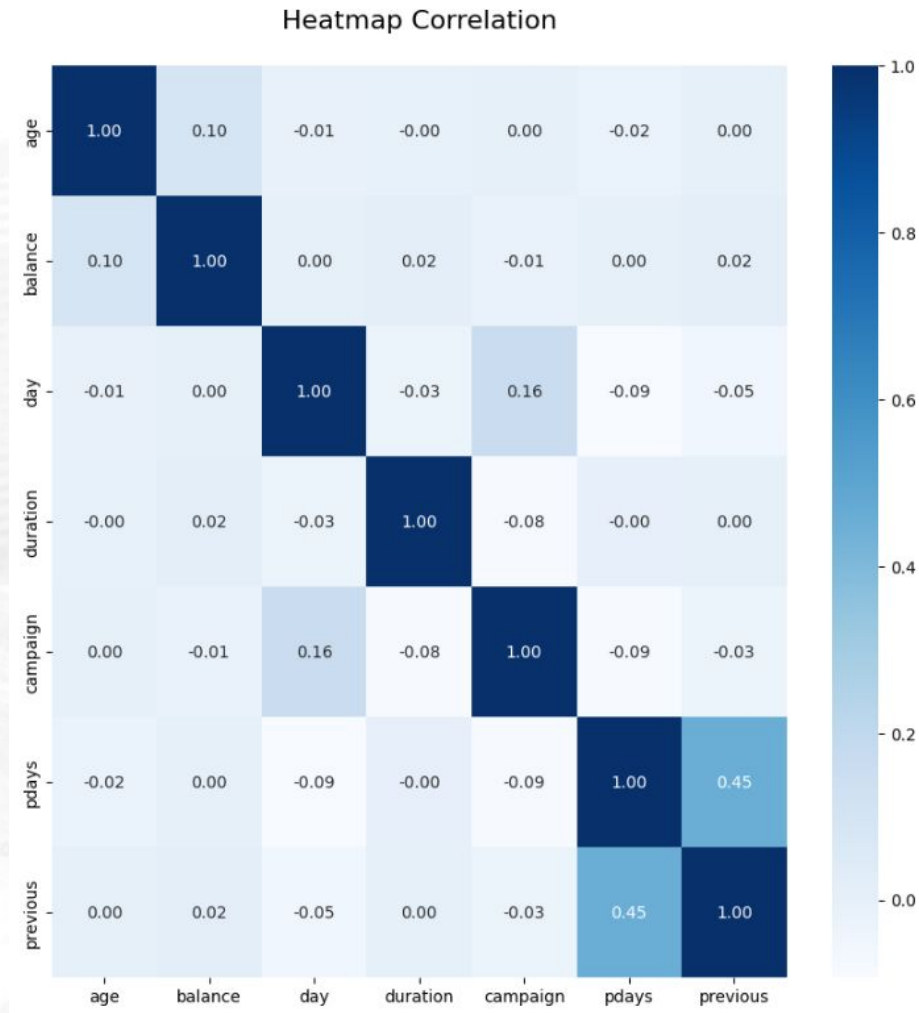
# MULTIVARIATE ANALYSIS

# Categorical Features

Countplot for Categorical Features



# Heatmap



- Korelasi antara **pdays** dan **previous** juga tinggi namun tidak cukup tinggi hingga bisa dihapus salah satunya.
- Tidak ada kolom numerical yang berkorelasi kuat satu dengan lainnya.

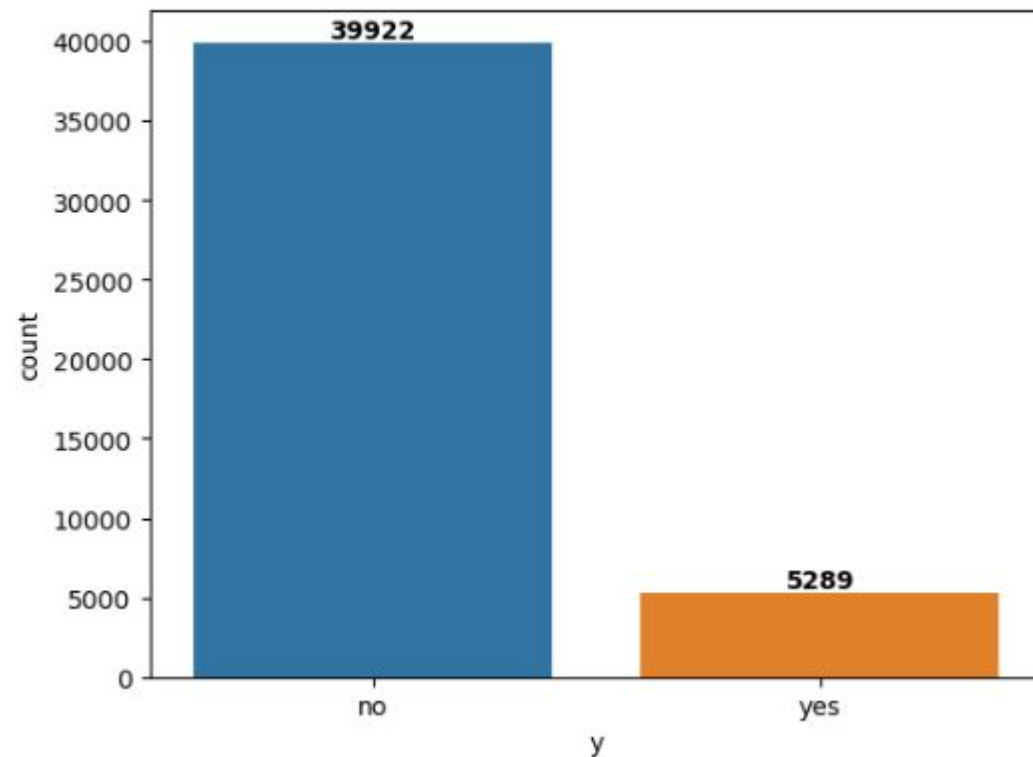




# BUSINESS INSIGHT

# Insights

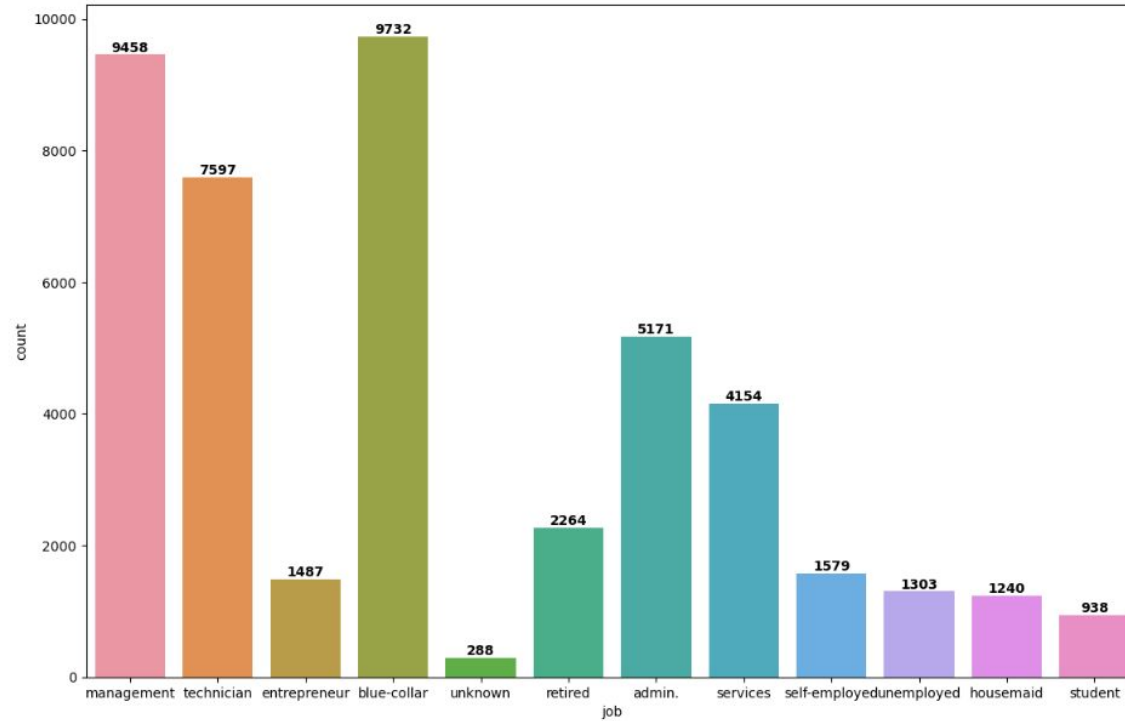
```
ax = sns.countplot(x='y', data=df)
for label in ax.containers :
    ax.bar_label(label, fontweight = 'bold')
```



Terjadi ketimpangan antara orang yang melakukan deposit dan yang tidak, terlihat dari plot diatas jumlah yang melakukan deposit hanya 5289 customer sedangkan **yang belum melakukan deposit ada 39922 customer atau sebanyak 88%**. Hal ini tentu menjadi sebuah pekerjaan untuk tim marketing untuk mengkonversi customer yang belum deposit karena masih ada potensi yang besar.

# Insights

```
plt.figure(figsize=(14,9))
ab = sns.countplot(x='job', data=df)
for label in ab.containers:
    ab.bar_label(label, fontweight='bold')
```

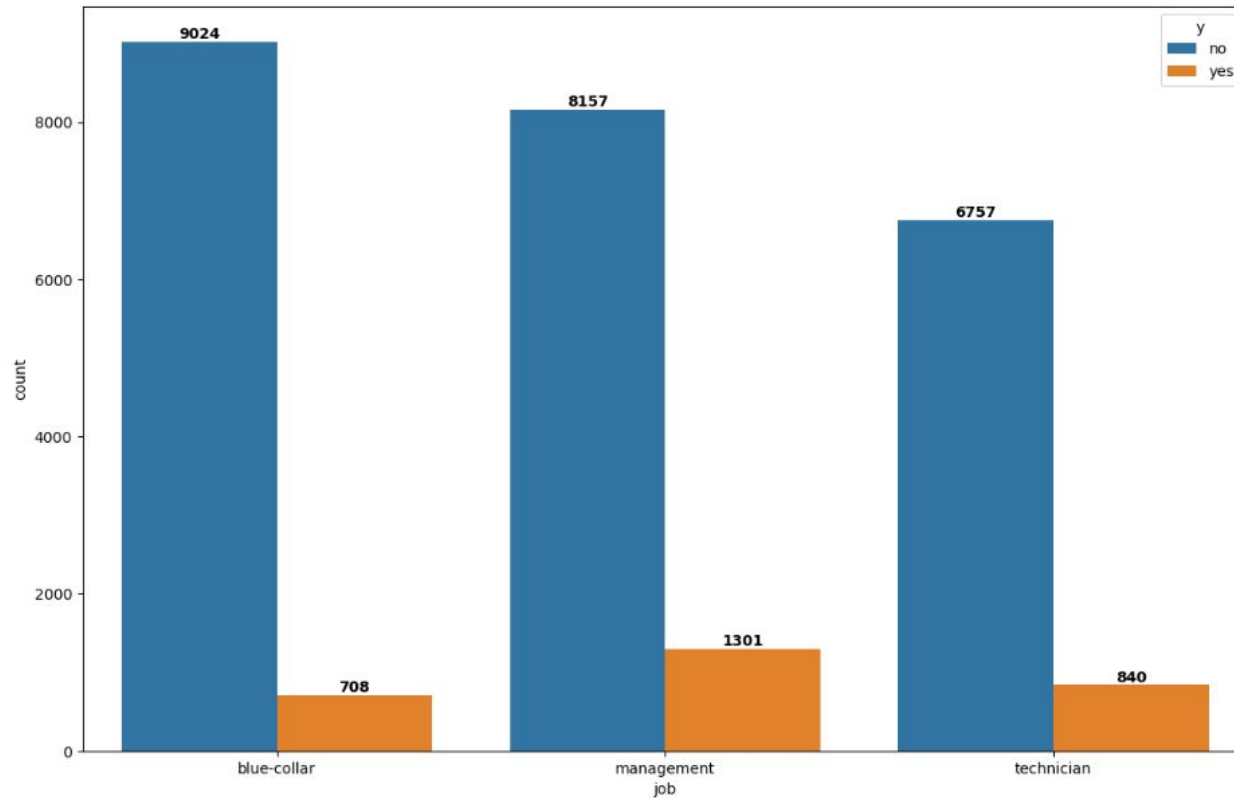


Seperti management, technician dan blue collar menjadi pekerjaan terbanyak dari customer bank.



# Insights

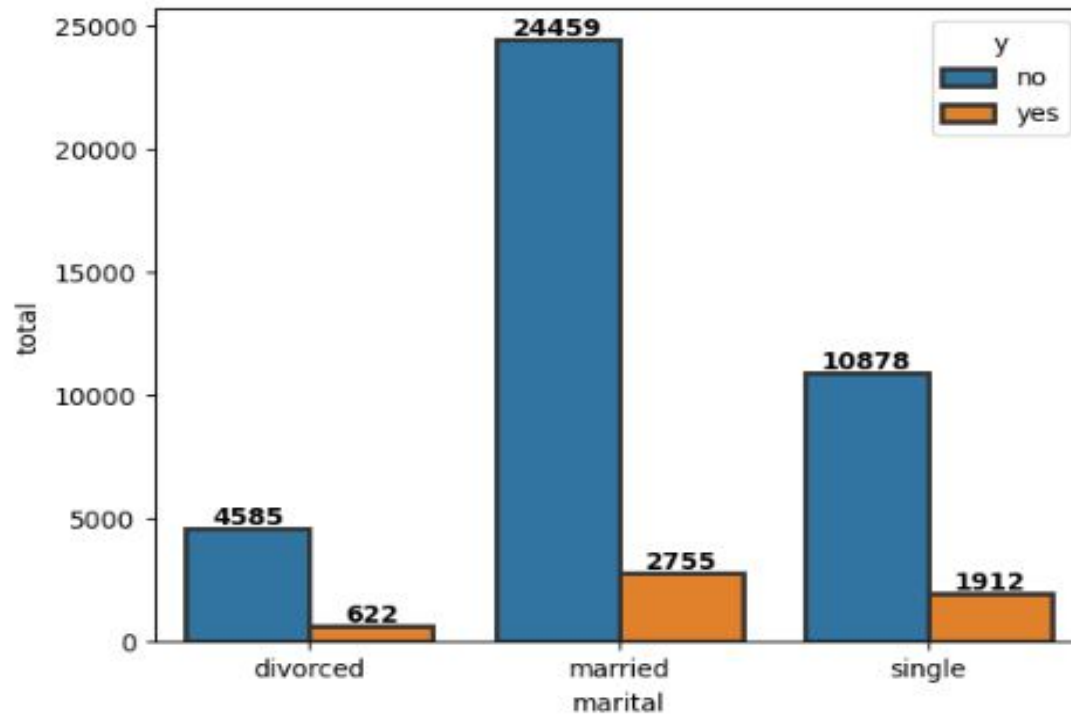
```
plt.figure(figsize=(14,9))
ac = sns.countplot(x='job', hue='y', data=df,
order=df.job.value_counts().iloc[:3].index)
for label in ac.containers :
    ac.bar_label(label, fontweight = 'bold')
```



Terjadi ketimpangan customer yang bekerja sebagai blue-collar, management dan technician di mana banyak dari mereka belum melakukan deposit.

# Insights

```
marital_chart = sns.barplot( x = 'marital',
                             y = 'total',
                             hue = 'y',
                             data = marital_count,
                             edgecolor = '0.2',
                             linewidth = 2
                             )
for label in marital_chart.containers :
    marital_chart.bar_label(label, fontweight = 'bold')
```

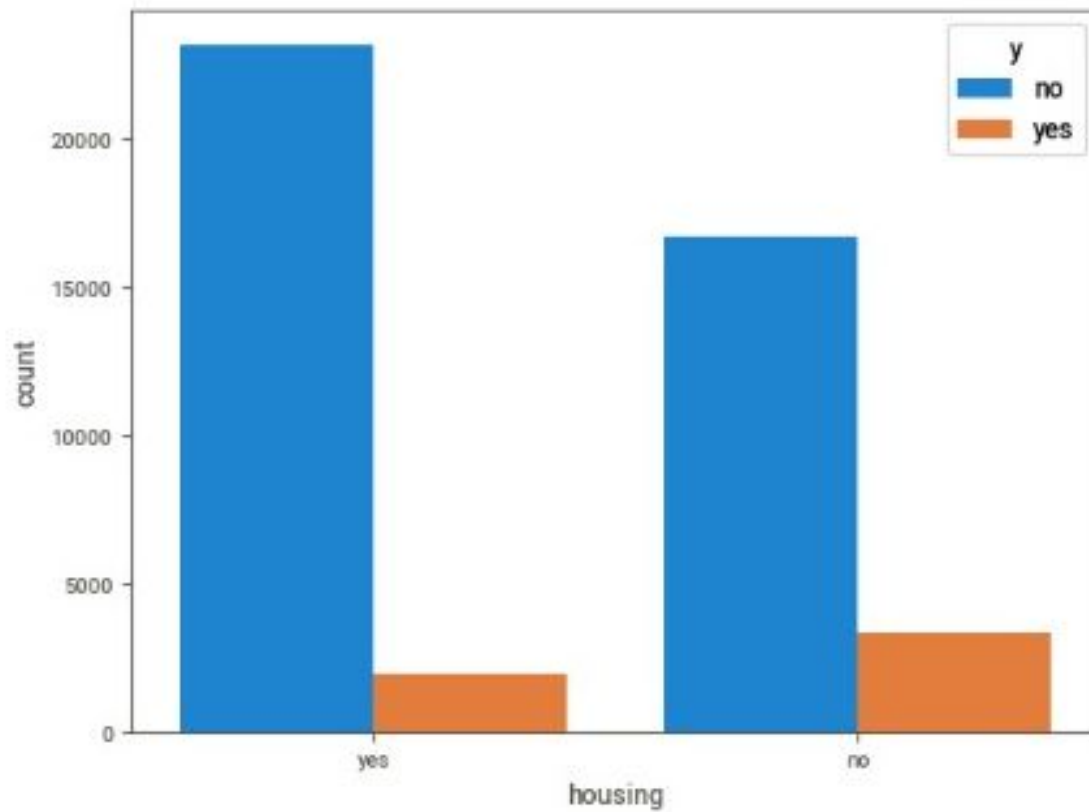


Customer berstatus married mendominasi data dengan jumlah terbanyak dibandingkan dengan customer berstatus divorce atau single. Akan tetapi jumlah customer dengan status married yang sudah membuka deposit terbilang cukup sedikit yaitu hanya sebanyak 10% dari jumlah customer berstatus married.

# Insights

```
sns.countplot(data=df, x='housing', hue='y')
```

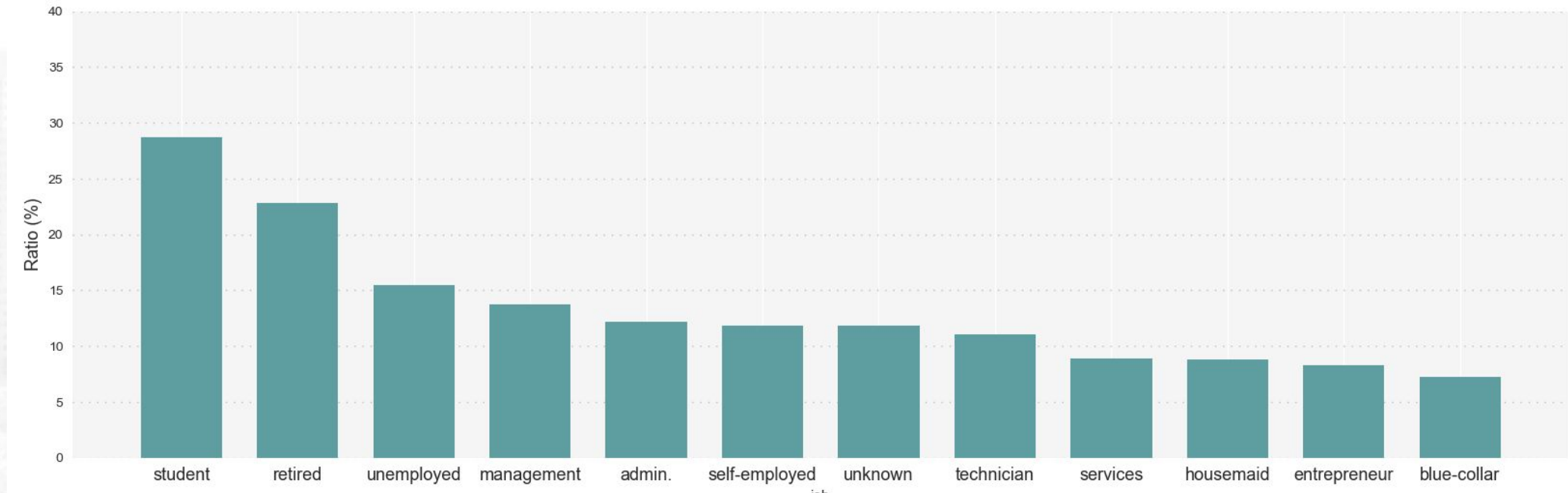
```
<AxesSubplot:xlabel='housing', ylabel='count'>
```



Pengguna yang mempunyai cicilan rumah memiliki kemungkinan yang rendah untuk deposit di bank, sedangkan yang tidak memiliki cicilan rumah memiliki kemungkinan lebih besar untuk deposit di bank.

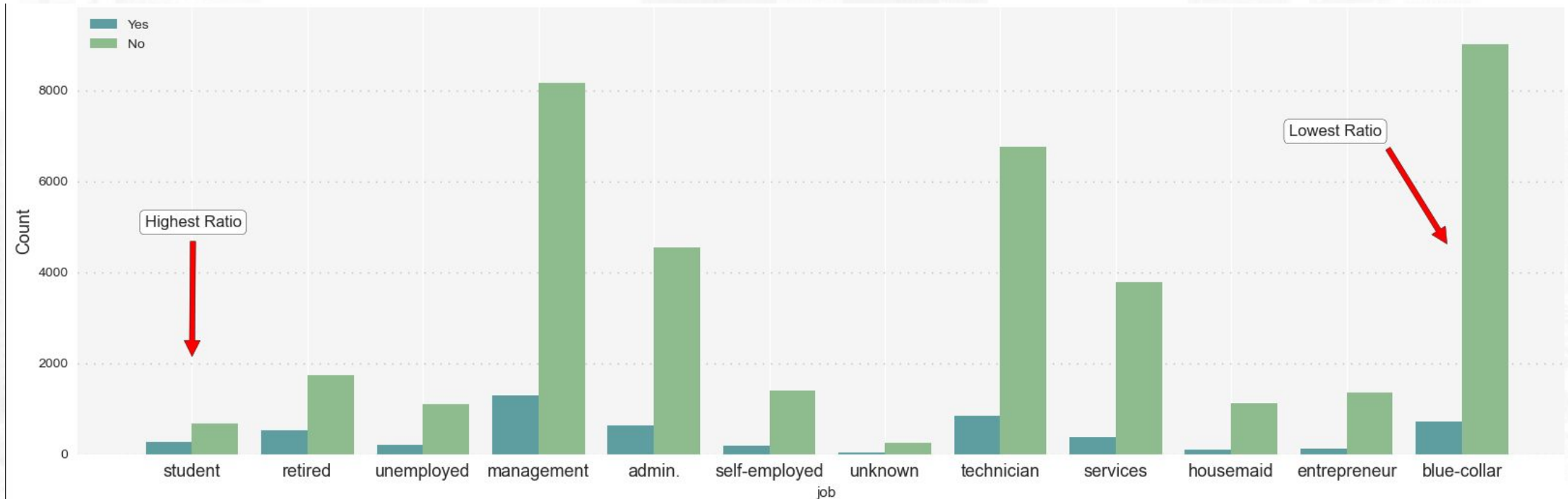


# Insights



Nasabah yang berstatus sebagai pelajar memiliki rasio tertinggi dalam membuka rekening deposito berjangka, dengan rasio sebesar 28,68%

# Insights

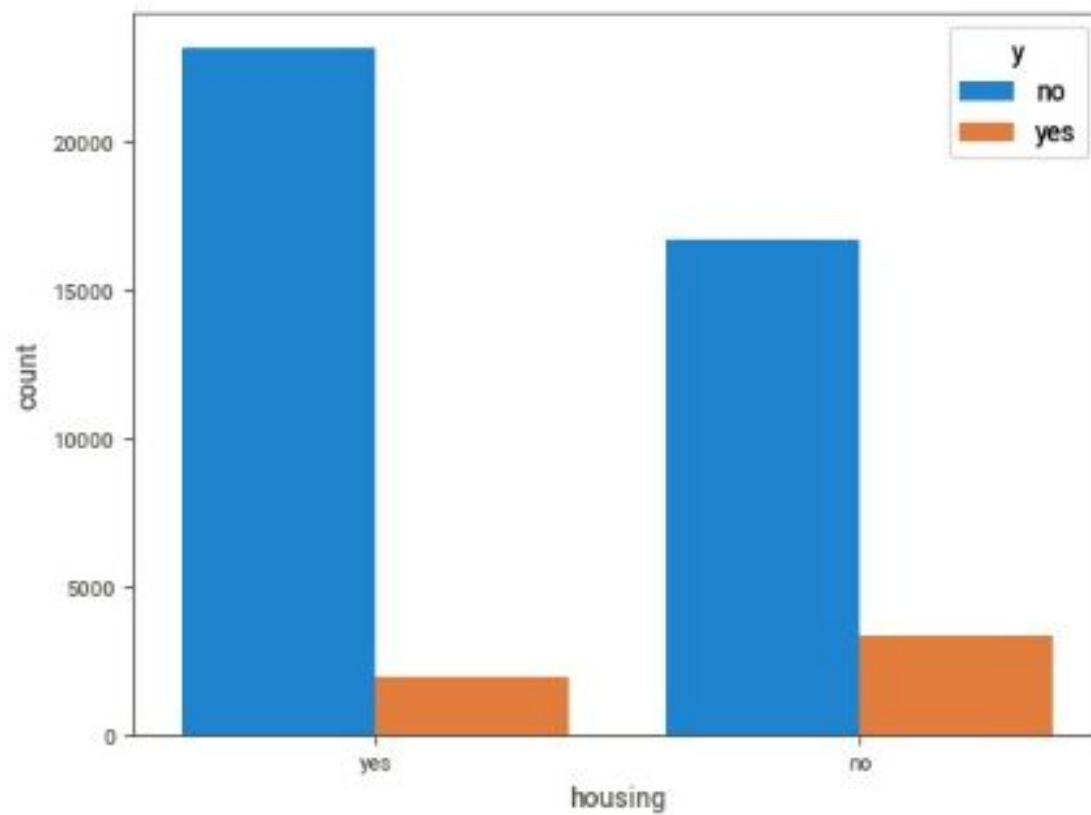


Jenis pekerjaan nasabah yang paling banyak dihubungi ialah blue-collar, merupakan jenis pekerjaan dengan rasio pembukaan deposito terendah. Sedangkan jenis pekerjaan nasabah yang sedikit dihubungi student, justru memiliki rasio pembukaan deposito tertinggi. Oleh karena itu, pelajar dapat dijadikan target campaign selanjutnya yang utama karena memiliki rasio penerimaan pembukaan deposito berjangka yang tinggi.

# Business Recommendation

```
sns.countplot(data=df, x='housing', hue='y')
```

```
<AxesSubplot:xlabel='housing', ylabel='count'>
```

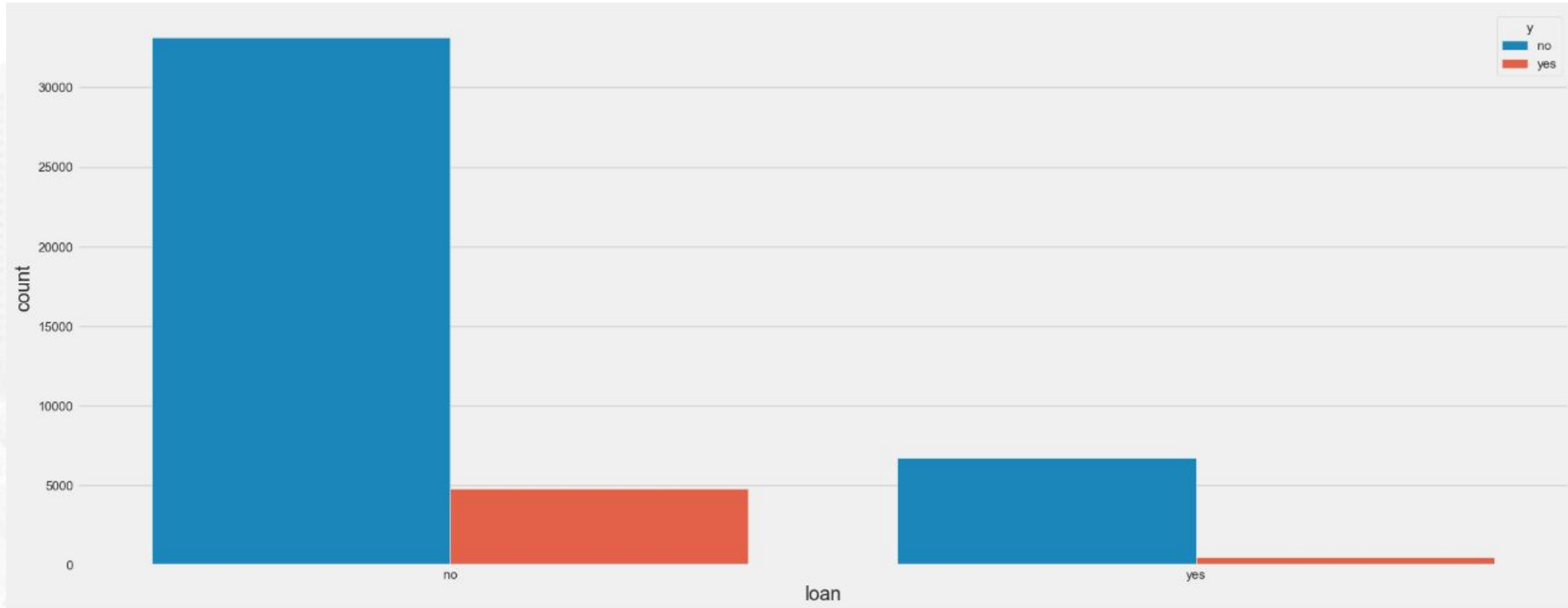


Produk deposit ditawarkan kepada pengguna yang tidak memiliki cicilan rumah terlebih dahulu.

# Business Recommendation

```
sns.countplot(data = df, x = 'loan', hue = 'y')
```

<Axes: xlabel='loan', ylabel='count'>



Karena berdasarkan grafik yang membuka deposit adalah yang tidak mempunyai hutang, maka kami merekomendasikan agar bank memberikan penawaran deposit terhadap konsumen yang tidak mempunyai hutang.



# DATA CLEANSING

# Handle Missing Value

```
# melihat informasi data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 17 columns):
#   Column      Non-Null Count  Dtype
---  -
0   age         45211 non-null  int64
1   job         45211 non-null  object
2   marital     45211 non-null  object
3   education   45211 non-null  object
4   default     45211 non-null  object
5   balance     45211 non-null  int64
6   housing     45211 non-null  object
7   loan        45211 non-null  object
8   contact     45211 non-null  object
9   day         45211 non-null  int64
10  month       45211 non-null  object
11  duration    45211 non-null  int64
12  campaign    45211 non-null  int64
13  pdays      45211 non-null  int64
14  previous    45211 non-null  int64
15  poutcome    45211 non-null  object
16  y           45211 non-null  object
dtypes: int64(7), object(10)
memory usage: 5.9+ MB
```

- Missing value tidak perlu di-handle karena **tidak ada kolom yang memiliki nilai kosong**.

# Handle Duplicate Data

```
# melihat jumlah data duplikat  
df.duplicated().sum()
```

```
0
```

- Data duplikat tidak perlu di-handle karena **tidak ada data yang terduplikat**.

# Handle Outliers

```
#melakukan filter outliers
filtered_entries = np.array([True] * len(df))
for kol in ['balance', 'campaign', 'pdays', 'previous']:
    zscore = abs(stats.zscore(df[kol]))
    filtered_entries = (zscore < 3) & filtered_entries

df1 = df[filtered_entries]

print(f'Jumlah baris sebelum memfilter outlier: {len(df)}')
print(f'Jumlah baris setelah memfilter outlier: {len(df1)}')
```

Jumlah baris sebelum memfilter outlier: 45211  
 Jumlah baris setelah memfilter outlier: 41446

- Melakukan handling outlier **dengan Z-score**.
- Jumlah data setelah di-handling outlier menjadi 41446.



# Feature Transformation

```
df.describe()
```

	age	balance	day	duration	campaign	pdays	previous
count	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000
mean	40.936210	1362.272058	15.806419	258.163080	2.763841	40.197828	0.580323
std	10.618762	3044.765829	8.322476	257.527812	3.098021	100.128746	2.303441
min	18.000000	-8019.000000	1.000000	0.000000	1.000000	-1.000000	0.000000
25%	33.000000	72.000000	8.000000	103.000000	1.000000	-1.000000	0.000000
50%	39.000000	448.000000	16.000000	180.000000	2.000000	-1.000000	0.000000
75%	48.000000	1428.000000	21.000000	319.000000	3.000000	-1.000000	0.000000
max	95.000000	102127.000000	31.000000	4918.000000	63.000000	871.000000	275.000000

```
# mengubah nilai minus pada kolom balance menjadi 0
```

```
df.loc[df['balance'] < 0, 'balance'] = 0
df['balance'].min()
```

```
0
```

- Pada gambar di samping dapat dilihat bahwa kolom *balance* memiliki min value **-8019** , sementara itu kolom *balance* berisi data saldo tahunan rata-rata (dalam euro). Sehingga akan aneh ketika saldo tahunan rata-rata nasabah adalah negatif.
- Value kolom *balance* yang negatif akan di-handle dengan **diubah menjadi value 0**.

# Feature Encoding

```
for cat in ['job_filter']:
    onehots = pd.get_dummies(df[cat], prefix=cat)
    df = df.join(onehots)
```

outcome	y	job_filter	job_filter_admin.	job_filter_blue-collar	job_filter_management	job_filter_others	job_filter_services	job_filter_technician
unknown	no	management	0	0	1	0	0	0
unknown	no	technician	0	0	0	0	0	1
unknown	no	others	0	0	0	1	0	0
unknown	no	blue-collar	0	1	0	0	0	0
unknown	no	others	0	0	0	1	0	0
...	...	...	...	...	...	...	...	...
unknown	yes	technician	0	0	0	0	0	1
unknown	yes	others	0	0	0	1	0	0
success	yes	others	0	0	0	1	0	0
unknown	no	blue-collar	0	1	0	0	0	0
other	no	others	0	0	0	1	0	0

- Dilakukan OHE pada kolom job

# Handle Class Imbalance

```
# melihat jumlah nasabah yang menerima tawaran dan menolak  
df['y'].value_counts(normalize=True)*100
```

```
no      88.30152  
yes     11.69848  
Name: y, dtype: float64
```

- Jumlah nasabah yang menerima (yes) dan menolak (no) berbeda jauh sehingga **data tidak seimbang** (imbalance).



# FEATURE ENGINEERING



# Feature Selection

```
columns_to_drop = ['day', 'duration']  
df_dropped = df.drop(columns=columns_to_drop)
```

- **Drop column 'day'**  
Hal ini dikarenakan column ini tidak terlalu relevan, dan lebih terwakilkan oleh column 'month'.
- **Drop column 'duration'**  
Hal ini dikarenakan 'duration' pada dataset hanya sebagai historis lama waktu telpon, sehingga tidak berpengaruh dengan campaign yang sekarang.

# Feature Extraction

- **Tidak ada feature baru yang ditambahkan**, dikarenakan feature yang sudah ada sudah cukup mewakili.

# Feature Tambahan



- **Investment**

informasi jenis investasi yang dimiliki oleh nasabah.

(categorical: "stocks", "bonds", "property", "precious metal", "cryptocurrency", "others", "no";

note: "no" berarti belum memiliki investasi)



- **Expenditure**

informasi pengeluaran bulanan rata-rata.

(numeric)



- **Dependent**

informasi jumlah tanggungan nasabah.

(numeric, 0 berarti tidak memiliki tanggungan)



- **Risk tolerance**

informasi seberapa besar toleransi risiko nasabah.

(categorical: conservative, moderate, aggressive)



# Feature Tambahan

- **Investment**

informasi jenis investasi yang dimiliki oleh nasabah.  
(categorical: "stocks", "bonds", "property", "precious metal",  
"cryptocurrency", "others", "no";  
note: "no" berarti belum memiliki investasi)

- **Expenditure**

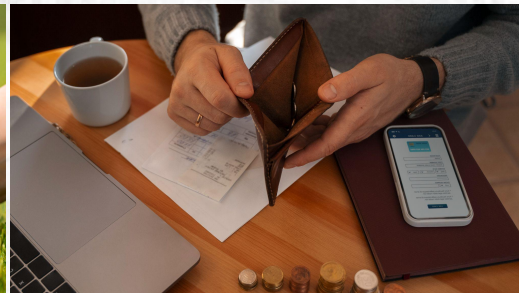
informasi pengeluaran bulanan rata-rata.  
(numeric)

- **Dependent**

informasi jumlah tanggungan nasabah.  
(numeric, 0 berarti tidak memiliki tanggungan)

- **Risk tolerance**

informasi seberapa besar toleransi risiko nasabah.  
(categorical: conservative, moderate, aggressive)





# MODELING



# Split Data Test & Train

```
#Split data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

```
y_test.value_counts()
```

```
0    11144
1     1503
Name: y, dtype: int64
```

```
y_train.value_counts()
```

```
0    26153
1     3355
Name: y, dtype: int64
```

- Membagi data menjadi *test* dan *train* dengan `train_test_split` dan `random_state = 42`.

# Modeling

```
# Algoritma Machine Learning
logreg = LogisticRegression()
dt = DecisionTreeClassifier()
rf = RandomForestClassifier(random_state=42)
clf = AdaBoostClassifier()
xg = XGBClassifier()
lgbm = LGBMClassifier(random_state=42, class_weight='balanced')
knn = KNeighborsClassifier()
nb = GaussianNB()
MLA = [logreg, dt, rf, knn, clf, xg, lgbm, nb]
```

- Dalam pemodelan digunakan 8 algoritma machine learning, yaitu Logistic Regression, Decision Tree, Random Forest, Ada Boost, XGB (eXtreme Gradient Boosting), LGBM (Light Gradient Boosting Machine), K-Nearest Neighbours, dan Gaussian Naive Bayes.

# Model Evaluation

	MLA used	Train Accuracy (%)	Test Accuracy (%)	Train Precision (%)	Test Precision (%)	Train Recall (%)	Test Recall (%)	CV recall (recall train)	CV recall (recall test)	Train F1-Score (%)	Train F1-Score (%)	roc-auc (test prob)	roc-auc (train prob)	Test F1-Score (%)
0	LogisticRegression	90.413	82.887	98.325	33.957	82.227	42.211	0.823	0.815	89.558	89.558	0.71	0.947	37.637
1	DecisionTreeClassifier	99.963	33.626	99.985	13.09	99.942	78.474	1.0	0.879	99.963	99.963	0.529	1.0	22.438
2	RandomForestClassifier	99.963	68.857	99.973	21.324	99.954	57.466	1.0	0.88	99.963	99.963	0.694	1.0	31.106
3	KNeighborsClassifier	92.62	75.998	93.523	23.102	91.582	41.306	0.917	0.888	92.542	92.542	0.663	0.982	29.631
4	AdaBoostClassifier	90.872	36.892	94.723	14.486	86.567	84.809	0.87	0.85	90.462	90.462	0.678	0.952	24.745
5	XGBClassifier	94.543	17.683	98.289	12.771	90.664	98.255	0.913	0.863	94.323	94.323	0.705	0.983	22.604
6	LGBMClassifier	93.386	23.385	97.758	13.339	88.808	95.734	0.895	0.858	93.068	93.068	0.678	0.975	23.415
7	GaussianNB	77.777	61.684	73.11	17.282	87.875	56.303	0.877	0.87	79.815	79.815	0.626	0.885	26.446

- Berdasarkan perhitungan *metrics model* dari beberapa algoritma di atas, **digunakan metrik berupa recall** dikarenakan metrik tersebut best-fit dibandingkan metrik lain yang cenderung *overfitting*. Selain itu, recall dipilih karena kami ingin data prediksi dan data asli sejalan yang menghasilkan nilai *True Positive* yang besar, di mana akan menekan *cost* kesalahan ketika melakukan *campaign*.



# Model Evaluation

	MLA used	Train Accuracy (%)	Test Accuracy (%)	Train Precision (%)	Test Precision (%)	Train Recall (%)	Test Recall (%)	CV recall (recall train)	CV recall (recall test)	Train F1-Score (%)	Train F1-Score (%)	roc-auc (test prob)	roc-auc (train prob)	Test F1-Score (%)
0	LogisticRegression	90.413	82.887	98.325	33.957	82.227	42.211	0.823	0.815	89.558	89.558	0.71	0.947	37.637
1	DecisionTreeClassifier	99.963	33.626	99.985	13.09	99.942	78.474	1.0	0.879	99.963	99.963	0.529	1.0	22.438
2	RandomForestClassifier	99.963	68.857	99.973	21.324	99.954	57.466	1.0	0.88	99.963	99.963	0.694	1.0	31.106
3	KNeighborsClassifier	92.62	75.998	93.523	23.102	91.582	41.306	0.917	0.888	92.542	92.542	0.663	0.982	29.631
4	AdaBoostClassifier	90.872	36.892	94.723	14.486	86.567	84.809	0.87	0.85	90.462	90.462	0.678	0.952	24.745
5	XGBClassifier	94.543	17.683	98.289	12.771	90.664	98.255	0.913	0.863	94.323	94.323	0.705	0.983	22.604
6	LGBMClassifier	93.386	23.385	97.758	13.339	88.808	95.734	0.895	0.858	93.068	93.068	0.678	0.975	23.415
7	GaussianNB	77.777	61.684	73.11	17.282	87.875	56.303	0.877	0.87	79.815	79.815	0.626	0.885	26.446

- Berdasarkan percobaan dari beberapa algoritma yang kami lakukan, **AdaBoostClassifier merupakan algoritma terbaik di *dataset* kami**. Cross-validation (CV) di algoritma tersebut sudah *best-fit*, di mana hasil CV pada data *train* didapatkan 87%, sedangkan di data *test* sebesar 85%, sehingga pada kasus ini, kami menggunakan algoritma AdaBoostClassifier.



# Hyperparameter Tuning

```
from sklearn.model_selection import GridSearchCV

grid = dict()
grid['n_estimators'] = [10,50,100,500]
grid['learning_rate'] = [0.0001, 0.001, 0.01, 0.1, 1.0]
grid['algorithm'] = ['SAMME', 'SAMME.R']

grid_search = GridSearchCV(estimator=AdaBoostClassifier(), param_grid=grid, n_jobs=-1, cv=5, scoring='recall')

grid_result = grid_search.fit(X_train_std, ytrain_smt)

print("Best : %f using %s" % (grid_result.best_score_, grid_result.best_params_))

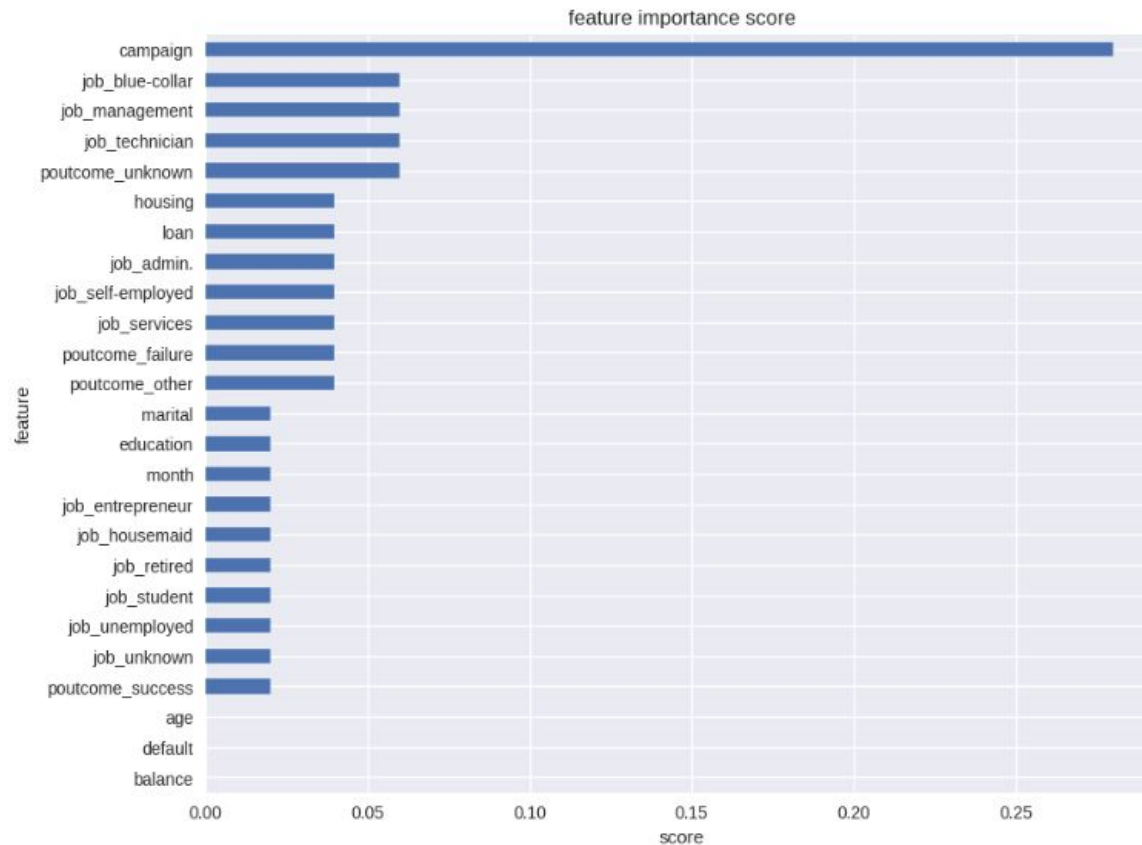
Best : 0.882912 using {'algorithm': 'SAMME', 'learning_rate': 0.01, 'n_estimators': 500}
```

- **Score score *recall* terbaik adalah 0.89**, di mana didapat dengan algoritma SAMME, learning rate = 0.01 dan n\_estimator = 500.



# FEATURE IMPORTANCE

# Feature Importance



- Berdasarkan hasil yang didapatkan dari visualisasi *feature importance*, **kolom *campaign* merupakan *feature* yang paling berpengaruh terhadap target**, yaitu untuk membuka deposito. Yang berarti peluang *client* membuka deposito sejalan dengan semakin seringnya kontak yang dilakukan oleh pihak bank.
- Hal ini sejalan dengan objective dari model yang dibuat di mana bertujuan untuk memprediksi nasabah yang akan membuka deposito sehingga *campaign* hanya dilakukan kepada nasabah tersebut.
- Dari *business insight* di atas, bisa dilakukan *action* berupa **mengurangi jumlah kontak terhadap nasabah yang memiliki nilai *campaign* rendah**.

# Referensi

- <https://www.semanticscholar.org/paper/Human-Age-Group-Classification-Using-Facial-Bhat-V.K.Patil/19ddb412336ce633c1fe21544605c7bd65ff8d66>
- <https://ojk.go.id/id/kanal/perbankan/pages/suku-bunga-dasar.aspx>
- <https://www.cnbcindonesia.com/mymoney/20230614154614-72-445896/cek-bunga-deposito-hari-ini-ada-yang-5-lebih>