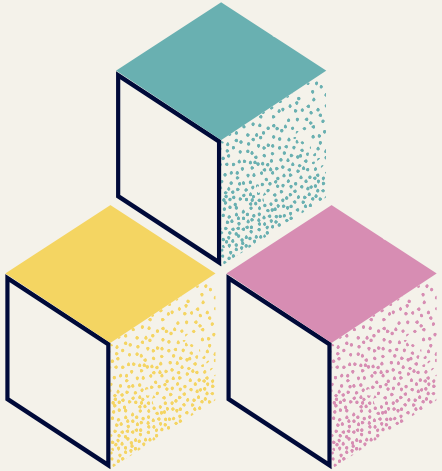


TELECOM

Predict Customer Churn/Non Churn

Septa Dwi Cahya

https://colab.research.google.com/drive/1Yk4-5dDhE5FR2OG-cTb6LKxkWiTYZRZF

OUTLINE

01

Data Overview

Overview tentang dataset yang digunakan

02

EDA

Eksplorasi variabel pada dataset

03

Data Preprocessing

Penanganan/pemrosesan dataset sebelum dilakukan modelling

04

Modelling

Pemodelan dataset dengan metode machine learning

05

Predict

Melakukan prediksi pada "data test" untuk klasifikasi customer churn atau non-churn

06

Conclusion

Kesimpulan dari pemodelan dan prediksi

01

Data Overview

Data Overview

Dataset berisi:

- 4250 sample
- 19 variabel penjelas
- 1 variabel respon (yaitu "churn" yang menunjukkan kelas dari sampel apakah termasuk customer churn atau tidak)

- ➔ Dataset ini akan digunakan untuk membuat model machine learning
- ➔ Kemudian memilih model machine learning terbaik
- ➔ Model terbaik akan digunakan untuk memprediksi data baru.

Visit this link for full syntax Python:
https://colab.research.google.com/drive/1Yk4-5dDhE5FR2OG-cTb6LKxkWiTYZRZF

Data Overview

Variabel Penjelas	Keterangan
state	Kode 2 huruf negara tempat tinggal customer AS
account_length	Jumlah bulan customer telah bersama penyedia telekomunikasi saat ini
area_code	3 digit area code
international_plan	customer memiliki paket international
voice_mail_plan	customer memiliki paket voice mail
number_mail_messages	Jumlah pesan dengan voice mail
total_day_minutes	Total menit panggilan hari
total_day_calls	Jumlah total panggilan hari
total_day_charge	Total biaya panggilan hari
total_eve_minutes	Total menit panggilan malam
total_eve_minutes	Jumlah total panggilan malam
total_eve_charge	Total biaya panggilan malam
total_night_minutes	Total menit panggilan malam
total_night_calls	Jumlah total panggilan malam
total_night_charge	Total biaya panggilan malam
total_intl_minutes	Total menit panggilan international
total_intl_calls	Jumlah total panggilan international
total_intl_charge	total biaya panggilan international
number customer_service_calls	Jumlah panggilan ke customer service

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4250 entries, 0 to 4249
Data columns (total 20 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   state                           4250 non-null   object 
 1   account_length                  4250 non-null   int64  
 2   area_code                       4250 non-null   object 
 3   international_plan              4250 non-null   object 
 4   voice_mail_plan                 4250 non-null   object 
 5   number_vmail_messages           4250 non-null   int64  
 6   total_day_minutes               4250 non-null   float64
 7   total_day_calls                 4250 non-null   int64  
 8   total_day_charge                4250 non-null   float64
 9   total_eve_minutes               4250 non-null   float64
 10  total_eve_calls                 4250 non-null   int64  
 11  total_eve_charge                4250 non-null   float64
 12  total_night_minutes             4250 non-null   float64
 13  total_night_calls               4250 non-null   int64  
 14  total_night_charge              4250 non-null   float64
 15  total_intl_minutes              4250 non-null   float64
 16  total_intl_calls                4250 non-null   int64  
 17  total_intl_charge               4250 non-null   float64
 18  number_customer_service_calls   4250 non-null   int64  
 19  churn                           4250 non-null   object 
dtypes: float64(8), int64(7), object(5)
memory usage: 664.2+ KB
```

```
df.isnull().sum().sum()
```

```
0
```

```
df[df.duplicated()]
```

Dataset tidak memuat:

- Data null
- Data duplikat

PROBLEM

"data test" yang belum diklasifikasikan ke dalam customer churn atau non-churn.

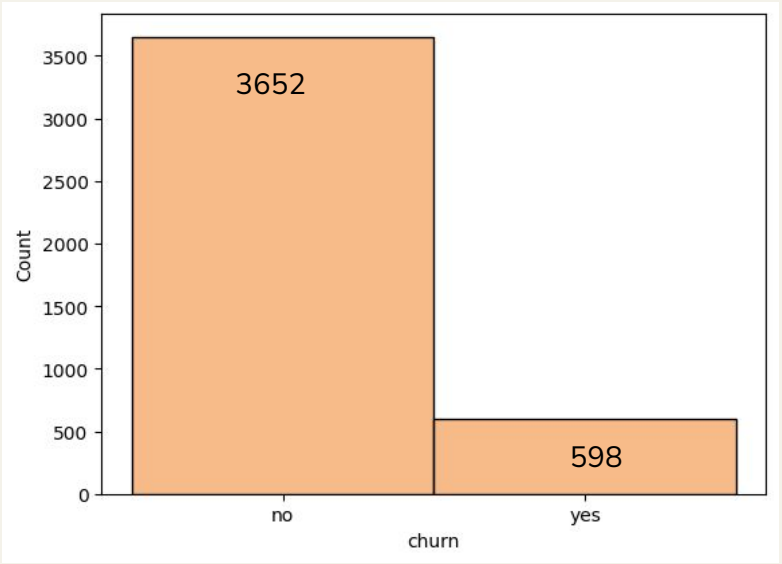
SOLUTION

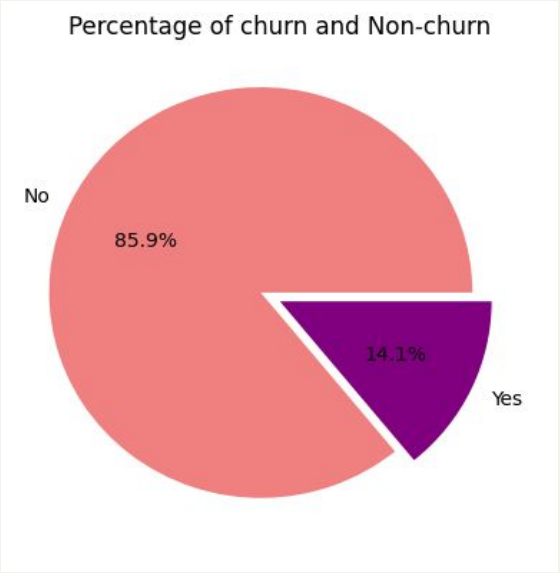
Memprediksi klasifikasi customer churn atau non-churn dengan model machine learning terbaik.

02

Exploratory Data Analysis (EDA)

Churn

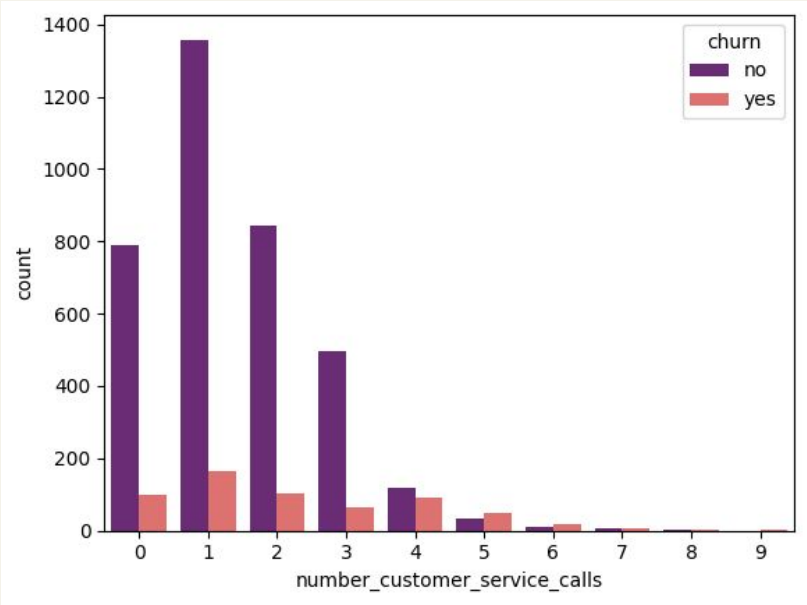




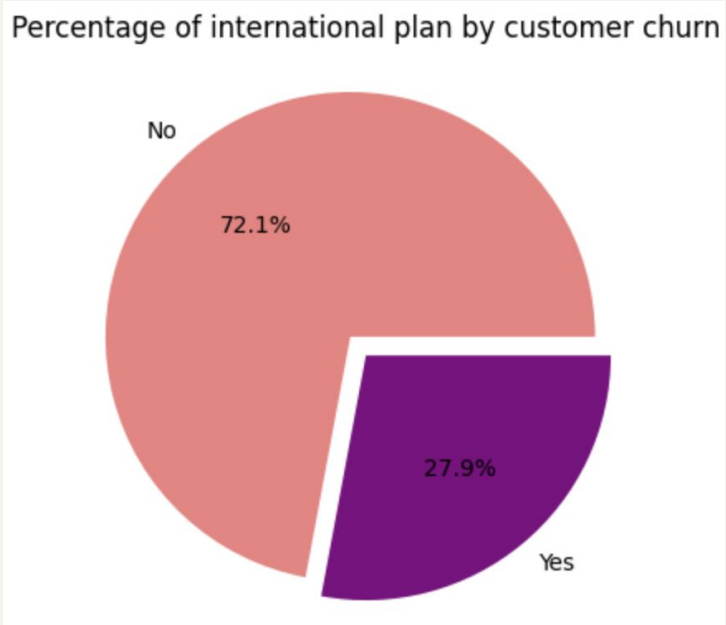
Berdasarkan bar chart dan pie chart di atas, dapat diketahui bahwa terjadi ketidakseimbangan kelas data (***imbalanced data***) yang masuk kategori "churn" dan "non-churn".

Ada sekitar 14.1% pelanggan yang termasuk customer churn, dan 85.9% pelanggan yang termasuk customer non-churn.

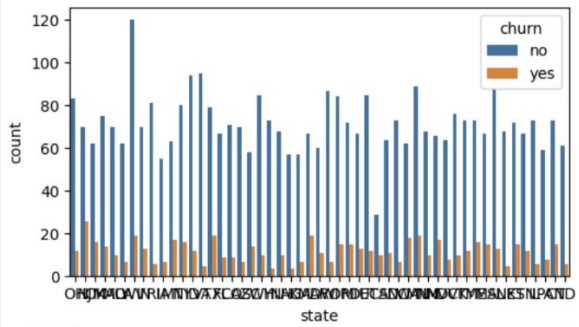
Churn by number customer service calls

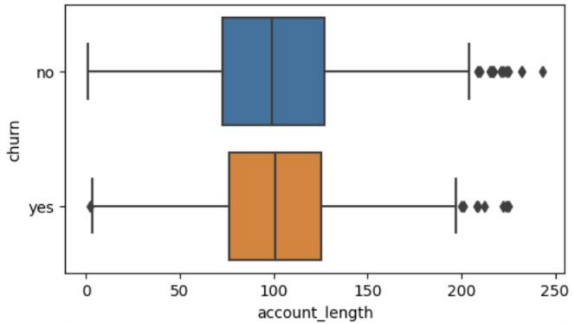


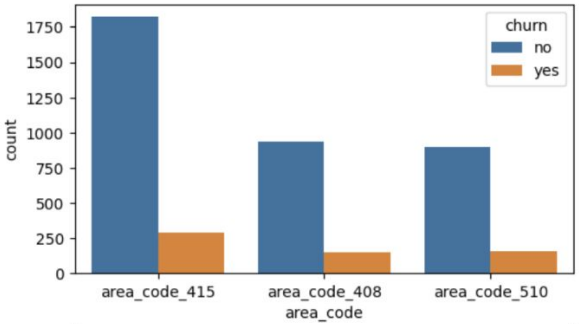
Customer yang memiliki jumlah panggilan ke customer service lebih dari 4 memiliki kecenderungan untuk melakukan churn.

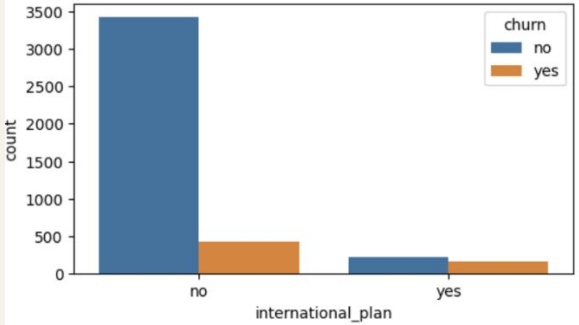


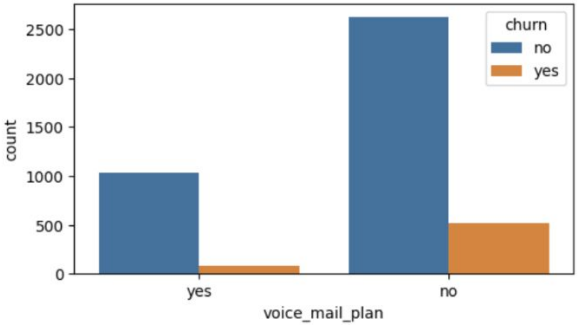
Customer yang melakukan churn 72.1% tidak memiliki paket international.

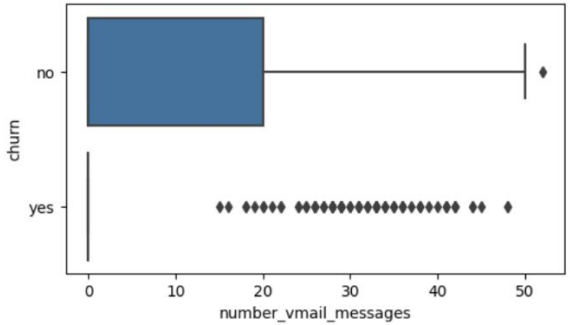

churn
no
yes
count
state

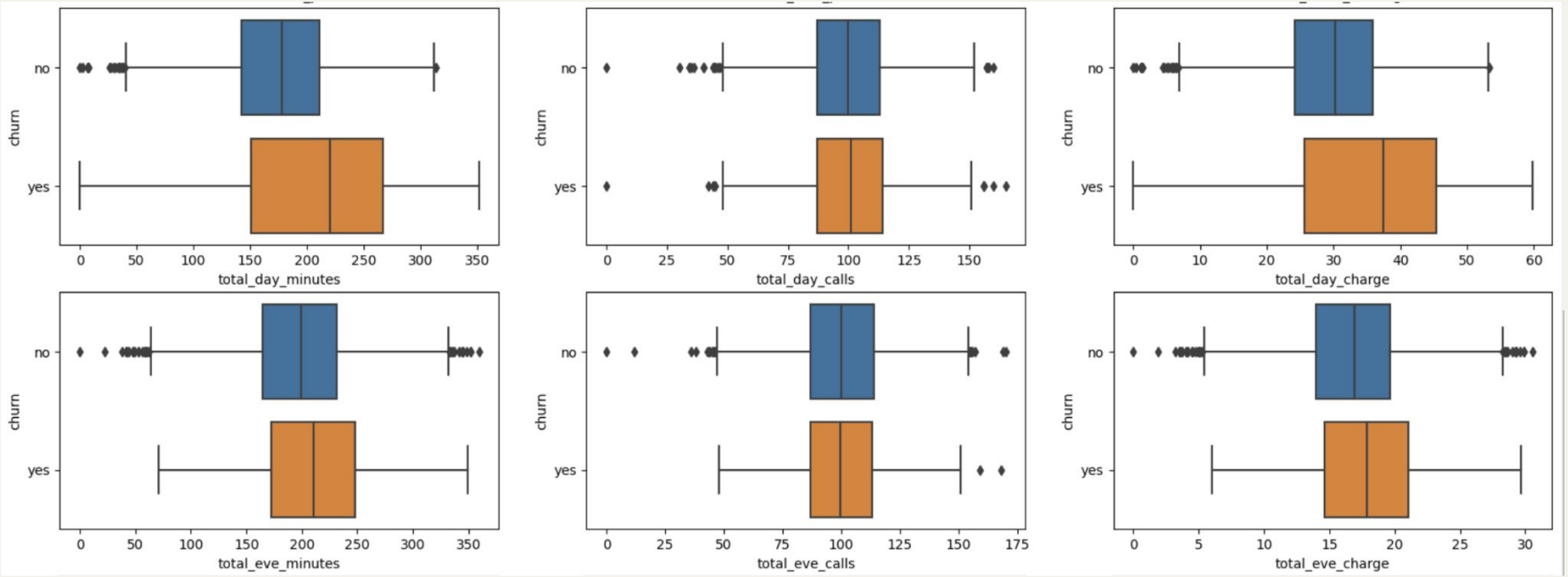

churn
no
yes
account_length

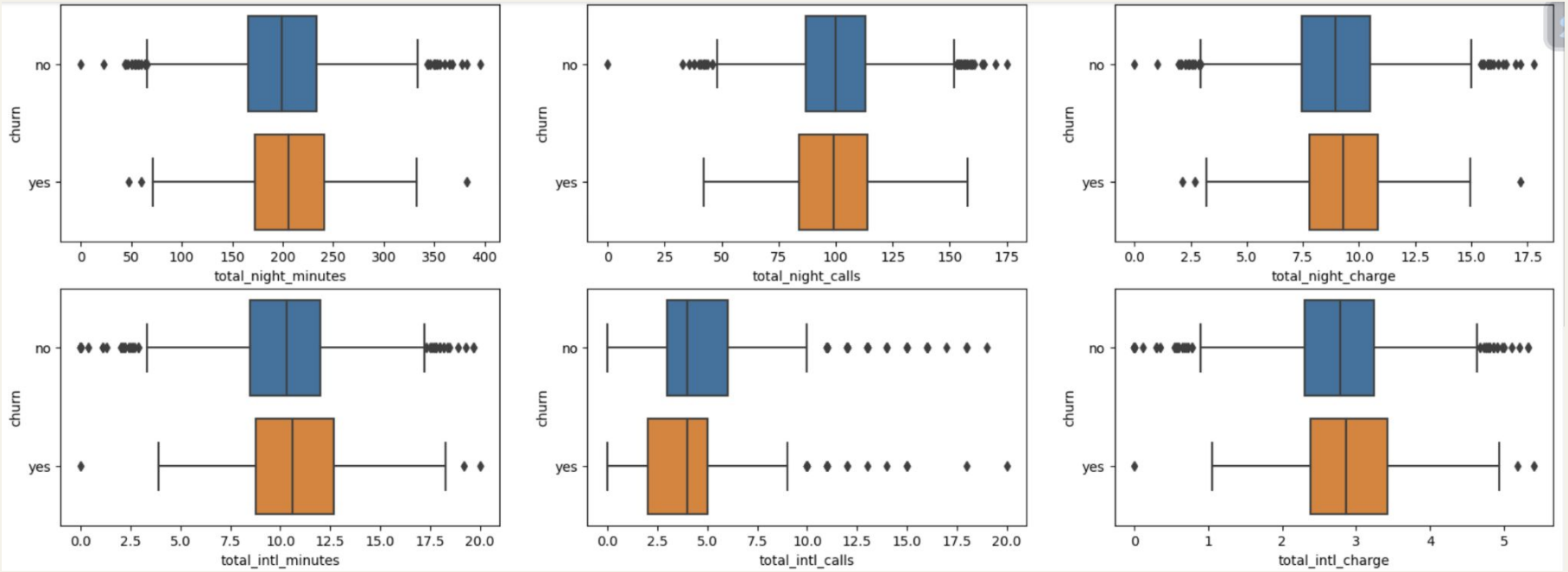

churn
no
yes
count
area_code_415
area_code_408
area_code_510
area_code

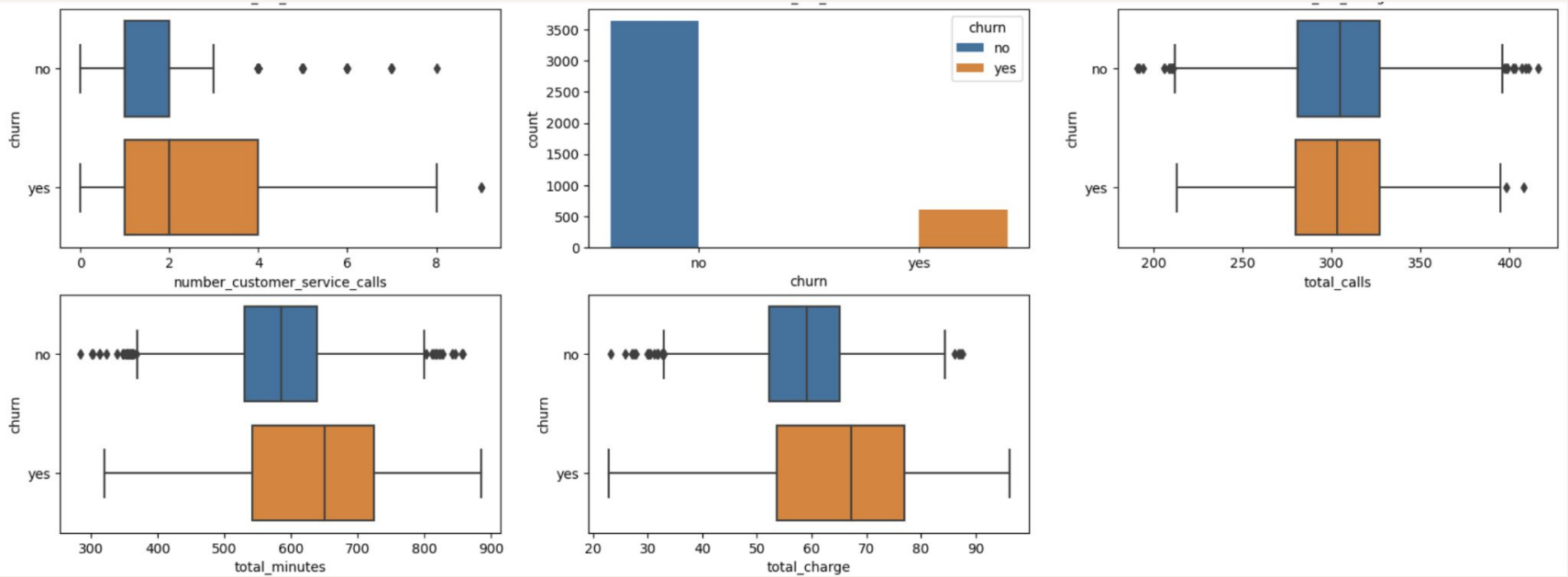

churn
no
yes
count
international_plan


churn
no
yes
count
voice_mail_plan


churn
no
yes
number_vmail_messages


churn
no
yes
0 50 100 150 200 250 300 350
total_day_minutes
churn
no
yes
0 25 50 75 100 125 150
total_day_calls
churn
no
yes
0 10 20 30 40 50 60
total_day_charge
churn
no
yes
0 50 100 150 200 250 300 350
total_eve_minutes
churn
no
yes
0 25 50 75 100 125 150 175
total_eve_calls
churn
no
yes
0 5 10 15 20 25 30
total_eve_charge


churn
no
yes
0 50 100 150 200 250 300 350 400
total_night_minutes
0 25 50 75 100 125 150 175
total_night_calls
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5
total_night_charge
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5 20.0
total_intl_minutes
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5 20.0
total_intl_calls
0 1 2 3 4 5
total_intl_charge



Visualisasi pada variabel-variabel dataset, mayoritas menunjukkan adanya pencilan.

03

Data Preprocessing

Feature Extraction

Total Calls = total day calls + total evening calls + total night calls + total international calls

Total Charges = total day charges + total evening charges + total night charges + total international charges

Total Minutes = total day minutes + total evening minutes + total night minutes + total international minutes

Label Encoding

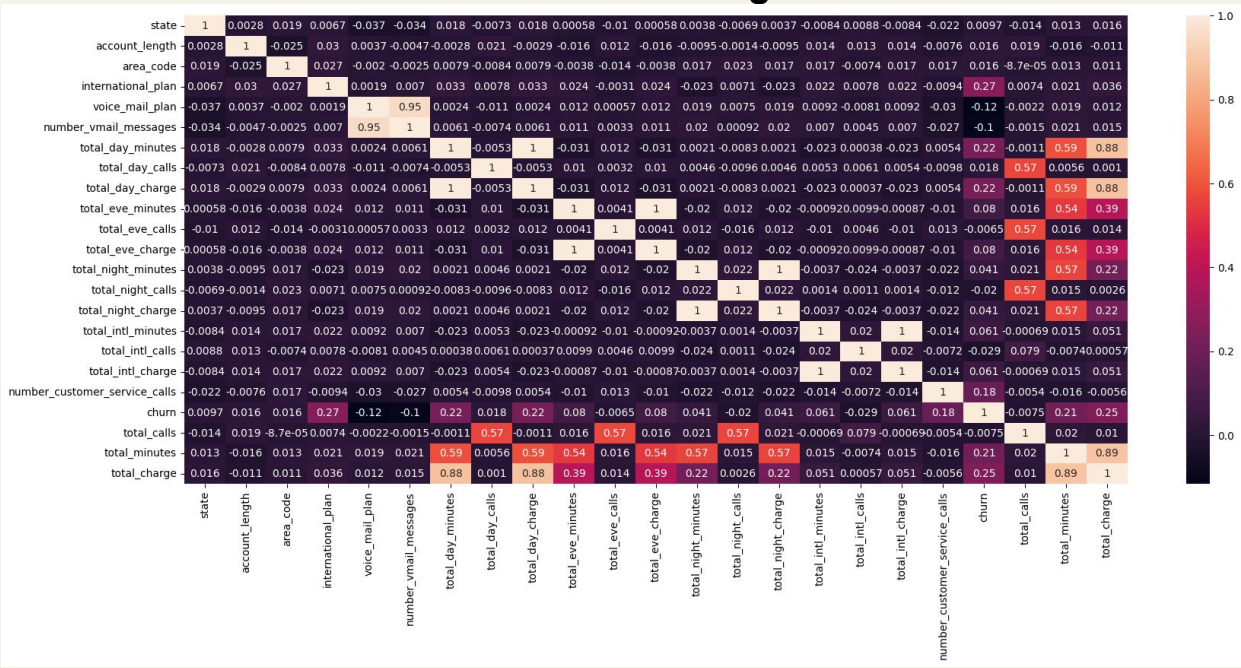
```
[ ]  from sklearn import preprocessing
     le=preprocessing.LabelEncoder()
```

```
[ ]  df['churn']=le.fit_transform(df.churn.values)
     df['state']=le.fit_transform(df.state.values)
     df['international_plan']=le.fit_transform(df.international_plan.values)
     df['voice_mail_plan']=le.fit_transform(df.voice_mail_plan.values)
     df['area_code']=le.fit_transform(df.area_code.values)

     df_test['state']=le.fit_transform(df_test.state.values)
     df_test['international_plan']=le.fit_transform(df_test.international_plan.values)
     df_test['voice_mail_plan']=le.fit_transform(df_test.voice_mail_plan.values)
     df_test['area_code']=le.fit_transform(df_test.area_code.values)
```

Korelasi antar variabel

Handling Outlier with zscore



Korelasi tertinggi yaitu churn dengan international plan yaitu sebesar 0,27 lalu total charge sebesar 0,25

Korelasi variabel indepenent dengan churn lebih besar saat handling outliers menggunakan zscore

Standarisasi Variabel

```
[ ]  from sklearn.preprocessing import StandardScaler
```

```
df1['account_length'] = StandardScaler().fit_transform(df1['account_length'].values.reshape(len(df1), 1))
df1['number_vmail_messages'] = StandardScaler().fit_transform(df1['number_vmail_messages'].values.reshape(len(df1), 1))
df1['number_customer_service_calls'] = StandardScaler().fit_transform(df1['number_customer_service_calls'].values.reshape(len(df1), 1))
df1['total_calls'] = StandardScaler().fit_transform(df1['total_calls'].values.reshape(len(df1), 1))
df1['total_minutes'] = StandardScaler().fit_transform(df1['total_minutes'].values.reshape(len(df1), 1))
df1['total_charge'] = StandardScaler().fit_transform(df1['total_charge'].values.reshape(len(df1), 1))
df1['total_charge'] = StandardScaler().fit_transform(df1['total_charge'].values.reshape(len(df1), 1))

df_test['account_length'] = StandardScaler().fit_transform(df_test['account_length'].values.reshape(len(df_test), 1))
df_test['number_vmail_messages'] = StandardScaler().fit_transform(df_test['number_vmail_messages'].values.reshape(len(df_test), 1))
df_test['number_customer_service_calls'] = StandardScaler().fit_transform(df_test['number_customer_service_calls'].values.reshape(len(df_test), 1))
df_test['total_calls'] = StandardScaler().fit_transform(df_test['total_calls'].values.reshape(len(df_test), 1))
df_test['total_minutes'] = StandardScaler().fit_transform(df_test['total_minutes'].values.reshape(len(df_test), 1))
df_test['total_charge'] = StandardScaler().fit_transform(df_test['total_charge'].values.reshape(len(df_test), 1))
```

04

Modelling

Sebelum dilakukan pemodelan, data dibagi menjadi:

- data training (75%)
- data testing (25%)

- Data training digunakan sebagai data yang dilatih untuk membuat model,
- sedangkan data testing digunakan untuk memvalidasi model tersebut.

Pemodelan Machine Learning

Regresi Logistik

```
Confusion Matrix :
[[889  22]
 [119  24]]
Classification Report :
              precision    recall  f1-score   support

           0       0.88      0.98      0.93       911
           1       0.52      0.17      0.25       143

    accuracy                           0.87      1054
   macro avg       0.70      0.57      0.59      1054
weighted avg       0.83      0.87      0.84      1054

Accuracy: 0.87
ROC AUC: 0.81
Cross Validation accuracy: 0.866 +/- 0.007
```

Gradient Boosting

```
Accuracy Score of Gradient Classifier is : 0.9430740037950665
Confusion Matrix :
[[904   7]
 [ 53  90]]
Classification Report :
              precision    recall  f1-score   support

           0       0.94      0.99      0.97       911
           1       0.93      0.63      0.75       143

    accuracy                           0.94      1054
   macro avg       0.94      0.81      0.86      1054
weighted avg       0.94      0.94      0.94      1054

ROC AUC: 0.90
[0.94312796 0.93838863 0.94075829 0.94075829 0.94075829 0.95486936
 0.95724466 0.95486936 0.93349169 0.95961995]
Cross Validation accuracy: 0.946 +/- 0.009
```

Ada Boost

```
Confusion Matrix :
[[895  16]
 [ 70  73]]
Classification Report :
              precision    recall  f1-score   support

           0       0.93      0.98      0.95       911
           1       0.82      0.51      0.63       143

    accuracy                           0.92      1054
   macro avg       0.87      0.75      0.79      1054
weighted avg       0.91      0.92      0.91      1054

Accuracy: 0.92
ROC AUC: 0.88
Cross Validation accuracy: 0.948 +/- 0.008
```

Dari akurasi ketiga pemodelan yang dilakukan di atas, dapat disimpulkan bahwa model **Gradient Boosting** merupakan model terbaik diantara ketiganya (karena menghasilkan akurasi prediksi yang tertinggi yaitu sekitar 94.3%)

05

Predict

Selanjutnya, melakukan prediksi pada "data test" untuk mengklasifikasikan customer churn atau non-churn. "data test" ini berisi 750 sample yang akan diprediksi sebagai customer churn atau non-churn menggunakan model terbaik yang telah diperoleh sebelumnya yaitu Gradient Boosting.

Result

Berdasarkan prediksi pada "data test" menggunakan Gradient Boosting, dapat diketahui bahwa:

- 681 pelanggan yang diprediksi sebagai customer non-churn
- 69 pelanggan yang diprediksi sebagai customer churn.

Result

Hal ini menunjukkan bahwa:

- sekitar 90.8% pelanggan diprediksi masih bertahan menggunakan provider sebelumnya.
- sedangkan 9.2% pelanggan berpindah ke provider lain.

06

Conclusion

- Dari 3 model machine learning (Regresi Logistik, Gradient Boosting, Ada Boost) yang digunakan pada Dataset, model machine learning terbaik yaitu **Gradient Boosting** dengan akurasi model sekitar 94.3%.
- Hasil prediksi model **Gradient Boosting** pada “data test” menunjukkan bahwa sekitar 90.8% pelanggan diprediksi masih bertahan menggunakan provider sebelumnya, sedangkan 9.2% pelanggan berpindah ke provider lain.

Thanks!