TELCO CHURN PREDICTION

Created By **Arsyadana Al 'Aziz**

Dataset Kaggle: https://www.kaggle.com/datasets/blastchar/telco-customer-churn

Overview

1. To find out how many customers Churn
2. Using several machine learning methods to find the best model

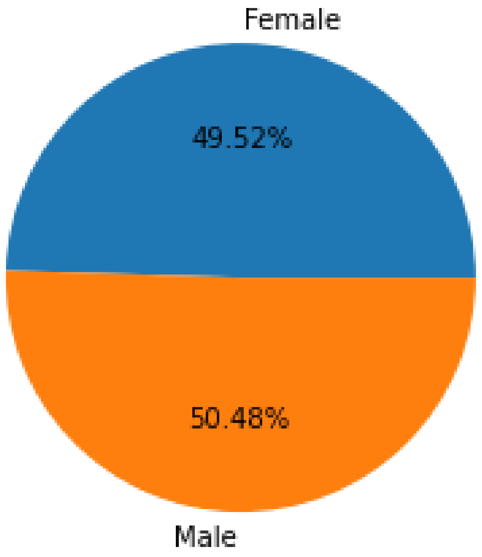
EXPLORATORY DATA ANALYSIS

Check Data and Simple Describe

```
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
 15  Contract          7043 non-null   object
 16  PaperlessBilling  7043 non-null   object
 17  PaymentMethod     7043 non-null   object
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object
 20  Churn             7043 non-null   object
dtypes: float64(1), int64(2), object(18)
```

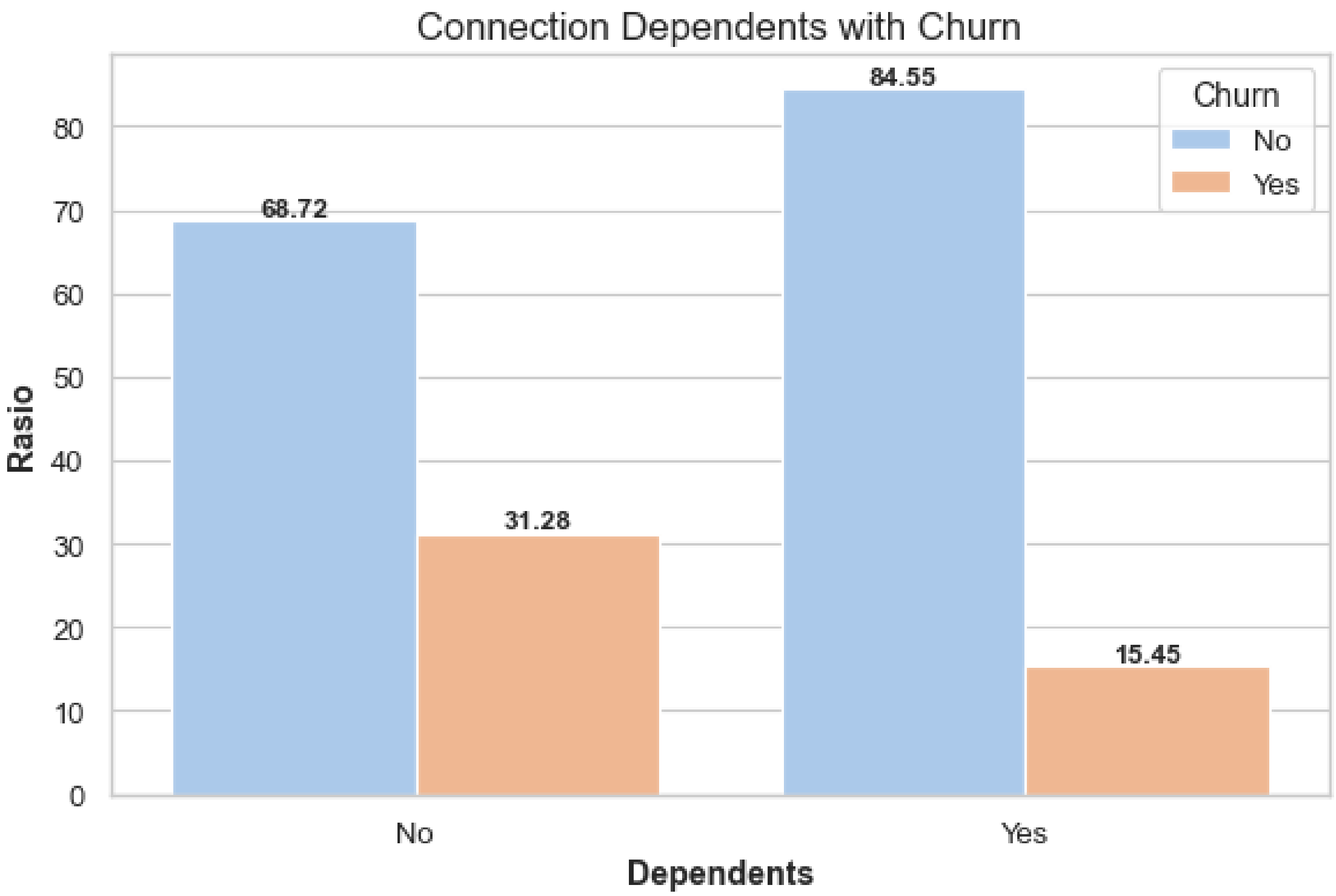
```
1  df.duplicated().sum()
✓ 0.1s

0
```

It can be seen that the available data already has all values and there are no duplicates data



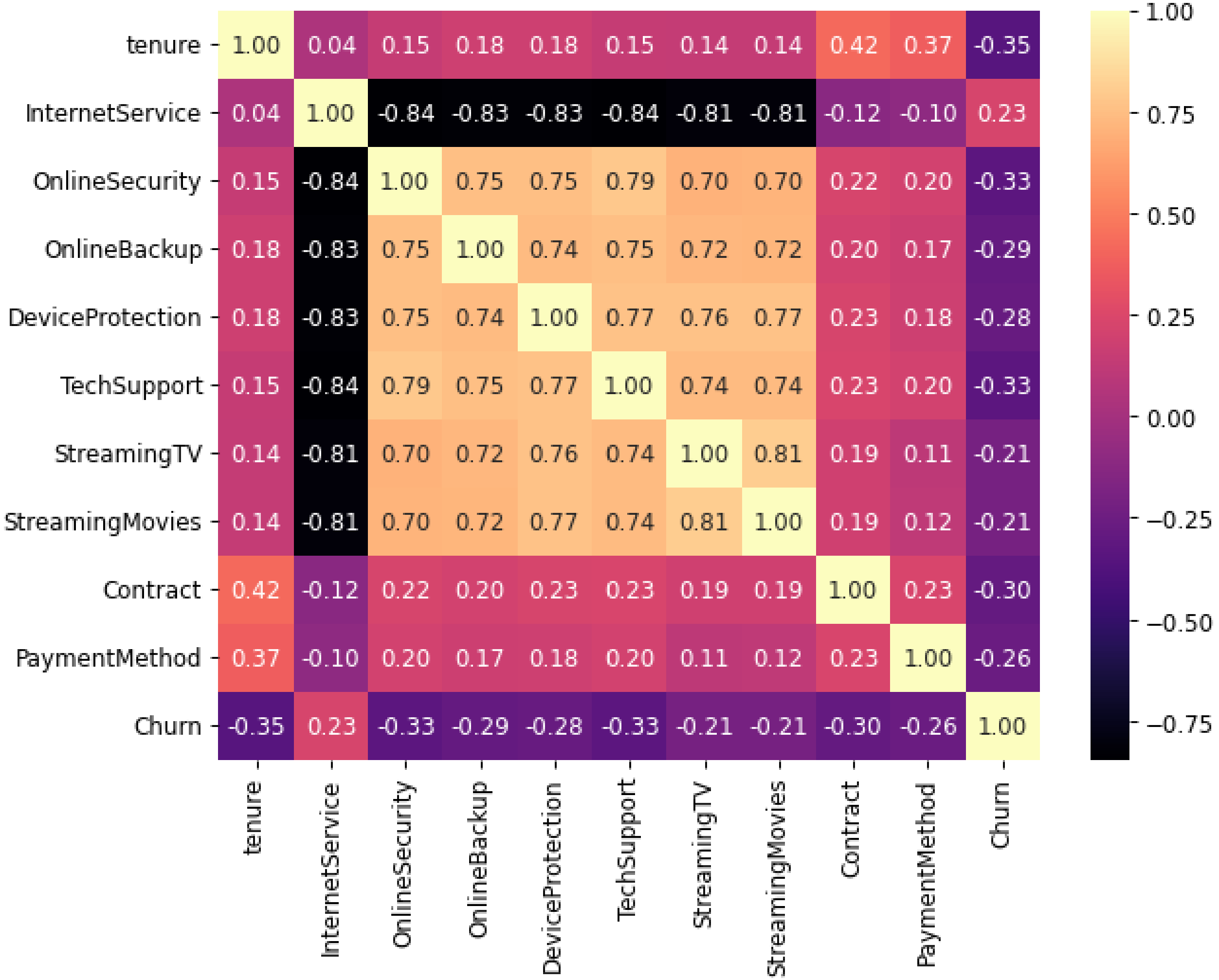
From the data it is known that 50,48% are men while the remaining 49,52% are women

Connection Dependents With Churn



It can be seen from the data, if customer Dependents (Yes) more have potential to not churn (84,55%), whereas if they are Dependents (No) the percentage of Churn is 31,28%.

Correlation



Some data correlations that are considered to have an effect on churn. It can be seen that the largest positive correlation is the internet serive (0,23), and largest negative correlation is the tenure (-0,35)

Data Processing

Encoding data because a lot of data is categorical type, so need encoding to change it the numeric data

```
# Melakukan encoding untuk beberapa data yang bersifat object
encod=['gender', 'Partner', 'Dependents',
       'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport',
       'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling',
       'PaymentMethod', 'Churn']
noencod=['customerID','SeniorCitizen','tenure','MonthlyCharges','TotalCharges']
```

Encoding process

```
#Looping untuk encoding
for i in encod:
    kond= [(df[i]=='Male'),(df[i]=='Female'),(df[i]=='Yes'),
        (df[i]=='No'),(df[i]=='No internet service'),(df[i]==''),(df[i]=='Month-to-month'),
        (df[i]=='Two year'),(df[i]=='Electronic check'),(df[i]=='Mailed check')]
    beta=[1,0,1,0,2,4,0,1,0,1]
    data1[i]=np.select(kond, beta, default=3)
for j in noencod:
    data1[j]=df[j]
display(data1)
```

Data Processing

```python
data1.sort_values(by=['TotalCharges'])
```

✓ 0.7s Python

OnlineSecurity	...	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
1	...	1	0	1	1	1	0	1	80.85		0
0	...	2	2	2	2	1	0	1	25.35		0
0	...	2	2	2	2	1	0	1	20.00		0
0	...	2	2	2	2	1	0	1	20.25		0
0	...	2	2	2	2	3	1	1	19.70		0
...	...	...	...	...	...	...	...	...	...	...	...
0	...	2	2	2	2	3	0	1	19.40	997.75	0

Cleaning Data because there is data that has a missing value ( )

```python
# Setelah dicek masih terdapat data non-clean yakni berupa data kosong bukan NaN, sehingga perlu dilakukan drop terhadap kondisi tersebut
data1['TotalCharges'].replace(' ', np.nan, inplace=True)
data1.dropna(subset=['TotalCharges'], inplace=True)
```

Drop missing value for data

MACHINE LEARNING MODEL

Split Data

```
X = data1.drop(columns=["customerID","Churn"])
y = data1[["Churn"]]


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)


scaler = MinMaxScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
data train: (5274, 19)
data test (1758, 19)
```

I use 25% test data because the available data is only around 7043 data, so it is necessary attention to the amount of test data.

Evaluation

Method	Recall
Decision Tree	0,721
Random Forest	0,784
SVM	0,787
Logistic Regression	0,791

of the several machine learning methods, we take the Recall because False Positif better than False Negative.

THANK YOU