Exploratory Data Analysis Result on Telco Customer Churn Dataset

Ahmad Ichsan Baihaqi

Data Science Bootcamp Batch 7 of dibimbing.id

ahmadichsanbaihaqi@gmail.com

Keywords: Exploratory Data Analysis; Churn; Predictor;

Abstract. Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypotheses and to check assumptions with the help of summary statistics and graphical representations [1]. This EDA aims to get insight about the correlation of our target feature (churn) among other features given in the dataset. Furthermore, the output of this EDA is to get a list of features that suit to be the predictor of our target feature. This analysis result includes data profiling, cleansing, manipulation/feature engineering, visualization and the EDA itself. The results show that the highest positive correlation value is +0.19 (PaperlessBilling and MonthlyCharges), followed by +0.15 (SeniorCitizen) and +0.11 (PaymentMethod). Then, the highest negative correlation value is -0.40 (Contract), followed by -0.35 (Tenure) and -0.29 (OnlineSecurity).

Introduction

Dataset used in this analysis is the Telco Customer Churn dataset which can be found here https://www.kaggle.com/blastchar/telco-customer-churn. Each row represents a customer, each column contains customer's attributes described on the column Metadata. The data set includes information about:

- Customers who left within the last month – the column is called Churn
- Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies
- Customer account information – how long they've been a customer, contract, payment method, paperless billing, monthly charges, and total charges
- Demographic info about customers – gender, age range, and if they have partners and dependents

The target feature of this analysis is the Churn feature.

Analysis

Dataframe information. To begin with, based on Fig 1 below, there are 7043 rows (observations) of data with 21 columns (features). Among the given columns in the dataset, there is a possible dropped column, which is a customerID. This is because the value is unique and has high cardinality and won't affect our analysis on the target feature.

In terms of data cleansing, column names are one of things that we need to check. The column names in the dataset have inconsistent naming. Some of them are lowercase (e.g. gender, tenure), some of them camelCase (e.g. customerID) and the rest is PascalCase. Variability and inconsistencies often lead to confusion, error and loss of time [2]. Since we will drop the

customerID column, later we would only handle the lowercase column name into PascalCase (follow the majority) in order to uniform the column name convention.

```
<class 'pandas.core.frame.DataFrame'>
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)
memory usage: 1.1+ MB
```

Fig 1. The dataframe's info by doing df.info()

Based on Fig 1, our data consist of 1 float feature, 2 integer features and 18 categorical features. Since we still have categorical features, later we need to do categorical data encoding. We are doing categorical data encoding because machine learning models require all input and output variables to be numeric [3].

By doing df.info(), we could retrieve information about null/missing value. As we can see above, there are no columns with null/missing values. But, don’t fall for this information easily. Look at the data types. Why the data type of TotalCharges is categorical instead of floats like MonthlyCharges? Let's see our 15 first data sorted by smallest TotalCharges in Fig 2.

	tenure	TotalCharges
936	0	
3826	0	
4380	0	
753	0	
5218	0	
3331	0	
6754	0	
6670	0	
1340	0	
488	0	
1082	0	
105	5	100.2
4459	1	100.25
1723	6	100.35
2124	5	100.4

Fig 2. Total charges missing value

Based on the above observation, there are rows containing string (whitespace). This is not detected as missing value by pandas. But, this whitespace is indeed a missing value. Then, how to handle this missing value? If we take a look at Fig 2, all rows with empty TotalCharges are when the tenure is zero. Tenure represents the number of months the customer has stayed with the company. It means, the TotalCharges is empty because the customer is not charged yet. So, to handle the empty value of TotalCharges, we can fill it with zero.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 20 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Gender            7043 non-null   int64
 1   SeniorCitizen     7043 non-null   int64
 2   Partner           7043 non-null   int64
 3   Dependents        7043 non-null   int64
 4   Tenure            7043 non-null   int64
 5   PhoneService      7043 non-null   int64
 6   MultipleLines     7043 non-null   int64
 7   InternetService   7043 non-null   int64
 8   OnlineSecurity    7043 non-null   int64
 9   OnlineBackup      7043 non-null   int64
 10  DeviceProtection  7043 non-null   int64
 11  TechSupport       7043 non-null   int64
 12  StreamingTV       7043 non-null   int64
 13  StreamingMovies   7043 non-null   int64
 14  Contract          7043 non-null   int64
 15  PaperlessBilling  7043 non-null   int64
 16  PaymentMethod     7043 non-null   int64
 17  MonthlyCharges    7043 non-null   float64
 18  TotalCharges      7043 non-null   float64
 19  Churn             7043 non-null   int64
dtypes: float64(2), int64(18)
memory usage: 1.1 MB
```

Fig 3. Dataframe information after cleansing

Fig 3 above shows us the result after data cleansing and data encoding. The customerID feature is dropped. All column names in PascalCase. Categorical features now are in integers.

Finding unique value. By finding unique values, we could identify, if a feature has unique values equal with the total data, it is possible that those features will not be related with our target feature (e.g. customerID which dropped before). On the other hand, if those features only have one unique value, it means those features also won't affect our target analysis. Then, we can drop these columns.

	features	total_unique
0	Gender	2
1	SeniorCitizen	2
2	Partner	2
3	Dependents	2
4	Tenure	73
5	PhoneService	2
6	MultipleLines	3
7	InternetService	3
8	OnlineSecurity	3
9	OnlineBackup	3
10	DeviceProtection	3
11	TechSupport	3
12	StreamingTV	3
13	StreamingMovies	3
14	Contract	3
15	PaperlessBilling	2
16	PaymentMethod	4
17	MonthlyCharges	1585
18	TotalCharges	6531
19	Churn	2

Fig 4. Unique value on each features

No features with 1 unique value nor have unique value equal with the total data. Thus, the rest of this feature might contribute to our target feature.

Target feature visualization. Before we dig down which feature can be used as the predictors of our target feature, it's a good thing to visualize our target feature data distribution first. Over 7043 entries in the dataset, there are 1869 customers who are churn and 5174 customers who are retained.

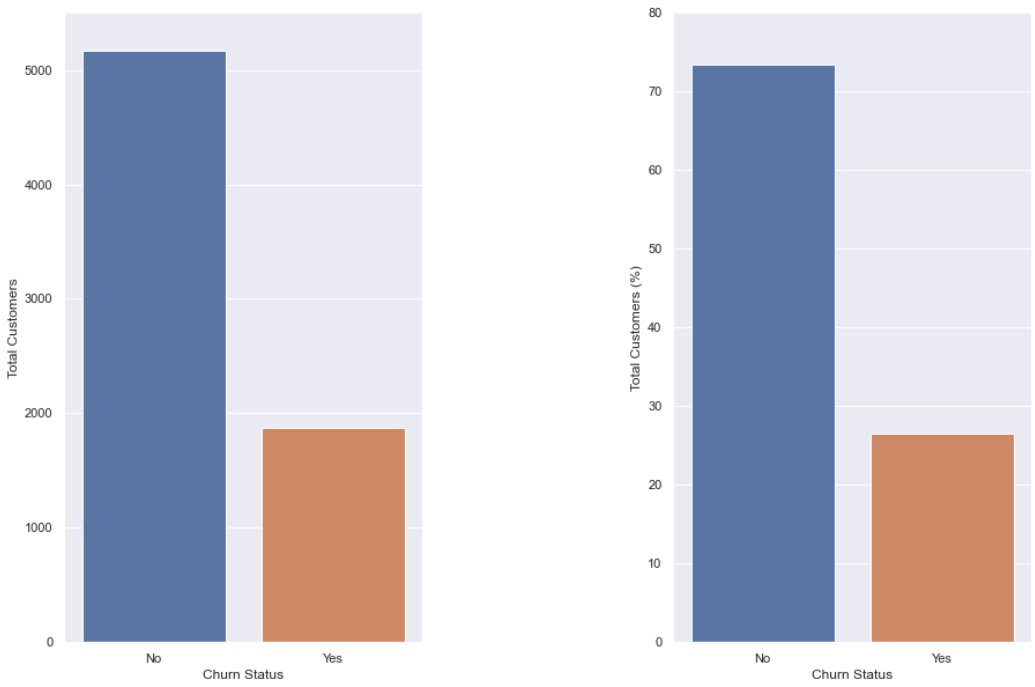

Fig 5. Churn distribution in the dataset

Feature correlation analysis. Correlation analysis between features and our target feature is useful to identify which feature has high positive correlation and which feature has high negative correlation. If a feature has high positive correlation with our target feature, it means that the increasing (or the decreasing) in predictor value will also increase (or decrease) our target feature value. On the other hand, if a feature has a high negative correlation with our target feature, it means that the increasing (or the decreasing) in predictor value will decrease (or increase) our target feature value. Correlation analysis can be achieved by plotting our features with heatmap.

Fig 6 shows us the correlation among features in our dataset. This plot gives us an attractive visual, so we could easily identify which feature combination has high positive correlation, which feature combination has no correlation and which feature combination has high negative correlation. By taking a look into the scalar on the right, we can understand that if the tile is getting dark, it represents a high negative correlation. On the other hand, the white tile represents high positive correlation.

From the heatmap in Fig 6, we can conclude that the highest positive correlation value is +0.19 (PaperlessBilling and MonthlyCharges), followed by +0.15 (SeniorCitizen) and +0.11

(PaymentMethod). On the other hand, the highest negative correlation value is -0.40 (Contract), followed by -0.35 (Tenure) and -0.29 (OnlineSecurity).

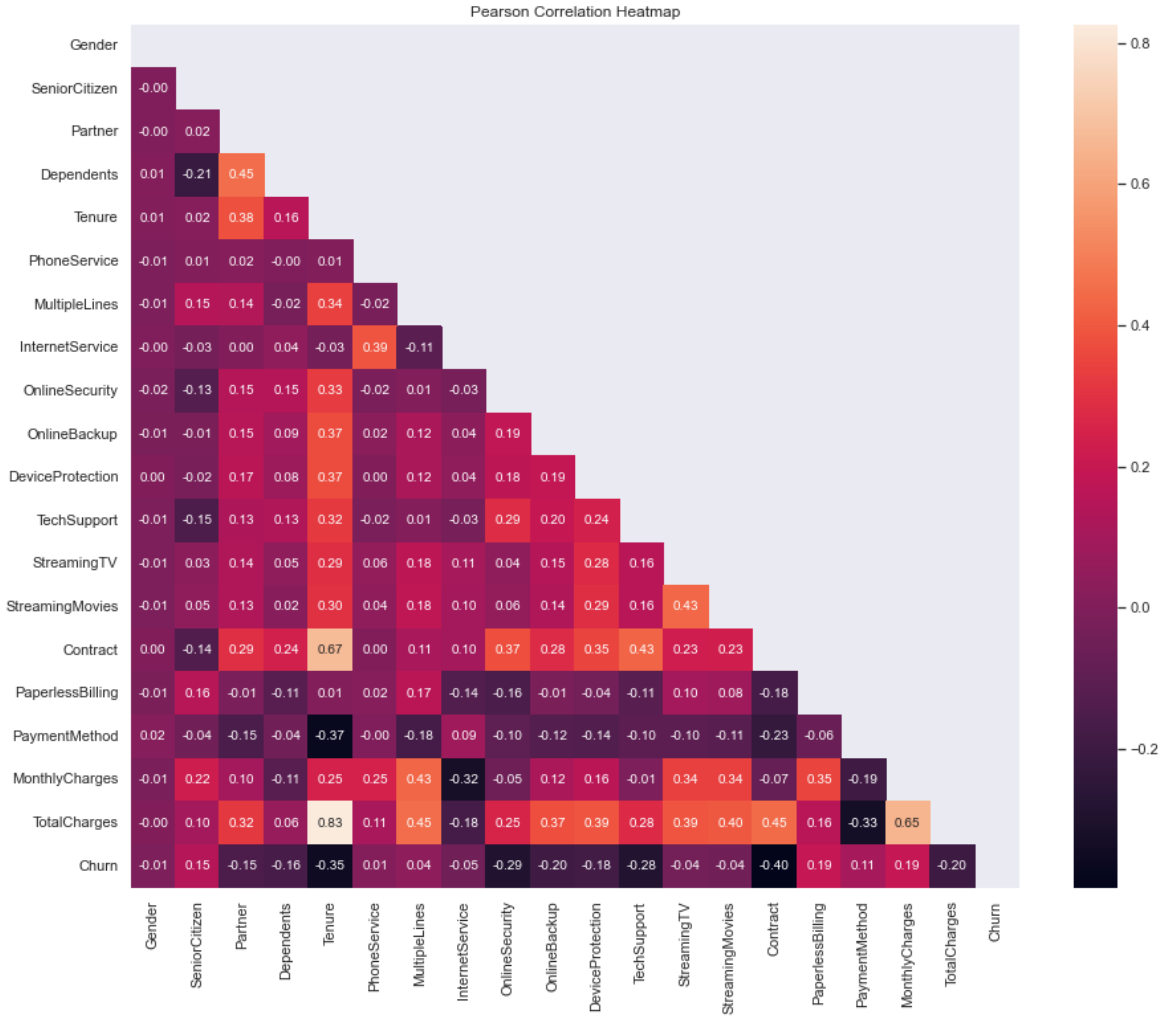

Fig 6. Heatmap plot between features in the dataset

Churn distribution by Paperless Billing. In Fig 7.1 and Fig 7.2, we can get information that customers with paperless billing are (approx) 2 times more likely to leave Telco (churn), than clients with paper billing. This indicator is significant and should be taken into account as a predictor feature.

	Churn	
	count	mean
PaperlessBilling		
No	2872	0.163301
Yes	4171	0.335651

Fig 7.1. Churn rate (in numbers) between paperless billing and paper billing

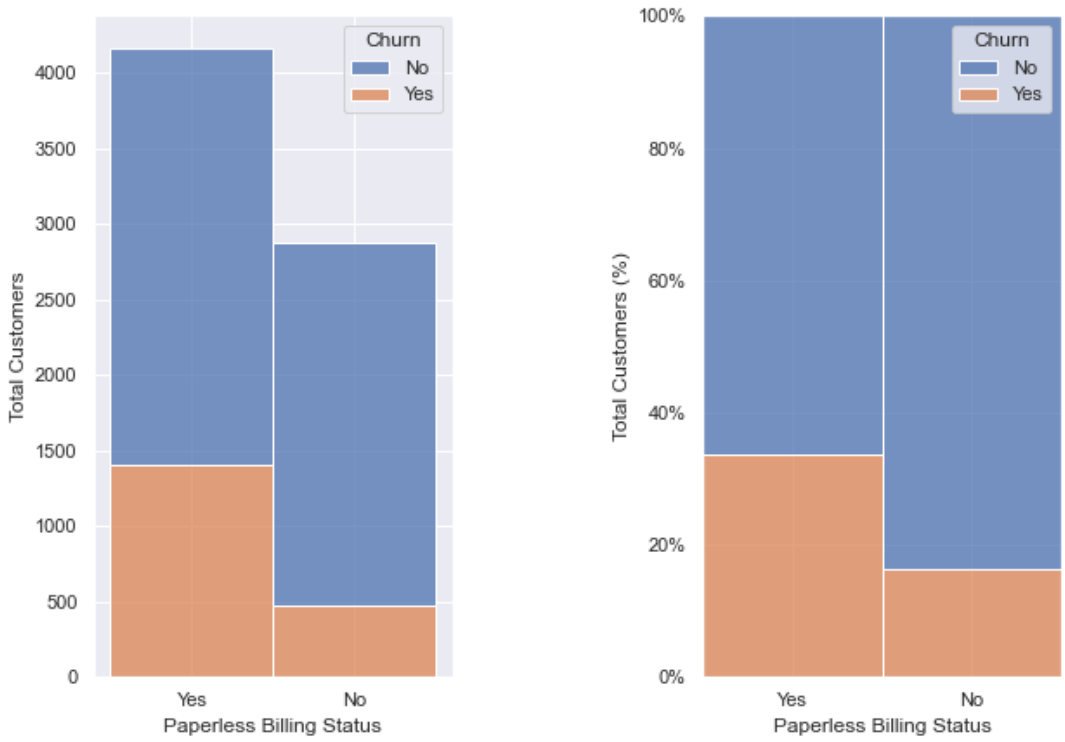

Fig 7.2. Churn rate distribution over paperless billing status

Churn distribution by senior citizens. Based on Fig 8.2, Telco customers are dominated by non-senior citizens. Moreover, based on the ratio, Fig 8.1, senior citizens tend to leave Telco 1.7 times more than non-senior citizens. This feature can be considered as a good predictor for our target feature.

	Churn	
	count	mean
SeniorCitizen		
No	5901	0.236062
Yes	1142	0.416813

Fig 8.1. Churn rate (in numbers) between senior citizen and non-senior citizen

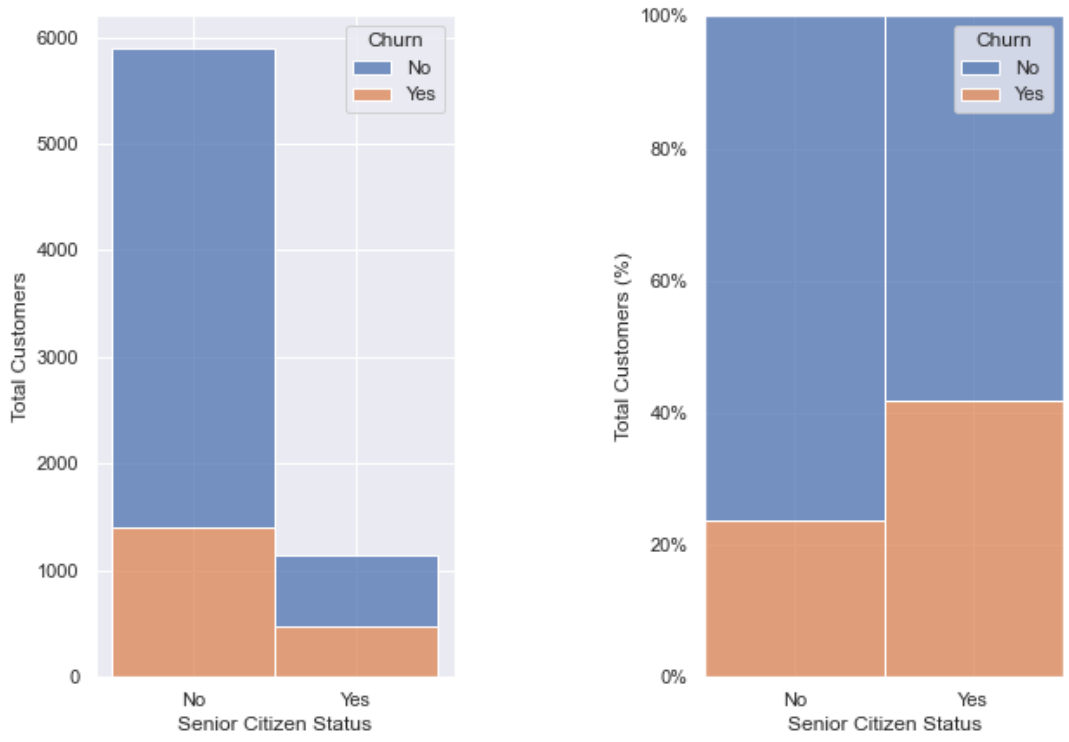

Fig 8.2. Churn rate distribution over senior citizen status

Churn distribution by payment method. Based on Fig 9.1, there are four payment methods available for Telco customers:

a. Electronic check
b. Mailed check
c. Bank transfer (automatic)
d. Credit card (automatic)

Most of the customers use electronic checks. Customers which use electronic checks tend to leave Telco 2.6 times more than the other payment method. This feature is a good candidate for our target feature predictor.

	Churn	
	count	mean
PaymentMethod		
Bank transfer (automatic)	1544	0.167098
Credit card (automatic)	1522	0.152431
Electronic check	2365	0.452854
Mailed check	1612	0.191067

Fig 9.1. Churn rate (in numbers) among different payment method

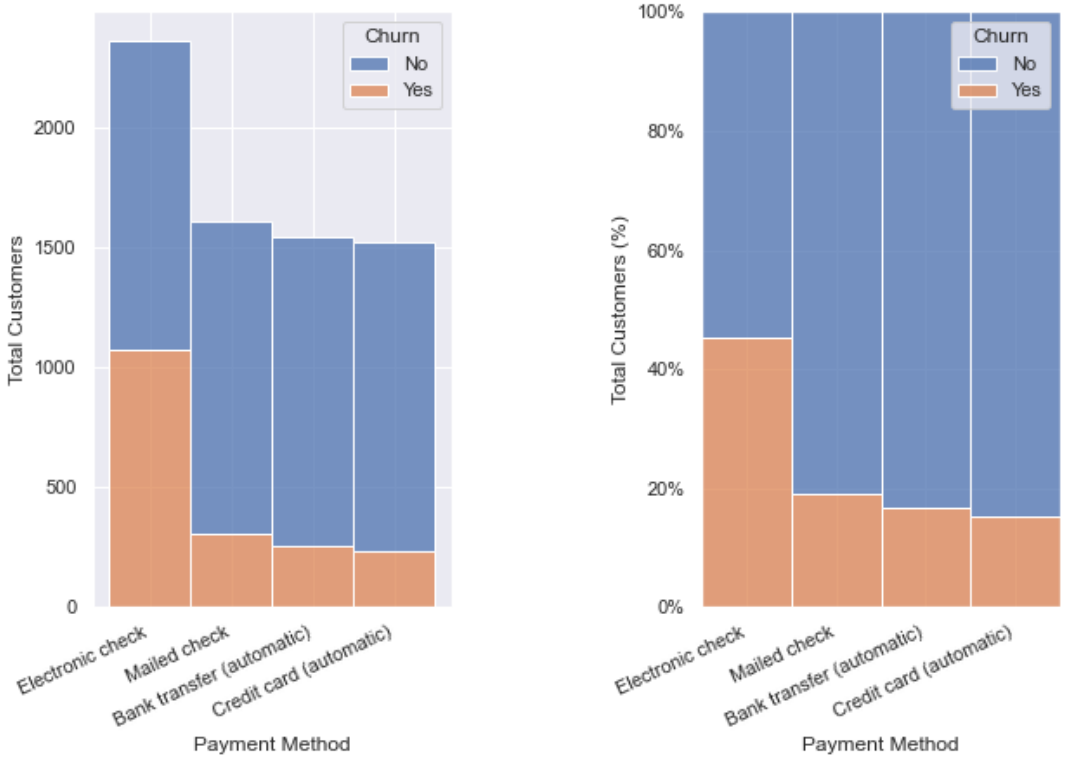

Fig 9.2. Churn rate distribution among different payment method

Alluvial diagram between payment method, senior citizen and churn features. The alluvial diagram is a type of flowchart that represents changes in a network structure over time. Alluvial diagrams serve to compare, correlate, distribute and identify trends over time [4].

	Payment Method	Senior Citizen	Occurence from Total Population (%)	Occurence from Total Churn (%)
0	Bank transfer (automatic)	Non-Senior	2.910691	10.968432
1	Bank transfer (automatic)	Senior Citizen	0.752520	2.835741
2	Credit card (automatic)	Non-Senior	2.413744	9.095773
3	Credit card (automatic)	Senior Citizen	0.880307	3.317282
4	Electronic check	Non-Senior	10.705665	40.342429
5	Electronic check	Senior Citizen	4.500923	16.960942
6	Mailed check	Non-Senior	3.748403	14.125201
7	Mailed check	Senior Citizen	0.624734	2.354200

Fig 10. Churn rate (in numbers) among different combination of payment method and senior citizen

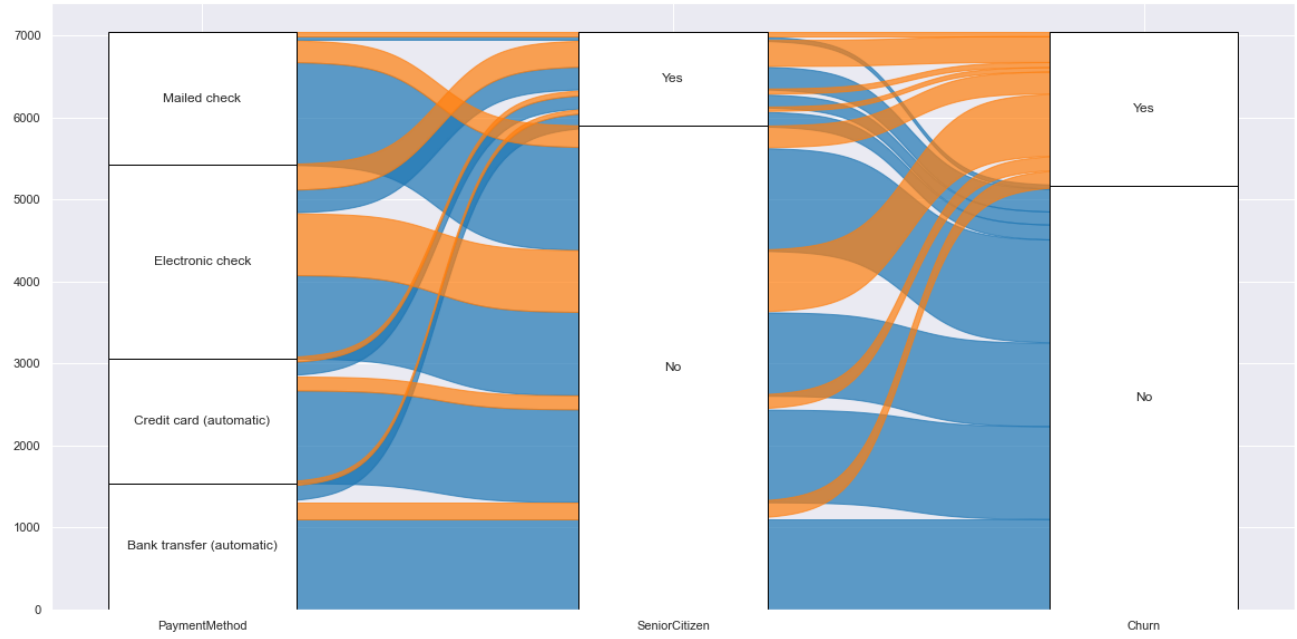

Fig 11. Alluvial diagram between payment method, senior citizen and churn

On churn distribution by senior citizens, senior citizen customers tend to churn more than non-senior citizen customers. Then, on churn distribution by payment method, customers using electronic checks tend to churn more than those who use the other method. Here, in Fig 11, writer has visualized the distribution of combinations between SeniorCitizen, PaymentMethod and Churn to find out if different SeniorCitizen will have segmentation in PaymentMethod they

are using. The highest churn ratio is the non-senior citizens customers which use electronic check with 40.3% occurence from total churn and 10.7% occurence from total population. This could happen because electronic check ease of use is not favorable among non-senior citizens since it will take much more time to do payment with a check rather than credit card or bank transfer.

Churn distribution by monthly charges. Distribution of churn customers based on MonthlyCharges (Fig 12) is quite interesting. Customers who paid 20 - 30 USD, 60 - 70 USD and above 110 USD have low churn rate. Otherwise, the churn rate is high. This distribution is possibly affected by other features, such as contracts. Different types of contracts might have different monthly charges.

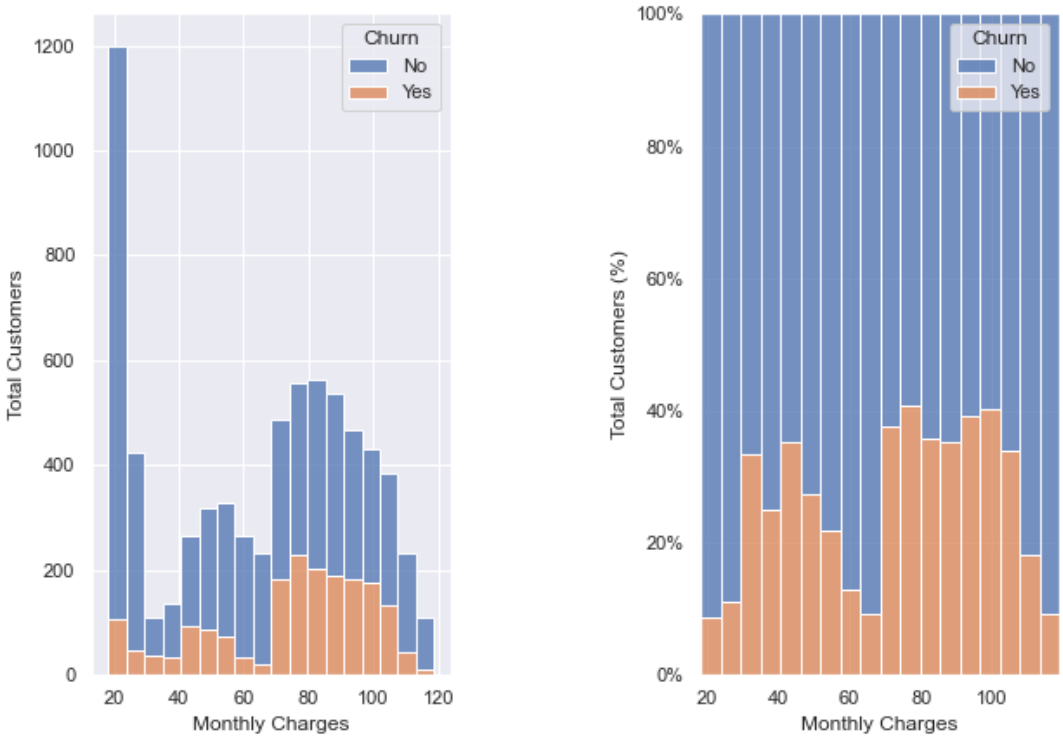

Fig 12. Churn rate distribution over monthly charges

Contract distribution over monthly charges. As written in the previous statement, different types of contracts might have different monthly charges. Now, let's see how contracts are distributed in monthly charges data.

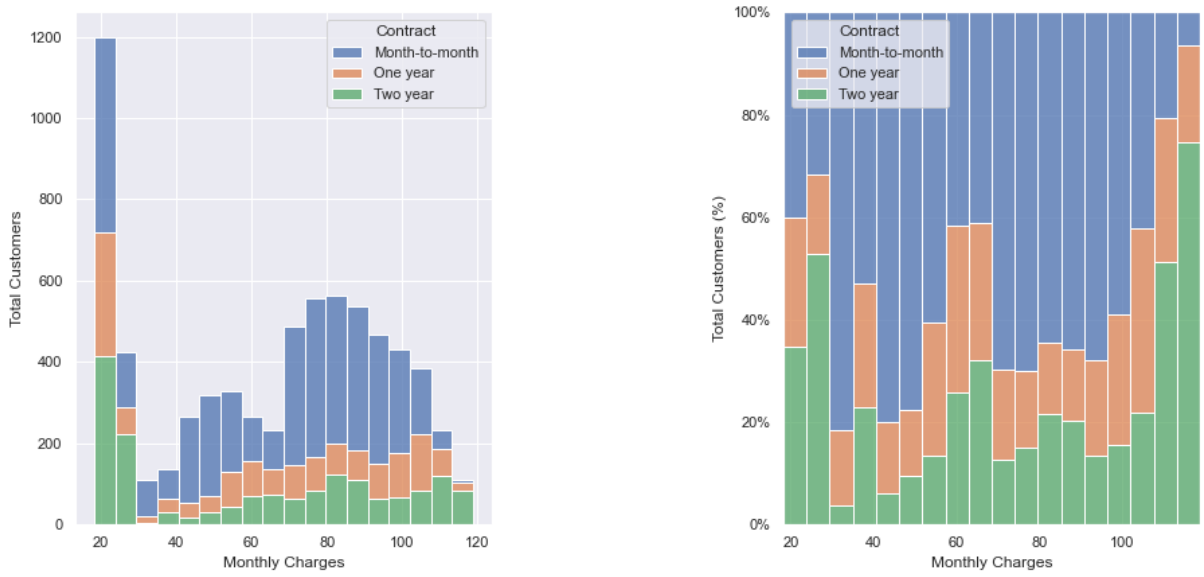

Fig 13. Contract distribution over monthly charges

It's quite interesting that our previous claim is not proven based on the above diagram. Above diagram tells us that the Month-to-Month contract mostly has low monthly charges. It means, the high churn rate for higher monthly charges is not because the contract is Month-to-Month.

Churn distribution by contract. The Month-to-Month contract type has a high churn rate, thus the contract feature is a good candidate for predictor. The high negative correlation means from Month-to-Month (0 index), One year (1 index) until Two year (2 index), the churn rate is decreased.

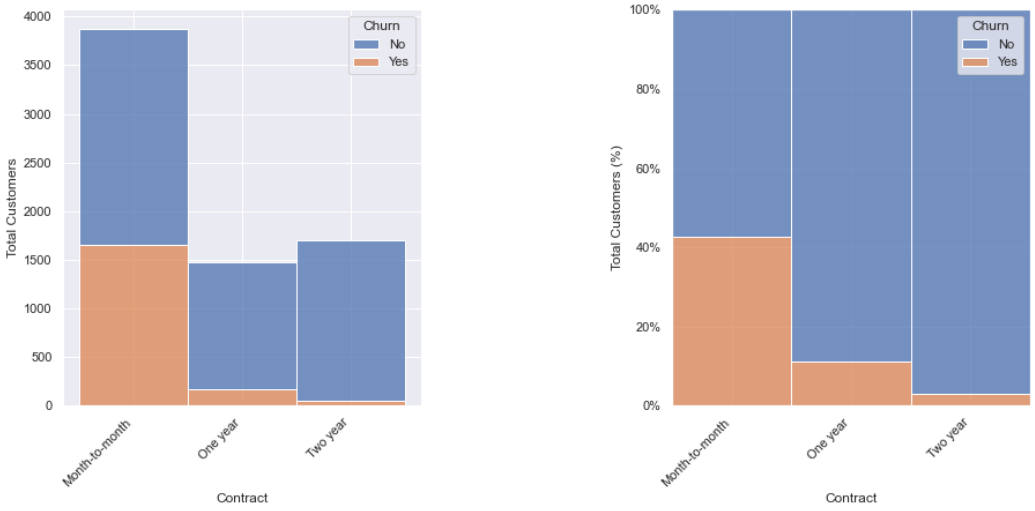

Fig 14. Churn distribution by contract type

Churn distribution by tenure. Tenure and churn have high negative correlation between them. It means, the longer the customer stays (high tenure value) the churn rate is decreasing. Thus, tenure is a good candidate as a predictor.

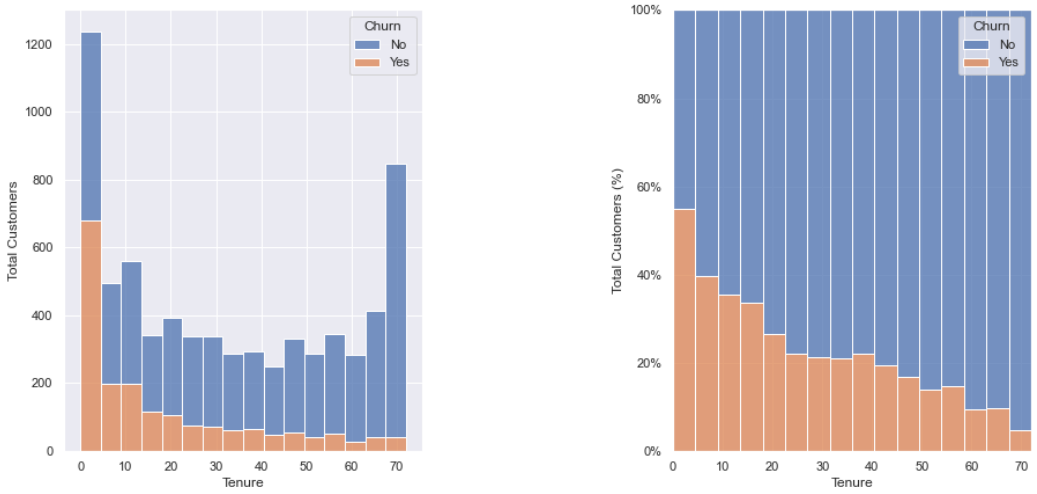

Fig 15. Churn distribution by tenure

Churn distribution by online security. The correlation between OnlineSecurity and Churn is a negative correlation. But, it's not the highest. See Fig 16, the rate is not always decreasing. But, the churn rate trend is decreasing. The high negative correlation in OnlineSecurity is a good candidate as a predictor feature.

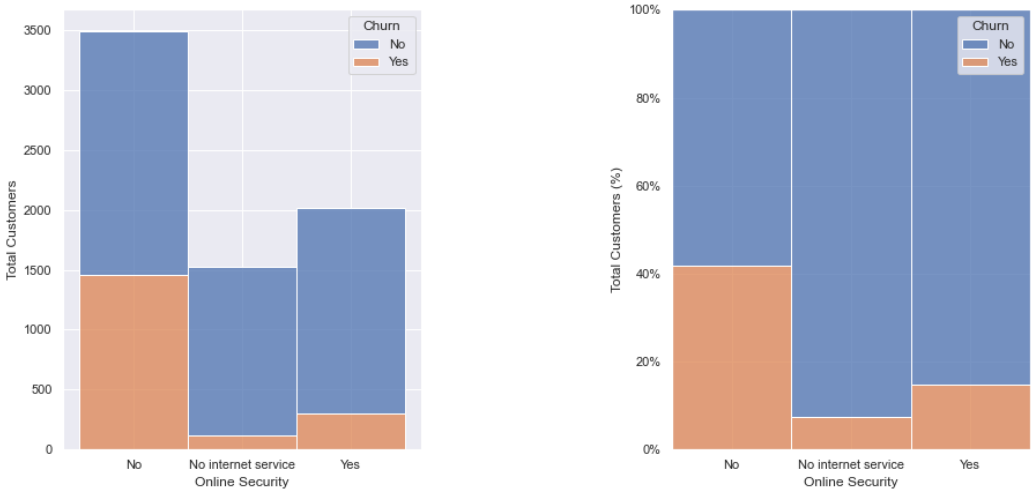

Fig 16. Churn distribution by online security

Conclusion

The highest positive correlation value is +0.19 (PaperlessBilling and MonthlyCharges), followed by +0.15 (SeniorCitizen) and +0.11 (PaymentMethod). The highest negative correlation value is -0.40 (Contract), followed by -0.35 (Tenure) and -0.29 (OnlineSecurity). Both positive and negative correlation would affect the target feature, thus it will be a good candidate as a predictor feature.

References

[1] Patil, P. (2018, March 23). *What is Exploratory Data Analysis?*. Retrieved from https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15.

[2] Bond, M. (2013, Oct 18). *What's In a Name: The Importance of Consistent Data Naming Conventions*. Retrieved from https://interworks.com/blog/mbond/2013/10/18/whats-name-importance-consistent-data-naming-conventions/.

[3] Browniee, J. (2020, June 12). *Ordinal and One-Hot Encodings for Categorical Data*. Retrieved from https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/.

[4] Munévar, J. P. G. (2021, Sept 30). *What is an alluvial diagram?*. Retrieved from https://www.datasketch.co/blog/data-visualization-alluvial-diagram/.

Appendix

1. GitHub repository of the project: https://github.com/ahmadichsan/python-task-d16
2. Google colab: https://bit.ly/telco-eda-ichsan