CUSTOMER CHURN PREDICTION

Churn rate:

Churn rate is a critical metric of customer satisfaction. Low churn rates mean happy customers; high churn rates mean customers are leaving you. A small rate of monthly/quarterly churn compounds over time. 1% monthly churn quickly translates to almost 12% yearly churn.Churn is a good indicator of growth potential.

• Predict which customers are going to churn
• Know why they are going to churn
CHURN

You can classify churn as:

1.Customer and revenue churn

2.Voluntary and involuntary churn

1.Customer and revenue churn:

Customer churn is simply the rate at which customers cancel their subscriptions. Also known as subscriber churn or logo churn, its value is represented in percentages. On the other hand, revenue churn is the loss in your monthly recurring revenue (MRR) at the beginning of the month. Customer churn and revenue churn aren’t always the same

2.Voluntary and involuntary Churn:

Voluntary churn is when the customer decides to cancel and takes the necessary steps to exit the service. It could be caused by dissatisfaction, or not receiving the value they expected. Involuntary churn happens due to situations such as expired payment details, server errors, insufficient funds, and other unpredictable predicaments.

The overall scope to build an ML-powered application to forecast customer churn is generic to standardized ML project structure that includes the following steps:

1.Defining problem and goal:

It’s essential to understand what insights you need to get from the analysis and prediction. Understand the problem and collect requirements, stakeholder pain points, and expectations.

2.Establishing data source:

Next, specify data sources that will be necessary for the modeling stage. Some popular sources of churn data are CRM systems, analytics services, and customer feedback.

3.Data preparation, exploration, and preprocessing:

Raw historical data for solving the problem and building predictive models needs to be transformed into a format suitable for machine learning algorithms. This step can also improve overall results by increasing the quality of data.

4.Modeling and testing:

This covers the development and performance validation of customers churn prediction models with various machine learning algorithms.

5.Deployment and monitoring:

This is the last stage in applying machine learning for churn rate prediction. Here, the most suitable model is sent into production. It can be either integrated into existing software, or become the core of a newly built application.

Dataset:

The sample data tracks a fictional telecommunications company, Telco. It's customer churn data sourced by the IBM Developer Platform, and it's available here. It includes a target label indicating whether or not the customer left within the last month, and other dependent features that cover demographics, services that each customer has signed up for, and customer account information. It has data for 7043 clients, with 20 features.

You can find this entire project on my Github.

Exploratory data analysis (EDA)

Let's critically explore the data to discover patterns and visualize how the features interact with the label (Churn or not).

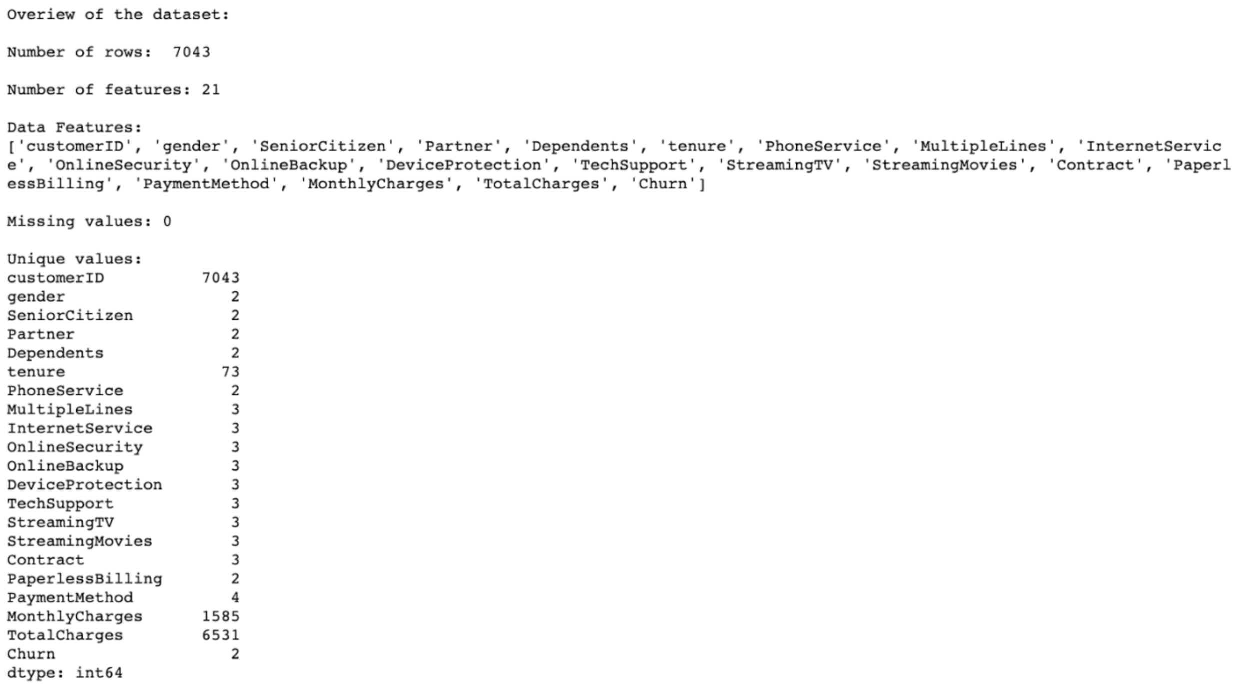
Read also

Exploratory Data Analysis for Natural Language Processing:

Let's first import libraries for EDA, load the data, and print the first five rows:

```
#Import libraries
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
pd.set_option('display.max_columns', None)

import plotly.express as px #for visualization
import matplotlib.pyplot as plt #for visualization

#Read the dataset
data_df = pd.read_csv("../data/churn.csv")

#Get overview of the data
def dataoveriew(df, message):
    print(f'{message}:n')
    print('Number of rows: ', df.shape[0])
    print("nNumber of features:", df.shape[1])
    print("nData Features:")
    print(df.columns.tolist())
    print("nMissing values:", df.isnull().sum().values.sum())
    print("nUnique values:")
    print(df.nunique())

dataoveriew(data_df, 'Overview of the dataset')
```

Output:



```
Overiew of the dataset:

Number of rows:  7043

Number of features: 21

Data Features:
['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents', 'tenure', 'PhoneService', 'MultipleLines', 'InternetServic
e', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract', 'Paperl
essBilling', 'PaymentMethod', 'MonthlyCharges', 'TotalCharges', 'Churn']

Missing values: 0

Unique values:
customerID          7043
gender                 2
SeniorCitizen          2
Partner                2
Dependents             2
tenure                73
PhoneService           2
MultipleLines          3
InternetService        3
OnlineSecurity         3
OnlineBackup           3
DeviceProtection       3
TechSupport            3
StreamingTV            3
StreamingMovies        3
Contract               3
PaperlessBilling       2
PaymentMethod          4
MonthlyCharges      1585
TotalCharges        6531
Churn                  2
dtype: int64
```

The dataset has 7043 rows and 21 columns.

There are 17 categorical features:

- **CustomerID:** Customer ID unique for each customer
- **Gender:** Whether the customer is a male or a female
- **SeniorCitizen:** Whether the customer is a senior citizen or not (1, 0)
- **Partner:** Whether the customer has a partner or not (Yes, No)
- **Dependent:** Whether the customer has dependents or not (Yes, No)
- **PhoneService:** Whether the customer has a phone service or not (Yes, No)
- **MultipeLines:** Whether the customer has multiple lines or not (Yes, No, No phone service)
- **InternetService:** Customer's internet service provider (DSL, Fiber optic, No)
- **OnlineSecurity:** Whether the customer has online security or not (Yes, No, No internet service)
- **OnlineBackup:** Whether the customer has an online backup or not (Yes, No, No internet service)
- **DeviceProtection:** Whether the customer has device protection or not (Yes, No, No internet service)
- **TechSupport:** Whether the customer has tech support or not (Yes, No, No internet service)
- **StreamingTV:** Whether the customer has streaming TV or not (Yes, No, No internet service)
- **StreamingMovies:** Whether the customer has streaming movies or not (Yes, No, No internet service)
- **Contract:** The contract term of the customer (Month-to-month, One year, Two years)
- **PaperlessBilling:** The contract term of the customer (Month-to-month, One year, Two years)
- **PaymentMethod:** The customer's payment method (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic))
- **Tenure:** Number of months the customer has stayed with the company
- **MonthlyCharges:** The amount charged to the customer monthly
- **TotalCharges:** The total amount charged to the customer
- **Churn:** Whether the customer churned or not (Yes or No)

These features can also be subdivided into:

Demographic customer information:

- gender , SeniorCitizen , Partner , Dependents

Services that each customer has signed up for:

- PhoneService , MultipleLines , InternetService , OnlineSecurity , OnlineBackup , DeviceProtection , TechSupport , StreamingTV , StreamingMovies,

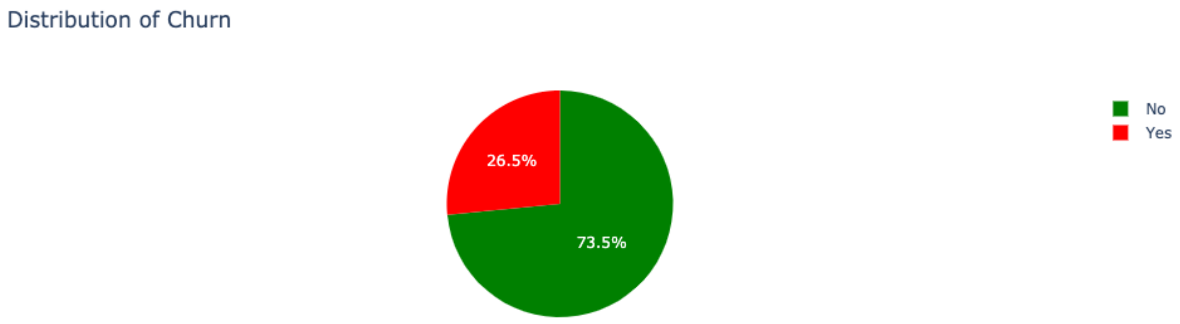
Customer account information:

- tenure , Contract , PaperlessBilling , PaymentMethod , MonthlyCharges , TotalCharges

Let's explore the target variable.

```
target_instance = data_df["Churn"].value_counts().to_frame()
target_instance = target_instance.reset_index()
target_instance = target_instance.rename(columns={'index': 'Category'})
fig = px.pie(target_instance, values='Churn', names='Category',
color_discrete_sequence=["green", "red"],
            title='Distribution of Churn')
fig.show()
```

Output:



We're trying to predict users that left the company in the previous month. It's a binary classification problem with an unbalanced target.

- Churn: No – 73.5%
- Churn: Yes – 26.5%

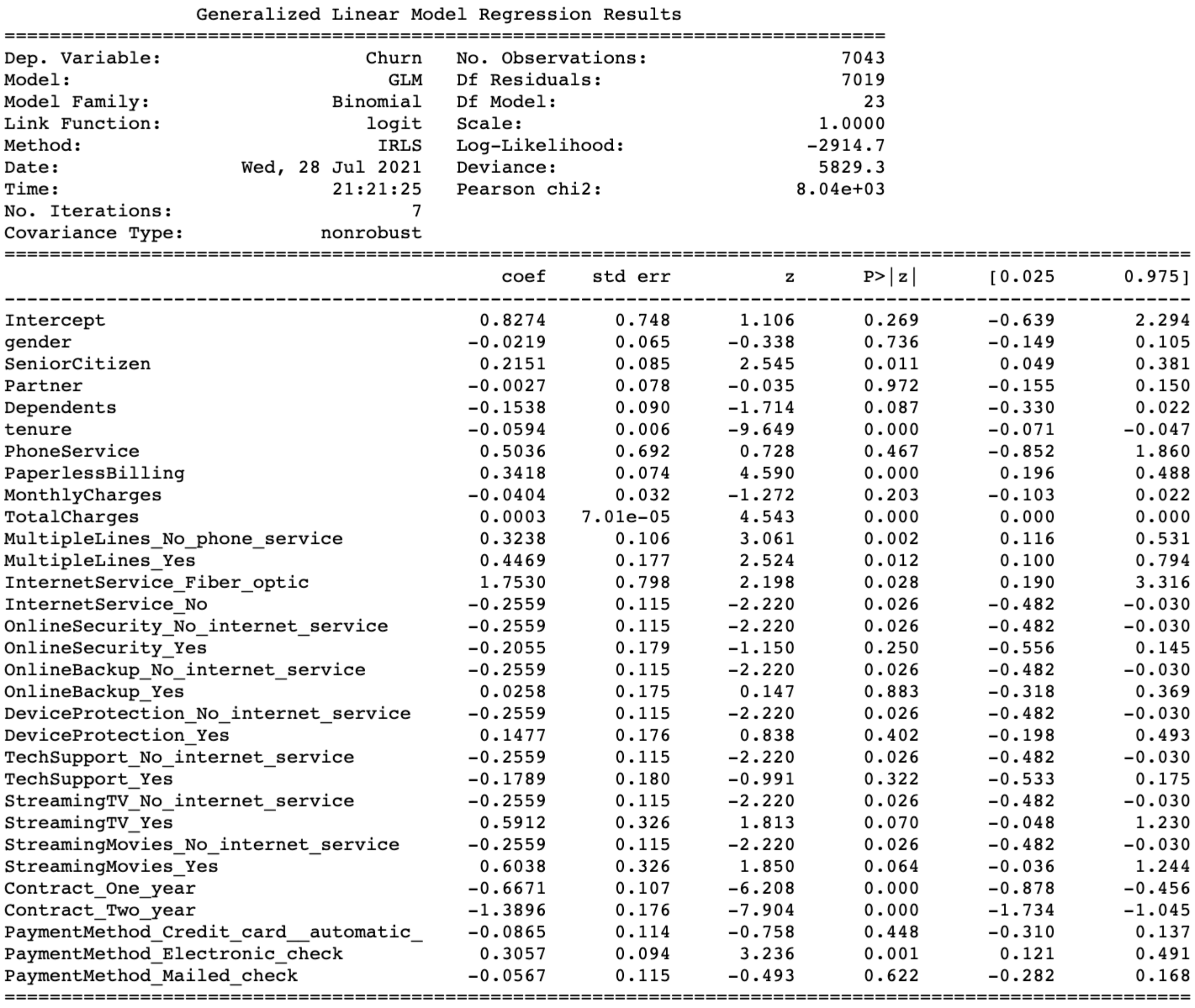
Let's use the generalized linear model (GLM) to gain some statistics of the respective features with the target:

```
import statsmodels.api as sm
import statsmodels.formula.api as smf

#Change variable name separators to '_'
all_columns = [column.replace(" ", "_").replace("(", "_").replace(")", "_").replace("-", "_") for
column in data_df.columns]


#Effect the change to the dataframe column names
data_df.columns = all_columns

#Prepare it for the GLM formula
glm_columns = [e for e in all_columns if e not in ['customerID', 'Churn']]
```

glm_columns = ' + '.join(map(str, glm_columns))

#Fiting it to the Generalized Linear Model
glm_model = smf.glm(formula=f'Churn ~ {glm_columns}', data=data_df,
family=sm.families.Binomial())
res = glm_model.fit()
print(res.summary())

Output:



```
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:                  Churn   No. Observations:                 7043
Model:                            GLM   Df Residuals:                     7019
Model Family:                Binomial   Df Model:                           23
Link Function:                  logit   Scale:                          1.0000
Method:                          IRLS   Log-Likelihood:                -2914.7
Date:                Wed, 28 Jul 2021   Deviance:                       5829.3
Time:                        21:21:25   Pearson chi2:                 8.04e+03
No. Iterations:                     7
Covariance Type:            nonrobust
=========================================================================================================
                                            coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------------------------------
Intercept                                 0.8274      0.748      1.106      0.269      -0.639       2.294
gender                                   -0.0219      0.065     -0.338      0.736      -0.149       0.105
SeniorCitizen                             0.2151      0.085      2.545      0.011       0.049       0.381
Partner                                  -0.0027      0.078     -0.035      0.972      -0.155       0.150
Dependents                               -0.1538      0.090     -1.714      0.087      -0.330       0.022
tenure                                   -0.0594      0.006     -9.649      0.000      -0.071      -0.047
PhoneService                              0.5036      0.692      0.728      0.467      -0.852       1.860
PaperlessBilling                          0.3418      0.074      4.590      0.000       0.196       0.488
MonthlyCharges                           -0.0404      0.032     -1.272      0.203      -0.103       0.022
TotalCharges                              0.0003   7.01e-05      4.543      0.000       0.000       0.000
MultipleLines_No_phone_service            0.3238      0.106      3.061      0.002       0.116       0.531
MultipleLines_Yes                         0.4469      0.177      2.524      0.012       0.100       0.794
InternetService_Fiber_optic               1.7530      0.798      2.198      0.028       0.190       3.316
InternetService_No                       -0.2559      0.115     -2.220      0.026      -0.482      -0.030
OnlineSecurity_No_internet_service       -0.2559      0.115     -2.220      0.026      -0.482      -0.030
OnlineSecurity_Yes                       -0.2055      0.179     -1.150      0.250      -0.556       0.145
OnlineBackup_No_internet_service         -0.2559      0.115     -2.220      0.026      -0.482      -0.030
OnlineBackup_Yes                          0.0258      0.175      0.147      0.883      -0.318       0.369
DeviceProtection_No_internet_service     -0.2559      0.115     -2.220      0.026      -0.482      -0.030
DeviceProtection_Yes                      0.1477      0.176      0.838      0.402      -0.198       0.493
TechSupport_No_internet_service          -0.2559      0.115     -2.220      0.026      -0.482      -0.030
TechSupport_Yes                          -0.1789      0.180     -0.991      0.322      -0.533       0.175
StreamingTV_No_internet_service          -0.2559      0.115     -2.220      0.026      -0.482      -0.030
StreamingTV_Yes                           0.5912      0.326      1.813      0.070      -0.048       1.230
StreamingMovies_No_internet_service      -0.2559      0.115     -2.220      0.026      -0.482      -0.030
StreamingMovies_Yes                       0.6038      0.326      1.850      0.064      -0.036       1.244
Contract_One_year                        -0.6671      0.107     -6.208      0.000      -0.878      -0.456
Contract_Two_year                        -1.3896      0.176     -7.904      0.000      -1.734      -1.045
PaymentMethod_Credit_card__automatic_    -0.0865      0.114     -0.758      0.448      -0.310       0.137
PaymentMethod_Electronic_check            0.3057      0.094      3.236      0.001       0.121       0.491
PaymentMethod_Mailed_check               -0.0567      0.115     -0.493      0.622      -0.282       0.168
=========================================================================================================
```