**[Predict Customer Churn Using Python & Machine Learning](https://medium.com/@randerson112358/predict-customer-churn-using-python-machine-learning-b92f39685f4c)**

[randerson112358](https://medium.com/@randerson112358?source=post_page-----b92f39685f4c--------------------------------)

[randerson112358](https://medium.com/@randerson112358?source=post_page-----b92f39685f4c--------------------------------)

[Apr 6·9 min read](https://medium.com/@randerson112358/predict-customer-churn-using-python-machine-learning-b92f39685f4c?source=post_page-----b92f39685f4c--------------------------------)

Build a model to develop a strategic retention plan



**Customer Churn**occurs when subscribers or customers stop doing business with a company or service. A business typically treats a customer as churned once a specific amount of time has passed since the customers last interaction with the business or service.

Retaining customers is obviously important for companies, because it boosts that companies revenue and helps the company to build a meaningful relationship with the customer. What might not be so obvious is that customer retention is actually more valuable than customer acquisition and there is a lot of data to back this claim.

5 Reasons Why Customer Retention Is Important

Below you can find 5 reasons why customer retention is important according to [dcrstrategies.com](https://www.dcrstrategies.com/customer-incentives/5-reasons-customer-retention-business/):

*1. Companies save money on marketing.*

*2. Repeat purchases from repeat customers means repeat profit.*

*3. Free word-of-mouth advertising.*

*4. Retained customers provide valuable feedback.*

*5. Previous customers will pay premium prices.*

In this article, I will attempt to create a model that can accurately predict / classify if a customer is likely to churn. I will also analyze the data to come up with a possible strategic retention plan.

The data set that will be used in this analysis will come from the Telco company.

If you prefer not to read this article and would like a video representation of it, you can check out the [**YouTube Video**](https://youtu.be/ocMd2loRfWE) . It goes through everything in this article with a little more detail and will help make it easy for you to start programming even if you don’t have the programming language Python installed on your computer. Or you can use both as supplementary materials for learning!

Programming:

The first thing that I like to do before writing a single line of code is to put in a description in comments of what the code does. This way I can look back on my code and know exactly what it does.

**# Description: This is a python program to predict customer churn**

Next import some libraries that will be used throughout this program.

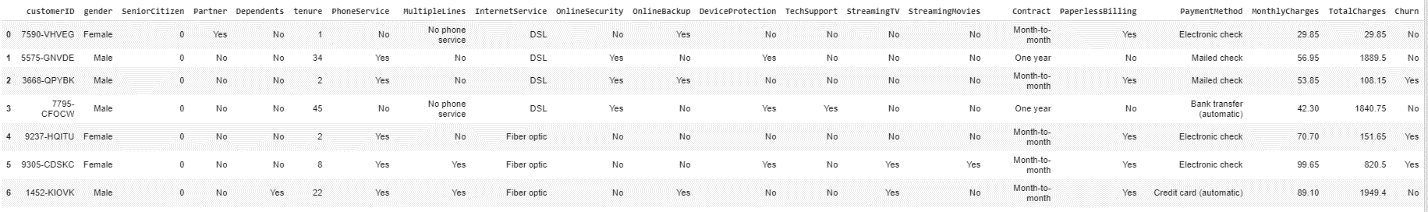
**#Import the library**  
import pandas as pd  
import numpy as np  
import sklearn  
import matplotlib.pyplot as plt  
import seaborn as sns  
from sklearn.preprocessing import LabelEncoder  
from sklearn.preprocessing import StandardScaler  
from sklearn.metrics import classification\_report  
from sklearn.linear\_model import LogisticRegression  
from sklearn.model\_selection import train\_test\_split  
%matplotlib inline

Load the data.

***#Load the data***   
**from** **google.colab** **import** files *# Use to load data on Google Colab*   
uploaded = files.upload() *# Use to load data on Google Colab*

Store the data into a data frame and print the first 7 rows of data.

**#Load the data into the data frame**  
df = pd.read\_csv('WA\_Fn-UseC\_-Telco-Customer-Churn.csv')  
df.head(7)



The first 7 rows of the data set

Analyze The Data

Get the number of rows and columns in the data set.

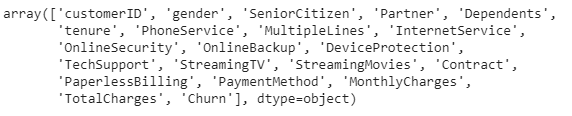
df.shape

Image for post

7043 rows, & 21 columns

So, the data set contains 7,043 customers and 21 data points on each customer. Next, I will show all of the column names in the data set.

**#Show all of the column names**  
df.columns.values



A list of all of the columns

Check for any missing data in the data set.

**#Check for na or missing data**  
df.isna().sum()

Image for post

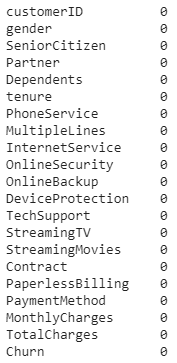
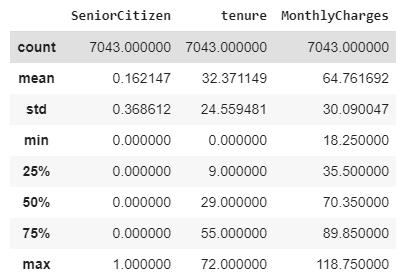


Image showing no missing data

I didn’t see any missing data. Let’s show some statistics on the data set.

**#Show statistics on the current data**   
df.describe()



Statistics on the data sets numeric values

From the statistics above, we can see that the longest tenure is 72 months or 6 years, and the maximum monthly charge is $118.75. The minimum monthly charge is about $30.09. The customer can expect to have a monthly charge of about $64.76. I am assuming the charges are in United States Dollars (USD).

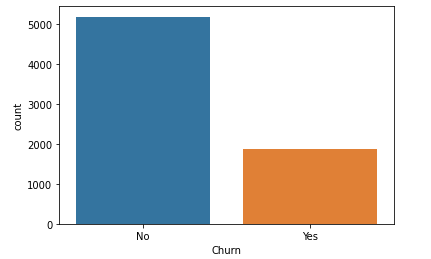
Next, get the number of customers that churned and were retained (did not churn).

**#Get the number of customers that churned**  
df['Churn'].value\_counts()

Image for post

About 5,174 customers were retained (did not churn) and 1,869 customers churned. Let’s show this count visually using a bar plot.

**#Visualize the count of customer churn**  
sns.countplot(df['Churn'])



Visual count of customer Churn

I want to know what percentage of customers are leaving.

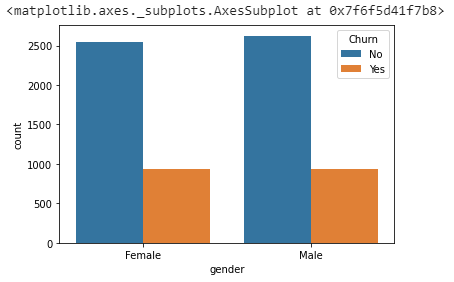
**#What percentage of customers are leaving ?**  
retained = df[df.Churn == 'No']  
churned = df[df.Churn == 'Yes']  
num\_retained = retained.shape[0]  
num\_churned = churned.shape[0]**#Print the percentage of customers that stayed and left**  
print( num\_retained / (num\_retained + num\_churned) \* 100 , "% of customers stayed with the company.")**#Print the percentage of customers that stayed and left**  
print( num\_churned / (num\_retained + num\_churned) \* 100,"% of customers left the company.")

Image for post

So, about 73.46% of the customers stayed or were retained and about 26.54% of the customers churned. This is important information for when I try to evaluate my model to predict customer churn, because it means that just by always guessing a random customer to have been retained from the data set, I have a 73.46% chance of guessing correctly. So, I want the model’s accuracy to classify/predict if a customer will churn to be higher than that percentage.

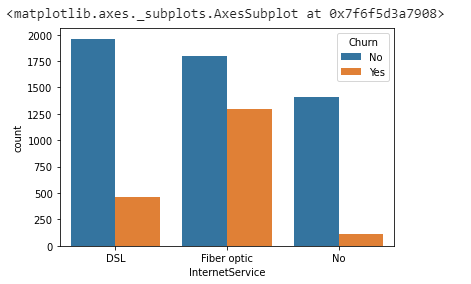
Let’s take a look at the churn count by gender.

**#Visualize the churn count for both Males and Females**  
sns.countplot(x='gender', hue='Churn',data = df)



From the plot above, it looks like gender does not play a role in customer churn. Let’s visualize the churn count for the internet service.

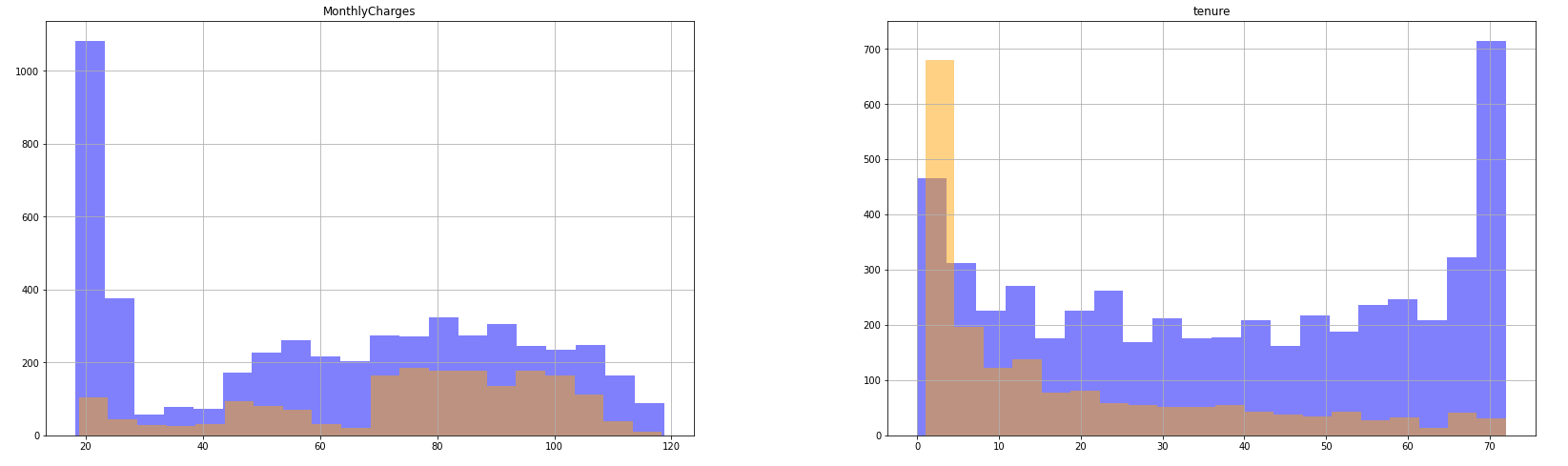
**#Visualize the churn count for the internet service**  
sns.countplot(x='InternetService', hue='Churn', data = df)



The chart above is interesting, because it helps me to discriminate retained and churned customers, it shows that most customers that churned had the Fiber optic internet service, and the most customers that were retained had DSL internet service. Maybe the company should only provide DSL as the internet service or stop providing Fiber optics for it’s internet service.

Next, I want to take a look visually at the tenure and monthly charges columns to see if there is any discrimination for customer churn. To do this I will create a histogram plot.

numerical\_features = ['tenure', 'MonthlyCharges']  
fig, ax = plt.subplots(1, 2, figsize=(28, 8))  
df[df.Churn == 'No'][numerical\_features].hist(bins=20, color="blue", alpha=0.5, ax=ax)  
df[df.Churn == 'Yes'][numerical\_features].hist(bins=20, color="orange", alpha=0.5, ax=ax)



From the two charts above, I can clearly see that there is some discrimination in the data. The monthly charges chart (on the left) shows that most of the loyal customers that stayed with the company had a monthly charge between $20 and $30. Most of the customers that churned had a monthly charge of $70 to $100. Maybe the company should lower the monthly charges to retain customers.

The tenure chart (on the right) shows some discrimination as well. From the chart, I can see that most of the customers that churned had between 1 and 9 months with the company, while most of the retained customers had a tenure between 24 and 72 months which is 2 to 6 years. So, it may be in the companies best interest to try everything they can to keep their customers for at least 2 years.

Data Processing & Cleaning

Now it’s time to do some data processing and cleaning. First I want to get rid of columns that are unnecessary. Immediately, I can see that the customerID column will not add any more value to the model or the analysis seeing as how it’s just an ID for the customer. For now that will be the only column that I remove from the data set that will be used to create the model, but there could be more.

**#Remove the unnecessary column customerID**  
cleaned\_df = df = df.drop('customerID', axis=1)

Let’s take a look at the number of rows and columns in the new data set.

**#Look at the number of rows and cols in the new data set**  
cleaned\_df.shape

Image for post

7,043 rows and 20 columns

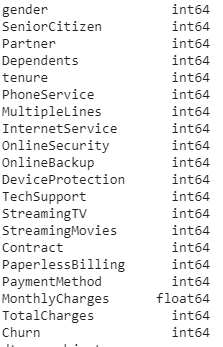
I can see that the new data set contains 7,043 rows of data and 20 columns (one less than the number of columns in the original data set).

Next, I will convert all non-numeric columns / categorical columns to numerical columns.

**#Convert all the non-numeric columns to numerical data types**  
for column in cleaned\_df.columns:  
 if cleaned\_df[column].dtype == np.number:  
 continue  
 cleaned\_df[column] = LabelEncoder().fit\_transform(cleaned\_df[column])

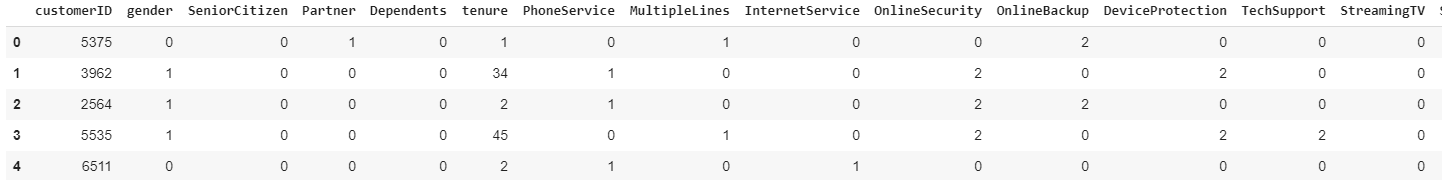
I want to check that the conversion was done successfully, so let’s take a look at the data set’s data types, and show a few rows with the numerical data.

**#Check the new data set data types**  
cleaned\_df.dtypes



All numerical data types from the new data set

**#Show the first 5 rows of the new data set**  
cleaned\_df.head()



Sample of the first 5 rows of the new data set

Perfect ! The conversion worked and all of the data types are numerical values.

I want to scale the data to be values between 0 and 1 inclusively .

**#Scale the cleaned data**X = cleaned\_df.drop('Churn', axis = 1)   
y = cleaned\_df['Churn']**#Standardizing/scaling the features**  
X = StandardScaler().fit\_transform(X)

Now that the data has been scaled, I need to split the data into training and testing data sets. Exactly 80% of the original cleaned data will be used for training the model and 20% will be used for testing the model.

**#Split the data into 80% training and 20% testing**x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

Create The Model

Finally, it’s time to create and train the model. The statistical model that I will use will be Logistic Regression. This model is good for binary classification.

**#Create the model**model = LogisticRegression()**#Train the model**  
model.fit(x\_train, y\_train)

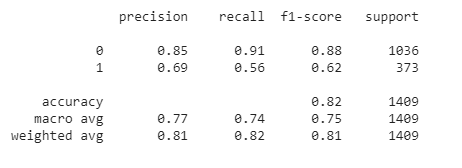
Evaluate The Model

To evaluate the logistic regression model, I will print the predictions and look at the test statistics like precision, recall and the f-1 score.

predictions = model.predict(x\_test)**#printing the predictions**  
print(predictions)

Image for post

**#Check precision, recall, f1-score**  
print( classification\_report(y\_test, predictions) )



From the report, I can see that the recall of the model is about 91% meaning the model correctly identified about 91% of the customers that were retained and missed about 9%.

The precision of the model was about 85% and the f1-score was about 88%. The accuracy of the model was about 82% which is better than the 73.46% that I could’ve done just by guessing a customer would always stay with the company.

The company may want to lower it’s monthly charges at least for new customers for the first 2 years and stop providing fiber optics internet service, this may be a good strategy to help retain their customers and reduce customer Churn.

Maybe with some more analysis on the data and tweaking of the program, I can improve this models performance and accuracy score.

If you are interested in reading more on machine learning to immediately get started with problems and examples then I strongly recommend you check out [**Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**](https://www.amazon.com/gp/product/1491962291?ie=UTF8&tag=medium074-20&camp=1789&linkCode=xm2&creativeASIN=1491962291). It is a great book for helping beginners learn how to write machine learning programs, and understanding machine learning concepts.



[Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems](https://www.amazon.com/gp/product/1491962291?ie=UTF8&tag=medium074-20&camp=1789&linkCode=xm2&creativeASIN=1491962291)

Thanks for reading this article I hope its helpful to you all ! If you enjoyed this article and found it helpful please leave some claps to show your appreciation. Keep up the learning, and if you like machine learning, mathematics, computer science, programming or algorithm analysis, please visit and subscribe to my [YouTube](https://www.youtube.com/channel/UCaV_0qp2NZd319K4_K8Z5SQ) channels ([randerson112358](https://www.youtube.com/channel/UCaV_0qp2NZd319K4_K8Z5SQ) & [compsci112358](https://www.youtube.com/channel/UCbmb5IoBtHZTpYZCDBOC1CA) ).