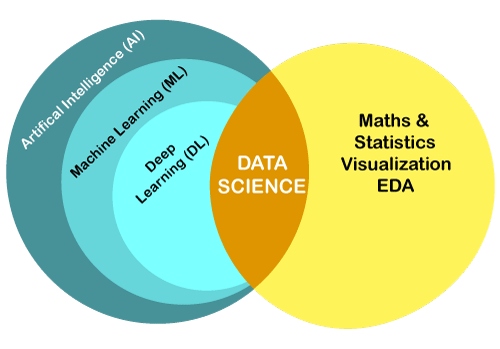
# **Customer Churn Analysis**

**Datatrained Data Science Project**



**ABSTRACT**

**Problem Statement:**

* **Customer churn is when a company’s customers stop doing business with that company. Businesses are very keen on measuring churn because keeping an existing customer is far less expensive than acquiring a new customer. New business involves working leads through a sales funnel, using marketing and sales budgets to gain additional customers. Existing customers will often have a higher volume of service consumption and can generate additional customer referrals.**
* [Customer churn](https://www.chargebee.com/resources/glossaries/what-is-customer-churn/) (also known as logo churn) is the ratio of the number of customers lost during a given period (typically a month or a year) and the number of customers present at the beginning of that period. It is usually expressed in percentage terms as an annual or monthly figure.

**Customer Churn Prevention**

* Customer retention can be achieved with good customer service and products. But the most effective way for a company to prevent attrition of customers is to truly know them. The vast volumes of data collected about customers can be used to build churn prediction models. Knowing who is most likely to defect means that a company can prioritise focused marketing efforts on that subset of their customer base.
* Preventing customer churn is critically important to the telecommunications sector, as the barriers to entry for switching services are so low.
* Preventing customer churn is critically important to the telecommunications sector, as the barriers to entry for switching services are so low.
* Monitor your churn numbers regularly – anything that isn’t measured can’t be improved
* Look for the dropoff points – where in your app users begin to disengage
* Identify the causes of churn so you can improve the user experience and keep users coming back
* Customer attrition (a.k.a customer churn) is one of the biggest expenditures of any organization. If we could figure out why a customer leaves and when they leave with reasonable accuracy, it would immensely help the organization to strategize their retention initiatives manifold. Let’s make use of a customer transaction dataset from the github to understand the key steps involved in predicting customer attrition in Python.
* Supervised Machine Learning is nothing but learning a function that maps an input to an output based on example input-output pairs. A supervised machine learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. Given that we have data on current and prior customer transactions in the telecom dataset, this is a standardized supervised classification problem that tries to predict a binary outcome (Y/N).
* By the end of this article, let’s attempt to solve some of the key business challenges pertaining to customer attrition like say, (1) what is the likelihood of an active customer leaving an organization? (2) what are key indicators of a customer churn? (3) what retention strategies can be implemented based on the results to diminish prospective customer churn?
* In real-world, we need to go through seven major stages to successfully predict customer churn:
* Section A: Data Preprocessing
* Section B: Data Evaluation
* Section C: Model Selection
* Section D: Model Evaluation
* Section E: Model Improvement
* Section F: Future Predictions
* Section G: Model Deployment
* To understand the business challenge and the proposed solution, I would recommend you to [download](https://github.com/srees1988/predict-churn-py) the dataset and to code with me. Feel free to ask me if you have any questions as you work along. Let’s look into each one of these aforesaid steps in detail here below

**Standard libraries for data analysis:**   
import numpy as np  
import matplotlib.pyplot as plt  
import pandas as pd  
from scipy.stats import norm, skew  
from scipy import stats  
import statsmodels.api as sm# sklearn modules for data preprocessing:from sklearn.impute import SimpleImputer  
from sklearn.preprocessing import LabelEncoder, OneHotEncoder  
from sklearn.compose import ColumnTransformer  
from sklearn.preprocessing import OneHotEncoder  
from sklearn.model\_selection import train\_test\_split  
from sklearn.preprocessing import StandardScaler#sklearn modules for Model Selection:from sklearn import svm, tree, linear\_model, neighbors  
from sklearn import naive\_bayes, ensemble, discriminant\_analysis, gaussian\_process  
from sklearn.neighbors import KNeighborsClassifier  
from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis  
from xgboost import XGBClassifier  
from sklearn.linear\_model import LogisticRegression  
from sklearn.svm import SVC  
from sklearn.neighbors import KNeighborsClassifier  
from sklearn.naive\_bayes import GaussianNB  
from sklearn.tree import DecisionTreeClassifier  
from sklearn.ensemble import RandomForestClassifier#sklearn modules for Model Evaluation & Improvement:  
   
from sklearn.metrics import confusion\_matrix, accuracy\_score   
from sklearn.metrics import f1\_score, precision\_score, recall\_score, fbeta\_score  
from statsmodels.stats.outliers\_influence import variance\_inflation\_factor  
from sklearn.model\_selection import cross\_val\_score  
from sklearn.model\_selection import GridSearchCV  
from sklearn.model\_selection import ShuffleSplit  
from sklearn.model\_selection import KFold  
from sklearn import feature\_selection  
from sklearn import model\_selection  
from sklearn import metrics  
from sklearn.metrics import classification\_report, precision\_recall\_curve  
from sklearn.metrics import auc, roc\_auc\_score, roc\_curve  
from sklearn.metrics import make\_scorer, recall\_score, log\_loss  
from sklearn.metrics import average\_precision\_score#Standard libraries for data visualization:import seaborn as sn  
from matplotlib import pyplot  
import matplotlib.pyplot as plt  
import matplotlib.pylab as pylab  
import matplotlib   
%matplotlib inline  
color = sn.color\_palette()  
import matplotlib.ticker as mtick  
from IPython.display import display  
pd.options.display.max\_columns = None  
from pandas.plotting import scatter\_matrix  
from sklearn.metrics import roc\_curve#Miscellaneous Utilitiy Libraries:  
   
import random  
import os  
import re  
import sys  
import timeit  
import string  
import time  
from datetime import datetime  
from time import time  
from dateutil.parser import parse  
import joblib

**Overview of the Problem**

We will examine customer data from IBM Sample Data Sets with the aim of building and comparing several customer churn prediction models.

**Dataset Attributes**

* The dataset contains the data of the customer. On the basis of the data we have to predict the total charges by the customer. The dataset contains the data like 'customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents','tenure', 'PhoneService', 'MultipleLines', 'InternetService','OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport','StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling','PaymentMethod', 'MonthlyCharges', 'TotalCharges' and'Churn‘.
* Target is to predict the "Total charges" paid by the customer.
* Churn rate (sometimes called attrition rate), in its broadest sense, is a measure of the number of individuals or items moving out of a collective group over a specific period. It is one of two primary factors that determine the steady-state level of customers a business will support. The term is used in many contexts, but is most widely applied in business with respect to a contractual customer base, for example in businesses with a subscriber-based service model such as mobile telephone networks and pay TV operators. The term is also used to refer to participant turnover in peer-to-peer networks. Churn rate is an input into customer lifetime value modeling, and can be part of a simulator used to measure return on marketing investment using marketing mix modeling.

**Steps to be followed (tentative time required)**

* Understand the problem and objective (1 hour)
* Understand the data and develop some business sense. (4-5 hours)
* EDA (if you think is required in this case). (5-6 Hours)
* Data Cleaning and preparation (4-5 Hours)
* Feature engineering (4-5 Hours)
* Model Building (try various techniques and at the end justify why you chose a technique over others) (3-4 hours)
* Testing and Cross-validation (3-4 hours)
* Final results, recommendations and plots/visualizations. (4-5 hours)
* BONUS: Any other insights or recommendations that you can give from the data which will help the business. (Subjective)
* Preparing the deck: 6-7 hours
* [Actual time might vary from person to person and step to step, this is just indicative] The final solution should be in form of a deck showing all the steps above. It will be judged on the following criteria:
* How well have you adhered to the modelling process discipline?
* Do your results make business sense, how have you used business intuition to take decisions during the modelling exercise, including but not limited to the following: -
* Deciding segmentation (if you choose to have segmentation)
* EDA, Feature engineering
* Choosing variables to be put in models
* Deciding a cut-off
* Performance of your model on test data.
* Precision
* Recall
* Random forest regressor & Cross validation (best model with more than 90% accuracy)
* Model Saving
* Conclusion
* The project and presentation will be assessed & graded on completion. Details on this will be provided separately.

**Elaboration of all steps**

* Importing required libraries such as pandas, numpy, matplotlib.pyplot, seaborn, warnings.
* Checking shape of the data set using command df.shape.
* Checking data type present in the dataset using command df.dtypes.
* From the head & column methods, we get an idea that this is a telco customer churn dataset where each record entails the nature of subscription, tenure, frequency of payment and churn (signifying their current status).
* Always keep an eye onto the missing values in a dataset. The missing values could mess up model building and accuracy. Hence we need to take care of missing values (if any) before we compare and select a model.
* Checking null values present in the dataset using command df.isnull().sum().
* Identify unique values: ‘Payment Methods’ and ‘Contract’ are the two categorical variables in the dataset. When we look into the unique values in each categorical variables, we get an insight that the customers are either on a month-to-month rolling contract or on a fixed contract for one/two years. Also, they are paying bills via credit card, bank transfer or electronic checks.
* Check target variable distribution: Let’s look at the distribution of churn values. This is quite a simple yet crucial step to see if the dataset upholds any class imbalance issues. As you can see below, the data set is imbalanced with a high proportion of active customers compared to their churned counterparts.
* Clean the dataset.
* Take care of missing data: As we saw earlier, the data provided has no missing values and hence this step is not required for the chosen dataset. I would like to showcase the steps here for any future references.
* Find the average and fill missing values programmatically: If we had any missing values in the numeric columns of the dataset, then we should find the average of each one of those columns and fill their missing values. Here’s a snippet of code to do the same step programmatically.
* Revalidate NA’s: It’s always a good practice to revalidate and ensure that we don’t have any more null values in the dataset.
* Making DataFrame for the Nominal Data.
* Data Visualization.
* For the nomial categorical data we will use countplot as it will give the frequency of the classes of the columns.
* Making dataframe of the ordinal data.
* Encoding of DataFrame. For this import from sklearn.preprocessing import OrdinalEncoder. Machine Learning algorithms can typically only have numerical values as their independent variables. Hence label encoding is quite pivotal as they encode categorical labels with appropriate numerical values. Here we are label encoding all categorical variables that have only two unique values. Any categorical variable that has more than two unique values are dealt with Label Encoding and one-hot Encoding in the subsequent sections.
* Statistical Summary of Dataset using command df.describe().
* Statistical Summary helps in checking outliers present in the dataset.
* If mean is more than 50th median & large difference between max & 75th median for any of the columns respectively indicates that outliers present in those columns.
* Correlation of the columns with the target columns using command df.corr()
* Heat map is used to visualize the correlation of the columns with the target columns.
* A few observations can be made based on the histograms for numerical variables:
* Gender distribution shows that the dataset features a relatively equal proportion of male and female customers. Almost half of the customers in our dataset are female whilst the other half are male.
* Most of the customers in the dataset are younger people.
* Not many customers seem to have dependents whilst almost half of the customers have a partner.
* There are a lot of new customers in the organization (less than 10 months old) followed by a loyal customer segment that stays for more than 70 months on average.
* Most of the customers seem to have phone service and 3/4th of them have opted for paperless Billing
* Checking skewness present in the dataset using command df.skew()
* Keeping +/-0.5 as the range for skewness, here are the columns which does not lie within this range.
* Senior Citizen- categorical.
* Dependents- categorical.
* Phone Service- categorical.
* Contract- categorical.
* Total Charges- target variable.
* Churn- categorical.
* Outliers Check using boxplot.
* For outlier removal import z score & after outlier removal calculate percentage of data loss.
* Separating the columns into features and target. Let’s try to explore and visualize our data set by doing distribution of independent variables to better understand the patterns in the data and to potentially form some hypothesis.
* Scaling the data using Min-Max Scaler

Liberaries required for this are:

**from** sklearn.preprocessing **import** MinMaxScaler

mms**=**MinMaxScaler()

**from** sklearn.linear\_model **import** LinearRegression

lr**=**LinearRegression()

**from** sklearn.metrics **import** r2\_score

**from** sklearn.model\_selection **import** train\_test\_split

* Find the best random\_State.
* Regularization.
* Ensemble technique.
* Liberaries required are:
* **from** sklearn.model\_selection **import** GridSearchCV
* **from** sklearn.ensemble **import** RandomForestRegressor
* We are getting model accuracy and cross validation both as 99.8% which shows our model is performing extremely well.
* Model Saving via command import pickle.
* Conclusion.
* Github Link: <https://github.com/Tannu19/world-happiness-proj/blob/main/Customer%20Churn%20Analysis%20(Tannu%20Dass).ipynb>