For this lab we will build a model on customer churn binary classification problem. You will be using **DATA_Customer-Churn** file that you can find in this **LINK**.

Scenario

You are working as an analyst with this internet service provider. You are provided with this historical data about your company's customers and their churn trends. Your task is to build a machine learning model that will help the company identify customers that are more likely to default/churn and thus prevent losses from such customers.

Instructions

In this lab, we will first take a look at the degree of imbalance in the data and correct it using the techniques we learned on the class. Here is the list of steps to be followed (building a simple model without balancing the data):

Round 1

- Import the required libraries and modules that you would need.
- Read that data into Python and call the dataframe churnData.
- Check the datatypes of all the columns in the data.You would see that the column TotalCharges is object type. Convert this column into numeric type using pd.to_numeric function.
- Check for null values in the dataframe. Replace the null values.
- Use the following features: tenure, SeniorCitizen, MonthlyCharges and TotalCharges:
    - Split the data into a training set and a test set.
    - Scale the features either by using MinMaxScaler or a standard scaler.
- (Optional) Encode the categorical variables so you can use them for modeling later.

Round 2

- (Optional) Fit a logistic Regression model on the training data.
- Fit a Knn Classifier (NOT KnnRegressor please!)model on the training data.
- Fit a Decision Tree Classifier on the training data.
- Compare the accuracy, precision, recall for the previous models on both the train and test sets.

Round 3

- apply K-fold cross validation on your models built before, and check the model score. Note: So far we have not balanced the data.

Round 4

- fit a Random forest Classifier on the data and compare the accuracy.
- tune the hyper parameters with Gridsearch and check the results. retrain the final mode with the best parameters found.

Managing imbalance in the dataset

- Check for the imbalance.
- Use the resampling strategies used in class for upsampling and downsampling to create a balance between the two classes.
- Each time fit the model and check the accuracy of the model.