Homework 1

A problem in data mining is to transform categorical variables into efficient numerical features. This focus is warranted due to the ubiquity of categorical data in real-world applications but, on the contrary, the development of many data mining methods based on the assumption of having numerical variables. This stage, which is transforming categorical variables to numeric variables is known as encoding. In the Iris classification application, we observed that the class variable was already encoded. In this exercise, we will practice encoding in conjunction with a dataset collected for a business application.

Working with clients and understanding the factors that can improve the business is part of *marketing data science*. One goal in marketing data science is to keep customers (customer retention), which could be easier in some aspects than attracting new customers. In this exercise, we see a case study on customer retention.

A highly competitive market is communication services. Following the break-up of AT&T in 80's, customers had a choice to keep their carrier (AT&T) or move to another carrier. AT&T tried to identify factors relating to customer choice of carriers. For this purpose, they collected customer data from a number of sources including phone interviews, household billing, and service information recorded in their databases. The dataset that we use here is based on this effort and is taken from https://www.oreilly.com/library/view/marketing-data-science/9780133887662/. It includes nine feature variables and a class variable (`pick`). The goal is to construct a classifier that predicts customers who switch to another service provider or stay with AT&T.

The dataset is stored in `att.csv` , which is part of the "data" folder. Below is description of the variables in the dataset:

- `pick` : customer choice (AT&T or OCC [short for Other Common Carrier])
- `income` : household income in thousands of dollars
- `moves` : number of times the household moved in the last five years
- `age` : age of respondent in years (18-24, 25-34, 35-44, 45-54, 55-64, 65+)
- `education` : `<HS` (less than high school); `HS` (high school graduate); `Voc` (vocational school); `Coll` (Some College); `BA` (college graduate); `>BA` (graduate school)

- `employment`: `D` (disabled); `F` (full-time); `P` (part-time); `H` (homemaker); `R` (retired); `S` (student); `U` (unemployed)
- `usage`: average phone usage per month
- `nonpub`: does the household have an unlisted phone number
- `reachout`: does the household participate in the AT&T "Reach Out America" phone service plan?
- `card`: does the household have an "AT&T Calling Card"?

1. Load the data and pick only variables named `employment`, `usage`, `reachout`, and `card` as features to use in training the classifier (keep "pick" because it is the class variable).
2. Find the rate of missing value for each variable (features and class variable).
3. Remove any feature vector and its corresponding class with at least one missing

entry. What is the sample size now?

4. Use stratified random split to divide the data into train and test sets where the test set size is 20% of the dataset. Set `random_state=100`.
5. There are various strategies for encoding with different pros and cons. An efficient strategy in this regard is ordinal encoding. In this strategy, a variable with N categories, will be converted to a numeric variable with integers from 0 to $N – 1$. Use `OrdinalEncoder()` and `LabelEncoder()` transformers to encode the categorical features and the class variable, respectively— `LabelEncoder` works similarly to `OrdinalEncoder` except that it accepts one-dimensional arrays and Series (so use that to encode the class variable). **Hint**: you need to fit encoder on training set and transform both training and test sets.
6. Train a 7NN classifier using encoded training set and evaluate that on the encoded test set.

Below we list all the modules and classes that will be needed in this homework.

In [1]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier as KNN

from sklearn.preprocessing import OrdinalEncoder
from sklearn.preprocessing import LabelEncoder
```

End