

1. (3 points) A classifier with high accuracy on the testing set is preferred.
  - (a) True
  - (b) False
  - (c) It depends
  - (d) All of the above
  - (e) None of the above
2. (3 points) Accuracy is the only statistics that we need to consider when evaluating classifiers.
  - (a) True
  - (b) False
  - (c) It depends
  - (d) All of the above
  - (e) None of the above

Consider the `Customer_Churn.csv` datafile. Each row represents a customer, each column contains customer's attributes described on the column Metadata. The data set includes information about:

- Customers who left within the last month, the column is called `Churn`.
- Services that each customer has signed up for phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies.
- Customer account information: how long they've been a customer, contract, payment method, paperless billing, monthly charges, and total charges.
- Demographic info about customers: gender, age range, and if they have partners and dependent

3. In **Python**, answer the following:

- (a) (3 points) Using the `pandas` library, read the csv file and create a data-frame called `churn`.
- (b) (3 points) Using the `where` function from the `numpy` library, create a variable called `Churn_num` that takes the value of 1 when `Churn = Yes` and 0 when `Churn = No`.
- (c) (4 points) Define the input variables, `X`, as `SeniorCitizen`, `tenure`, and `MonthlyCharges`, and the target variable, `Y`, as `Churn_num`. Split the data into training (80%) and testing (20%). Make you sure you split the data taking into account the proportion of 0s and 1s.
- (d) (6 points) Using the `train` dataset, build a logistic regression model called `logit_md`. Using the `logit_md` model, predict the likelihood of churn in the `test` dataset. Using 0.35 as cutoff, recode the predictions. That is, if the likelihood of churn is greater than 0.35, recode it as 1; on the other hand, if the likelihood of churn is less than or equal to 0.35, recode it as 0 (use the `where` function from `numpy`). Compare the predictions and actuals using accuracy and recall.
- (e) (6 points) Using the `train` dataset, build a random forest classifier called `RF_md`. Using the `RF_md` model, predict the likelihood of churn in the `test` dataset. Using 0.35 as cutoff, recode the predictions. That is, if the likelihood of churn is greater than 0.35, recode it as 1; on the other hand, if the likelihood of churn is less than or equal to 0.35, recode it as 0 (use the `where` function from `numpy`). Compare the predictions and actuals using accuracy and recall.

- (f) (6 points) Using the `train` dataset, build a gradient boosting classifier called `GBC_md`. Using the `GBC_md` model, predict the likelihood of churn in the `test` dataset. Using 0.35 as cutoff, recode the predictions. That is, if the likelihood of churn is greater than 0.35, recode it as 1; on the other hand, if the likelihood of churn is less than or equal to 0.35, recode it as 0 (use the `where` function from `numpy`). Compare the predictions and actuals using accuracy and recall.
- (g) (3 points) Considering accuracy and recall, what model would you select to make predictions? logistic? random forest? or gradient boosting? Explain.