EMORY UNIVERSITY

# Introduction to Business Data Analytics

**Homework #1** **Due: turned in by Monday Sep 17 9am**

# Adil Ashish Kumar

(put your name above)

Note: This is an individual homework. Discussing this homework with your classmates is a violation of the Honor Code. If you borrow code from somewhere else, please add a comment in your code to make it clear what the source of the code is (e.g., a URL would sufficient). If you borrow code and you don’t provide the source, it is a violation of the Honor Code.

Total grade: ________ out of ___150___ points

***This homework has <u>seven</u> questions. Please answer them all and submit your assignment using Canvas. In particular, you need to submit:***

- ***a) This file with your answers. Please transform the doc file to a PDF file. Then, submit the PDF File in the Assignment labeled as "Homework 1"***
- ***b) Your code and/or Rapidminer repositories for the hands-on exercise. Submit the code/rapidminer repositories in the Assignment labeled as "Homework 1_Code & Repositories"***

**1) (24 points) Choose the data technology (Q, H, U, or S) that is most appropriate for each of the following business questions/scenarios.**

Q – SQL Querying
H – Statistical Hypothesis Testing
U – Unsupervised Data Mining/Pattern Finding
S – Supervised Data Mining

a) __H For my on-line advertising the decision tree model yields a response rate of 0.5% and the old, manual targeting model yields 0.3%. Is the decision tree model really better?

b) __Q I want to know which of my customers are the most profitable.

c) __Q I need to get data on all my on-line customers who were exposed to the special offer, including their registration data, their past purchases, and whether or not they purchased in the 15 days following the exposure.

d)__U I would like to segment my customers into groups based on their demographics and prior purchase activity. I am not focusing on improving a particular task, but would like to generate ideas.

e) __S I have a budget to target 10,000 existing customers with a special offer. I would like to identify those customers most likely to respond to the special offer.

f) __U I want to know what characteristics differentiate my most profitable customers.

**2) (16 points) Label each case as describing either data mining (DM), or the use of the results of data mining (Use).**

g) ___Use Choose customers who are most likely to respond to an on-line ad.

h) ___DM Discover rules that indicate when an account has been defrauded.

i) ___DM Find patterns indicating what customer behavior is more likely to lead to response to an on-line ad.

j) ___Use Estimate probability of default for a credit application.

**3) (15 points) MTC (MegaTelCo) has decided to use supervised learning to address its problem of churn in its wireless phone business. As a consultant to MTC, you realize that a main task in the business understanding/data understanding phases of the data mining process is to define the target variable. In one or two sentences, please suggest a definition for the target variable. Be as precise as possible—someone else will be implementing your suggestion. *(Remember: it should make sense from a business point of view, and it should be reasonable that MTC would have data available to know the value of the target variable for historical customers.)***

Assuming we look at data for a certain time frame, say 2018-2019 full year, the target variable can indicate if the customer discontinued the service or not. In this case, a value of 1 could indicate that customer churned, and a value of 0 would indicate that customer did not churn.

**4) (20 points) A predictive model has been applied to a test dataset and has classified 87 records as fraudulent (31 correctly so) and 953 as non-fraudulent (919 correctly so).**

- **Present the confusion matrix for this scenario.**

| | **Actual Fraudulent** | **Actual Non-Fraudulent** |
|---|---|---|
| **Predicted Fraudulent** | **31** | **56** |
| **Predicted Non-Fraudulent** | **34** | **919** |

- **Calculate the error rate and accuracy rate.**

**Error rate = (56+34) / (31+34+56+919) = 8.653%**

**Accuracy rate = (919+31) / (31+34+56+919) = 91.346%**

- **Calculate the precision, recall, and f-measure values for each of the two outcome classes (i.e., fraudulent and non-fraudulent records);**

**Precision fraudulent = 31/ (31+56) = 35.63%**

**Recall fraudulent= 31 / (31+34) = 47.69%**

**F fraudulent= 2pr/ p+r = 0.4078**

**Precision Non fraudulent = 919 / (919+34) = 96.43%**

**Recall Non fraudulent = 919 / (919+56) = 94.25%**

**F Non Fraudulent = 0.9532**

- **Also, calculate the accuracy rate that would be achieved by naïve (majority) rule on this data.**

| | **Actual Fraudulent** | **Actual Non-Fraudulent** |
|---|---|---|
| **Predicted Fraudulent** | **0** | **0** |
| **Predicted Non-Fraudulent** | **65** | **975** |

**Accuracy = 975/ (975+65)= 93.75 %**

**5) (50 points) [Implement this exercise with both RapidMiner(20 points) and Python (30 points)] Use the decision tree classification technique on the *HW1* dataset. This dataset is provided on the course website and contains data about consumers and their decisions to terminate a contract (i.e., consumer churn problem).**

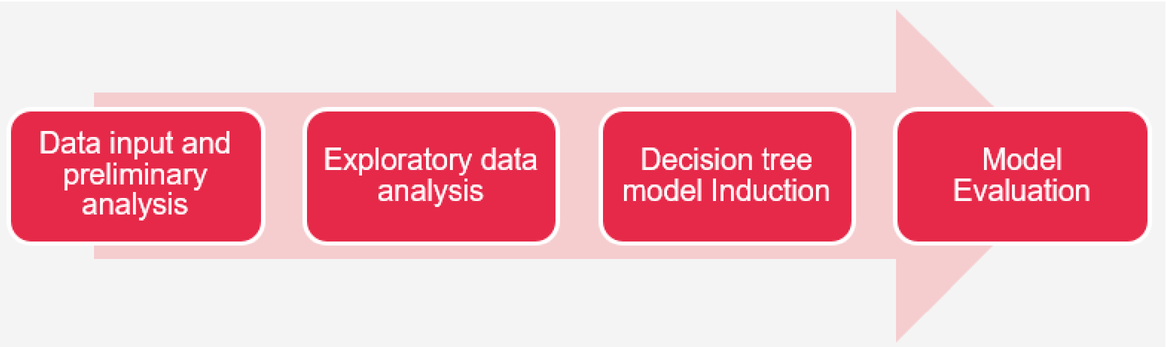
**Data description:**

| Col. | Var. Name | Var. Description |
|---|---|---|
| 1 | revenue | Mean monthly revenue in dollars |
| 2 | outcalls | Mean number of outbound voice calls |
| 3 | incalls | Mean number of inbound voice calls |
| 4 | months | Months in Service |
| 5 | eqpdays | Number of days the customer has had his/her current equipment |
| 6 | webcap | Handset is web capable |
| 7 | marryyes | Married (1=Yes; 0=No) |
| 8 | travel | Has traveled to non-US country (1=Yes; 0=No) |
| 9 | pcown | Owns a personal computer (1=Yes; 0=No) |
| 10 | creditcd | Possesses a credit card (1=Yes; 0=No) |
| 11 | retcalls | Number of calls previously made to retention team |
| 12 | churndep | Did the customer churn (1=Yes; 0=No) |

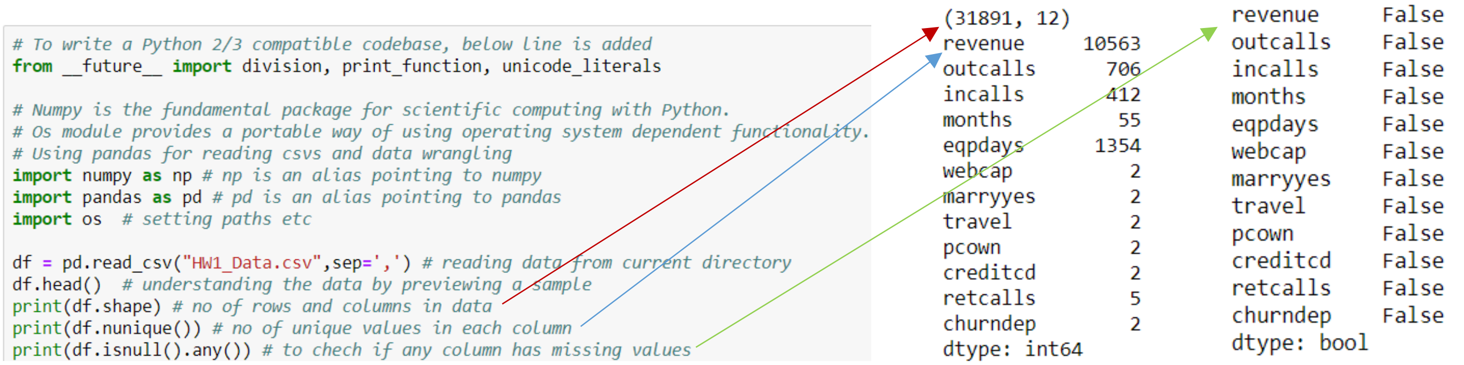
**Build a decision tree model that predicts whether a consumer will terminate his/her contract. In particular, I would like for you to create a decision tree using entropy with no max depth. Explore how well the decision trees perform for several different parameter values (e.g., for different splitting criteria). Interpret the model (decision tree) that provides the best predictive performance.**
**Some possible issues / hints to think about: using training vs. test datasets.**
**Present a brief overview of your predictive modeling process, explorations, and discuss your results. Make sure you present information about the model "goodness" (please report the confusion matrix, predictive accuracy, classification error, precision, recall, f-measure).**

**Present a brief overview of your predictive modeling process. That is, you need to lay out the steps you have taken in order to build and evaluate the decision tree model. For instance, how did you explore the data set before you built the model? Write this report in a way that the upper level management of the team would understand what you are doing. Why is the decision tree an appropriate model for this problem? How can we evaluate the predictive ability of the decision tree? If you build decision trees with different splitting criteria, which decision tree would you prefer to use in practice?**

**Brief overview of modeling process:**


Data input and preliminary analysis
Exploratory data analysis
Decision tree model Induction
Model Evaluation

## 1) Data Input and Preliminary analysis

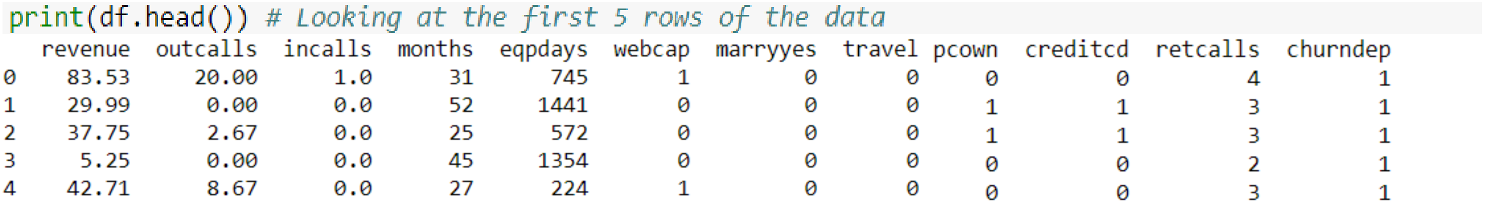


```
# To write a Python 2/3 compatible codebase, below line is added
from __future__ import division, print_function, unicode_literals

# Numpy is the fundamental package for scientific computing with Python.
# Os module provides a portable way of using operating system dependent functionality.
# Using pandas for reading csvs and data wrangling
import numpy as np # np is an alias pointing to numpy
import pandas as pd # pd is an alias pointing to pandas
import os  # setting paths etc

df = pd.read_csv("HW1_Data.csv",sep=',') # reading data from current directory
df.head()  # understanding the data by previewing a sample
print(df.shape) # no of rows and columns in data
print(df.nunique()) # no of unique values in each column
print(df.isnull().any()) # to chech if any column has missing values
```

```
(31891, 12)
revenue      10563
outcalls       706
incalls        412
months          55
eqpdays       1354
webcap           2
marryyes         2
travel           2
pcown            2
creditcd         2
retcalls         5
churndep         2
dtype: int64
```

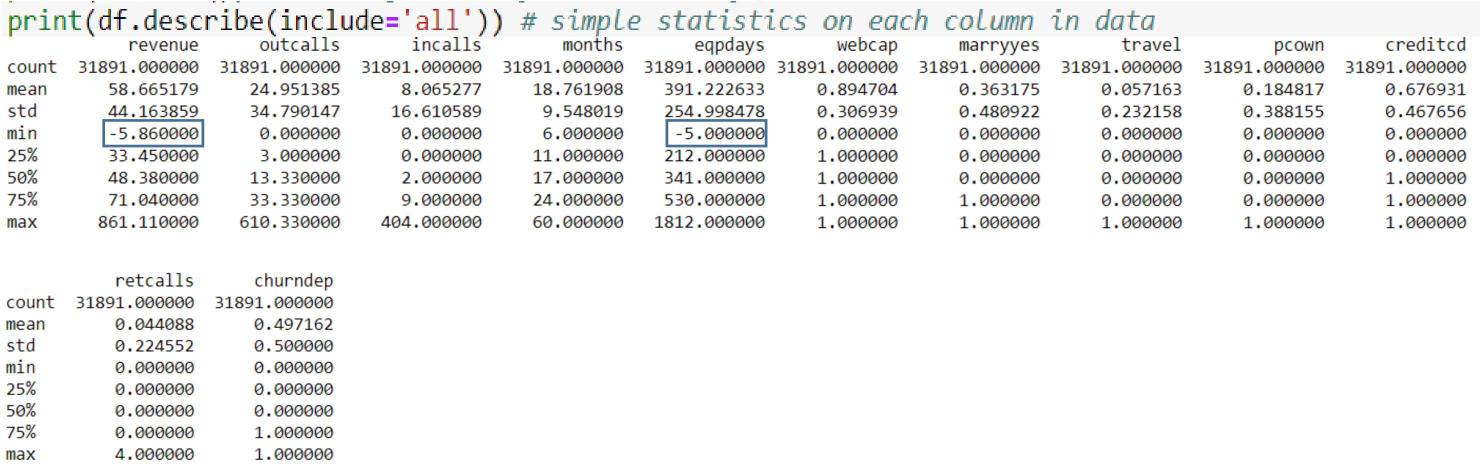
```
revenue      False
outcalls     False
incalls      False
months       False
eqpdays      False
webcap       False
marryyes     False
travel       False
pcown        False
creditcd     False
retcalls     False
churndep     False
dtype: bool
```

In this step, the data (csv file) is read in python and certain packages are imported to help read and prepare the data. A quick preliminary analysis is also done on the data to get an idea about the structure of the data. The data has 31891 rows and 12 columns. We can also see the no of unique values in each column of the data above. This helps to detect if there are any data issues. Looking at the number of unique values in each column, the data seems to be in alignment with the data description shared. There are no missing values in the data, which indicates no missing value treatment needs to be carried out.



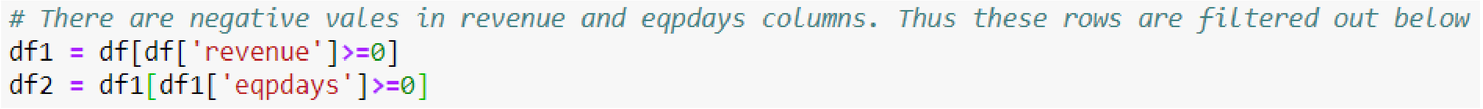
```
print(df.head()) # Looking at the first 5 rows of the data
   revenue  outcalls  incalls  months  eqpdays  webcap  marryyes  travel  pcown  creditcd  retcalls  churndep
0    83.53     20.00      1.0      31      745       1         0       0      0         0         4         1
1    29.99      0.00      0.0      52     1441       0         0       0      1         1         3         1
2    37.75      2.67      0.0      25      572       0         0       0      1         1         3         1
3     5.25      0.00      0.0      45     1354       0         0       0      0         0         2         1
4    42.71      8.67      0.0      27      224       1         0       0      0         0         3         1
```

Above, a sample of the data is printed to get an idea of the structure of the data. Though there is no customer ID, its important to note that each row corresponds to an individual customer and his/her details. All variables are numerically coded, so no transformation is needed for categorical variables.



```
print(df.describe(include='all')) # simple statistics on each column in data
            revenue      outcalls       incalls        months       eqpdays        webcap      marryyes        travel         pcown      creditcd
count  31891.000000  31891.000000  31891.000000  31891.000000  31891.000000  31891.000000  31891.000000  31891.000000  31891.000000  31891.000000
mean      58.665179     24.951385      8.065277     18.761908    391.222633      0.894704      0.363175      0.057163      0.184817      0.676931
std       44.163859     34.790147     16.610589      9.548019    254.998478      0.306939      0.480922      0.232158      0.388155      0.467656
min       -5.860000      0.000000      0.000000      6.000000     -5.000000      0.000000      0.000000      0.000000      0.000000      0.000000
25%       33.450000      3.000000      0.000000     11.000000    212.000000      1.000000      0.000000      0.000000      0.000000      0.000000
50%       48.380000     13.330000      2.000000     17.000000    341.000000      1.000000      0.000000      0.000000      0.000000      1.000000
75%       71.040000     33.330000      9.000000     24.000000    530.000000      1.000000      1.000000      0.000000      0.000000      1.000000
max      861.110000    610.330000    404.000000     60.000000   1812.000000      1.000000      1.000000      1.000000      1.000000      1.000000

           retcalls      churndep
count  31891.000000  31891.000000
mean       0.044088      0.497162
std        0.224552      0.500000
min        0.000000      0.000000
25%        0.000000      0.000000
50%        0.000000      0.000000
75%        0.000000      1.000000
max        4.000000      1.000000
```

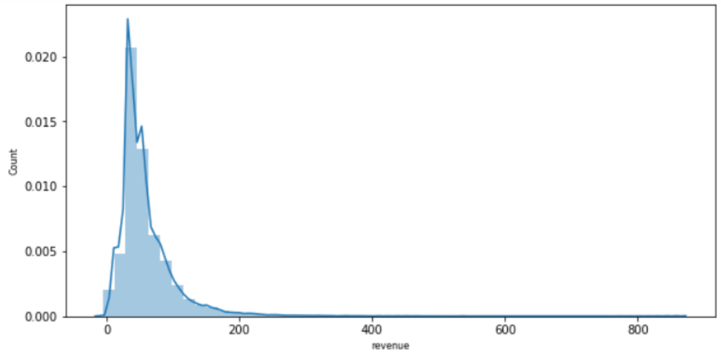
The describe function describes each column in the data and gives simple statistics. There are negative values in the revenue and eqpdays column, which does not make sense based on the data description. Thus these rows are deleted from the data.

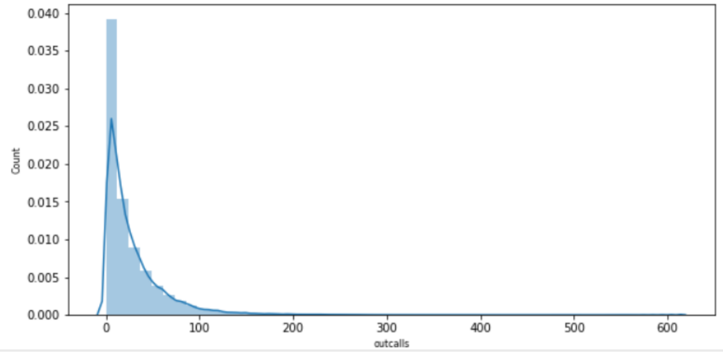


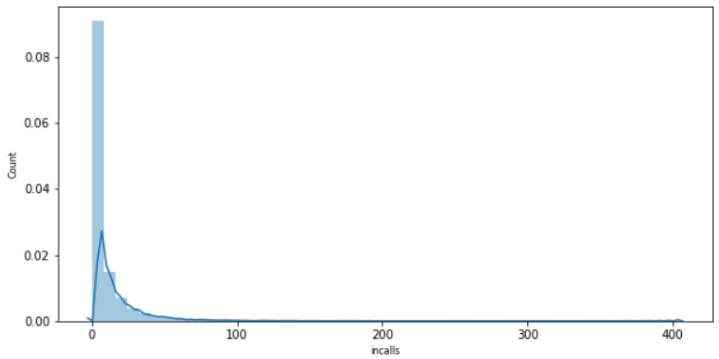
```
# There are negative vales in revenue and eqpdays columns. Thus these rows are filtered out below
df1 = df[df['revenue']>=0]
df2 = df1[df1['eqpdays']>=0]
```

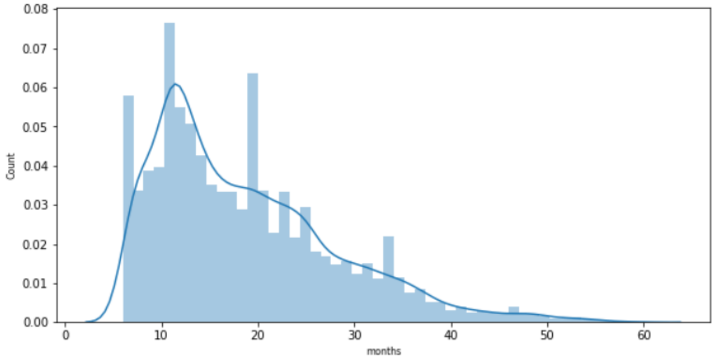
2) Exploratory data analysis & Visualization

To get an idea of the distribution of each variable in the dataset, **histogram plots** are created for the numeric variables and **count plots** for the categorical variables. The revenue variable seems skewed to the right, indicating a huge chunk of low revenue customers in the data. There are very few customers with revenue > 400$. This may look like outliers but should not be treated as outliers since it could be the case that there are very few customers who are very high revenue generators.

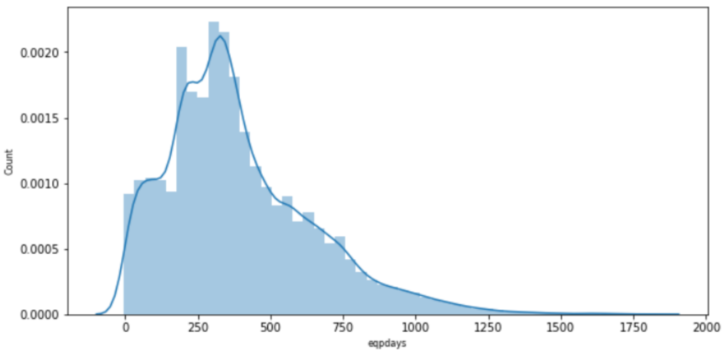


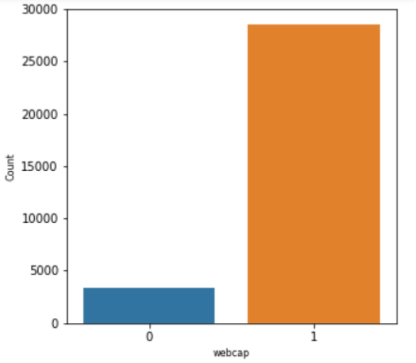


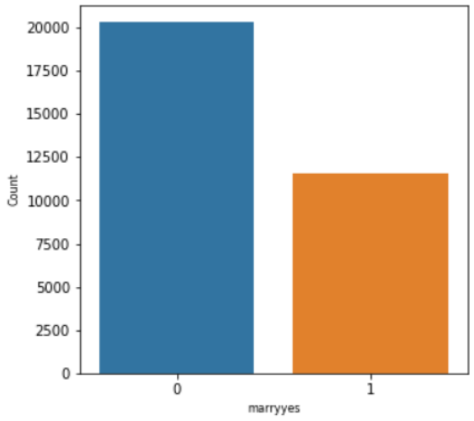


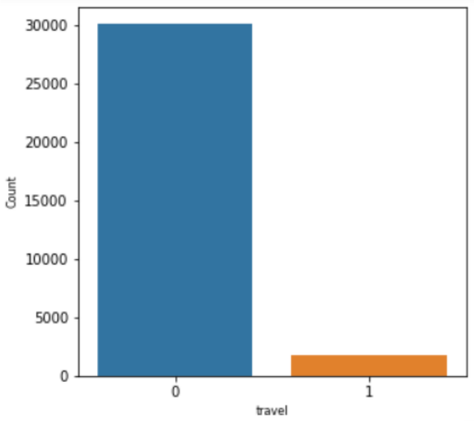


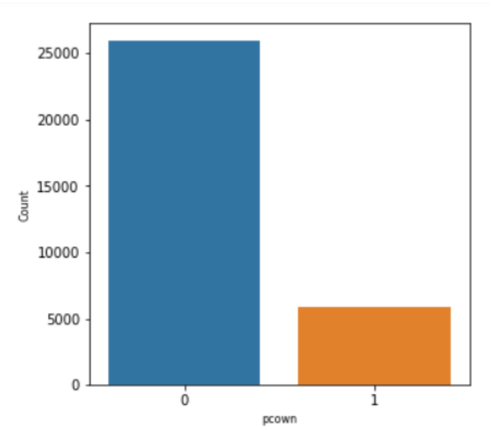
Both incalls and outcalls are skewed to right, which align with the revenue distribution. Months histogram has a more symmetric distribution and shows that the data seems to have a decent spread of old and new customers. The eqpdays has a similar distribution as months, which makes sense.

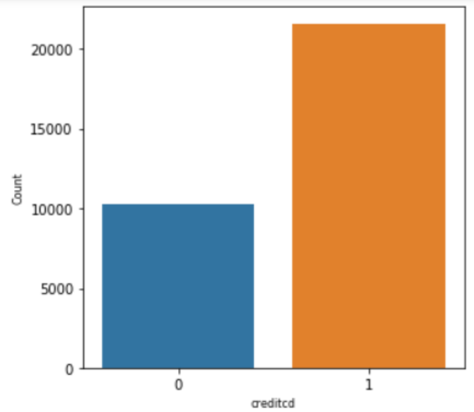


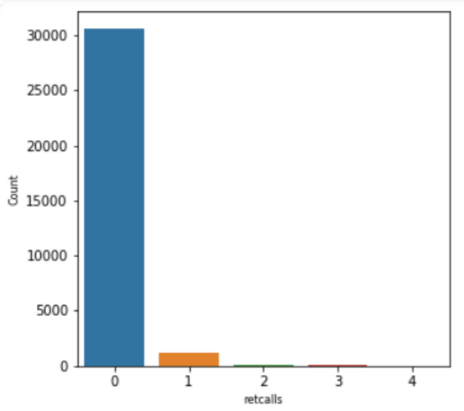


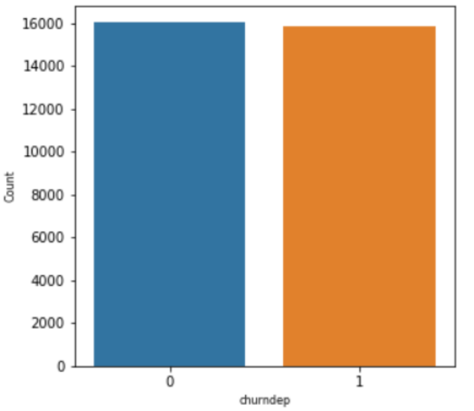




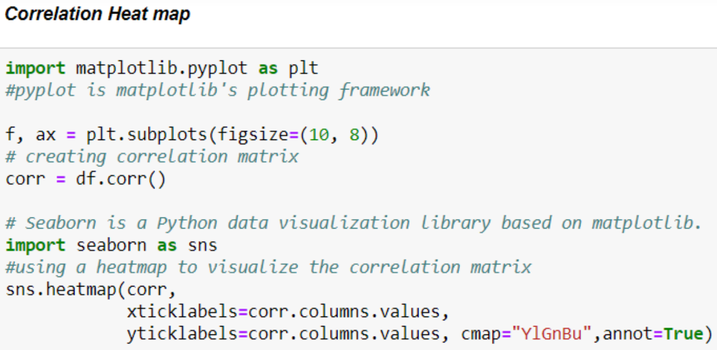






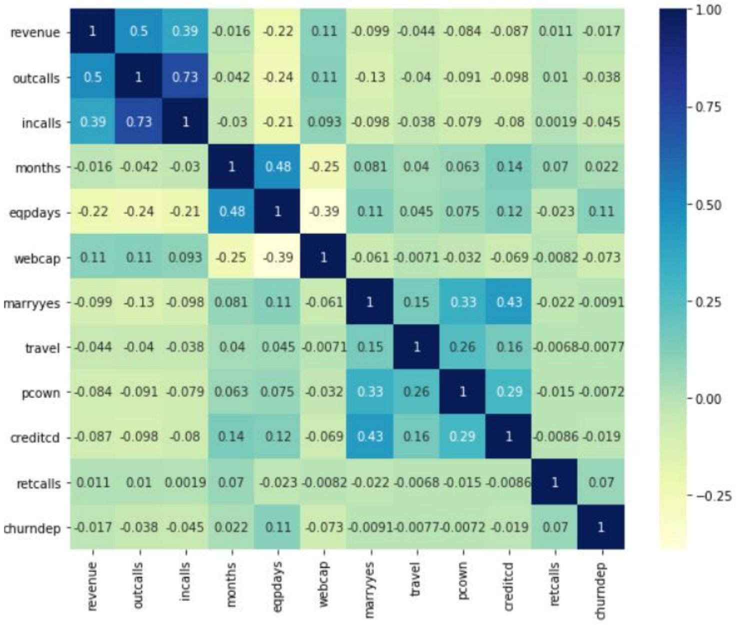


Among categorical variables, most customers seem to have web capable phones. About 63% of the customers are unmarried. Very few customers seem to have traveled to non US countries. Only about 18% of customers own PCs.About 67% of customers own a credit card. Most customers seem to have made 0 calls to the retention team. The target variable churndep has a near perfect balanced class distribution.



***Correlation Heat map***

```
import matplotlib.pyplot as plt
#pyplot is matplotlib's plotting framework

f, ax = plt.subplots(figsize=(10, 8))
# creating correlation matrix
corr = df.corr()

# Seaborn is a Python data visualization library based on matplotlib.
import seaborn as sns
#using a heatmap to visualize the correlation matrix
sns.heatmap(corr,
            xticklabels=corr.columns.values,
            yticklabels=corr.columns.values, cmap="YlGnBu",annot=True)
```

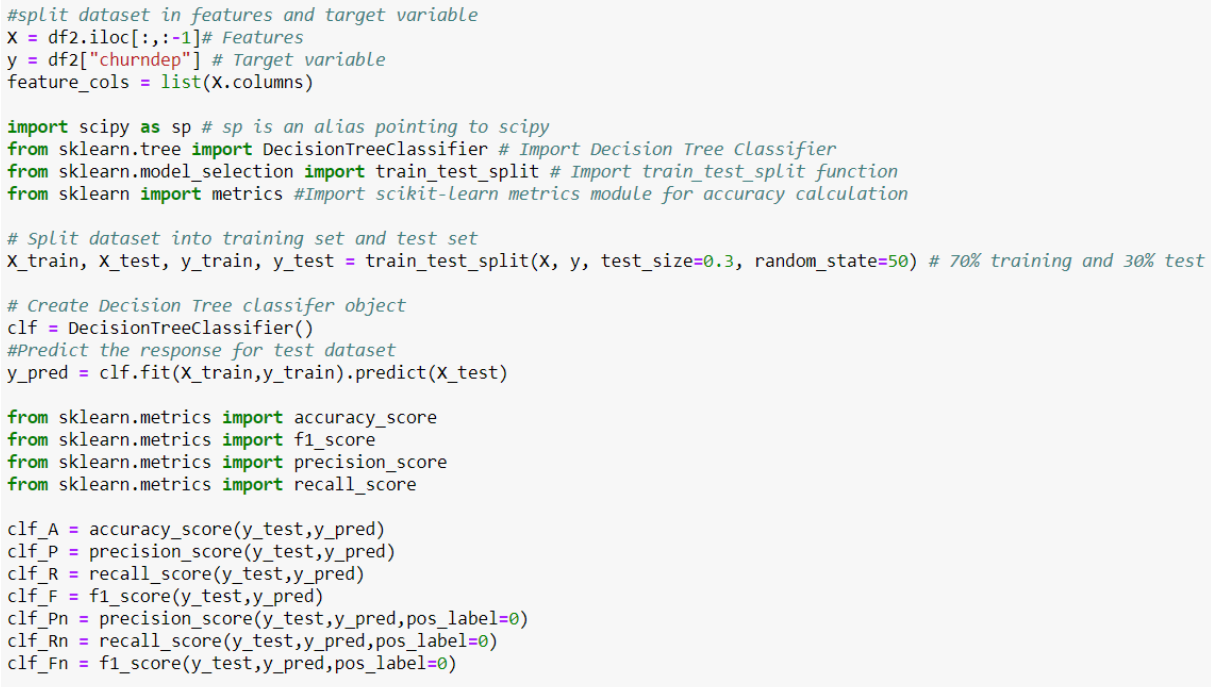


| | revenue | outcalls | incalls | months | eqpdays | webcap | marryyes | travel | pcown | creditcd | retcalls | churndep |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| revenue | 1 | 0.5 | 0.39 | -0.016 | -0.22 | 0.11 | -0.099 | -0.044 | -0.084 | -0.087 | 0.011 | -0.017 |
| outcalls | 0.5 | 1 | 0.73 | -0.042 | -0.24 | 0.11 | -0.13 | -0.04 | -0.091 | -0.098 | 0.01 | -0.038 |
| incalls | 0.39 | 0.73 | 1 | -0.03 | -0.21 | 0.093 | -0.098 | -0.038 | -0.079 | -0.08 | 0.0019 | -0.045 |
| months | -0.016 | -0.042 | -0.03 | 1 | 0.48 | -0.25 | 0.081 | 0.04 | 0.063 | 0.14 | 0.07 | 0.022 |
| eqpdays | -0.22 | -0.24 | -0.21 | 0.48 | 1 | -0.39 | 0.11 | 0.045 | 0.075 | 0.12 | -0.023 | 0.11 |
| webcap | 0.11 | 0.11 | 0.093 | -0.25 | -0.39 | 1 | -0.061 | -0.0071 | -0.032 | -0.069 | -0.0082 | -0.073 |
| marryyes | -0.099 | -0.13 | -0.098 | 0.081 | 0.11 | -0.061 | 1 | 0.15 | 0.33 | 0.43 | -0.022 | -0.0091 |
| travel | -0.044 | -0.04 | -0.038 | 0.04 | 0.045 | -0.0071 | 0.15 | 1 | 0.26 | 0.16 | -0.0068 | -0.0077 |
| pcown | -0.084 | -0.091 | -0.079 | 0.063 | 0.075 | -0.032 | 0.33 | 0.26 | 1 | 0.29 | -0.015 | -0.0072 |
| creditcd | -0.087 | -0.098 | -0.08 | 0.14 | 0.12 | -0.069 | 0.43 | 0.16 | 0.29 | 1 | -0.0086 | -0.019 |
| retcalls | 0.011 | 0.01 | 0.0019 | 0.07 | -0.023 | -0.0082 | -0.022 | -0.0068 | -0.015 | -0.0086 | 1 | 0.07 |
| churndep | -0.017 | -0.038 | -0.045 | 0.022 | 0.11 | -0.073 | -0.0091 | -0.0077 | -0.0072 | -0.019 | 0.07 | 1 |

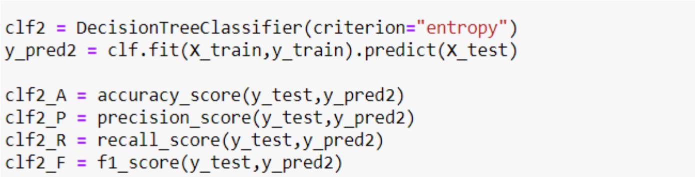
The first thing to look at is the correlation of the independent variables with the target variable churndep. Its interesting to note that, there are no significant correlations between the independent variables and churndep. The variable with highest positive correlation with churndep is eqpdays (0.11), and the magnitude indicates a rather weak correlation. The variable with least correlation with churndep is webcap (-0.073). Since this value is almost 0, it indicates that the 2 variables are not correlated. Apart from eqpdays and retcalls, most independent variables have a correlation of almost 0 with target variable. While these indicate no correlation with target variable, it is possible that there may be a cumulative effect of the independent variables in predicting churndep.

Amongst the correlation values between independent variables, the highest positive correlation is observed between incalls and outcalls (0.73). This indicates a strong correlation between the two variables and means that customers with higher incoming calls make a high number of outgoing calls. Overall, there are no cases where there is very high correlation(~0.9) among the independent variables, indicating that there is no multicollinearity in the dataset.

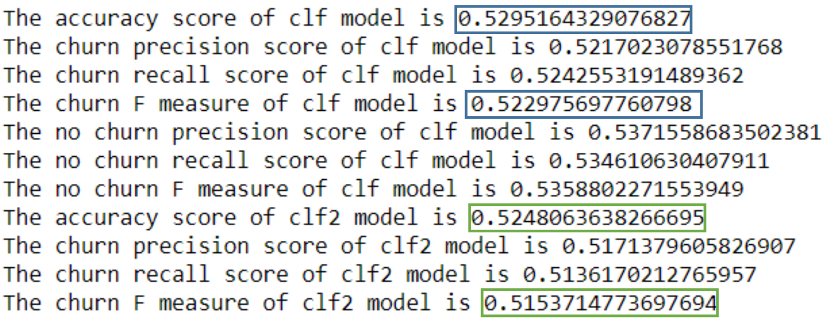
3) Decision tree model induction



```
#split dataset in features and target variable
X = df2.iloc[:,:-1]# Features
y = df2["churndep"] # Target variable
feature_cols = list(X.columns)

import scipy as sp # sp is an alias pointing to scipy
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn.model_selection import train_test_split # Import train_test_split function
from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation

# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=50) # 70% training and 30% test

# Create Decision Tree classifer object
clf = DecisionTreeClassifier()
#Predict the response for test dataset
y_pred = clf.fit(X_train,y_train).predict(X_test)

from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score

clf_A = accuracy_score(y_test,y_pred)
clf_P = precision_score(y_test,y_pred)
clf_R = recall_score(y_test,y_pred)
clf_F = f1_score(y_test,y_pred)
clf_Pn = precision_score(y_test,y_pred,pos_label=0)
clf_Rn = recall_score(y_test,y_pred,pos_label=0)
clf_Fn = f1_score(y_test,y_pred,pos_label=0)
```



```
clf2 = DecisionTreeClassifier(criterion="entropy")
y_pred2 = clf.fit(X_train,y_train).predict(X_test)

clf2_A = accuracy_score(y_test,y_pred2)
clf2_P = precision_score(y_test,y_pred2)
clf2_R = recall_score(y_test,y_pred2)
clf2_F = f1_score(y_test,y_pred2)
```



```
The accuracy score of clf model is 0.5295164329076827
The churn precision score of clf model is 0.5217023078551768
The churn recall score of clf model is 0.5242553191489362
The churn F measure of clf model is 0.522975697760798
The no churn precision score of clf model is 0.5371558683502381
The no churn recall score of clf model is 0.534610630407911
The no churn F measure of clf model is 0.5358802271553949
The accuracy score of clf2 model is 0.5248063638266695
The churn precision score of clf2 model is 0.5171379605826907
The churn recall score of clf2 model is 0.5136170212765957
The churn F measure of clf2 model is 0.5153714773697694
```

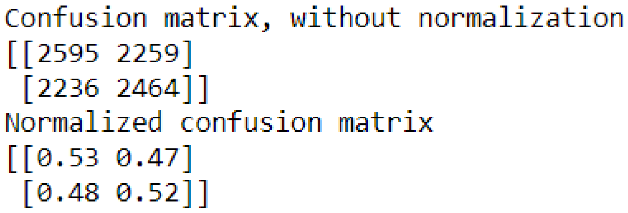
In this step, the features and target variables are first identified and assigned to variables X and y respectively. The data is split in ratio 70:30 for train and test data respectively. The first decision tree model "clf" is created with default splitting criteria “gini” indicating that gini impurity is used to measure quality of split. No max_depth value is specified, so there is no limit on the number of splits in our decision tree. For this model, the accuracy after predicting churn on the test data is 52.95%. The churn F measure for the model is 0.523. The classification error for this model would be 1-Accuracy score= 47.05%.

The second decision tree model “clf2” is created with splitting criteria “entropy” indicating that it uses information gain to measure quality of split. The max_depth value is again unspecified. The accuracy after predicting churn on the test data is 52.48%. The churn F measure for the model is 0.5153. The classification error for this model would be 1-Accuracy score = 47.52%.

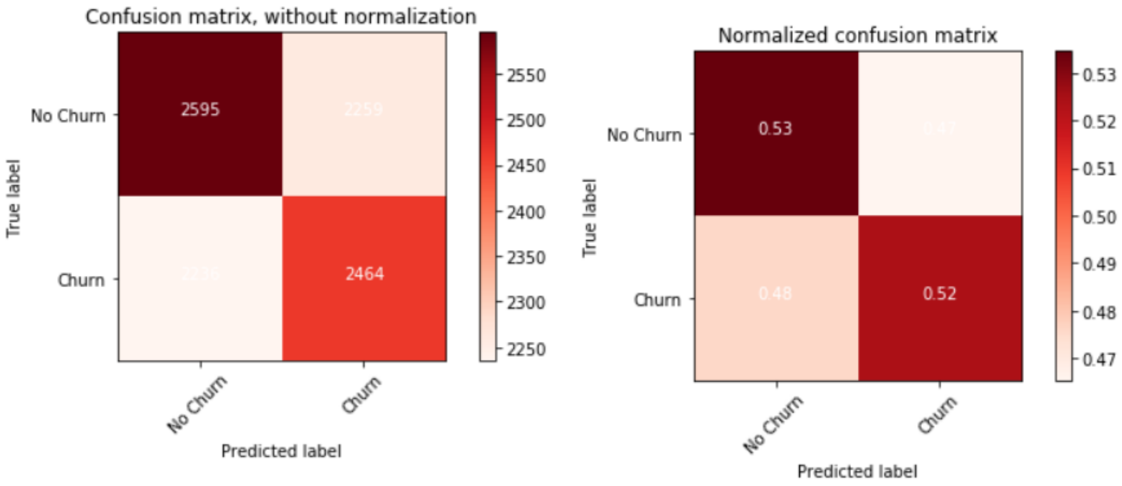
4) Model Evaluation:

Both models have similar accuracy and churn F measure. I would pick model “clf”, since it has a slightly better churn F measure score. The reason for preferring F Measure is because it takes into account both the precision and recall scores of the model, while the accuracy measure does not do so.

In this dataset, we notice from our EDA that there is no clear linear relationship between the independent variables and target variable. Decision tree model is a good choice in this problem because it will not only predict if a customer will churn, but the decision tree will also help understand how the independent variables are shaping the prediction. The rules from the decision tree model can help identify possible reasons for customer churn. Being a white box model, decision tree is easy to interpret and fits well in this problem. Below are the goodness of fit measures for the preferred model clf:



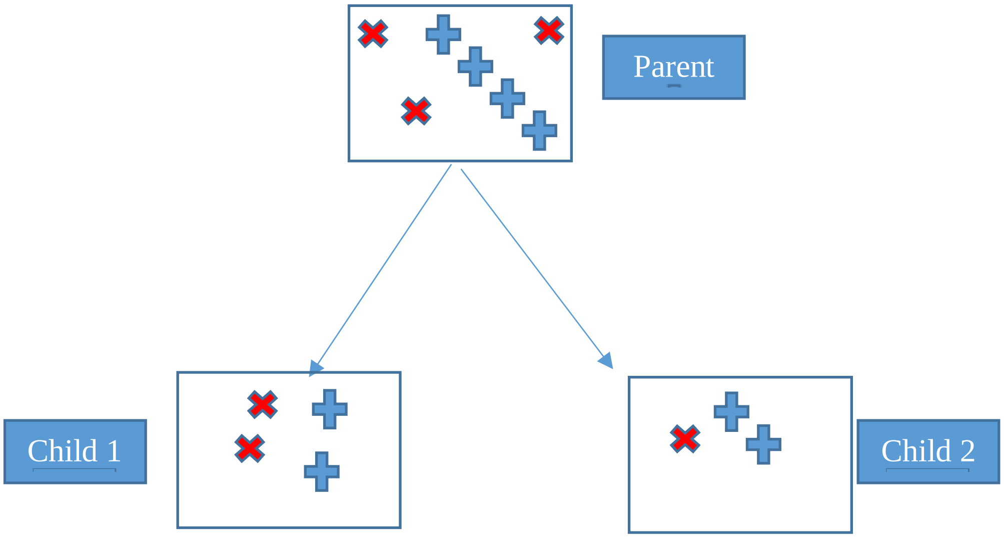
```
Confusion matrix, without normalization
[[2595 2259]
 [2236 2464]]
Normalized confusion matrix
[[0.53 0.47]
 [0.48 0.52]]
```



The confusion matrix is a good indicator of how the model is performing in predicting both classes – churn and no churn on the test dataset. The model accurately classifies 53% of customers who actually do not churn and classifies 52% of the customers who actually do churn. Of the customers who did not actually churn, the model wrongly classifies 47% of them as churned. Of the customers who actually churned, the model wrongly classifies 48% of the customers as not churned. Based on what is more important to the company, the model can be tweaked to get a higher precision or recall score. In this case I have chosen the model with better churn F measure.

**6) (20 points) Is a node's entropy generally lower or greater than its parent's? Is it ever possible for a node's entropy to be higher than its parent's entropy? Please justify your answer. Be precise but concise.**

Generally, the entropy of a child node is smaller than that of the parent node. It is possible for entropy of child node to be higher than that of parent node. Let us look at the below example:



Parent entropy = -3/7 log(3/7) – 4/7log(4/7) = 0.5238+0.4613 = 0.985

Child1 entropy = -0.5 log(0.5) – 0.5 log(0.5) = 1

Child2 entropy = -0.33log (1/3) – 0.67log(2/3) = 0.5283 + 0.3899 = 0.91

In above case, the child1 node has entropy higher than the parent node.

**7) (5 points) What are the differences between supervised and unsupervised methods in machine learning? Is the decision tree algorithm a supervised or an unsupervised method? Be precise but concise.**

In supervised methods, the goal is to predict a target variable using independent variables. In unsupervised methods, there is no target variable to predict. Rather the goal is to find patterns in the data. Decision tree algorithm is a supervised method since we can predict a target variable with the help of the rules from the decision tree. The outcome of a decision tree is either classification or regression, hence it is supervised.