pd.get\_dummies(obj\_data, columns=["State"]).head()

pd.get\_dummies(obj\_data, columns=["Phone"]).head()

pd.get\_dummies(obj\_data, columns=["Int'l Plan"]).head()

pd.get\_dummies(obj\_data, columns=["VMail Plan"]).head()

param\_grid = [

{'criterion': ['gini', 'entropy'], 'min\_samples\_split': [2,4,6,8]}

]

from sklearn.model\_selection import GridSearchCV

simple\_tree = DecisionTreeClassifier()

grid\_model = GridSearchCV(simple\_tree, param\_grid)

grid\_model.fit(X\_train,y\_train)

pd.get\_dummies(obj\_data, columns=["Churn?"]).head()

pd.get\_dummies(obj\_data, columns=["State", "Phone", "Int'l Plan", "VMail Plan", "Churn?"], prefix=["State", "Phone", "Int'l Plan", "VMail Plan", "Churn?"]).head()

df["Int'l Plan"] = np.where(df["Int'l Plan"] == 'yes',1,0)

df["VMail Plan"] = np.where(df["VMail Plan"] == 'yes',1,0)

####################################################################################

### Código que resultou em Accuracy de 0,92

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

df = pd.read\_csv('churn.csv')

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

# Drop the columns that we have decided won't be used in prediction

df = df.drop(["Phone", "Area Code"], axis=1)

df=pd.get\_dummies(df, columns=["State"], prefix=["State"])

df["Int'l Plan"] = np.where(df["Int'l Plan"] == 'yes',1,0)

df["VMail Plan"] = np.where(df["VMail Plan"] == 'yes',1,0)

# Separa a variavel objectivo do dataset

X = df.drop('Churn?',axis=1)

y = df['Churn?']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.30)

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

dtree = DecisionTreeClassifier(min\_samples\_split=4)

dtree2=DecisionTreeClassifier(criterion='entropy',min\_samples\_split=4)

dtree.fit(X\_train,y\_train)

dtree2.fit(X\_train,y\_train)

predictions = dtree.predict(X\_test)

predictions2 = dtree2.predict(X\_test)

from sklearn.metrics import classification\_report,confusion\_matrix, accuracy\_score

print(classification\_report(y\_test,predictions))

#print('accuracy={0}'.format(accuracy\_score(y\_test,predictions)))

print(classification\_report(y\_test,predictions2))

#print('accuracy={0}'.format(accuracy\_score(y\_test,predictions2)))

print(confusion\_matrix(y\_test,predictions))

print(confusion\_matrix(y\_test,predictions2))

print('accuracy={0}'.format(accuracy\_score(y\_test,predictions)))

print('accuracy={0}'.format(accuracy\_score(y\_test,predictions2)))

####################################################################################