Module 3 Assignment

Gabi Rivera || 14Nov2022 || ADS502-01

In [2]:
```
import os
os.getcwd()
```

Out[2]:
```
'/Users/gabirivera/Desktop/MSADS2/ADS502-01/Module3/Assignment'
```

In [3]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.naive_bayes import MultinomialNB
import statsmodels.tools.tools as stattools
```

Data Science Using Python and R: Chapter 7 Hands-On Analysis

1. Using the training data set, create a C5.0 model (Model 1) to predict a customer's Income using Marital Status and Capital Gains and Losses. Obtain the predicted responses.

In [4]:
```
import statsmodels.tools.tools as stattools
from sklearn.tree import DecisionTreeClassifier, export_graphviz
from sklearn import tree
```

In [5]:
```
adult_tr= pd.read_csv('adult_ch6_training', sep = ',')
adult_tr.head()
```

Out[5]:

	Marital status	Income	Cap_Gains_Losses
0	Never-married	<=50K	0.02174
1	Divorced	<=50K	0.00000
2	Married	<=50K	0.00000
3	Married	<=50K	0.00000
4	Married	<=50K	0.00000

In [6]:
```
y = adult_tr[['Income']]

mar_np = np.array(adult_tr['Marital status'])
(mar_cat, mar_cat_dict) = stattools.categorical(mar_np, drop=True, dictnames = True)
mar_cat_pd = pd.DataFrame(mar_cat)
X = pd.concat((adult_tr[['Cap_Gains_Losses']], mar_cat_pd), axis = 1)

mar_cat_dict

X_names = ["Cap_Gains_Losses", "Divorced", "Married", "Never-married",
           "Separated", "Widowed"]

y_names = ["<=50K", ">50K"]
```

```
/Users/gabirivera/opt/anaconda3/lib/python3.8/site-packages/statsmodels/tools/tools.py:1
52: FutureWarning: categorical is deprecated. Use pandas Categorical to represent catego
```

```
rical data and can get_dummies to construct dummy arrays. It will be removed after relea
se 0.13.
  warnings.warn(
```

In [7]:

```
c50_01 = DecisionTreeClassifier(criterion="entropy", max_leaf_nodes=5).fit(X,y)
```

```
/Users/gabirivera/opt/anaconda3/lib/python3.8/site-packages/sklearn/utils/validation.py:
1858: FutureWarning: Feature names only support names that are all strings. Got feature
names with dtypes: ['int', 'str']. An error will be raised in 1.2.
  warnings.warn(
```

In [8]:

```
c50_01.predict(X)
```

```
/Users/gabirivera/opt/anaconda3/lib/python3.8/site-packages/sklearn/utils/validation.py:
1858: FutureWarning: Feature names only support names that are all strings. Got feature
names with dtypes: ['int', 'str']. An error will be raised in 1.2.
  warnings.warn(
```

Out[8]:

```
array(['<=50K', '<=50K', '<=50K', ..., '<=50K', '<=50K', '<=50K'],
      dtype=object)
```

Data Science Using Python and R: Chapter 8 Hands-On Analysis

1. Run the Naïve Bayes classifier to classify persons as living or dead based on sex and education.

In [25]:

```
fn_train = pd.read_csv("framingham_nb_training.csv", sep = ',')
fn_train.head()
```

Out[25]:

	Sex	Educ	Death
0	2	3	0
1	2	2	0
2	1	1	0
3	2	1	0
4	2	1	0

In [26]:

```
fn_test = pd.read_csv("framingham_nb_test.csv", sep = ',')
fn_test.head()
```

Out[26]:

	Sex	Educ	Death
0	1	1	0
1	1	2	0
2	1	3	0
3	1	1	0
4	2	2	0

Contingency table: Death based on sex

In [30]:

```
t1 = pd.crosstab(fn_train['Death'], fn_train['Sex'])
t1['Total'] = t1.sum(axis=1)
t1.loc['Total'] = t1.sum()
t1
```

Out[30]:

Sex	1	2	Total

Death			
0	184	266	450
1	308	242	550
Total	492	508	1000

Contingency table: Death based on sex

```
In [31]: t2 = pd.crosstab(fn_train['Death'], fn_train['Educ'])
         t2['Total'] = t2.sum(axis=1)
         t2.loc['Total'] = t2.sum()
         t2
```

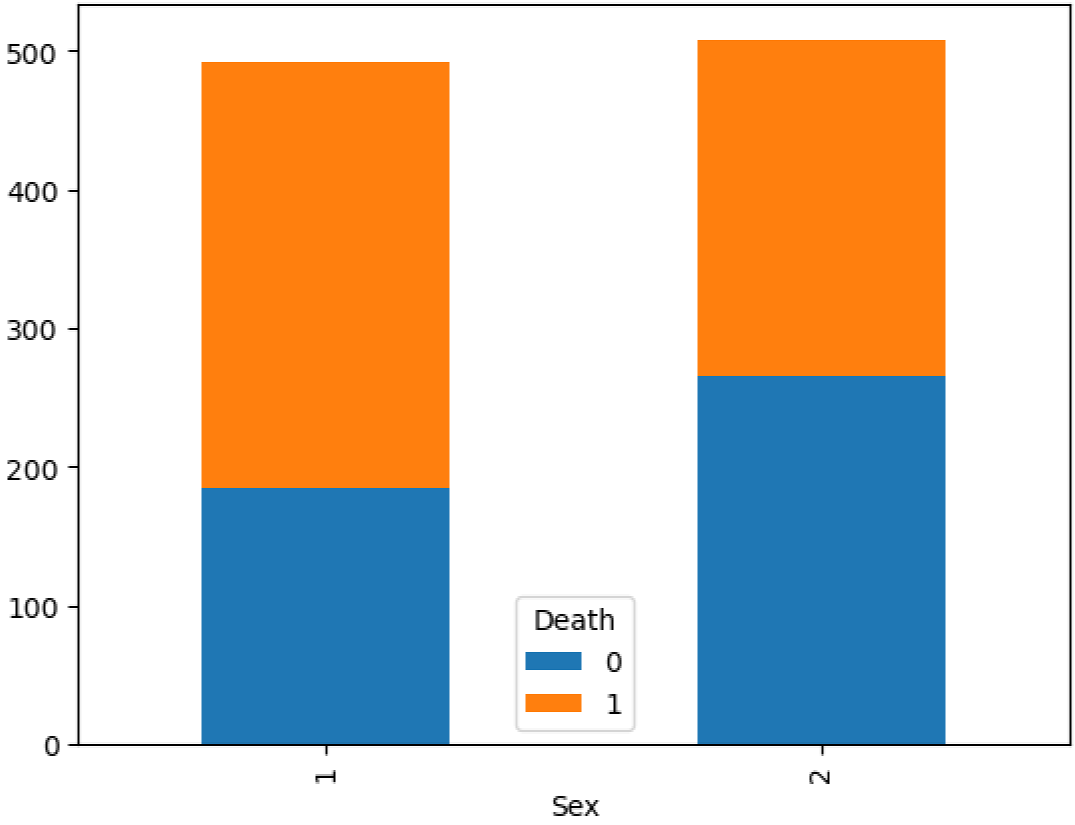
Out[31]:

Educ	1	2	3	4	Total
Death					
0	173	146	84	47	450
1	287	135	80	48	550
Total	460	281	164	95	1000

Plot: Death based on sex

```
In [29]: t1_plot = pd.crosstab(fn_train['Sex'], fn_train['Death'])
         t1_plot.plot(kind='bar', stacked = True)
```

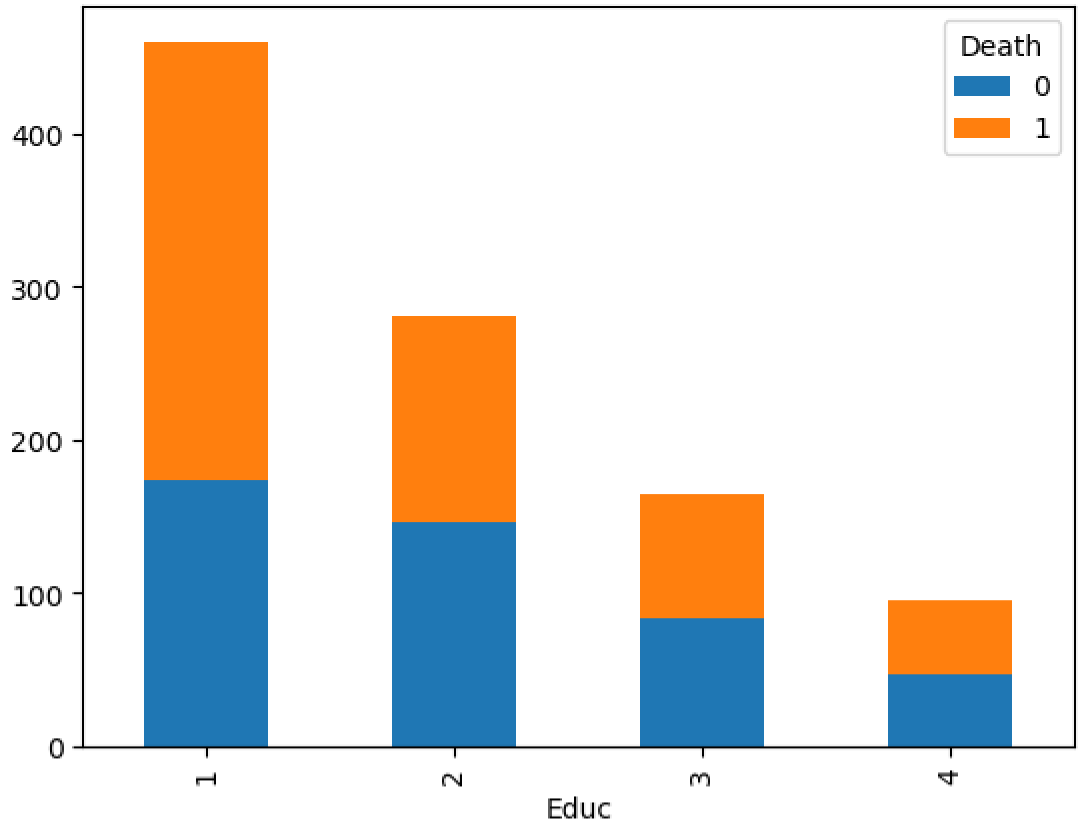
```
Out[29]: <AxesSubplot:xlabel='Sex'>
```



```
In [ ]: Plot: Death based on education
```

```
In [33]: t2_plot = pd.crosstab(fn_train['Educ'], fn_train['Death'])
         t2_plot.plot(kind='bar', stacked = True)
```

```
Out[33]: <AxesSubplot:xlabel='Educ'>
```



Naive Bayes dataset prep:

```
In [42]: X_Sex_ind = np.array(fn_train['Sex'])
         (X_Sex_ind , X_Sex_ind_dict) = stattools.categorical(X_Sex_ind,drop=True, dictnames = Tr
         X_Sex_ind = pd.DataFrame(X_Sex_ind)

         X_Educ_ind = np.array(fn_train['Educ'])
         (X_Educ_ind , X_Educ_ind_dict) = stattools.categorical(X_Educ_ind, drop=True, dictnames
         X_Educ_ind = pd.DataFrame(X_Educ_ind)



         X = pd.concat((X_Sex_ind, X_Educ_ind), axis = 1)
         Y = fn_train['Death']
```

```
/Users/gabirivera/opt/anaconda3/lib/python3.8/site-packages/statsmodels/tools/tools.py:1
52: FutureWarning: categorical is deprecated. Use pandas Categorical to represent catego
rical data and can get_dummies to construct dummy arrays. It will be removed after relea
se 0.13.
  warnings.warn(
```

```
In [51]: nb_01 = MultinomialNB().fit(X, Y)
```

1. Evaluate the Naïve Bayes model on the framingham_nb_test data set. Display the results in a contingency table. Edit the row and column names of the table to make the table more readable. Include a total row and column.

Naïve Bayes model on the framingham_nb_test data set:

```
In [43]: X_Sex_ind_test = np.array(fn_test['Sex'])
         (X_Sex_ind_test, X_Sex_ind_dict_test) = stattools.categorical(X_Sex_ind_test,drop=True,
         X_Sex_ind_test = pd.DataFrame(X_Sex_ind_test)
```

```
X_Educ_ind_test = np.array(fn_test['Educ'])
(X_Educ_ind_test, X_Educ_ind_dict_test) = stattools.categorical(X_Educ_ind_test, drop=Tr
X_Educ_ind_test = pd.DataFrame(X_Educ_ind_test)

X_test = pd.concat((X_Sex_ind_test, X_Educ_ind_test), axis = 1)
Y_predicted = nb_01.predict(X_test)
```

Naive Bayes contingency table:

In [49]:

```
ypred = pd.crosstab(fn_test['Death'], Y_predicted, rownames = ['Actual'], colnames = ['P

ypred['Total'] = ypred.sum(axis=1); ypred.loc['Total'] = ypred.sum(); ypred
```

Out[49]:

Predicted	**0**	**1**	**Total**
Actual			
0	203	322	525
1	105	370	475
Total	308	692	1000

1. According to your table in the previous exercise, find the following values for the Naïve Bayes model:

a. Accuracy

In [56]:

```
Accuracy_NB = ((203+370) / 1000) * 100
Accuracy_NB
```

Out[56]:

```
57.3
```

b. Error rate

In [57]:

```
Error_rate_NB = (100 - Accuracy_NB)
Error_rate_NB
```

Out[57]:

```
42.7
```

1. According to your contingency table, find the following values for the Naïve Bayes model:

a. How often it correctly classifies dead persons.

In [61]:

```
Specificity_NB = (203/ 525)*100
round(Specificity_NB, 1)
```

Out[61]:

```
38.7
```

b. How often it correctly classifies living persons.

In [63]:

```
Sensitivity_NB = (370/475)*100
round(Sensitivity_NB, 1)
```

Out[63]:

```
77.9
```