S3 - HOPE logistic regression-V2

December 4, 2020

0.1 Import data from DB.

[1]:

```
import pandas as pd
import numpy as np
```

[2]:

```
dfOrg = pd.read_csv('hope_dataset_cleaned.csv')

print(dfOrg.shape[0])
```

```
1243
```

[3]:

```
dfOrg.head(10)
```

[3]:

```
   pedido.data.attributes.age pedido.data.attributes.diagnostic_main  \
0                        75.0                     FISTULA PERITONEAL
1                        75.0                     FISTULA PERITONEAL
2                        75.0                     FISTULA PERITONEAL
3                        75.0                     FISTULA PERITONEAL
4                        75.0                     FISTULA PERITONEAL
5                        75.0                     FISTULA PERITONEAL
6                        75.0                     FISTULA PERITONEAL
7                        75.0                     FISTULA PERITONEAL
8                        75.0                     FISTULA PERITONEAL
9                        75.0                     FISTULA PERITONEAL

  pedido.data.attributes.gender  articulo  respuesta.articlesRevisedYear  \
0                          male  27395425                           2018
1                          male  28560554                           2018
2                          male  28641726                           2017
3                          male  26245344                           2016
4                          male  28942543                           2018
5                          male  24782153                           2014
6                          male  28002229                           2018
7                          male  27505109                           2017
8                          male  24850546                           2015
9                          male  29371050                           2019

   respuesta.articlesRevisedMonth  \
0                              1
1                              4
2                             12
3                             12
4                              6
5                              6
6                              9
7                              4
8                              1
9                              4

                            respuesta.pubmed_keys  utilidad
0  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       1.0
1  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
2  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
3  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
4  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
5  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
6  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
7  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
8  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
9  Abdomen,Adenocarcinoma,Antiemetics,Blood Cultu…       NaN
```

Expand pubmed_keys attribute

[4]:

```python
dfOrg['respuesta.pubmed_keys'] = dfOrg['respuesta.pubmed_keys'].apply(lambda x :
 ↪ str(x).split(','))

dfOrg = dfOrg.explode('respuesta.pubmed_keys').reset_index(drop=True)

dfOrg.head(10)
```

[4]:

```
   pedido.data.attributes.age pedido.data.attributes.diagnostic_main  \
0                        75.0                     FISTULA PERITONEAL
1                        75.0                     FISTULA PERITONEAL
2                        75.0                     FISTULA PERITONEAL
3                        75.0                     FISTULA PERITONEAL
4                        75.0                     FISTULA PERITONEAL
5                        75.0                     FISTULA PERITONEAL
6                        75.0                     FISTULA PERITONEAL
7                        75.0                     FISTULA PERITONEAL
8                        75.0                     FISTULA PERITONEAL
9                        75.0                     FISTULA PERITONEAL

  pedido.data.attributes.gender  articulo  respuesta.articlesRevisedYear  \
0                          male  27395425                           2018
1                          male  27395425                           2018
```

```
2                          male  27395425                           2018
3                          male  27395425                           2018
4                          male  27395425                           2018
5                          male  27395425                           2018
6                          male  27395425                           2018
7                          male  27395425                           2018
8                          male  27395425                           2018
9                          male  27395425                           2018

   respuesta.articlesRevisedMonth respuesta.pubmed_keys  utilidad
0                               1               Abdomen       1.0
1                               1        Adenocarcinoma       1.0
2                               1           Antiemetics       1.0
3                               1         Blood Culture       1.0
4                               1             Catharsis       1.0
5                               1              Diuresis       1.0
6                               1               Fistula       1.0
7                               1           Gastrectomy       1.0
8                               1     Incisional Hernia       1.0
9                               1            Intestines       1.0
```

0.2 Transform (factorice) from Categories to continuous atributes

Transform 'pedido.data.attributes.diagnostic_main' atribute

[5]:

```
dataDiagnosticMain, categoriesDiagnosticMain = pd.factorize(dfOrg['pedido.data.
 ↪attributes.diagnostic_main'])

dfOrg['pedido.data.attributes.diagnostic_main'] = dataDiagnosticMain
```

Transform 'gender' atribute

[6]:

```
dataGender, categoriesGender = pd.factorize(dfOrg['pedido.data.attributes.
 ↪gender'])

dfOrg['pedido.data.attributes.gender'] = dataGender
```

Transform 'respuesta.pubmed_keys' atribute

[7]:

```
categoriesORGPubMedKeys = dfOrg['respuesta.pubmed_keys'].value_counts()

print("total: " + str(categoriesORGPubMedKeys.size))
```

```
total: 353
```

[8]:

```
dataPubMedKeys, categoriesPubMedKeys = pd.factorize(dfOrg['respuesta.
 ↪pubmed_keys'])
```

```
dfOrg['respuesta.pubmed_keys'] = dataPubMedKeys
```

[9]:

```
dfOrg.head(10)
```

[9]:

```
   pedido.data.attributes.age  pedido.data.attributes.diagnostic_main  \
0                        75.0                                       0
1                        75.0                                       0
2                        75.0                                       0
3                        75.0                                       0
4                        75.0                                       0
5                        75.0                                       0
6                        75.0                                       0
7                        75.0                                       0
8                        75.0                                       0
9                        75.0                                       0

   pedido.data.attributes.gender  articulo  respuesta.articlesRevisedYear  \
0                              0  27395425                           2018
1                              0  27395425                           2018
2                              0  27395425                           2018
3                              0  27395425                           2018
4                              0  27395425                           2018
5                              0  27395425                           2018
6                              0  27395425                           2018
7                              0  27395425                           2018
8                              0  27395425                           2018
9                              0  27395425                           2018

   respuesta.articlesRevisedMonth  respuesta.pubmed_keys  utilidad
0                               1                      0       1.0
1                               1                      1       1.0
2                               1                      2       1.0
3                               1                      3       1.0
4                               1                      4       1.0
5                               1                      5       1.0
6                               1                      6       1.0
7                               1                      7       1.0
8                               1                      8       1.0
9                               1                      9       1.0
```

[10]:

```
print("age NaN => " + str(dfOrg[pd.isnull(dfOrg['pedido.data.attributes.age'])].
 ↪shape[0]))
print("diagnostic_main NaN => " + str(dfOrg[pd.isnull(dfOrg['pedido.data.
 ↪attributes.diagnostic_main'])].shape[0]))
print("gender NaN => " + str(dfOrg[pd.isnull(dfOrg['pedido.data.attributes.
 ↪gender'])].shape[0]))
print("articulo NaN => " + str(dfOrg[pd.isnull(dfOrg['articulo'])].shape[0]))
print("articlesRevisedYear NaN => " + str(dfOrg[pd.isnull(dfOrg['respuesta.
 ↪articlesRevisedYear'])].shape[0]))
print("articlesRevisedMonth NaN => " + str(dfOrg[pd.isnull(dfOrg['respuesta.
 ↪articlesRevisedMonth'])].shape[0]))
print("pubmed_keys NaN => " + str(dfOrg[pd.isnull(dfOrg['respuesta.
 ↪pubmed_keys'])].shape[0]))
print("utilidad NaN => " + str(dfOrg[pd.isnull(dfOrg['utilidad'])].shape[0]))
```

```
age NaN => 10
diagnostic_main NaN => 0
gender NaN => 0
articulo NaN => 0
articlesRevisedYear NaN => 0
articlesRevisedMonth NaN => 0
pubmed_keys NaN => 0
utilidad NaN => 14758
```

Remove row with age eq NaN

[11]:

```
dfOrg = dfOrg[pd.notnull(dfOrg['pedido.data.attributes.age'])]
```

0.3 Standardize the Data

[12]:

```
from sklearn.preprocessing import StandardScaler

features = dfOrg.columns.drop(['utilidad'])

# Separating out the features
x = dfOrg.loc[:, features].values

featuresTransformed = StandardScaler().fit_transform(x)

dfStandarized = pd.DataFrame(featuresTransformed, index=dfOrg.index,␣
 ↪columns=features)
dfStandarized['utilidad'] = dfOrg['utilidad']

dfStandarized.head(10)
```

[12]:

```
   pedido.data.attributes.age  pedido.data.attributes.diagnostic_main  \
0                    1.285887                                -1.503163
1                    1.285887                                -1.503163
2                    1.285887                                -1.503163
3                    1.285887                                -1.503163
4                    1.285887                                -1.503163
5                    1.285887                                -1.503163
6                    1.285887                                -1.503163
7                    1.285887                                -1.503163
```

```
8                    1.285887                               -1.503163
9                    1.285887                               -1.503163

   pedido.data.attributes.gender  articulo  respuesta.articlesRevisedYear  \
0                            0.0  -0.00421                       0.633249
1                            0.0  -0.00421                       0.633249
2                            0.0  -0.00421                       0.633249
3                            0.0  -0.00421                       0.633249
4                            0.0  -0.00421                       0.633249
5                            0.0  -0.00421                       0.633249
6                            0.0  -0.00421                       0.633249
7                            0.0  -0.00421                       0.633249
8                            0.0  -0.00421                       0.633249
9                            0.0  -0.00421                       0.633249

   respuesta.articlesRevisedMonth  respuesta.pubmed_keys  utilidad
0                       -1.463658              -1.089722       1.0
1                       -1.463658              -1.080463       1.0
2                       -1.463658              -1.071203       1.0
3                       -1.463658              -1.061944       1.0
4                       -1.463658              -1.052684       1.0
5                       -1.463658              -1.043424       1.0
6                       -1.463658              -1.034165       1.0
7                       -1.463658              -1.024905       1.0
8                       -1.463658              -1.015646       1.0
9                       -1.463658              -1.006386       1.0
```

0.4 Separe data by utilidad is defined

[13]:

```
dfDataSetComplete = dfStandarized[pd.notnull(dfOrg['utilidad'])]

print(dfDataSetComplete.shape[0])

dfDataSetToPredict = dfStandarized[pd.isnull(dfOrg['utilidad'])]

print(dfDataSetToPredict.shape[0])
```

```
830
14748
```

[14]:

```
dfDataSetComplete.head(10)
```

[14]:

```
   pedido.data.attributes.age  pedido.data.attributes.diagnostic_main  \
0                    1.285887                               -1.503163
1                    1.285887                               -1.503163
2                    1.285887                               -1.503163
3                    1.285887                               -1.503163
```

```
4                    1.285887                                -1.503163
5                    1.285887                                -1.503163
6                    1.285887                                -1.503163
7                    1.285887                                -1.503163
8                    1.285887                                -1.503163
9                    1.285887                                -1.503163

   pedido.data.attributes.gender  articulo  respuesta.articlesRevisedYear  \
0                            0.0  -0.00421                       0.633249
1                            0.0  -0.00421                       0.633249
2                            0.0  -0.00421                       0.633249
3                            0.0  -0.00421                       0.633249
4                            0.0  -0.00421                       0.633249
5                            0.0  -0.00421                       0.633249
6                            0.0  -0.00421                       0.633249
7                            0.0  -0.00421                       0.633249
8                            0.0  -0.00421                       0.633249
9                            0.0  -0.00421                       0.633249

   respuesta.articlesRevisedMonth  respuesta.pubmed_keys  utilidad
0                       -1.463658              -1.089722       1.0
1                       -1.463658              -1.080463       1.0
2                       -1.463658              -1.071203       1.0
3                       -1.463658              -1.061944       1.0
4                       -1.463658              -1.052684       1.0
5                       -1.463658              -1.043424       1.0
6                       -1.463658              -1.034165       1.0
7                       -1.463658              -1.024905       1.0
8                       -1.463658              -1.015646       1.0
9                       -1.463658              -1.006386       1.0
```

0.5 Logistic Regression

[15]:

```
from sklearn import linear_model
from sklearn.model_selection import train_test_split
from sklearn import model_selection
from sklearn.metrics import accuracy_score
```

[16]:

```
dfDataSetComplete.describe()
```

[16]:

```
       pedido.data.attributes.age  pedido.data.attributes.diagnostic_main  \
count                  830.000000                              830.000000
mean                     0.323607                               -0.039986
std                      1.006114                                0.650873
min                     -1.554838                               -1.503163
25%                     -1.006628                               -0.586347
50%                      0.737677                                0.101264
```

```
75%                      1.236049                                0.215866
max                      1.584910                                1.820293

       pedido.data.attributes.gender    articulo  \
count                          830.0  830.000000
mean                             0.0    0.076390
std                              0.0    0.742731
min                              0.0   -1.880409
25%                              0.0   -0.309977
50%                              0.0    0.301868
75%                              0.0    0.553395
max                              0.0    1.215055

       respuesta.articlesRevisedYear  respuesta.articlesRevisedMonth  \
count                     830.000000                      830.000000
mean                        0.093236                       -0.168806
std                         1.156247                        1.082831
min                        -2.656375                       -1.463658
25%                        -0.189157                       -1.178433
50%                         0.427648                       -0.893208
75%                         1.044452                        1.103364
max                         1.044452                        1.673814

       respuesta.pubmed_keys    utilidad
count             830.000000  830.000000
mean               -0.031101    0.583133
std                 0.925381    0.493338
min                -1.089722    0.000000
25%                -0.904532    0.000000
50%                -0.358219    1.000000
75%                 0.808482    1.000000
max                 2.049259    1.000000
```

We check the number of results

[17]:
```
dfDataSetComplete.groupby('utilidad').size()
```

[17]:
```
utilidad
0.0    346
1.0    484
dtype: int64
```

Choosed “age”, “diagnostic_main”, “month” and “pubmed_keys” attributes (based on PCA_V3 study)

[18]:
```
dataToTrain = dfDataSetComplete[["pedido.data.attributes.age",
    "pedido.data.attributes.diagnostic_main",
    "respuesta.articlesRevisedMonth",
    "respuesta.pubmed_keys",
    "utilidad"
]]


X = np.array(dataToTrain.drop(['utilidad'],1))
y = np.array(dataToTrain['utilidad'])
X.shape
```

[18]:
```
(830, 4)
```

[19]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
```

[20]:
```
model = linear_model.LogisticRegression()
model.fit(X_train,y_train)

model.score(X_train,y_train)
```

[20]:
```
0.5627009646302251
```

[21]:
```
# see: https://es.wikipedia.org/wiki/Validaci%C3%B3n_cruzada
kfold = model_selection.KFold(n_splits=10)
cv_results = model_selection.cross_val_score(model, X_train, y_train, cv=kfold,␣
 ↪scoring='accuracy')
msg = "%s: %f (%f)" % ("Logistic Regression", cv_results.mean(), cv_results.
 ↪std())
print(msg)
```

```
Logistic Regression: 0.554531 (0.059046)
```

[22]:
```
predictions = model.predict(X_test)
print(accuracy_score(y_test, predictions))
```

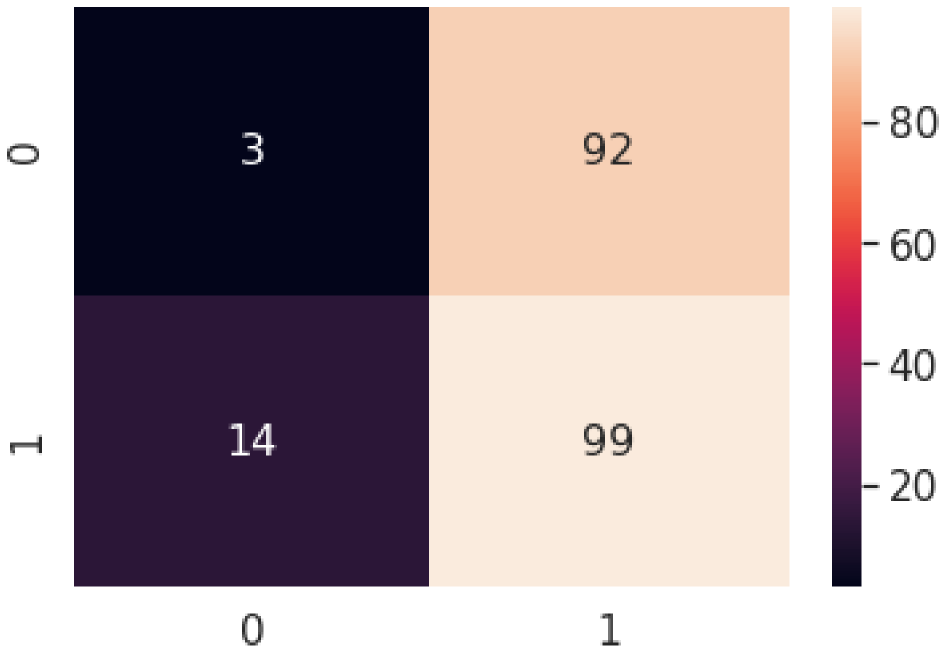
```
0.49038461538461536
```

[23]:
```
import seaborn as sn
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
```

[24]:
```
cf = confusion_matrix(y_test, predictions)

df_cm = pd.DataFrame(cf, range(2), range(2))
sn.set(font_scale=1.4) # for label size
sn.heatmap(df_cm, annot=True, annot_kws={"size": 16}) # font size

plt.show()
```



0.6 Run Prediction

[25]:
```
result = model.predict(dfDataSetToPredict[["pedido.data.attributes.age",
    "pedido.data.attributes.diagnostic_main",
    "respuesta.articlesRevisedMonth",
    "respuesta.pubmed_keys"
]])

result
```

[25]:
```
array([0., 0., 0., …, 1., 1., 1.])
```

0.7 Try with all atributes

[26]:
```
X = np.array(dfDataSetComplete.drop(['utilidad'],1))
y = np.array(dfDataSetComplete['utilidad'])
X.shape
```

[26]:
```
(830, 7)
```

[27]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
model = linear_model.LogisticRegression()
model.fit(X_train,y_train)

model.score(X_train,y_train)
```

```
[27]: 0.5530546623794212
```

```
[28]: kfold = model_selection.KFold(n_splits=10)
      cv_results = model_selection.cross_val_score(model, X_train, y_train, cv=kfold,␣
       ↪scoring='accuracy')
      msg = "%s: %f (%f)" % ("Logistic Regression", cv_results.mean(), cv_results.
       ↪std())
      print(msg)
```

```
Logistic Regression: 0.541654 (0.047477)
```

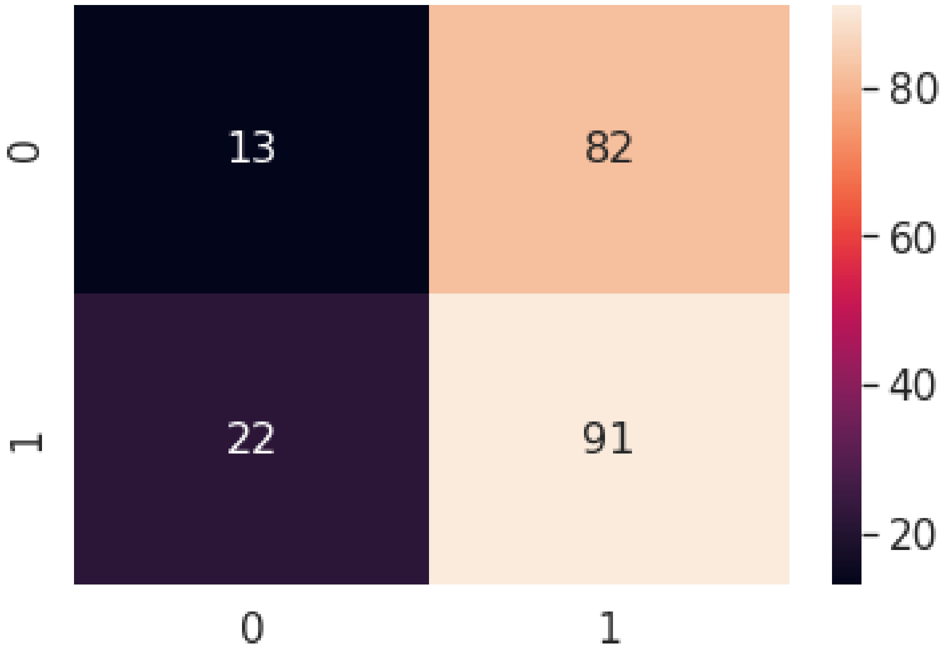
```
[29]: predictions = model.predict(X_test)
      print(accuracy_score(y_test, predictions))
```

```
0.5
```

```
[30]: cf = confusion_matrix(y_test, predictions)

      df_cm = pd.DataFrame(cf, range(2), range(2))
      sn.set(font_scale=1.4) # for label size
      sn.heatmap(df_cm, annot=True, annot_kws={"size": 16}) # font size

      plt.show()
```



```
[ ]:
```