Notebook8-checkpoint

May 22, 2020

[1]:
```
#Final Project Machine learning
```

[2]:
```
#For Machine Learning I will use the following two data sets
```

[5]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
%matplotlib inline
```

[6]:
```
cancer_type_df_1 = pd.read_csv('/home/haziz/Data/cancer-death-rates-by-age.csv')
cancer_type_df_1
```

[6]:
```
           Entity Code  Year  Under-5s (per 100,000)  \
0     Afghanistan  AFG  1990                15.827827
1     Afghanistan  AFG  1991                14.466737
2     Afghanistan  AFG  1992                11.643169
3     Afghanistan  AFG  1993                10.836940
4     Afghanistan  AFG  1994                11.640549
…             …  …    …                         …
6463     Zimbabwe  ZWE  2013                 7.687691
6464     Zimbabwe  ZWE  2014                 7.532188
6465     Zimbabwe  ZWE  2015                 7.430483
6466     Zimbabwe  ZWE  2016                 7.440329
6467     Zimbabwe  ZWE  2017                 7.307537

      Age-standardized (per 100,000)  \
0                          142.098540
1                          142.244809
2                          142.764864
3                          144.315587
4                          146.378153
…                                 …
6463                       177.273322
6464                       173.645053
6465                       170.957412
6466                       168.919629
```

```
6467                      166.321697

      All ages (not age-standardized) (per 100,000)  \
0                                          100.554579
1                                           97.843372
2                                           86.269536
3                                           77.755180
4                                           76.304587
…                                                 …
6463                                        78.863740
6464                                        78.031679
6465                                        77.641326
6466                                        77.831585
6467                                        77.749176

      70+ years old (per 100,000)  5-14 years old (per 100,000)  \
0                      926.438867                       6.165860
1                      927.141275                       5.993131
2                      931.368776                       5.864208
3                      937.067105                       6.114936
4                      943.803688                       6.291496
…                             …                              …
6463                  1290.218007                       6.942269
6464                  1268.341354                       6.934370
6465                  1251.483425                       6.712509
6466                  1232.083866                       6.701618
6467                  1209.876411                       6.592164

      50-69 years old (per 100,000)  15-49 years old (per 100,000)
0                         371.906483                       42.814564
1                         373.880797                       40.830377
2                         376.820796                       36.218385
3                         381.654423                       33.445856
4                         387.996533                       33.314720
…                                …                              …
6463                      467.327755                       36.506438
6464                      457.757157                       36.091182
6465                      449.814316                       35.993954
6466                      445.379406                       36.091215
6467                      438.714110                       36.042900

[6468 rows x 10 columns]
```

[7]:
```
cancer_type_df_1 = cancer_type_df_1[(cancer_type_df_1['Year'] >=2002)]
```

[8]:
```
cancer_type_df_1
```

[8]:

	Entity	Code	Year	Under-5s (per 100,000)	Age-standardized (per 100,000)	All ages (not age-standardized) (per 100,000)	70+ years old (per 100,000)	5-14 years old (per 100,000)
12	Afghanistan	AFG	2002	9.629008	154.779744	67.163280	982.926077	5.950292
13	Afghanistan	AFG	2003	12.123571	155.132961	65.421248	982.996623	7.835944
14	Afghanistan	AFG	2004	12.990653	155.201207	64.679520	981.647489	8.463892
15	Afghanistan	AFG	2005	12.564291	154.185158	62.144043	981.172033	8.412478
16	Afghanistan	AFG	2006	12.183719	153.465735	61.194630	978.305198	8.175820
…	…	…	…	…	…	…	…	…
6463	Zimbabwe	ZWE	2013	7.687691	177.273322	78.863740	1290.218007	6.942269
6464	Zimbabwe	ZWE	2014	7.532188	173.645053	78.031679		
6465	Zimbabwe	ZWE	2015	7.430483	170.957412	77.641326		
6466	Zimbabwe	ZWE	2016	7.440329	168.919629	77.831585		
6467	Zimbabwe	ZWE	2017	7.307537	166.321697	77.749176		

```
6464                  1268.341354                        6.934370
6465                  1251.483425                        6.712509
6466                  1232.083866                        6.701618
6467                  1209.876411                        6.592164

      50-69 years old (per 100,000)  15-49 years old (per 100,000)
12                        417.787357                        38.137765
13                        420.602521                        38.166120
14                        424.412499                        39.145357
15                        424.259712                        39.114839
16                        427.415811                        40.088267
…                                …                                …
6463                      467.327755                        36.506438
6464                      457.757157                        36.091182
6465                      449.814316                        35.993954
6466                      445.379406                        36.091215
6467                      438.714110                        36.042900

[3696 rows x 10 columns]
```

[9]:
```
#Picking the countries

cancer_type_df_1  = cancer_type_df_1 .query('Entity in ["Norway", "Switzerland",
 ↪", "Ireland" , "Germany" , "Australia" , "Iceland" , "United Kingdom" , "United
 ↪States" , "Finland" , "Japan", "Pakistan" , "Yemen" , "Liberia" , "Guinea",
 ↪"Congo", "Mozambique" , "Afghanistan" , "Zimbabwe" , "Syria" , "Iraq"] ')

cancer_type_df_1
```

[9]:
```
           Entity Code  Year  Under-5s (per 100,000)  \
12    Afghanistan  AFG  2002                9.629008
13    Afghanistan  AFG  2003               12.123571
14    Afghanistan  AFG  2004               12.990653
15    Afghanistan  AFG  2005               12.564291
16    Afghanistan  AFG  2006               12.183719
…             …  …    …                        …
6463     Zimbabwe  ZWE  2013                7.687691
6464     Zimbabwe  ZWE  2014                7.532188
6465     Zimbabwe  ZWE  2015                7.430483
6466     Zimbabwe  ZWE  2016                7.440329
6467     Zimbabwe  ZWE  2017                7.307537

      Age-standardized (per 100,000)  \
12                         154.779744
13                         155.132961
14                         155.201207
15                         154.185158
```

```
16                        153.465735
…                              …
6463                      177.273322
6464                      173.645053
6465                      170.957412
6466                      168.919629
6467                      166.321697

      All ages (not age-standardized) (per 100,000)  \
12                                          67.163280
13                                          65.421248
14                                          64.679520
15                                          62.144043
16                                          61.194630
…                                                …
6463                                        78.863740
6464                                        78.031679
6465                                        77.641326
6466                                        77.831585
6467                                        77.749176

      70+ years old (per 100,000)  5-14 years old (per 100,000)  \
12                     982.926077                       5.950292
13                     982.996623                       7.835944
14                     981.647489                       8.463892
15                     981.172033                       8.412478
16                     978.305198                       8.175820
…                             …                              …
6463                  1290.218007                       6.942269
6464                  1268.341354                       6.934370
6465                  1251.483425                       6.712509
6466                  1232.083866                       6.701618
6467                  1209.876411                       6.592164

      50-69 years old (per 100,000)  15-49 years old (per 100,000)
12                       417.787357                       38.137765
13                       420.602521                       38.166120
14                       424.412499                       39.145357
15                       424.259712                       39.114839
16                       427.415811                       40.088267
…                               …                              …
6463                     467.327755                       36.506438
6464                     457.757157                       36.091182
6465                     449.814316                       35.993954
6466                     445.379406                       36.091215
6467                     438.714110                       36.042900
```

```
[320 rows x 10 columns]
```

[10]:
```
cancer_type_df_1['Entity'] = cancer_type_df_1['Entity'].map({ "Norway": 0,␣
↪"Switzerland": 1, "Ireland": 2, "Germany": 3, "Australia": 4, "Iceland": 5,␣
↪"United Kingdom": 6, "United States": 7, "Finland": 8, "Japan": 9, "Pakistan":
↪ 10, "Yemen": 11, "Liberia": 12, "Guinea": 13, "Congo": 14, "Mozambique": 15,␣
↪"Afghanistan": 16, "Zimbabwe": 17, "Syria": 18, "Iraq": 19 })
```

```
/usr/lib/python3.6/site-packages/IPython/kernel/__main__.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  if __name__ == '__main__':
```

[11]:
```
cancer_type_df_1.isnull().sum()
```

[11]:
```
Entity                                          0
Code                                            0
Year                                            0
Under-5s (per 100,000)                          0
Age-standardized (per 100,000)                  0
All ages (not age-standardized) (per 100,000)   0
70+ years old (per 100,000)                     0
5-14 years old (per 100,000)                    0
50-69 years old (per 100,000)                   0
15-49 years old (per 100,000)                   0
dtype: int64
```

[12]:
```
cancer_type_df_1 = cancer_type_df_1.dropna()
```

[13]:
```
cancer_type_df_1.isnull().sum()
```

[13]:
```
Entity                                          0
Code                                            0
Year                                            0
Under-5s (per 100,000)                          0
Age-standardized (per 100,000)                  0
All ages (not age-standardized) (per 100,000)   0
70+ years old (per 100,000)                     0
5-14 years old (per 100,000)                    0
50-69 years old (per 100,000)                   0
15-49 years old (per 100,000)                   0
dtype: int64
```

[14]:
```
cancer_type_df_1
```

[14]:

	Entity	Code	Year	Under-5s (per 100,000)	\
12	16	AFG	2002	9.629008	
13	16	AFG	2003	12.123571	
14	16	AFG	2004	12.990653	
15	16	AFG	2005	12.564291	
16	16	AFG	2006	12.183719	
…	…	…	…	…	
6463	17	ZWE	2013	7.687691	
6464	17	ZWE	2014	7.532188	
6465	17	ZWE	2015	7.430483	
6466	17	ZWE	2016	7.440329	
6467	17	ZWE	2017	7.307537	

	Age-standardized (per 100,000)	\
12	154.779744	
13	155.132961	
14	155.201207	
15	154.185158	
16	153.465735	
…	…	
6463	177.273322	
6464	173.645053	
6465	170.957412	
6466	168.919629	
6467	166.321697	

	All ages (not age-standardized) (per 100,000)	\
12	67.163280	
13	65.421248	
14	64.679520	
15	62.144043	
16	61.194630	
…	…	
6463	78.863740	
6464	78.031679	
6465	77.641326	
6466	77.831585	
6467	77.749176	

	70+ years old (per 100,000)	5-14 years old (per 100,000)	\
12	982.926077	5.950292	
13	982.996623	7.835944	
14	981.647489	8.463892	
15	981.172033	8.412478	
16	978.305198	8.175820	

```
…                              …                                …
6463                  1290.218007                         6.942269
6464                  1268.341354                         6.934370
6465                  1251.483425                         6.712509
6466                  1232.083866                         6.701618
6467                  1209.876411                         6.592164

      50-69 years old (per 100,000)  15-49 years old (per 100,000)
12                        417.787357                        38.137765
13                        420.602521                        38.166120
14                        424.412499                        39.145357
15                        424.259712                        39.114839
16                        427.415811                        40.088267
…                                …                                …
6463                      467.327755                        36.506438
6464                      457.757157                        36.091182
6465                      449.814316                        35.993954
6466                      445.379406                        36.091215
6467                      438.714110                        36.042900

[320 rows x 10 columns]
```

[ ]:

[15]:
```
#Class Distribution
cancer_type_df_1.groupby('Entity').size()
```

[15]:
```
Entity
0     16
1     16
2     16
3     16
4     16
5     16
6     16
7     16
8     16
9     16
10    16
11    16
12    16
13    16
14    16
15    16
16    16
17    16
18    16
```

```
19    16
dtype: int64
```

[16]:
```
#cancer_type_df_1['Entity'] = cancer_type_df_5.fit_transform(data['Entity'])
cancer_type_df_1 = cancer_type_df_1.drop(['Code'],axis=1)
```

[17]:
```
cancer_type_df_1 = cancer_type_df_1.dropna()
```

[18]:
```
cancer_type_df_1.describe()
```

[18]:

```
           Entity         Year  Under-5s (per 100,000)  \
count  320.000000   320.000000              320.000000
mean     9.500000  2009.500000                6.059764
std      5.775312     4.616992                3.900891
min      0.000000  2002.000000                1.902172
25%      4.750000  2005.750000                3.049887
50%      9.500000  2009.500000                3.944563
75%     14.250000  2013.250000                8.375547
max     19.000000  2017.000000               18.231105

       Age-standardized (per 100,000)  \
count                      320.000000
mean                       127.173799
std                         27.940519
min                         56.616802
25%                        114.094224
50%                        132.309018
75%                        142.385689
max                        214.641780

       All ages (not age-standardized) (per 100,000)  \
count                                      320.000000
mean                                       142.366081
std                                         90.701380
min                                         30.078081
25%                                         54.281577
50%                                        135.083110
75%                                        217.944159
max                                        323.066029

       70+ years old (per 100,000)  5-14 years old (per 100,000)  \
count                   320.000000                     320.000000
mean                   1091.316900                       4.362954
std                     288.003087                       2.605084
min                     403.259280                       1.940484
25%                     928.641961                       2.744915
50%                    1130.680778                       3.469463
```

```
75%                    1298.660024                          5.210946
max                    1563.410433                         14.124861

       50-69 years old (per 100,000)  15-49 years old (per 100,000)
count                     320.000000                     320.000000
mean                      313.500645                      26.126200
std                        78.999059                       7.378051
min                       134.160438                      14.167761
25%                       272.466489                      21.057883
50%                       315.579799                      24.816131
75%                       360.328803                      29.649243
max                       551.251432                      46.265123
```

[19]:
```
#set up the data
X = cancer_type_df_1.drop('Entity',axis=1)
y = cancer_type_df_1['Entity']
```

[20]:
```
from sklearn.model_selection import train_test_split
```

[21]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y)
```

[22]:
```
from sklearn.tree import DecisionTreeClassifier
```

[23]:
```
model_cancer = DecisionTreeClassifier().fit(X, y)
```

[24]:
```
#Steps to test accuracy score
```

[25]:
```
y_pred_class = model_cancer.predict(X)
```

[26]:
```
from sklearn import metrics
```

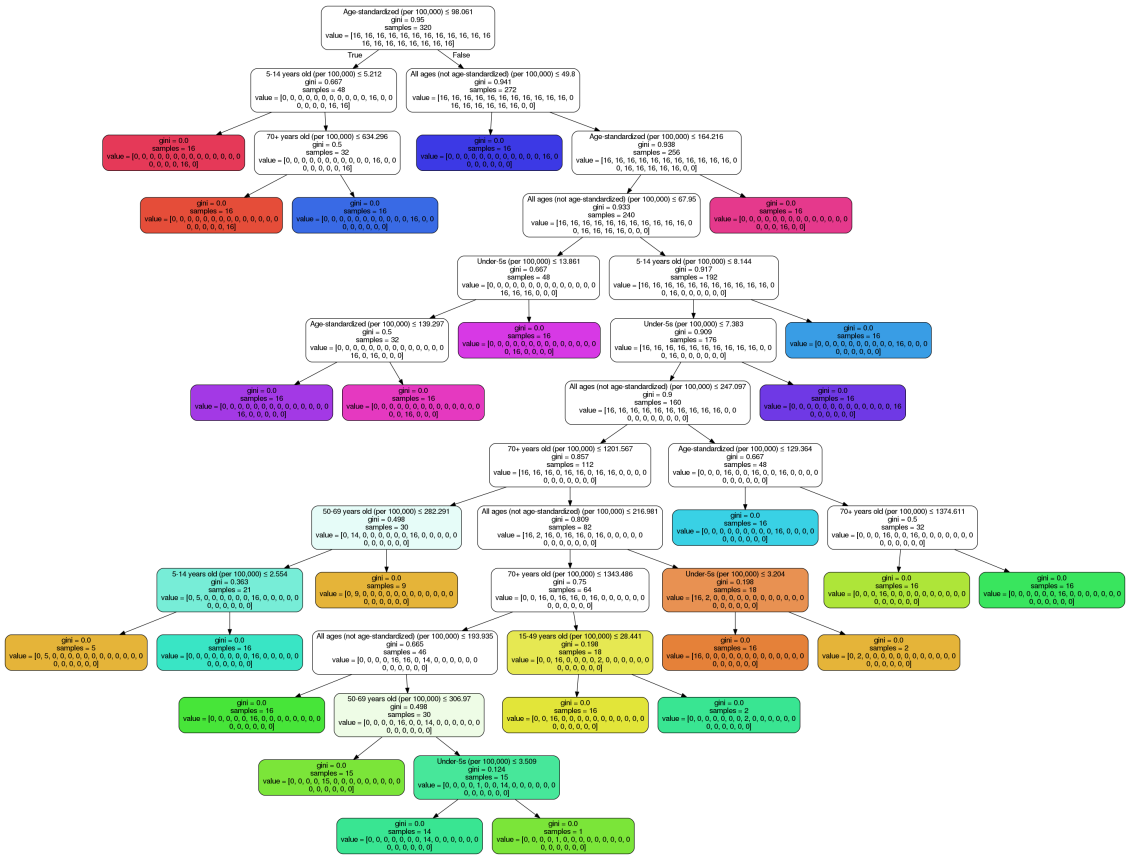
[27]:
```
import pydotplus
from sklearn.externals.six import StringIO
from IPython.display import Image
from sklearn.tree import export_graphviz
```

```
/usr/lib/python3.6/site-packages/sklearn/externals/six.py:31:
DeprecationWarning: The module is deprecated in version 0.21 and will be removed
in version 0.23 since we've dropped support for Python 2.7. Please rely on the
official version of six (https://pypi.org/project/six/).
  "(https://pypi.org/project/six/).", DeprecationWarning)
```

[28]:
```
dot_data = StringIO() #output your visualization to a PNG file
export_graphviz(model_cancer, #DATA
out_file=dot_data,
feature_names=X.columns,
filled=True,
```

```
rounded=True,
special_characters=True)
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
Image(graph.create_png()) # PNG saved to your local directory
```

[28]:



[44]:
```
from sklearn.metrics import accuracy_score
```

[46]:
```
print('Model accuracy score with criterion gini index: {0:0.4f}'.␣
 ↪format(accuracy_score(y, y_pred_class)))
```

```
Model accuracy score with criterion gini index: 1.0000
```

[32]:
```
from sklearn.metrics import classification_report,confusion_matrix
```

[33]:
```
from sklearn.neural_network import MLPClassifier
```

[36]:
```
print(confusion_matrix(y, y))
```

```
[[16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0 16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
```

```
[ 0  0  0 16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0 16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0 16  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0 16  0  0  0  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0 16  0  0  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0 16  0  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0 16  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0  0 16  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0 16  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0 16  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0  0 16  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0 16  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 16  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 16  0  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 16  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 16  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 16]]
```

[ ]:
```
from sklearn.metrics import classification_report
#checking for the accuracy
```

[42]:
```
print(classification_report(y, y_pred_class))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        16
           1       1.00      1.00      1.00        16
           2       1.00      1.00      1.00        16
           3       1.00      1.00      1.00        16
           4       1.00      1.00      1.00        16
           5       1.00      1.00      1.00        16
           6       1.00      1.00      1.00        16
           7       1.00      1.00      1.00        16
           8       1.00      1.00      1.00        16
           9       1.00      1.00      1.00        16
          10       1.00      1.00      1.00        16
          11       1.00      1.00      1.00        16
          12       1.00      1.00      1.00        16
          13       1.00      1.00      1.00        16
          14       1.00      1.00      1.00        16
          15       1.00      1.00      1.00        16
          16       1.00      1.00      1.00        16
          17       1.00      1.00      1.00        16
          18       1.00      1.00      1.00        16
          19       1.00      1.00      1.00        16

    accuracy                           1.00       320
   macro avg       1.00      1.00      1.00       320
```

```
weighted avg       1.00      1.00      1.00       320
```

[ ]: