In [1]:
```python
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.tree import export_graphviz
from six import StringIO
from IPython.display import Image
import pydotplus
```

In [2]:
```python
parameters = pd.read_csv(r"D:\Detecting parkinsons disease\cleaned-data.csv")
parameters.head()
```

Out[2]:

	Unnamed: 0	name	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DI
0	0	phon_R01_S01_1	119.992	157.302	74.997	0.00784	0.000070	0.00370	0.00554	0.011
1	1	phon_R01_S01_2	122.400	148.650	113.819	0.00968	0.000080	0.00465	0.00576	0.013
2	2	phon_R01_S01_3	116.682	131.111	111.555	0.01050	0.000090	0.00544	0.00576	0.016
3	3	phon_R01_S01_4	116.676	137.871	111.366	0.00997	0.000090	0.00502	0.00576	0.015
4	4	phon_R01_S01_5	116.014	141.781	110.655	0.01101	0.000037	0.00593	0.00576	0.017

5 rows × 25 columns

In [3]:
```python
columns = list(parameters.columns)
columns.remove('name')
columns
```

Out[3]:
```
['Unnamed: 0',
 'MDVP:Fo(Hz)',
 'MDVP:Fhi(Hz)',
 'MDVP:Flo(Hz)',
 'MDVP:Jitter(%)',
 'MDVP:Jitter(Abs)',
 'MDVP:RAP',
 'MDVP:PPQ',
 'Jitter:DDP',
 'MDVP:Shimmer',
 'MDVP:Shimmer(dB)',
 'Shimmer:APQ3',
 'Shimmer:APQ5',
 'MDVP:APQ',
 'Shimmer:DDA',
 'NHR',
 'HNR',
 'status',
 'RPDE',
 'DFA',
 'spread1',
 'spread2',
 'D2',
 'PPE']
```

In [4]:
```python
feature_columns = columns
feature_columns.remove('status')
feature_columns
```

Out[4]:
```
['Unnamed: 0',
 'MDVP:Fo(Hz)',
 'MDVP:Fhi(Hz)',
 'MDVP:Flo(Hz)',
 'MDVP:Jitter(%)',
 'MDVP:Jitter(Abs)',
 'MDVP:RAP',
 'MDVP:PPQ',
 'Jitter:DDP',
 'MDVP:Shimmer',
 'MDVP:Shimmer(dB)',
 'Shimmer:APQ3',
 'Shimmer:APQ5',
 'MDVP:APQ',
 'Shimmer:DDA',
 'NHR',
 'HNR',
 'RPDE',
 'DFA',
 'spread1',
 'spread2',
 'D2',
 'PPE']
```

In [5]:
```python
x = parameters[feature_columns]
y = parameters.status
x_train,x_test,y_train,y_test = train_test_split(x, y, test_size=0.3, random_state=1)
```

In [6]:
```python
classifier = DecisionTreeClassifier(criterion="entropy", max_depth=3)
classifier = classifier.fit(x_train,y_train)
y_pred = classifier.predict(x_test)
```

In [7]:
```python
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.9322033898305084
```

In [8]:
```python
dot_data = StringIO()
export_graphviz(classifier, out_file=dot_data,
                filled=True, rounded=True,
                special_characters=True,feature_names = feature_columns,class_names=['0','1'])
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('diabetes.png')
Image(graph.create_png())
```

```
---------------------------------------------------------------------------
InvocationException                       Traceback (most recent call last)
<ipython-input-8-0bdbd50bb4b1> in <module>
      4                 special_characters=True,feature_names = feature_columns,class_names=['0','1'])
      5 graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
----> 6 graph.write_png('diabetes.png')
      7 Image(graph.create_png())

D:\Anaconda\lib\site-packages\pydotplus\graphviz.py in <lambda>(path, f, prog)
   1808                 lambda path,
   1809                 f=frmt,
-> 1810                 prog=self.prog: self.write(path, format=f, prog=prog)
   1811             )
   1812

D:\Anaconda\lib\site-packages\pydotplus\graphviz.py in write(self, path, prog, format)
   1916
   1917             else:
-> 1918                 fobj.write(self.create(prog, format))
   1919         finally:
   1920             if close:

D:\Anaconda\lib\site-packages\pydotplus\graphviz.py in create(self, prog, format)
   1957             self.progs = find_graphviz()
   1958             if self.progs is None:
-> 1959                 raise InvocationException(
   1960                     'GraphViz\'s executables not found')
   1961

InvocationException: GraphViz's executables not found
```

In [9]:
```python
from dtreeviz.trees import dtreeviz
```

In [10]:
```python
viz = dtreeviz(classifier, x, y,
                target_name="target",
                feature_names=feature_columns,
                class_names=['0','1'])
```

In [11]:
```python
viz
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
D:\Anaconda\lib\site-packages\graphviz\backend.py in run(cmd, input, capture_output, check, encoding, quiet, **
kwargs)
    163     try:
--> 164         proc = subprocess.Popen(cmd, startupinfo=get_startupinfo(), **kwargs)
    165     except OSError as e:

D:\Anaconda\lib\subprocess.py in __init__(self, args, bufsize, executable, stdin, stdout, stderr, preexec_fn, c
lose_fds, shell, cwd, env, universal_newlines, startupinfo, creationflags, restore_signals, start_new_session,
 pass_fds, encoding, errors, text)
    853
--> 854             self._execute_child(args, executable, preexec_fn, close_fds,
    855                                 pass_fds, cwd, env,

D:\Anaconda\lib\subprocess.py in _execute_child(self, args, executable, preexec_fn, close_fds, pass_fds, cwd, e
nv, startupinfo, creationflags, shell, p2cread, p2cwrite, c2pread, c2pwrite, errread, errwrite, unused_restore_
signals, unused_start_new_session)
   1306             try:
-> 1307                 hp, ht, pid, tid = _winapi.CreateProcess(executable, args,
   1308                                          # no special security

FileNotFoundError: [WinError 2] The system cannot find the file specified

During handling of the above exception, another exception occurred:

ExecutableNotFound                        Traceback (most recent call last)
D:\Anaconda\lib\site-packages\IPython\core\formatters.py in __call__(self, obj)
    343             method = get_real_method(obj, self.print_method)
    344             if method is not None:
--> 345                 return method()
    346             return None
    347         else:

D:\Anaconda\lib\site-packages\dtreeviz\trees.py in _repr_svg_(self)
     35
     36     def _repr_svg_(self):
---> 37         return self.svg()
     38
     39     def svg(self):

D:\Anaconda\lib\site-packages\dtreeviz\trees.py in svg(self)
     39     def svg(self):
     40         """Render tree as svg and return svg text."""
---> 41         svgfilename = self.save_svg()
     42         with open(svgfilename, encoding='UTF-8') as f:
     43             svg = f.read()

D:\Anaconda\lib\site-packages\dtreeviz\trees.py in save_svg(self)
     52         tmp = tempfile.gettempdir()
     53         svgfilename = os.path.join(tmp, f"DTreeViz_{os.getpid()}.svg")
---> 54         self.save(svgfilename)
     55         return svgfilename
     56

D:\Anaconda\lib\site-packages\dtreeviz\trees.py in save(self, filename)
     77         cmd = ["dot", f"-T{format}", "-o", filename, dotfilename]
     78         # print(' '.join(cmd))
---> 79         run(cmd, capture_output=True, check=True, quiet=False)
     80
     81         if filename.endswith(".svg"):

D:\Anaconda\lib\site-packages\graphviz\backend.py in run(cmd, input, capture_output, check, encoding, quiet, **
kwargs)
    165     except OSError as e:
    166         if e.errno == errno.ENOENT:
--> 167             raise ExecutableNotFound(cmd)
    168         else:
    169             raise

ExecutableNotFound: failed to execute ['dot', '-Tsvg', '-o', 'C:\\Users\\Lenovo\\AppData\\Local\\Temp\\DTreeViz
_7268.svg', 'C:/Users/Lenovo/AppData/Local/Temp\\DTreeViz_7268'], make sure the Graphviz executables are on you
r systems' PATH
```

Out[11]:
```
<dtreeviz.trees.DTreeViz at 0x2c0c89f8460>
```

In [12]:
```python
from sklearn import datasets
from sklearn.tree import DecisionTreeRegressor
from sklearn import tree
```

In [13]:
```python
regr = DecisionTreeRegressor(max_depth=3, random_state=1234)
model = regr.fit(x, y)
```

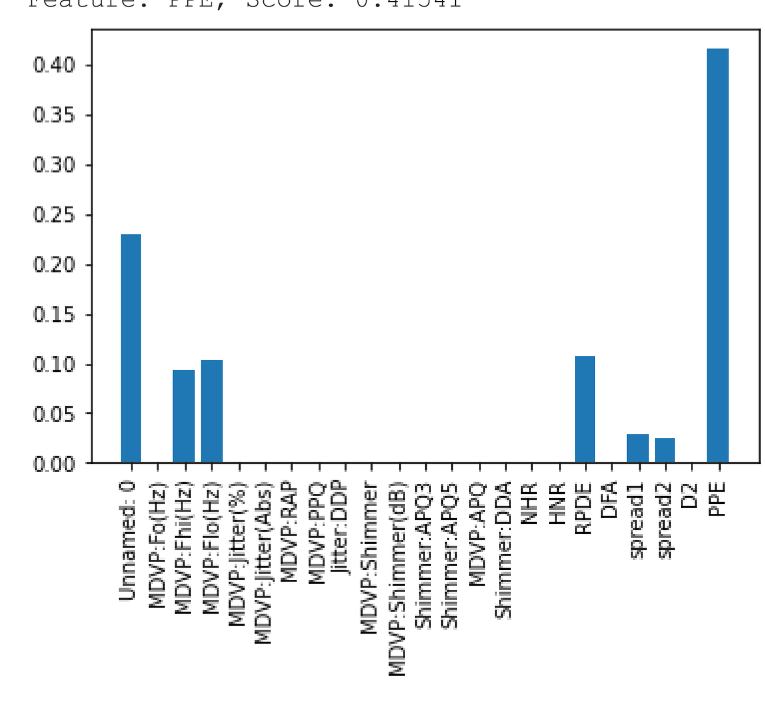
In [14]:
```python
viz = dtreeviz(regr, x, y,
                target_name="target",
                feature_names=feature_columns)
viz
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
D:\Anaconda\lib\site-packages\graphviz\backend.py in run(cmd, input, capture_output, check, encoding, quiet, **
kwargs)
    163     try:
--> 164         proc = subprocess.Popen(cmd, startupinfo=get_startupinfo(), **kwargs)
    165     except OSError as e:

D:\Anaconda\lib\subprocess.py in __init__(self, args, bufsize, executable, stdin, stdout, stderr, preexec_fn, c
lose_fds, shell, cwd, env, universal_newlines, startupinfo, creationflags, restore_signals, start_new_session,
 pass_fds, encoding, errors, text)
    853
--> 854             self._execute_child(args, executable, preexec_fn, close_fds,
    855                                 pass_fds, cwd, env,

D:\Anaconda\lib\subprocess.py in _execute_child(self, args, executable, preexec_fn, close_fds, pass_fds, cwd, e
nv, startupinfo, creationflags, shell, p2cread, p2cwrite, c2pread, c2pwrite, errread, errwrite, unused_restore_
signals, unused_start_new_session)
   1306             try:
-> 1307                 hp, ht, pid, tid = _winapi.CreateProcess(executable, args,
   1308                                          # no special security

FileNotFoundError: [WinError 2] The system cannot find the file specified

During handling of the above exception, another exception occurred:

ExecutableNotFound                        Traceback (most recent call last)
D:\Anaconda\lib\site-packages\IPython\core\formatters.py in __call__(self, obj)
    343             method = get_real_method(obj, self.print_method)
    344             if method is not None:
--> 345                 return method()
    346             return None
    347         else:

D:\Anaconda\lib\site-packages\dtreeviz\trees.py in _repr_svg_(self)
     35
     36     def _repr_svg_(self):
---> 37         return self.svg()
     38
     39     def svg(self):

D:\Anaconda\lib\site-packages\dtreeviz\trees.py in svg(self)
     39     def svg(self):
     40         """Render tree as svg and return svg text."""
---> 41         svgfilename = self.save_svg()
     42         with open(svgfilename, encoding='UTF-8') as f:
     43             svg = f.read()

D:\Anaconda\lib\site-packages\dtreeviz\trees.py in save_svg(self)
     52         tmp = tempfile.gettempdir()
     53         svgfilename = os.path.join(tmp, f"DTreeViz_{os.getpid()}.svg")
---> 54         self.save(svgfilename)
     55         return svgfilename
     56

D:\Anaconda\lib\site-packages\dtreeviz\trees.py in save(self, filename)
     77         cmd = ["dot", f"-T{format}", "-o", filename, dotfilename]
     78         # print(' '.join(cmd))
---> 79         run(cmd, capture_output=True, check=True, quiet=False)
     80
     81         if filename.endswith(".svg"):

D:\Anaconda\lib\site-packages\graphviz\backend.py in run(cmd, input, capture_output, check, encoding, quiet, **
kwargs)
    165     except OSError as e:
    166         if e.errno == errno.ENOENT:
--> 167             raise ExecutableNotFound(cmd)
    168         else:
    169             raise

ExecutableNotFound: failed to execute ['dot', '-Tsvg', '-o', 'C:\\Users\\Lenovo\\AppData\\Local\\Temp\\DTreeViz
_7268.svg', 'C:/Users/Lenovo/AppData/Local/Temp\\DTreeViz_7268'], make sure the Graphviz executables are on you
r systems' PATH
```

Out[14]:
```
<dtreeviz.trees.DTreeViz at 0x2c0c96561c0>
```

In [15]:
```python
from matplotlib import pyplot
importance = model.feature_importances_
for i,v in enumerate(importance):
        print('Feature: %s, Score: %.5f' % (feature_columns[i],v))
pyplot.bar(feature_columns,importance)
pyplot.xticks(rotation='vertical')
pyplot.show()
```

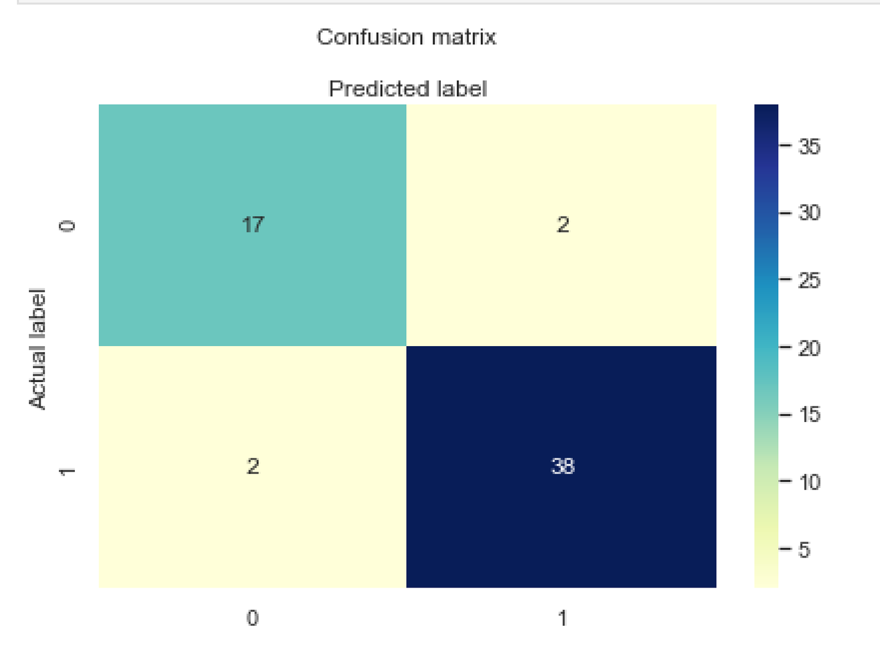
```
Feature: Unnamed: 0, Score: 0.22945
Feature: MDVP:Fo(Hz), Score: 0.00000
Feature: MDVP:Fhi(Hz), Score: 0.09236
Feature: MDVP:Flo(Hz), Score: 0.10253
Feature: MDVP:Jitter(%), Score: 0.00000
Feature: MDVP:Jitter(Abs), Score: 0.00000
Feature: MDVP:RAP, Score: 0.00000
Feature: MDVP:PPQ, Score: 0.00000
Feature: Jitter:DDP, Score: 0.00000
Feature: MDVP:Shimmer, Score: 0.00000
Feature: MDVP:Shimmer(dB), Score: 0.00000
Feature: Shimmer:APQ3, Score: 0.00000
Feature: Shimmer:APQ5, Score: 0.00000
Feature: MDVP:APQ, Score: 0.00000
Feature: Shimmer:DDA, Score: 0.00000
Feature: NHR, Score: 0.00000
Feature: HNR, Score: 0.00000
Feature: RPDE, Score: 0.10648
Feature: DFA, Score: 0.00000
Feature: spread1, Score: 0.02818
Feature: spread2, Score: 0.02559
Feature: D2, Score: 0.00000
Feature: PPE, Score: 0.41541
```



In [16]:
```python
from sklearn.metrics import confusion_matrix
cnf = confusion_matrix(y_test,y_pred)
cnf
```

Out[16]:
```
array([[17,  2],
       [ 2, 38]], dtype=int64)
```

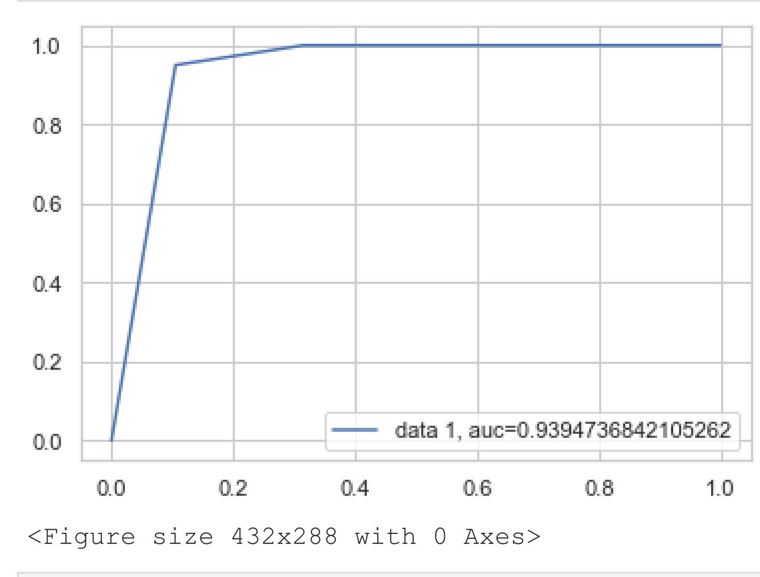
In [17]:
```python
import seaborn as sns
import numpy as np
sns.set(style="white")
sns.set(style="whitegrid", color_codes=True)
class_names=[0,1] # name  of classes
fig, ax = pyplot.subplots()
tick_marks = np.arange(len(class_names))
pyplot.xticks(tick_marks, class_names)
pyplot.yticks(tick_marks, class_names)
# create heatmap
sns.heatmap(pd.DataFrame(cnf), annot=True, cmap="YlGnBu" ,fmt='g')
ax.xaxis.set_label_position("top")
pyplot.tight_layout()
pyplot.title('Confusion matrix', y=1.1)
pyplot.ylabel('Actual label')
pyplot.xlabel('Predicted label')
pyplot.savefig("HeatMap")
```



In [18]:
```python
from sklearn.metrics import precision_score, recall_score
print("Precision : ",precision_score(y_test, y_pred))
print("Recall",recall_score(y_test, y_pred))
```

```
Precision :  0.95
Recall 0.95
```

In [19]:
```python
y_pred_proba = classifier.predict_proba(x_test)[::,1]
fpr, tpr, _ = metrics.roc_curve(y_test,  y_pred_proba)
auc = metrics.roc_auc_score(y_test, y_pred_proba)
pyplot.plot(fpr,tpr,label="data 1, auc="+str(auc))
pyplot.legend(loc=4)
pyplot.show()
pyplot.savefig('ROC')
```




```
<Figure size 432x288 with 0 Axes>
```

In [20]:
```python
#LogLoss
from sklearn.metrics import log_loss
logLoss=log_loss(y_test,y_pred)
print("Logloss: %.2f" % (logLoss))
```

```
Logloss: 2.34
```

In [21]:
```python
#F score
from sklearn.metrics import f1_score
f1 = f1_score(y_test, y_pred)
print('F1 score: %f' % f1)
```

```
F1 score: 0.950000
```

In [ ]: