Decision Tree Algorithm Project

January 20, 2020

```
                          Decision Tree Classification First Project
```

[1]:
```
# importing basic libraries
import numpy as np
import pandas as pd
from pandas import Series, DataFrame
from sklearn.model_selection import train_test_split
from sklearn import tree
from sklearn import metrics
# load dataset
diabetes = pd.read_csv("diabetes.csv")
diabetes.head()
```

[1]:
```
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0            6      148             72             35        0  33.6
1            1       85             66             29        0  26.6
2            8      183             64              0        0  23.3
3            1       89             66             23       94  28.1
4            0      137             40             35      168  43.1

   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50        1
1                     0.351   31        0
2                     0.672   32        1
3                     0.167   21        0
4                     2.288   33        1
```

[2]:
```
feat_colums = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',␣
 ↪'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age']
X = diabetes[feat_colums]
y = diabetes.Outcome
```

[3]:
```
#Divinding the dataset into training and test dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=1)
```

[4]:
```
#Creating Tree Object With Entropy Criterias
model = tree.DecisionTreeClassifier(criterion='entropy')
```

[5]:

```
#Decision Tree Classifier
model = model.fit(X_train,y_train)
```

[6]:

```
#Predict the response for test dataset
y_pred = model.predict(X_test)
```

[7]:

```
# Model Accuracy, how often is the classifier correct?
print("Accuracy Using Entropy Criterion:",metrics.accuracy_score(y_test,␣
 ↪y_pred))
```

```
Accuracy Using Entropy Criterion: 0.7077922077922078
```

[8]:

```
#Improving Model Accuracy By Trying Maximum Depth Of Node
model = tree.DecisionTreeClassifier(max_depth = 5)
model = model.fit(X_train,y_train)
y_pred = model.predict(X_test)
print("Accuracy Using Entropy Criterion:",metrics.accuracy_score(y_test,␣
 ↪y_pred))
metrics.accuracy_score(y_test, y_pred)
```

```
Accuracy Using Entropy Criterion: 0.7402597402597403
```

[8]:

```
0.7402597402597403
```

[9]:

```
#Improving Model Accuracy By Trying Maximum Depth Of Node
model = tree.DecisionTreeClassifier(max_depth = 5)
model = model.fit(X_train,y_train)
y_pred = model.predict(X_test)
print("Accuracy Using Entropy Criterion:",metrics.accuracy_score(y_test,␣
 ↪y_pred))
```

```
Accuracy Using Entropy Criterion: 0.7337662337662337
```

As can be seen that model has been optimised by trimming the depth of tree at 4

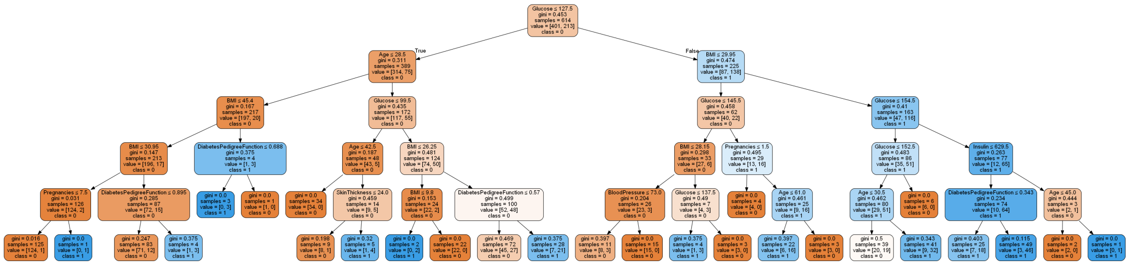
[10]:

```
#Plotting The Decision Tree
from sklearn.externals.six import StringIO
from IPython.display import Image
from sklearn.tree import export_graphviz
import pydotplus
import os
os.environ['PATH'] = os.environ['PATH']+';'+os.
 ↪environ['CONDA_PREFIX']+r"\Library\bin\graphviz"
dot_data = StringIO()
export_graphviz(model, out_file=dot_data,
                filled=True, rounded=True,
                special_characters=True, feature_names =␣
 ↪feat_colums,class_names=['0','1'])
```

```
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('diabetes.png')
Image(graph.create_png())
```

```
C:\Users\Trilo\Anaconda3\lib\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)
```

[10]:


Glucose ≤ 127.5 gini = 0.453 samples = 614 value = [401, 213] class = 0
True
Age ≤ 28.5 gini = 0.311 samples = 389 value = [314, 75] class = 0
False
BMI ≤ 29.95 gini = 0.474 samples = 225 value = [87, 138] class = 1
BMI ≤ 45.4 gini = 0.167 samples = 217 value = [197, 20] class = 0
Glucose ≤ 99.5 gini = 0.435 samples = 172 value = [117, 55] class = 0
Glucose ≤ 145.5 gini = 0.458 samples = 62 value = [40, 22] class = 0
Glucose ≤ 154.5 gini = 0.41 samples = 163 value = [47, 116] class = 1
BMI ≤ 30.95 gini = 0.147 samples = 213 value = [196, 17] class = 0
DiabetesPedigreeFunction ≤ 0.688 gini = 0.375 samples = 4 value = [1, 3] class = 1
Age ≤ 42.5 gini = 0.187 samples = 48 value = [43, 5] class = 0
BMI ≤ 26.25 gini = 0.481 samples = 124 value = [74, 50] class = 0
BMI ≤ 28.15 gini = 0.298 samples = 33 value = [27, 6] class = 0
Pregnancies ≤ 1.5 gini = 0.495 samples = 29 value = [13, 16] class = 1
Glucose ≤ 152.5 gini = 0.483 samples = 86 value = [35, 51] class = 1
Insulin ≤ 629.5 gini = 0.263 samples = 77 value = [12, 65] class = 1
Pregnancies ≤ 7.5 gini = 0.031 samples = 126 value = [124, 2] class = 0
DiabetesPedigreeFunction ≤ 0.895 gini = 0.285 samples = 87 value = [72, 15] class = 0
gini = 0.0 samples = 3 value = [0, 3] class = 1
gini = 0.0 samples = 1 value = [1, 0] class = 0
gini = 0.0 samples = 34 value = [34, 0] class = 0
SkinThickness ≤ 24.0 gini = 0.459 samples = 14 value = [9, 5] class = 0
BMI ≤ 9.8 gini = 0.153 samples = 24 value = [22, 2] class = 0
DiabetesPedigreeFunction ≤ 0.57 gini = 0.499 samples = 100 value = [52, 48] class = 0
BloodPressure ≤ 73.0 gini = 0.204 samples = 26 value = [23, 3] class = 0
Glucose ≤ 137.5 gini = 0.49 samples = 7 value = [4, 3] class = 0
gini = 0.0 samples = 4 value = [4, 0] class = 0
Age ≤ 61.0 gini = 0.461 samples = 25 value = [9, 16] class = 1
Age ≤ 30.5 gini = 0.462 samples = 80 value = [29, 51] class = 1
gini = 0.0 samples = 6 value = [6, 0] class = 0
DiabetesPedigreeFunction ≤ 0.343 gini = 0.234 samples = 74 value = [10, 64] class = 1
Age ≤ 45.0 gini = 0.444 samples = 3 value = [2, 1] class = 0
gini = 0.016 samples = 125 value = [124, 1] class = 0
gini = 0.0 samples = 1 value = [0, 1] class = 1
gini = 0.247 samples = 83 value = [71, 12] class = 0
gini = 0.375 samples = 4 value = [1, 3] class = 1
gini = 0.198 samples = 9 value = [8, 1] class = 0
gini = 0.32 samples = 5 value = [1, 4] class = 1
gini = 0.0 samples = 2 value = [0, 2] class = 1
gini = 0.0 samples = 22 value = [22, 0] class = 0
gini = 0.469 samples = 72 value = [45, 27] class = 0
gini = 0.375 samples = 28 value = [7, 21] class = 1
gini = 0.397 samples = 11 value = [8, 3] class = 0
gini = 0.0 samples = 15 value = [15, 0] class = 0
gini = 0.375 samples = 4 value = [1, 3] class = 1
gini = 0.0 samples = 3 value = [3, 0] class = 0
gini = 0.397 samples = 22 value = [6, 16] class = 1
gini = 0.0 samples = 3 value = [3, 0] class = 0
gini = 0.5 samples = 39 value = [20, 19] class = 0
gini = 0.343 samples = 41 value = [9, 32] class = 1
gini = 0.403 samples = 25 value = [7, 18] class = 1
gini = 0.115 samples = 49 value = [3, 46] class = 1
gini = 0.0 samples = 2 value = [2, 0] class = 0
gini = 0.0 samples = 1 value = [0, 1] class = 1