Practical 4:

Implement Decision Tree

Submitted By:

- Aman Kumar
- 2019UCO1634
- CoE - 2

```
# Load libraries
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

```
# load dataset
pima = pd.read_csv("diabetes.csv")
```

```
pima.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	6	148	72	35	0	33.6	0.627	50
1	1	85	66	29	0	26.6	0.351	31
2	8	183	64	0	0	23.3	0.672	32
3	1	89	66	23	94	28.1	0.167	21
4	0	137	40	35	168	43.1	2.288	33

```
#split dataset in features and target variable
feature_cols = ['Pregnancies', 'Insulin', 'BMI', 'Age','Glucose','BloodPressure','DiabetesPedigreeFunction']
X = pima[feature_cols]
y = pima['Outcome']
```

```
# Split dataset into training set and test set
# 70 - 30 split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)
```

```
# create classifier and train it
clf = DecisionTreeClassifier()
clf = clf.fit(X_train,y_train)

# predict response
y_pred = clf.predict(X_test)
```

```
# Model Accuracy
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.6536796536796536
```

```
# Improving the performance using different DT algorithms and limiting depth

clf2 = DecisionTreeClassifier(criterion="entropy", max_depth=3)
clf2 = clf2.fit(X_train,y_train)
y_pred2 = clf2.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred2))
```

```
Accuracy: 0.7705627705627706
```

```
clf3 = DecisionTreeClassifier(criterion="entropy", max_depth=4)
clf3 = clf3.fit(X_train,y_train)
y_pred3 = clf3.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred3))
```

```
Accuracy: 0.7878787878787878
```

```
clf4 = DecisionTreeClassifier(criterion="entropy", max_depth=5)
clf4 = clf4.fit(X_train,y_train)
y_pred4 = clf4.predict(X_test)
```

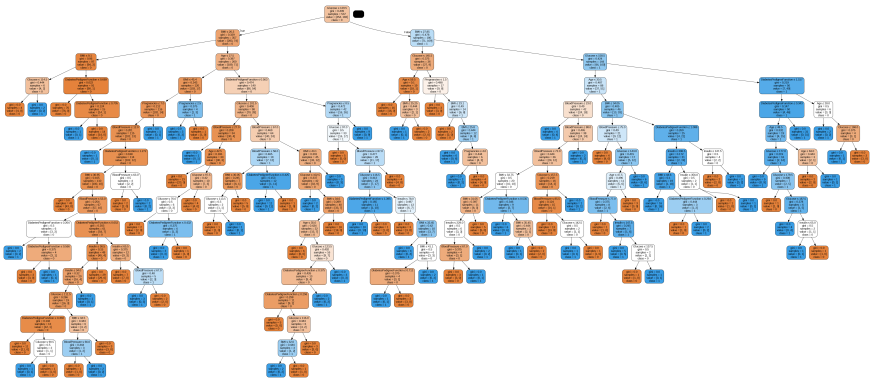
```
print("Accuracy:",metrics.accuracy_score(y_test, y_pred4))
```

```
Accuracy: 0.7835497835497836
```

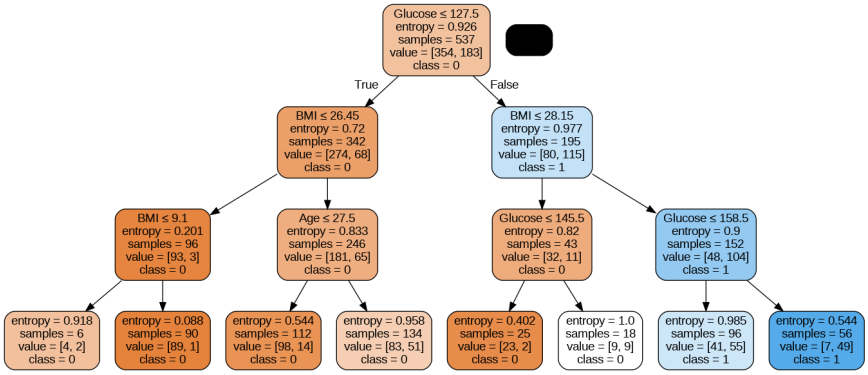
+ Code + Text

Visualization

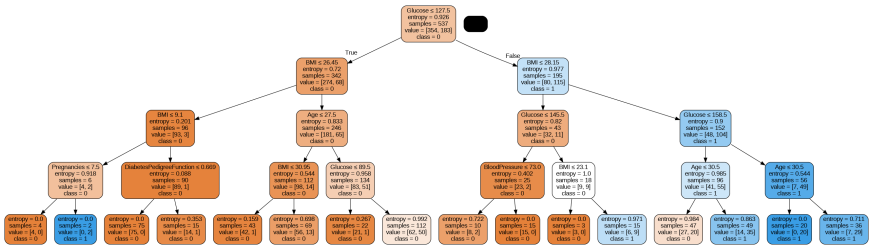
```
# displaying the decision tree

from sklearn.tree import export_graphviz
from six import StringIO
from IPython.display import Image
import pydotplus

dot_data = StringIO()
export_graphviz(clf, out_file=dot_data,
                filled=True, rounded=True,
                special_characters=True,feature_names = feature_cols,class_names=['0','1'])
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('diabetes.png')
Image(graph.create_png())
```



```
dot_data = StringIO()
export_graphviz(clf2, out_file=dot_data,
                filled=True, rounded=True,
                special_characters=True,feature_names = feature_cols,class_names=['0','1'])
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('diabetes.png')
Image(graph.create_png())
```




```
dot_data = StringIO()
export_graphviz(clf3, out_file=dot_data,
                filled=True, rounded=True,
                special_characters=True,feature_names = feature_cols,class_names=['0','1'])
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('diabetes.png')
Image(graph.create_png())
```



Colab paid products - Cancel contracts here