In [1]:
```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
from sklearn.model_selection import train_test_split
```

In [2]:
```
pima = pd.read_csv("diabetes.csv")

pima = pima.drop(0, axis=0)
pima.head()
```

Out[2]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
1	1	85	66	29	0	26.6	0.351
2	8	183	64	0	0	23.3	0.672
3	1	89	66	23	94	28.1	0.167
4	0	137	40	35	168	43.1	2.288
5	5	116	74	0	0	25.6	0.201

In [3]:

```

# Assuming 'pima' is your DataFrame
column_mapping = {
    'Pregnancies': 'pregnant',
    'Glucose': 'glucose',
    'BloodPressure': 'bp',
    'SkinThickness': 'skin',
    'Insulin': 'insulin',
    'BMI': 'bmi',
    'DiabetesPedigreeFunction': 'pedigree',
    'Age': 'age',
    'Outcome': 'Label'
}

pima.rename(columns=column_mapping, inplace=True)

# Display the DataFrame with highlighted column names using Markdown
#display(pima.style.set_table_styles([{
 #    'selector': 'th',
  #   'props': [('background-color', '#7CAE00'), ('color', 'white')]
#}]))
pima.head()
```

Out[3]:

	pregnant	glucose	bp	skin	insulin	bmi	pedigree	age	Label
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
5	5	116	74	0	0	25.6	0.201	30	0

In [4]:

```
#split dataset in features and target variable

feature_cols = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedigre

X = pima[feature_cols]  # Features
y = pima['Label']
```

In [5]:

```
# Assuming X and y are your features and target variable
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, randor
```

In [6]:

```

# Create Decision Tree classifier object
clf = DecisionTreeClassifier()

# Train Decision Tree Classifier
clf = clf.fit(X_train, y_train)

# Test Model
y_pred = clf.predict(X_test)
```

In [7]:

```
# Model Accuracy, how often is the classifier correct?

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

```
Accuracy: 0.7056277056277056
```

In [ ]:

In [8]:

```
!pip install graphviz
```

```
Requirement already satisfied: graphviz in c:\users\anas\anaconda3\lib\site-p
ackages (0.20.1)
```

In [64]:

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[64], line 11
      7 feature_names=X_train.columns, # Replace with your feature names clas
s_names=[str(x) for x in clf.classes_], # Convert class names to stri filled=
True, rounded=True, special_characters=True)
      9 #Create and display the graph graph = graphviz.Source(dot_data) grap
h.render("decision_tree") # Saves the visualization as a file (e.g., "decisio
n_tree.pdf")
---> 11 graph.view("decision_tree")

NameError: name 'graph' is not defined
```

In [8]:

```
from sklearn.tree import export_graphviz

import graphviz

#Export the decision tree to DOT format
dot_data = export_graphviz(clf , out_file=None,
                           feature_names=X_train.columns, # Replace with your fea
                           class_names=[str(x) for x in clf.classes_], # Convert
                           filled=True, rounded=True, special_characters=True)

#Create and display the graph
graph = graphviz.Source(dot_data)
graph.render("decision_tree") # Saves the visualization as a file (e.g., "decis

graph.view("decision_tree") # Opens the visualization using the default viewer
```

Out[8]:

```
'decision_tree.pdf'
```

In [11]:

```
import graphviz
print(graphviz.__version__)
```

```
0.20.1
```

In [11]:

```
#Optimizing Decision Tree Performance

#Create Decision Tree classifer object
clf = DecisionTreeClassifier(criterion="entropy", max_depth=3)

#Train Decision Tree Classifer
clf = clf.fit(X_train,y_train)

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

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

```
Accuracy: 0.6883116883116883
```

In [ ]: