

Lab Assignment 6

1. Create a Decision Tree Model and Perform the training and Testing(70:30) using 'Entropy' as tree split criteria. Display the confusion matrix and accuracy of the model both on train and test data. Display the tree as well. Use Diabetes_null.csv as a data set for the above model creation.

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.linear_model import LogisticRegression
```

In [2]:

```
df=pd.read_csv("C://Users/91947/OneDrive/Desktop/ML LAB/diabetes_null.csv")
```

In [3]:

```
df
```

Out[3]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFun
0	6	148.0	72.0	35.0	NaN	33.6	
1	1	85.0	66.0	29.0	NaN	26.6	
2	8	183.0	64.0	NaN	NaN	23.3	
3	1	89.0	66.0	23.0	94.0	28.1	
4	0	137.0	4.0	35.0	168.0	43.1	
...	
763	10	11.0	76.0	48.0	18.0	32.9	
764	2	122.0	7.0	27.0	NaN	36.8	
765	5	121.0	72.0	23.0	112.0	26.2	
766	1	126.0	6.0	NaN	NaN	3.1	
767	1	93.0	7.0	31.0	NaN	3.4	

768 rows × 9 columns

In [4]:

```
df.fillna(df.mean(), inplace=True)
```

In [5]:

df

Out[5]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigree
0	6	148.0	72.0	35.000000	105.659898	33.6	
1	1	85.0	66.0	29.000000	105.659898	26.6	
2	8	183.0	64.0	25.876155	105.659898	23.3	
3	1	89.0	66.0	23.000000	94.000000	28.1	
4	0	137.0	4.0	35.000000	168.000000	43.1	
...	
763	10	11.0	76.0	48.000000	18.000000	32.9	
764	2	122.0	7.0	27.000000	105.659898	36.8	
765	5	121.0	72.0	23.000000	112.000000	26.2	
766	1	126.0	6.0	25.876155	105.659898	3.1	
767	1	93.0	7.0	31.000000	105.659898	3.4	

768 rows × 9 columns

In [6]:

```
x1 = df.drop('Outcome', axis=1)
y1 = df['Outcome']
```

In [7]:

```
x1_train, x1_test, y1_train, y1_test = train_test_split(x1, y1, test_size=0.3, random_state=42)
```

In [8]:

```
from sklearn.tree import DecisionTreeClassifier
tree_model=DecisionTreeClassifier(criterion="entropy", random_state=42)
tree_model.fit(x1_train,y1_train)
```

Out[8]:

▼

DecisionTreeClassifier

DecisionTreeClassifier(criterion='entropy', random_state=42)

In [9]:

```
y1_pred = tree_model.predict(x1_test)
y1_pred_train = tree_model.predict(x1_train)
```

In [10]:

```
# Calculate the accuracy of the model
accuracy_null = accuracy_score(y1_test, y1_pred)
accuracy_null_train = accuracy_score(y1_train, y1_pred_train)

conf_matrix1 = confusion_matrix(y1_test, y1_pred)
conf_matrix1_train = confusion_matrix(y1_train, y1_pred_train)
```

In [11]:

```
print(f"Accuracy : {accuracy_null:.6f}")
print(f"Accuracy of training set : {accuracy_null_train:.6f}")

print("Confusion Matrix test:")
print(conf_matrix1)

print("Confusion Matrix training:")
print(conf_matrix1_train)
```

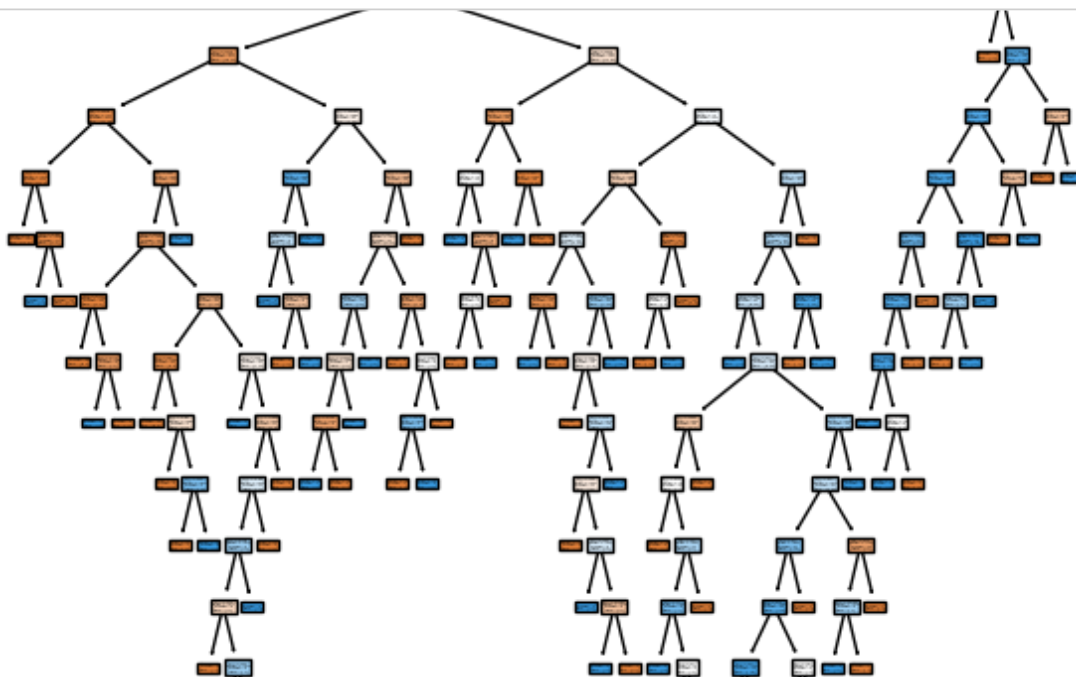
```
Accuracy : 0.675325
Accuracy of training set : 1.000000
Confusion Matrix test:
[[112  39]
 [ 36  44]]
Confusion Matrix training:
[[349   0]
 [   0 188]]
```

In [12]:

```
import matplotlib.pyplot as plt
from sklearn import tree
```

In [13]:

```
plt.figure(figsize=(7,6))
tree.plot_tree(tree_model,filled=True)
```



2. Create another Decision Tree model using Diabetes_zero.csv. Also display the confusion matrix and accuracy of this model.

In [14]:

```
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.linear_model import LogisticRegression
df = pd.read_csv("C://Users/91947/OneDrive/Desktop/ML LAB/diabetes_zero.csv")
# Split the data into features (X) and target (y)
x = df.drop('Outcome', axis=1)
y = df['Outcome']

# Split the data into a training set and a testing set
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=42)

# Decision Model
from sklearn.tree import DecisionTreeClassifier
tree_model = DecisionTreeClassifier(criterion="entropy")
tree_model.fit(x_train, y_train)

# Make predictions on the test set
y_pred = tree_model.predict(x_test)
y_pred_train = tree_model.predict(x_train)

# Calculate the accuracy of the model
accuracy_zero = accuracy_score(y_test, y_pred)
accuracy_zero_train = accuracy_score(y_train, y_pred_train)
conf_matrix = confusion_matrix(y_test, y_pred)
conf_matrix_train = confusion_matrix(y_train, y_pred_train)
print(f"Accuracy : {accuracy_zero:.6f}")
print(f"Accuracy of training set : {accuracy_zero_train:.6f}")
print("Confusion Matrix:")
print(conf_matrix)
print(conf_matrix_train)
```

Accuracy : 0.727273

Accuracy of training set : 1.000000

Confusion Matrix:

```
[[118  33]
 [ 30  50]]
[[349   0]
 [  0 188]]
```

In [15]:

```
tree_model.fit(x_train, y_train)
```

Out[15]:

```
DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy')
```

In [16]:

```
plt.figure(figsize=(7,6))
```

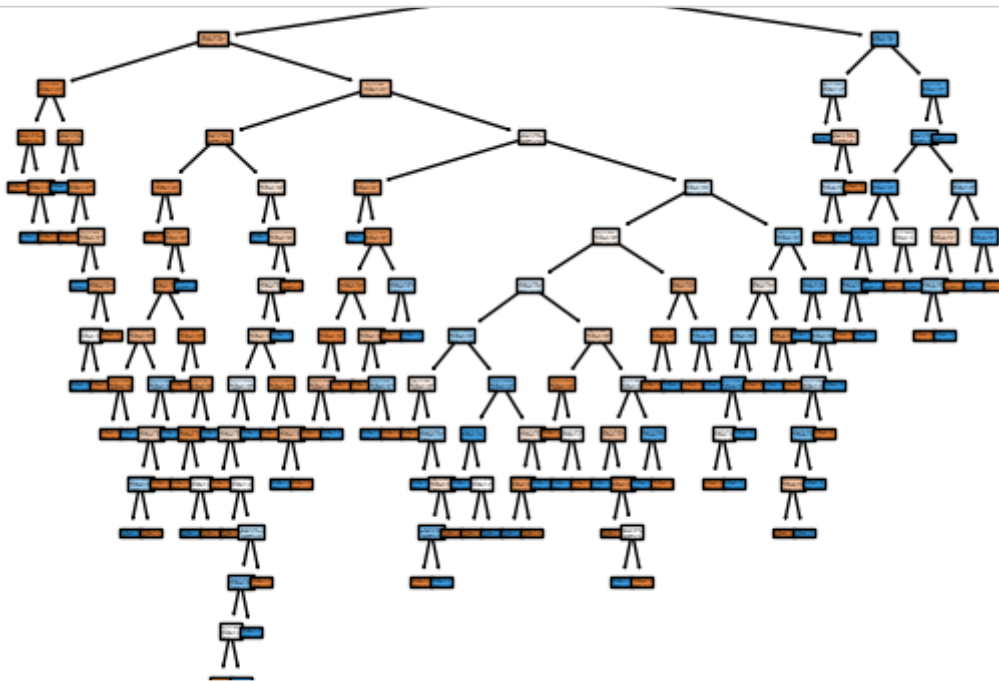
Out[16]:

<Figure size 700x600 with 0 Axes>

<Figure size 700x600 with 0 Axes>

In [17]:

```
tree.plot_tree(tree_model,filled=True)
```



3. Compare the accuracy of the above two models using Bar Chart.

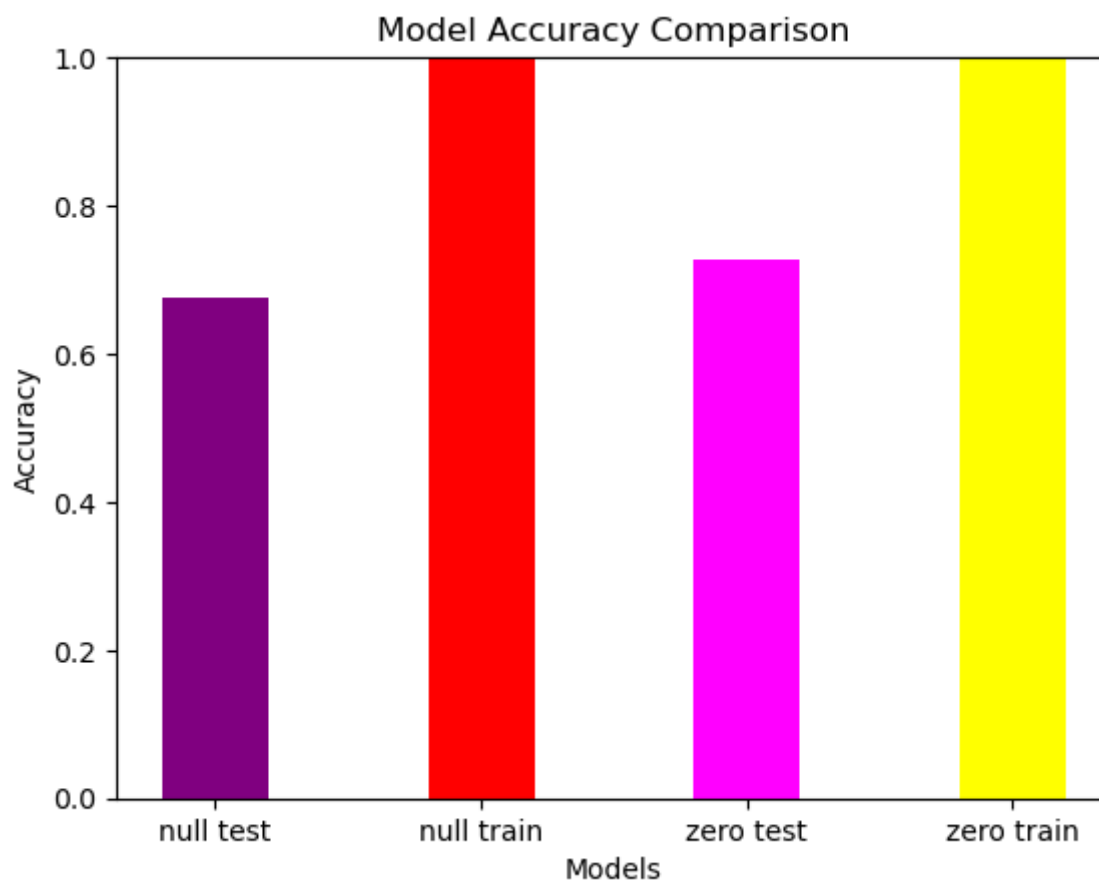
In [23]:

```
import matplotlib.pyplot as plt

accuracies = [accuracy_null, accuracy_null_train, accuracy_zero, accuracy_zero_train]
model_names = ['null test', 'null train', 'zero test', 'zero train']

plt.bar(model_names, accuracies, color=['purple', 'red', 'magenta', 'yellow'], width=0.4)
plt.xlabel('Models')
plt.ylabel('Accuracy')
plt.title('Model Accuracy Comparison')
plt.ylim([0, 1])

plt.show()
```



In []:

In []: