In [ ]:
```
#Name : Krishna Ranode
#Roll : 158
#PRn : 0124UITM2013

# Load libraries
import pandas as pd
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classif
from sklearn.model_selection import train_test_split # Import train_test_split
from sklearn import metrics #Import scikit-learn metrics module for accuracy c
```

In [36]:
```
col_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedigree'
# load dataset
pima = pd.read_csv("diabetes.csv", header=None, names=col_names)
```

In [37]:
```
pima.head()
```

Out[37]:

	pregnant	glucose	bp	skin	insulin	bmi	p
0	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigree
1	6	148	72	35	0	33.6	
2	1	85	66	29	0	26.6	
3	8	183	64	0	0	23.3	
4	1	89	66	23	94	28.1	

In [38]:
```
#split dataset in features and target variable
feature_cols = ['pregnant', 'insulin', 'bmi', 'age','glucose','bp','pedigree']
X = pima[feature_cols] # Features
y = pima.label # Target variable
```

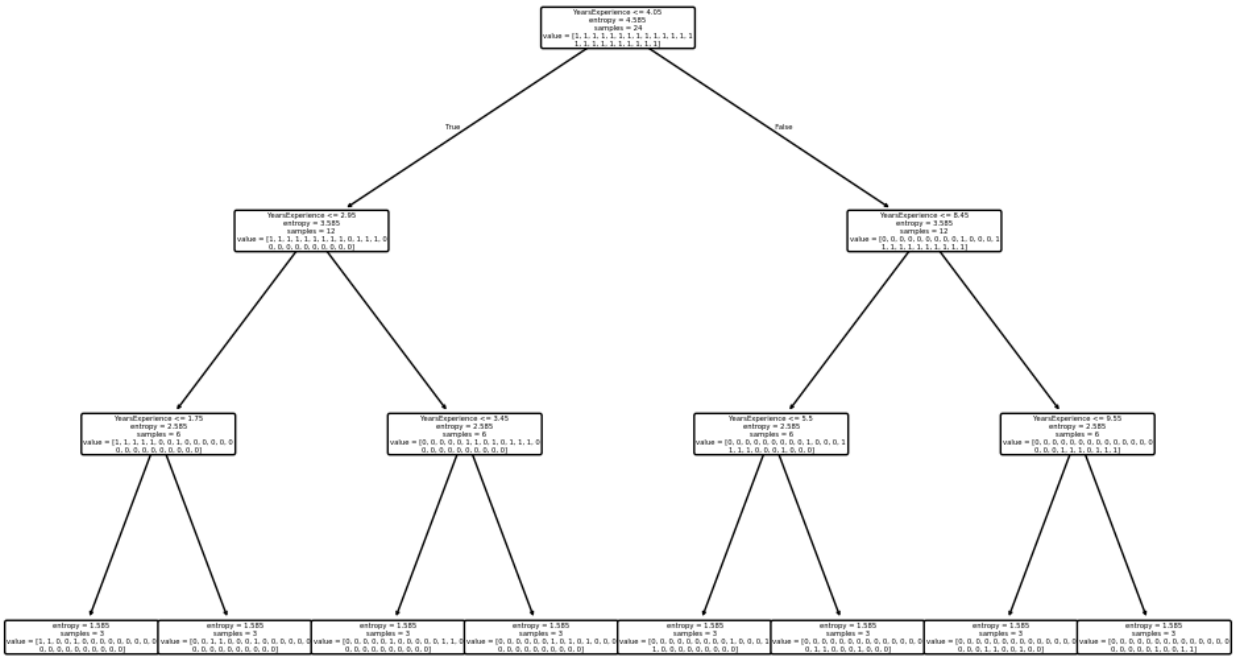
In [39]:
```
# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, randc
```

In [42]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn import metrics

# Load dataset
data = pd.read_csv("Salary_Data.csv")

# Features and target
X = data[['YearsExperience']]   # independent variable
y = data['Salary']              # dependent variable

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, randc

# Create Decision Tree Regressor
clf = DecisionTreeRegressor(random_state=1)
```

```
clf.fit(X_train, y_train)

# Predictions
y_pred = clf.predict(X_test)

# Evaluation
print("R2 Score:", metrics.r2_score(y_test, y_pred))
print("Mean Absolute Error:", metrics.mean_absolute_error(y_test, y_pred))
accuracy = metrics.r2_score(y_test, y_pred) * 100
print("Accuracy (R² %):", accuracy)

#print("Root Mean Squared Error:", metrics.mean_squared_error(y_test, y_pred,
```

```
R2 Score: 0.5132697810376604
Mean Absolute Error: 8985.416666666666
Accuracy (R² %): 51.32697810376604
```

In [47]:
```
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt

plt.figure(figsize=(12,8))
plot_tree(clf, feature_names=['YearsExperience'], filled=True, rounded=True)
plt.show()
```



In [51]:
```
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt

plt.figure(figsize=(12,8))
```

```
plot_tree(clf, feature_names=['YearsExperience'], filled=True, rounded=True)
plt.show()
```