

## Decision Tree

### Objective

create a synthetic dataset where the features are Age, Income and the Target is whether the person purchased a car or not

```
In [2]: # Import necessary Libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt
```

```
In [3]: # Create the dataset
data = {
    'Age': [25, 45, 35, 50, 23, 40, 60, 48, 33, 55],
    'Income': ['Low', 'High', 'Medium', 'High', 'Low', 'Medium', 'High', 'High', 'Medium', 'High'],
    'Purchased': [0, 1, 1, 1, 0, 1, 1, 1, 1, 1] # 0 means 'No', 1 means 'Yes'
}
```

```
In [6]: df = pd.DataFrame(data)
```

```
In [7]: df
```

Out[7]:

	Age	Income	Purchased
0	25	Low	0
1	45	High	1
2	35	Medium	1
3	50	High	1
4	23	Low	0
5	40	Medium	1
6	60	High	1
7	48	High	1
8	33	Medium	1
9	55	High	1

```
In [8]: # Convert categorical feature 'Income' into numerical values
df['Income'] = df['Income'].map({'Low': 1, 'Medium': 2, 'High': 3})
```

```
In [9]: # Features (Age and Income) and target (Purchased)
X = df[['Age', 'Income']] # Features
y = df['Purchased'] # Target
```

```
In [11]: #Split the dataset into training and testing sets
X_train,X_test,y_train,y_test = train_test_split(X,y, test_size =0.2,random_
```

```
In [12]: #create a decisiontree classifier with max depth 3
clf = DecisionTreeClassifier(max_depth = 3,random_state = 42)
```

```
In [13]: #Train the classifier with training data
clf.fit(X_train,y_train)
```

```
Out[13]: DecisionTreeClassifier(max_depth=3, random_state=42)
```

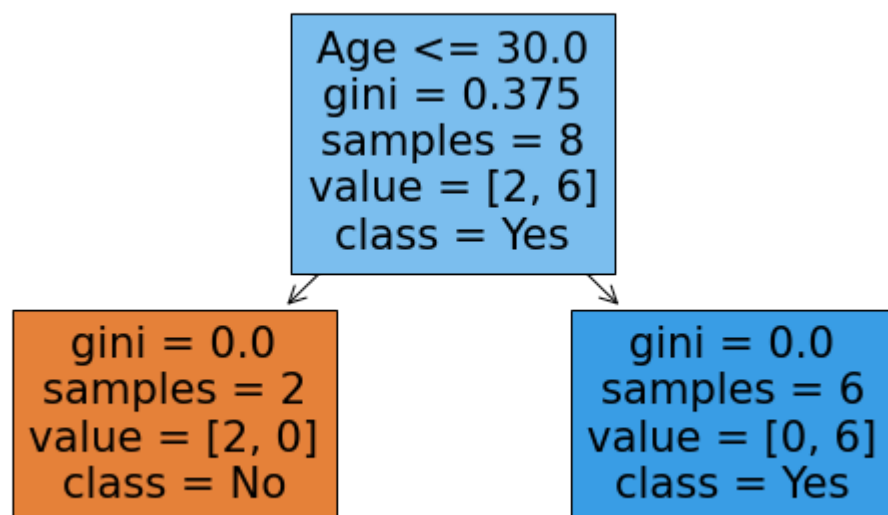
```
In [14]: #predict on the test set
y_pred = clf.predict(X_test)
```

```
In [21]: #evaluate the models perfomance
Accuracy = metrics.accuracy_score(y_test,y_pred)
print("Accuracy :",Accuracy)
```

```
Accuracy : 1.0
```

```
In [24]: plt.figure(figsize = [10,5])
plot_tree(clf,feature_names = ['Age','Income'],
          class_names = ['No','Yes'],filled = True)
```

```
Out[24]: [Text(0.5, 0.75, 'Age <= 30.0\ngini = 0.375\nsamples = 8\nvalue = [2, 6]\n          class = Yes'),
          Text(0.25, 0.25, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]\n          class = No'),
          Text(0.75, 0.25, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]\n          class = Yes')]
```



```
In [ ]:
```

