Problem:

We will use this dataset to try and predict gas consumptions (in millions of gallons) in 48 US states based upon gas tax (in cents), per capita income (dollars), paved highways (in miles) and the proportion of population with a drivers license.

Importing Libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

Importing Dataset

In [2]:
```
dataset = pd.read_csv('petrol_consumption.csv')
```

In [4]:
```
dataset.head()
```

Out[4]:

	Petrol_tax	Average_income	Paved_Highways	Population_Driver_licence(%)	Petrol_Consumption
0	9.0	3571	1976	0.525	541
1	9.0	4092	1250	0.572	524
2	9.0	3865	1586	0.580	561
3	7.5	4870	2351	0.529	414
4	8.0	4399	431	0.544	410

Dataset Analysis

In [5]:
```
dataset.describe()
```

Out[5]:

	Petrol_tax	Average_income	Paved_Highways	Population_Driver_licence(%)	Petrol_Consumption
count	48.000000	48.000000	48.000000	48.000000	48.000000
mean	7.668333	4241.833333	5565.416667	0.570333	576.770833
std	0.950770	573.623768	3491.507166	0.055470	111.885816
min	5.000000	3063.000000	431.000000	0.451000	344.000000
25%	7.000000	3739.000000	3110.250000	0.529750	509.500000
50%	7.500000	4298.000000	4735.500000	0.564500	568.500000
75%	8.125000	4578.750000	7156.000000	0.595250	632.750000
max	10.000000	5342.000000	17782.000000	0.724000	968.000000

In [6]:
```
dataset.isnull().sum()
```

Out[6]:
```
Petrol_tax                      0
Average_income                  0
Paved_Highways                  0
Population_Driver_licence(%)    0
Petrol_Consumption              0
dtype: int64
```

Preparing Data

In [7]:
```
X = dataset.drop('Petrol_Consumption', axis=1)
y = dataset['Petrol_Consumption']
```

In [8]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=
```

Training and Making Predictions

In [9]:
```
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor()
regressor.fit(X_train, y_train)
```

Out[9]:
```
DecisionTreeRegressor()
```

In [10]:
```
y_pred = regressor.predict(X_test)
```

Now let's compare some of our predicted values with the actual values and see how accurate we were:

In [11]:
```
df=pd.DataFrame({'Actual':y_test, 'Predicted':y_pred})
df
```

Out[11]:

	Actual	Predicted
29	534	541.0
4	410	414.0
26	577	574.0
30	571	554.0
32	577	574.0
37	704	554.0
34	487	628.0
40	587	524.0
7	467	414.0
10	580	498.0

Evaluating the Algorithm

To evaluate performance of the regression algorithm, the commonly used metrics are mean absolute error, mean squared error, and root mean squared error.

In [12]:
```
from sklearn import metrics
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```
```
Mean Absolute Error: 52.3
Mean Squared Error: 5625.5
Root Mean Squared Error: 75.00333325926255
```

The mean absolute error for our algorithm is 52.3, which is less than10% [(MBE/Mean*100, 52.3l/576.770*100)~9.07%] percent of the mean of all the values in the 'Petrol_Consumption' column. This means that our algorithm did a fine prediction job.

In [ ]:
```
```