Import Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Load the dataset

```
dataset = pd.read_csv('/content/IceCreamData.csv')
X = dataset['Temperature'].values
y = dataset['Revenue'].values
dataset.head(5)
```

	Temperature	Revenue
0	24.566884	534.799028
1	26.005191	625.190122
2	27.790554	660.632289
3	20.595335	487.706960
4	11.503498	316.240194

Next steps: View recommended plots

Splitting the dataset into the Training set and Test set

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.05)
```

Fitting the decision tree

```
# Fitting Decision Tree Regression to the dataset
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor()
```

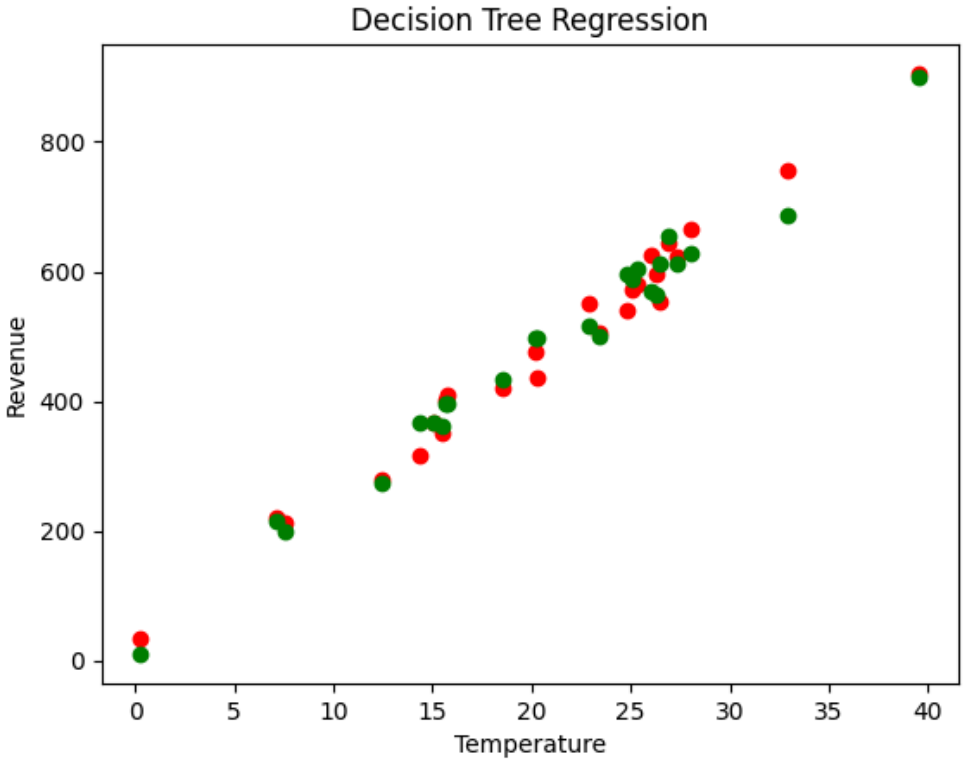
```
regressor.fit(X_train.reshape(-1,1), y_train.reshape(-1,1))
y_pred = regressor.predict(X_test.reshape(-1,1))
y_pred
df = pd.DataFrame({'Real Values':y_test.reshape(-1), 'Predicted Values':y_pred.reshape(-1)})
df
```

	Real Values	Predicted Values
0	212.483559	198.121563
1	540.977511	594.804871
2	625.190122	570.577875
3	420.966453	432.819795
4	581.262016	604.626673
5	550.055216	516.548601
6	221.400252	216.183462
7	350.629036	362.515216
8	506.432135	501.345330
9	665.672676	628.453211
10	437.251993	498.252146
11	402.455320	396.935648
12	755.818399	685.654655
13	905.477604	898.805423
14	554.742974	612.243721
15	596.889105	563.381633
16	644.488633	654.197406
17	475.538209	498.252146
18	409.493848	396.935648
19	623.248701	612.803770
20	366.247714	367.052376
21	279.866148	274.065619
22	571.434257	587.221246
23	315.646581	367.940744
24	32.546619	10.000000

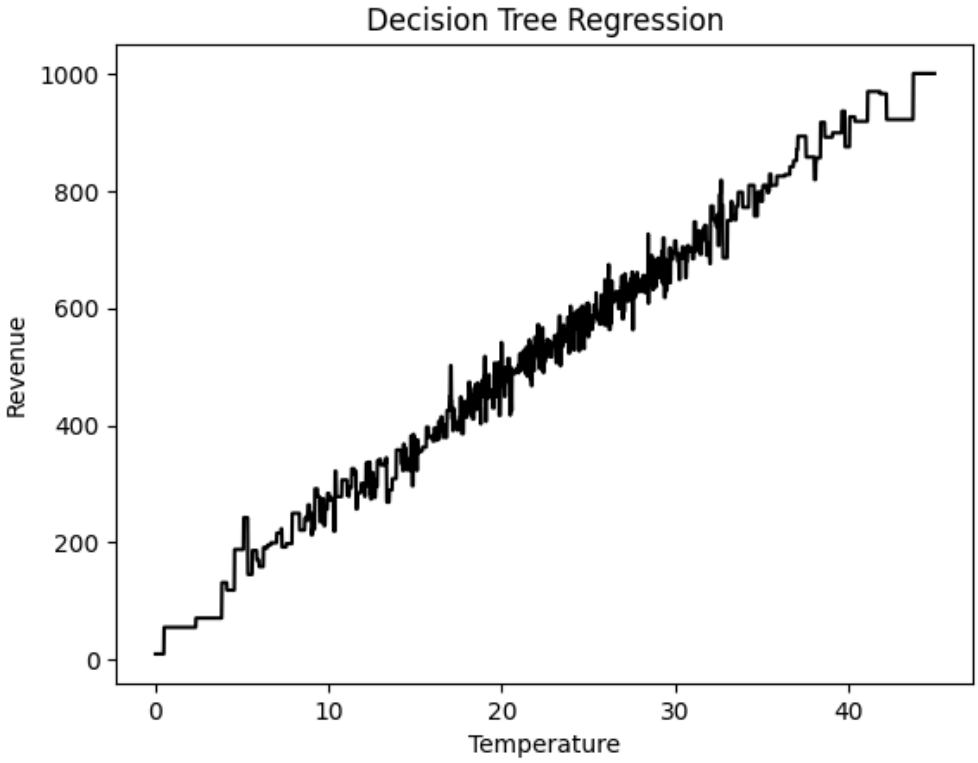
Next steps: View recommended plots

Visualization Decision Tree

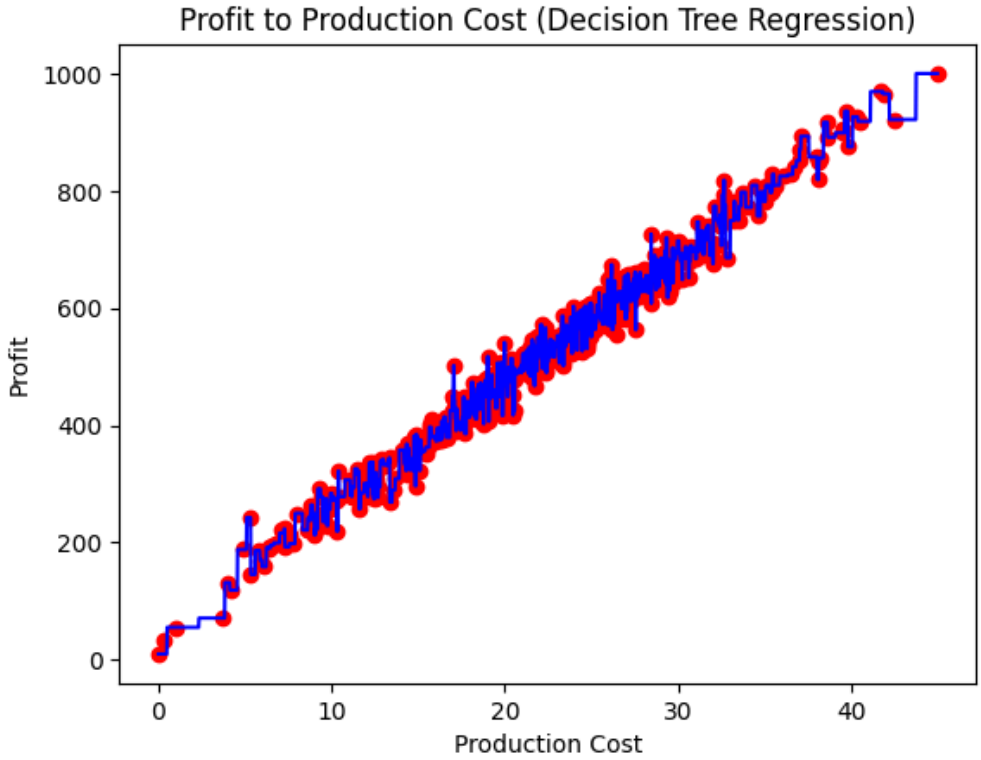
```
# Visualising the Decision Tree Regression Results
X_grid = np.arange(min(X), max(X), 0.01)
X_grid = X_grid.reshape((len(X_grid), 1))
plt.scatter(X_test, y_test, color = 'red')
plt.scatter(X_test, y_pred, color = 'green')
plt.title('Decision Tree Regression')
plt.xlabel('Temperature')
plt.ylabel('Revenue')
plt.show()
```




```
plt.plot(X_grid, regressor.predict(X_grid), color = 'black')
plt.title('Decision Tree Regression')
plt.xlabel('Temperature')
plt.ylabel('Revenue')
plt.show()
```




```
1
2   # arange for creating a range of values
3   # from min value of X to max value of X
4   # with a difference of 0.01 between two
5   # consecutive values
6   X_grid = np.arange(min(X), max(X), 0.01)
7
8   # reshape for reshaping the data into
9   # a len(X_grid)*1 array, i.e. to make
10  # a column out of the X_grid values
11  X_grid = X_grid.reshape((len(X_grid), 1))
12
13  # scatter plot for original data
14  plt.scatter(X, y, color = 'red')
15
16  # plot predicted data
17  plt.plot(X_grid, regressor.predict(X_grid), color = 'blue')
18  # specify title
19  plt.title('Profit to Production Cost (Decision Tree Regression)')
20
21  # specify X axis label
22  plt.xlabel('Production Cost')
23
24  # specify Y axis label
25  plt.ylabel('Profit')
26
27  # show the plot
28  plt.show()
```



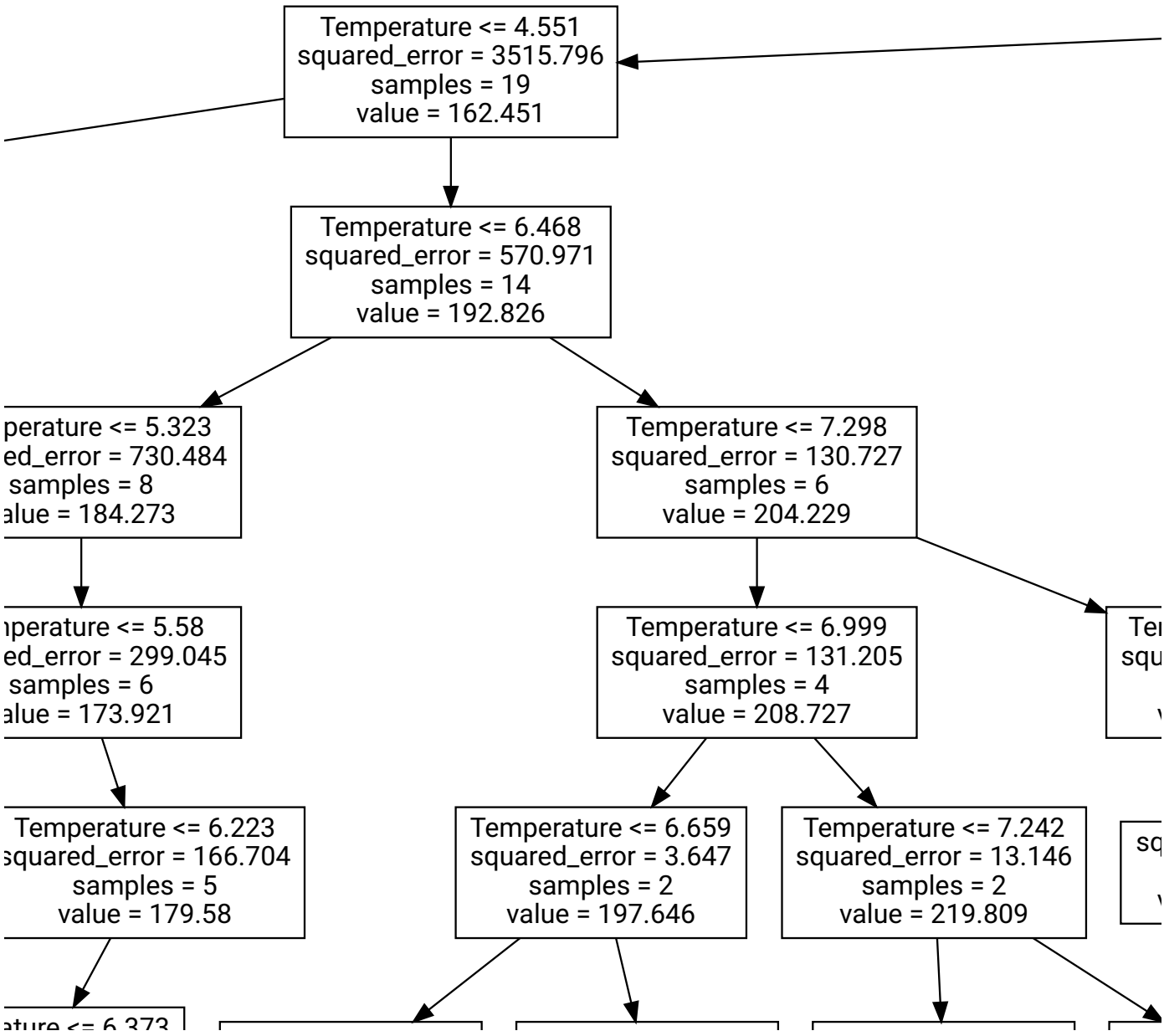

```
!pip install graphviz
```

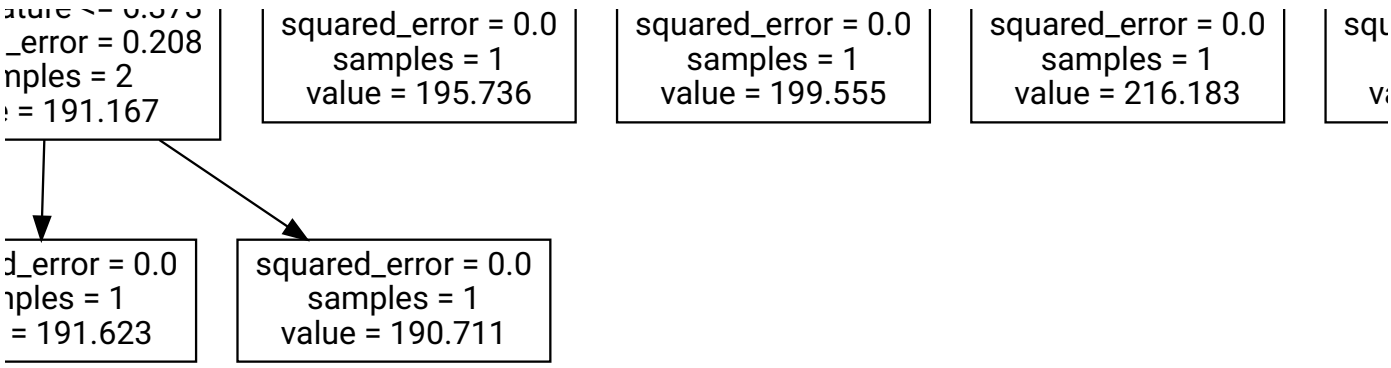
```
Requirement already satisfied: graphviz in /usr/local/lib/python3.10/dist-packages (0.20.3)
```

```
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier, export_graphviz
import graphviz

# Load the Iris dataset
iris = load_iris()
X = iris.data
y = iris.target

# Train a decision tree classifier
clf = DecisionTreeClassifier()
clf.fit(X, y)

# Export the decision tree as a DOT file
dot_data = export_graphviz(clf, out_file=None,
                           feature_names=iris.feature_names,
                           class_names=iris.target_names,
                           filled=True, rounded=True,
                           special_characters=True)

# Visualize the decision tree
graph = graphviz.Source(dot_data)
graph.render('decision_tree', format='jpg', cleanup=True)
```

```
'decision_tree.jpg'
```

```
# Export the decision tree as a DOT file
dot_data = export_graphviz(regressor, out_file= None,
                           feature_names=['Temperature'])

# Visualize the decision tree within the notebook
graph = graphviz.Source(dot_data)
graph
```


Temperature <= 4.551
squared_error = 3515.796
samples = 19
value = 162.451
Temperature <= 6.468
squared_error = 570.971
samples = 14
value = 192.826
perature <= 5.323
ed_error = 730.484
samples = 8
alue = 184.273
Temperature <= 7.298
squared_error = 130.727
samples = 6
value = 204.229
nperature <= 5.58
ed_error = 299.045
samples = 6
alue = 173.921
Temperature <= 6.999
squared_error = 131.205
samples = 4
value = 208.727
Temperature <= 6.223
squared_error = 166.704
samples = 5
value = 179.58
Temperature <= 6.659
squared_error = 3.647
samples = 2
value = 197.646
Temperature <= 7.242
squared_error = 13.146
samples = 2
value = 219.809


_error = 0.208
nples = 2
= 191.167
squared_error = 0.0
samples = 1
value = 195.736
squared_error = 0.0
samples = 1
value = 199.555
squared_error = 0.0
samples = 1
value = 216.183
d_error = 0.0
nples = 1
= 191.623
squared_error = 0.0
samples = 1
value = 190.711