```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import plot_tree


dt= pd.read_csv('/content/manufacturing_defect_dataset.csv')


d=DecisionTreeClassifier()
dt.head()
```

| ProductionVolume | ProductionCost | SupplierQuality | DeliveryDelay | DefectRate | QualityScor |
|---|---|---|---|---|---|
| 202 | 13175.403783 | 86.648534 | 1 | 3.121492 | 63.4634 |
| 535 | 19770.046093 | 86.310664 | 4 | 0.819531 | 83.6978 |
| 960 | 19060.820997 | 82.132472 | 0 | 4.514504 | 90.3505 |
| 370 | 5647.606037 | 87.335966 | 5 | 0.638524 | 67.6286 |
| 206 | 7472.222236 | 81.989893 | 3 | 3.867784 | 82.7283 |

Next steps: Generate code with dt View recommended plots

```
dt.isnull().sum()
```

```
ProductionVolume        0
ProductionCost          0
SupplierQuality         0
DeliveryDelay           0
DefectRate              0
QualityScore            0
MaintenanceHours        0
DowntimePercentage      0
InventoryTurnover       0
StockoutRate            0
WorkerProductivity      0
SafetyIncidents         0
EnergyConsumption       0
EnergyEfficiency        0
AdditiveProcessTime     0
AdditiveMaterialCost    0
DefectStatus            0
dtype: int64
```

```
x=dt[['ProductionVolume','ProductionCost','SupplierQuality','DefectRate','QualityScore',]]
y=dt['DefectStatus']
```

```
d.fit(x,y)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
columns_to_drop = ['DeliveryDelay','MaintenanceHours','DowntimePercentage','InventoryTurnove
```

```
dt.drop(columns_to_drop,axis=1)
```

| | ProductionVolume | ProductionCost | SupplierQuality | DefectRate | QualityScore | Defec |
|---|---|---|---|---|---|---|
| **0** | 202 | 13175.403783 | 86.648534 | 3.121492 | 63.463494 | |
| **1** | 535 | 19770.046093 | 86.310664 | 0.819531 | 83.697818 | |
| **2** | 960 | 19060.820997 | 82.132472 | 4.514504 | 90.350550 | |
| **3** | 370 | 5647.606037 | 87.335966 | 0.638524 | 67.628690 | |
| **4** | 206 | 7472.222236 | 81.989893 | 3.867784 | 82.728334 | |
| **...** | ... | ... | ... | ... | ... | |
| **3235** | 762 | 11325.689263 | 89.252385 | 2.667570 | 87.141681 | |
| **3236** | 335 | 5598.837988 | 95.701437 | 0.751272 | 95.562997 | |
| **3237** | 835 | 11736.177712 | 96.431554 | 4.899756 | 77.973442 | |
| **3238** | 302 | 13664.196210 | 91.089782 | 4.057665 | 95.755591 | |
| **3239** | 355 | 13563.605806 | 83.595956 | 2.705502 | 94.630965 | |

3240 rows × 6 columns

```
d.predict([[100,200,65,30.52,45.3]])
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not hav
  warnings.warn(
array([1])
```

```
l1=int(input("enter prduction volume"))
l2=float(input("production cost"))
l3=float(input("supplier quality"))
l4=float(input("defect rate"))
l5=float(input("quality score"))
out=d.predict([[l1,l2,l3,l4,l5]])
if out==True:
  print("positive")
else:
  print("negative")
```

```
enter prduction volume355
production cost13258.368
supplier quality53.65
defect rate1.003
quality score86.36
positive
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not hav
  warnings.warn(
```

```
from sklearn.tree import export_graphviz
from IPython.display import Image
```

ALGORITHM : DECISION TREE

Decision Tree Classifier Building in Scikit-learn.

Importing Required Libraries.

Loading Data.

Checking for null values

Feature Selection.

Splitting Data.

Building Decision Tree Model.

Predicting.

Visualizing Decision Trees.

This dataset provides insights into factors influencing defect rates in a manufacturing environment. Each record represents various metrics crucial for predicting high or low defect occurrences in production processes.

ProductionVolume: Number of units produced per day.Data Type: Integer.

ProductionCost: Cost incurred for production per day.Data Type: Float.

SupplierQuality: Quality ratings of suppliers.Data Type: Float (%).

DefectRate: Defects per thousand units produced.Data Type: Float.

QualityScore: Overall quality assessment.Data Type: Float (%).

Target Variable:

DefectStatus: Predicted defect status.Data Type: Binary (0 for Low Defects, 1 for High Defects).