Experiment No: 04

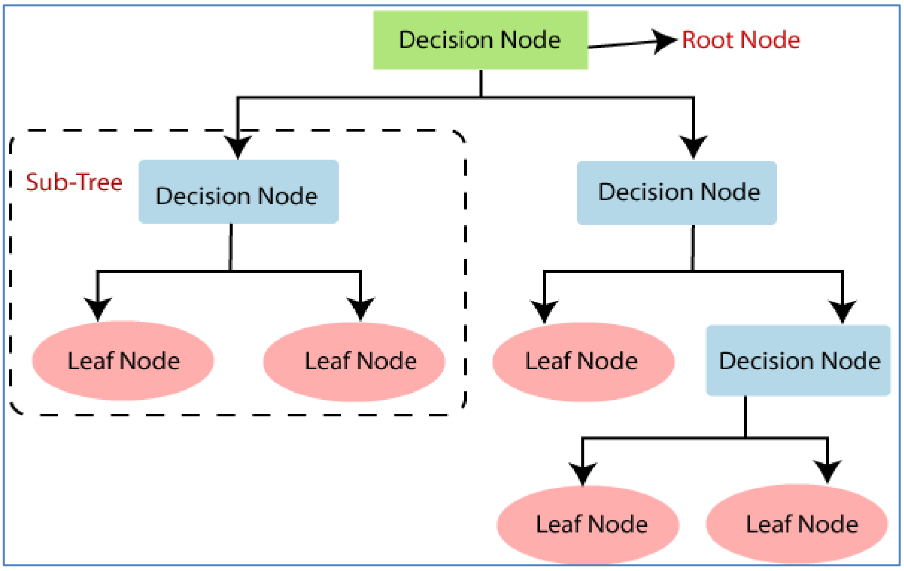
●**Aim**: **To implement Decision Tree Algorithms.**

●**Theory**:

Decision Tree

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

- In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.
- A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.



How does the Decision Tree algorithm Work?

- **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
- **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**
- **Step-3:** Divide the S into subsets that contains possible values for the best attributes.
- **Step-4:** Generate the decision tree node, which contains the best attribute.
- **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

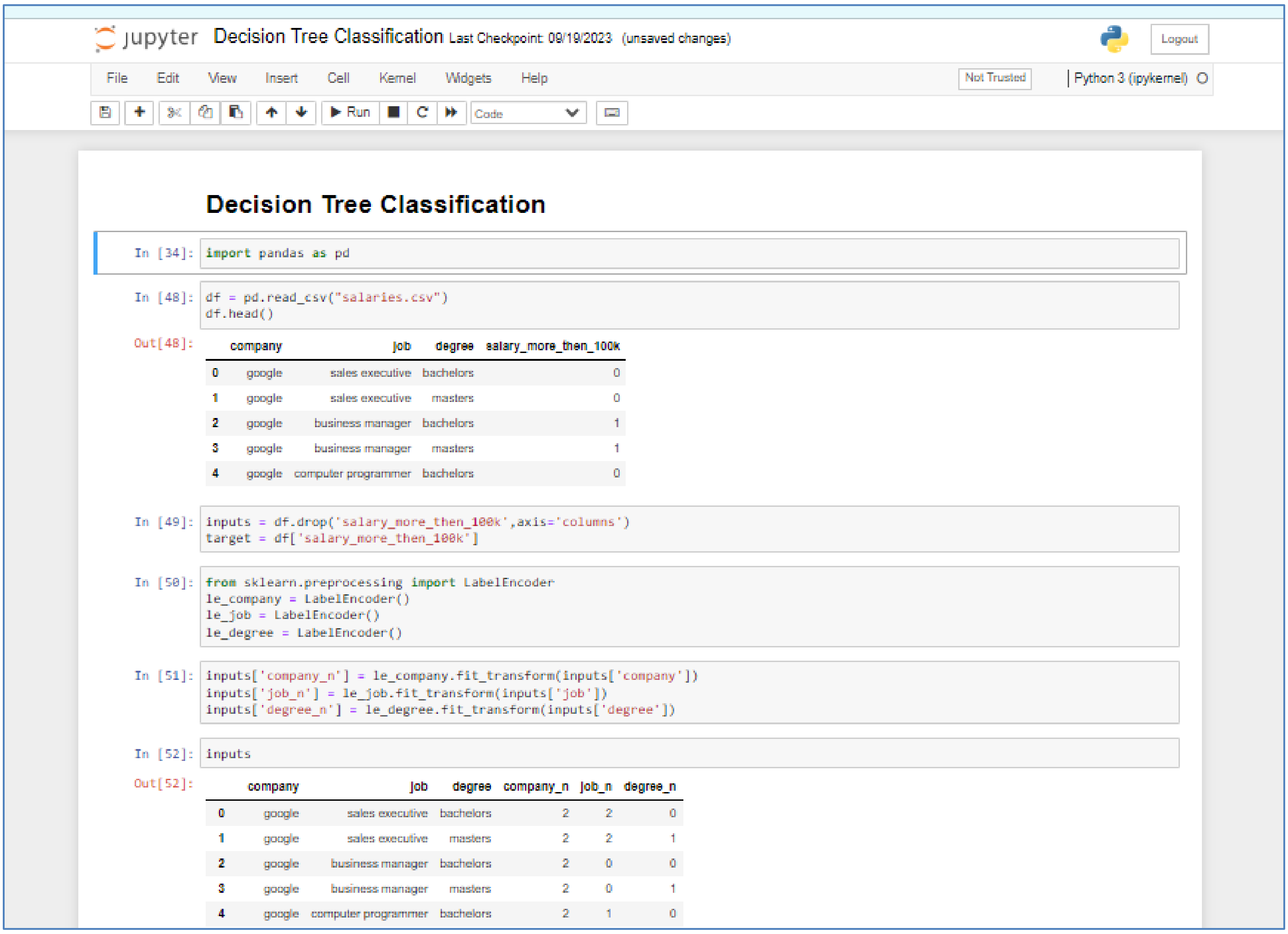
Advantages of the Decision Tree

- It is simple to understand as it follows the same process which a human follow while making any decision in real-life.
- It can be very useful for solving decision-related problems.
- It helps to think about all the possible outcomes for a problem.
- There is less requirement of data cleaning compared to other algorithms.

Disadvantages of the Decision Tree

- The decision tree contains lots of layers, which makes it complex.
- It may have an overfitting issue, which can be resolved using the Random Forest algorithm.
- For more class labels, the computational complexity of the decision tree may increase

Implementation of Decision Tree Algorithms:



Decision Tree Classification

In [34]:
```
import pandas as pd
```

In [48]:
```
df = pd.read_csv("salaries.csv")
df.head()
```

Out[48]:

	company	job	degree	salary_more_then_100k
0	google	sales executive	bachelors	0
1	google	sales executive	masters	0
2	google	business manager	bachelors	1
3	google	business manager	masters	1
4	google	computer programmer	bachelors	0

In [49]:
```
inputs = df.drop('salary_more_then_100k',axis='columns')
target = df['salary_more_then_100k']
```

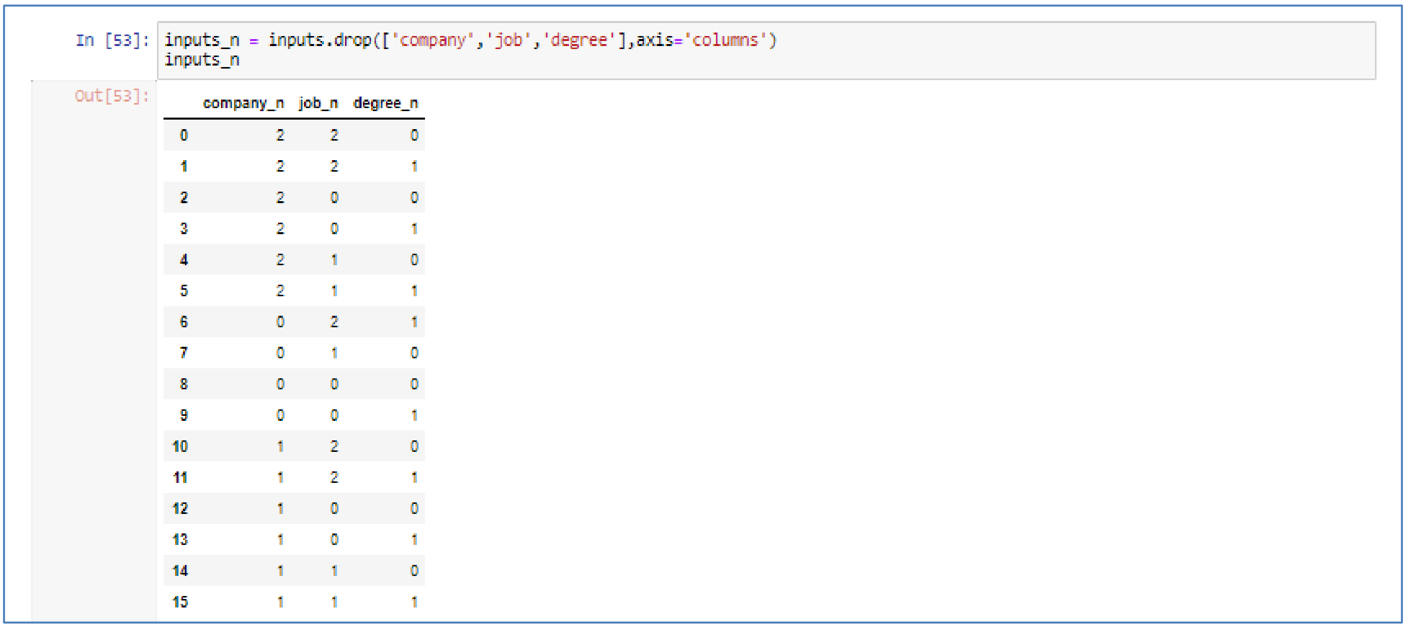
In [50]:
```
from sklearn.preprocessing import LabelEncoder
le_company = LabelEncoder()
le_job = LabelEncoder()
le_degree = LabelEncoder()
```

In [51]:
```
inputs['company_n'] = le_company.fit_transform(inputs['company'])
inputs['job_n'] = le_job.fit_transform(inputs['job'])
inputs['degree_n'] = le_degree.fit_transform(inputs['degree'])
```

In [52]:
```
inputs
```

Out[52]:

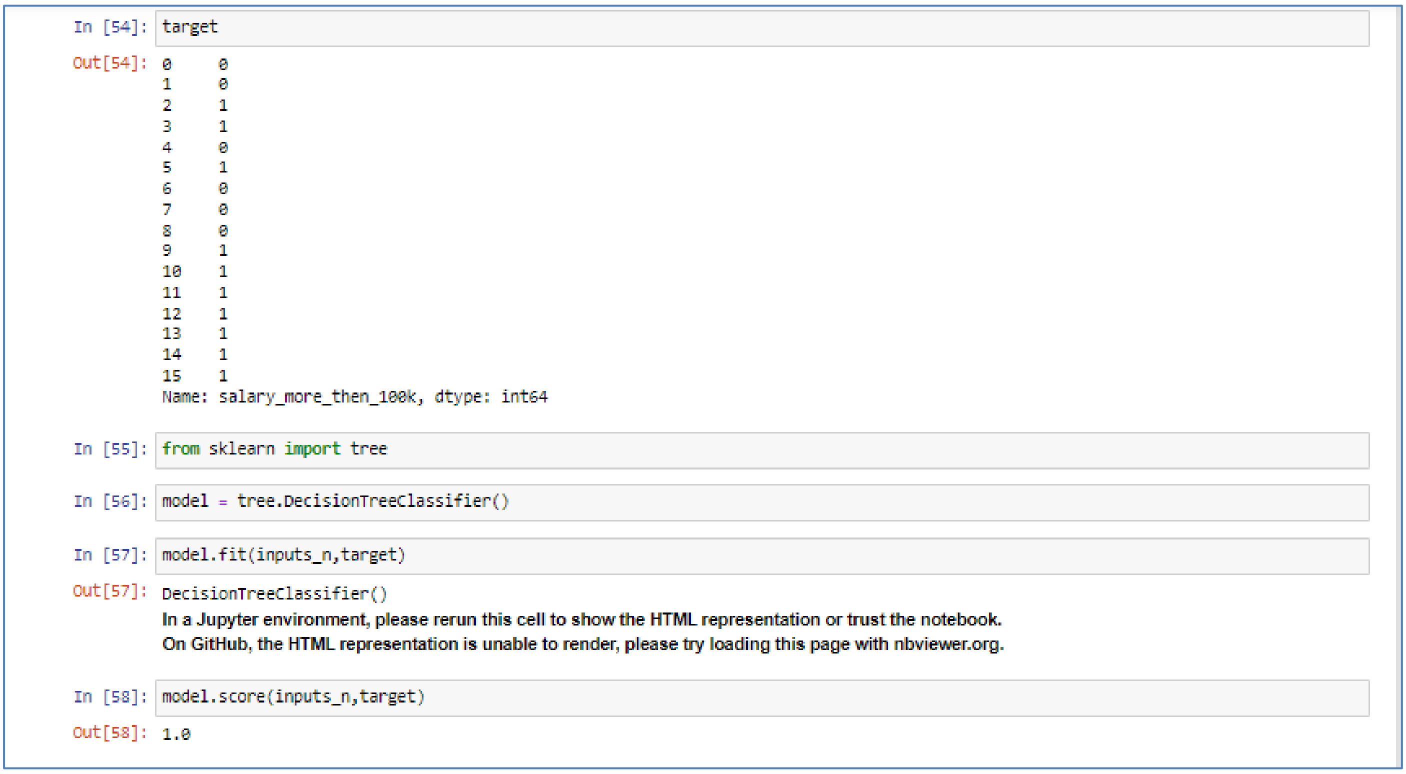
	company	job	degree	company_n	job_n	degree_n
0	google	sales executive	bachelors	2	2	0
1	google	sales executive	masters	2	2	1
2	google	business manager	bachelors	2	0	0
3	google	business manager	masters	2	0	1
4	google	computer programmer	bachelors	2	1	0



```
In [53]: inputs_n = inputs.drop(['company','job','degree'],axis='columns')
         inputs_n
```

Out[53]:

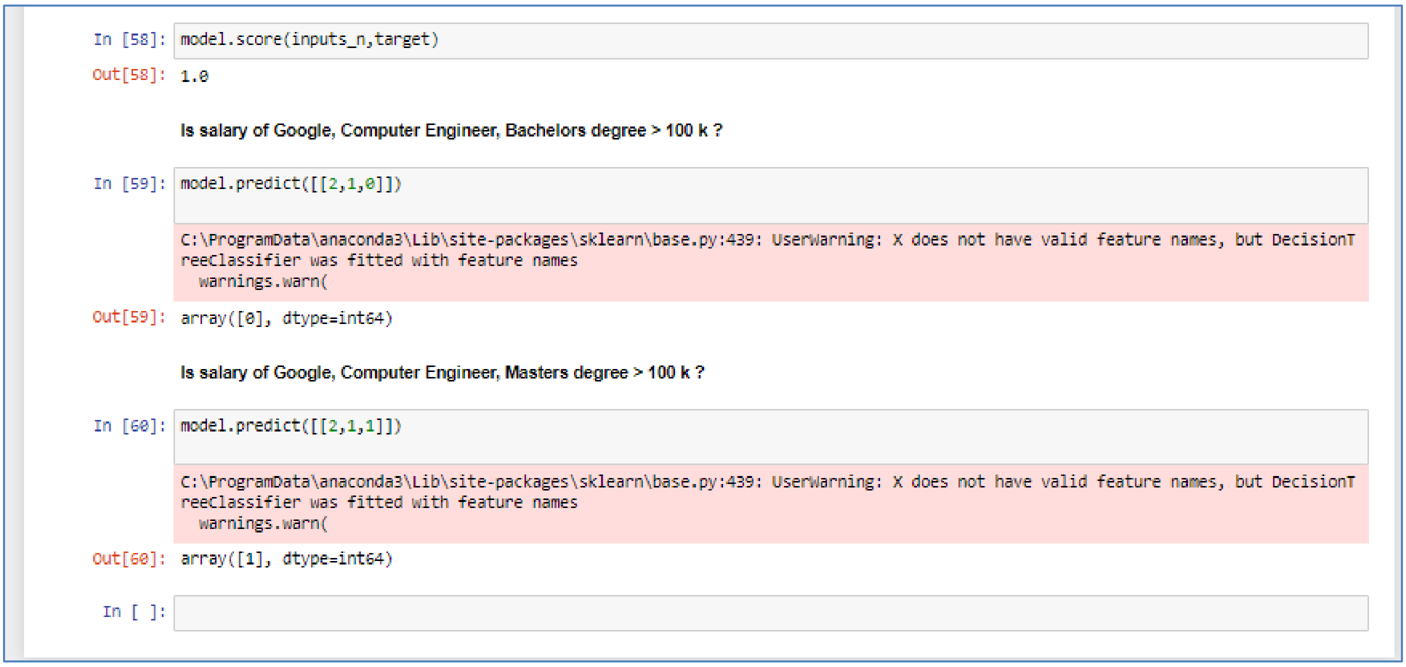
	company_n	job_n	degree_n
0	2	2	0
1	2	2	1
2	2	0	0
3	2	0	1
4	2	1	0
5	2	1	1
6	0	2	1
7	0	1	0
8	0	0	0
9	0	0	1
10	1	2	0
11	1	2	1
12	1	0	0
13	1	0	1
14	1	1	0
15	1	1	1



```
In [54]: target

Out[54]: 0     0
         1     0
         2     1
         3     1
         4     0
         5     1
         6     0
         7     0
         8     0
         9     1
         10    1
         11    1
         12    1
         13    1
         14    1
         15    1
         Name: salary_more_then_100k, dtype: int64

In [55]: from sklearn import tree

In [56]: model = tree.DecisionTreeClassifier()

In [57]: model.fit(inputs_n,target)

Out[57]: DecisionTreeClassifier()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
In [58]: model.score(inputs_n,target)

Out[58]: 1.0
```



```
In [58]: model.score(inputs_n,target)
Out[58]: 1.0
```

Is salary of Google, Computer Engineer, Bachelors degree > 100 k ?

```
In [59]: model.predict([[2,1,0]])

C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning: X does not have valid feature names, but DecisionT
reeClassifier was fitted with feature names
  warnings.warn(
Out[59]: array([0], dtype=int64)
```

Is salary of Google, Computer Engineer, Masters degree > 100 k ?

```
In [60]: model.predict([[2,1,1]])

C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning: X does not have valid feature names, but DecisionT
reeClassifier was fitted with feature names
  warnings.warn(
Out[60]: array([1], dtype=int64)

In [ ]:
```

● Conclusion:

Decision tree algorithms are interpretable and widely used in machine learning due to their simplicity and effectiveness, making them valuable tools for both beginners and experienced data scientists.