Experiment 09– Decision Tree Learning

Learning Objective: Apply Decision Tree Learning on given problem.

Tools: Python under Windows or Linux environment.

Theory:

A **decision tree** is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm.

Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal, but are also a popular tool in machine learning.

Advantages of decision trees:

Among decision support tools, decision trees (and influence diagrams) have several advantages. Decision trees:

- Are simple to understand and interpret. People are able to understand decision tree models after a brief explanation.
- Have value even with little hard data. Important insights can be generated based on experts describing a situation (its alternatives, probabilities, and costs) and their preferences for outcomes.
- Allow the addition of new possible scenarios.
- Help determine worst, best and expected values for different scenarios.
- Use a white box model. If a given result is provided by a model.
- Can be combined with other decision techniques.

Disadvantages of decision trees:

- For data including categorical variables with different number of levels, information gain in decision trees is biased in favor of those attributes with more levels.[6]
- Calculations can get very complex, particularly if many values are uncertain and/or if many outcomes are linked.

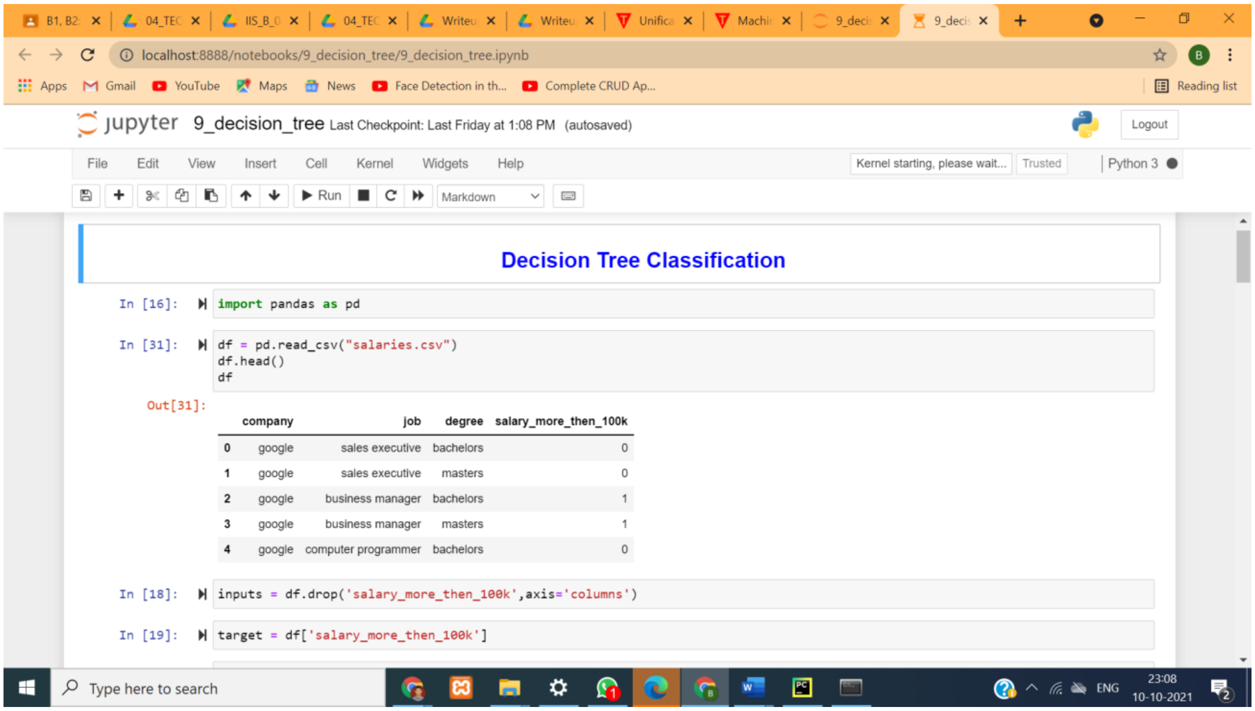
Applications

- `Agriculture
- Astronomy

Algorithm:

- *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.

Design:

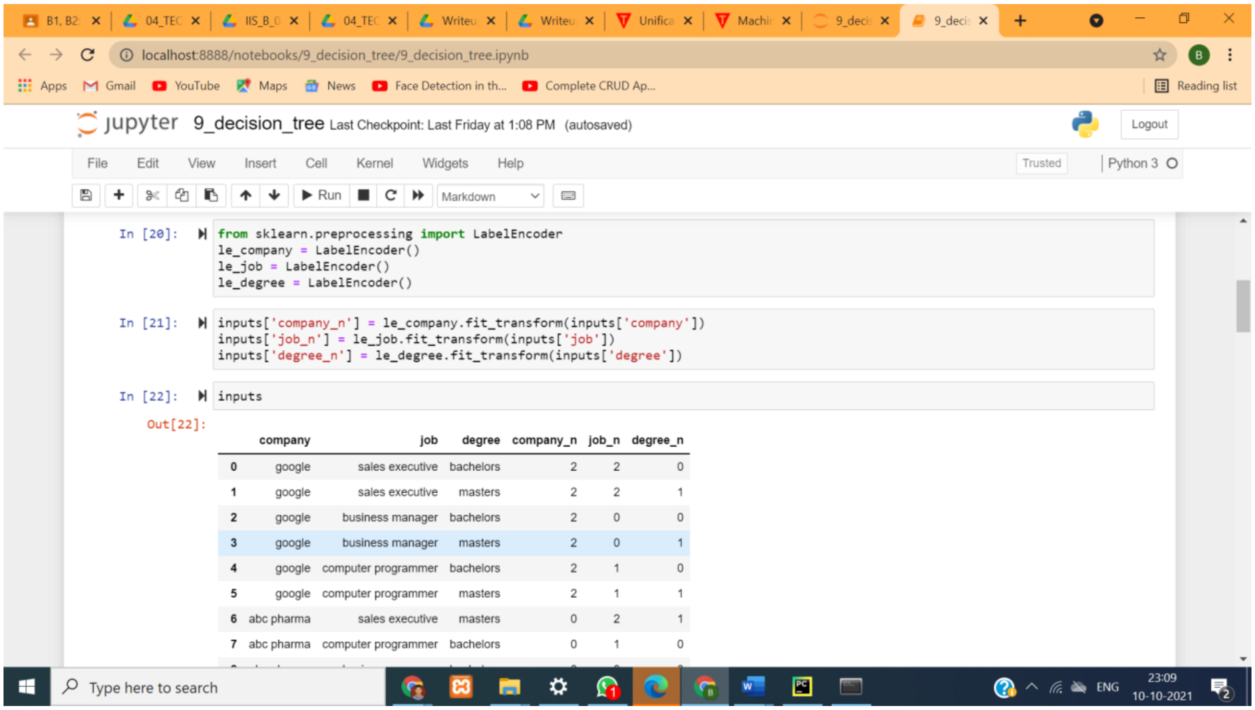


Decision Tree Classification

```
In [16]:  import pandas as pd

In [31]:  df = pd.read_csv("salaries.csv")
          df.head()
          df

Out[31]:
```

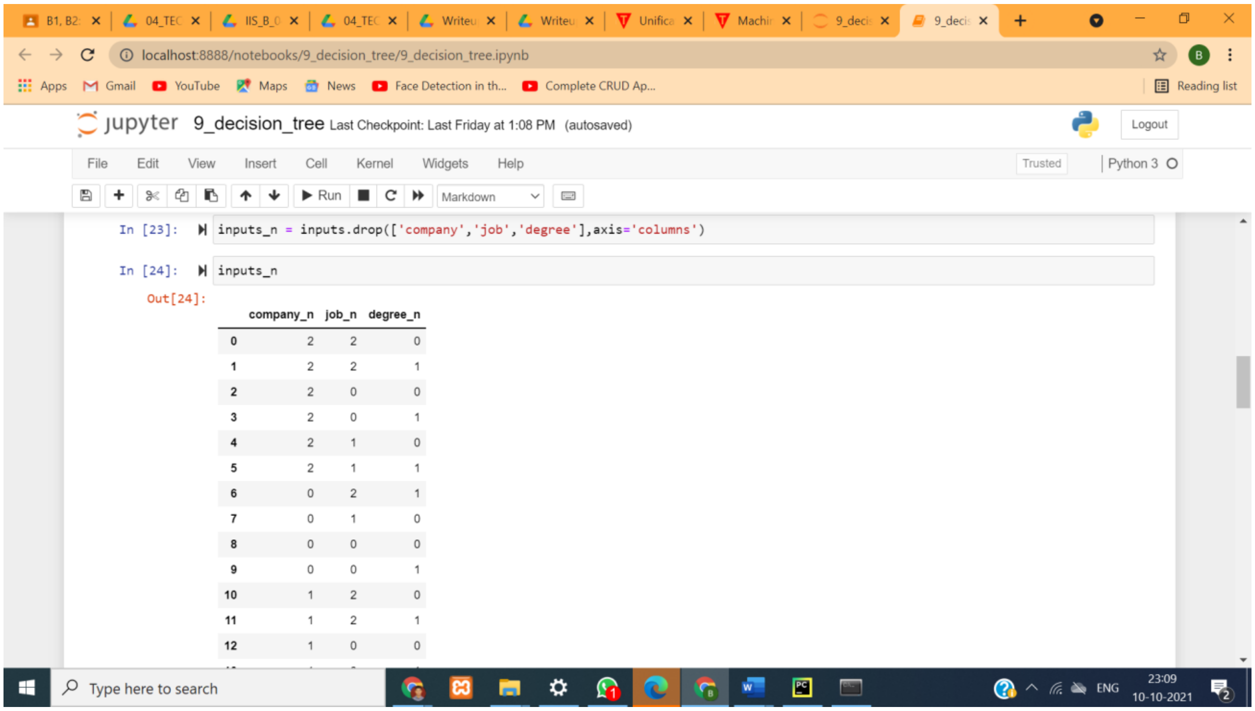
	company	job	degree	salary_more_then_100k
0	google	sales executive	bachelors	0
1	google	sales executive	masters	0
2	google	business manager	bachelors	1
3	google	business manager	masters	1
4	google	computer programmer	bachelors	0

```
In [18]:  inputs = df.drop('salary_more_then_100k',axis='columns')

In [19]:  target = df['salary_more_then_100k']
```


localhost:8888/notebooks/9_decision_tree/9_decision_tree.ipynb
jupyter 9_decision_tree Last Checkpoint: Last Friday at 1:08 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown
Type here to search
23:09
10-10-2021

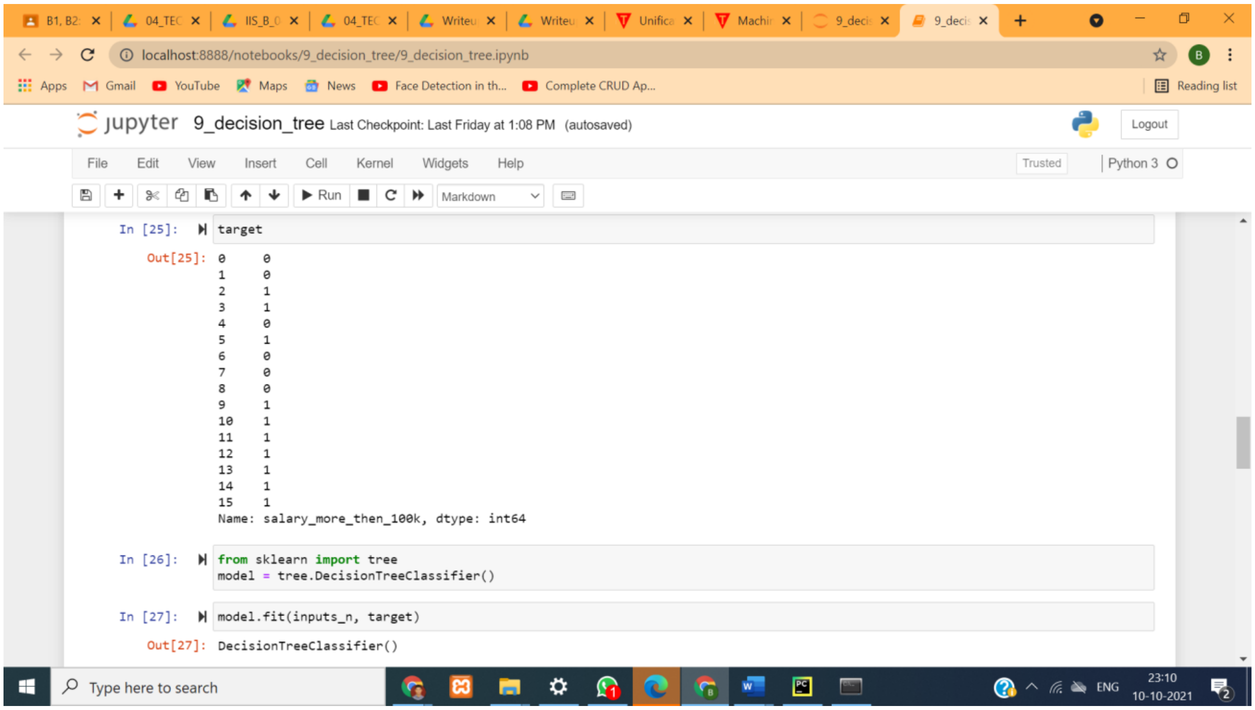
```
In [20]:  from sklearn.preprocessing import LabelEncoder
          le_company = LabelEncoder()
          le_job = LabelEncoder()
          le_degree = LabelEncoder()

In [21]:  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 [22]:  inputs

Out[22]:
```

	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
5	google	computer programmer	masters	2	1	1
6	abc pharma	sales executive	masters	0	2	1
7	abc pharma	computer programmer	bachelors	0	1	0

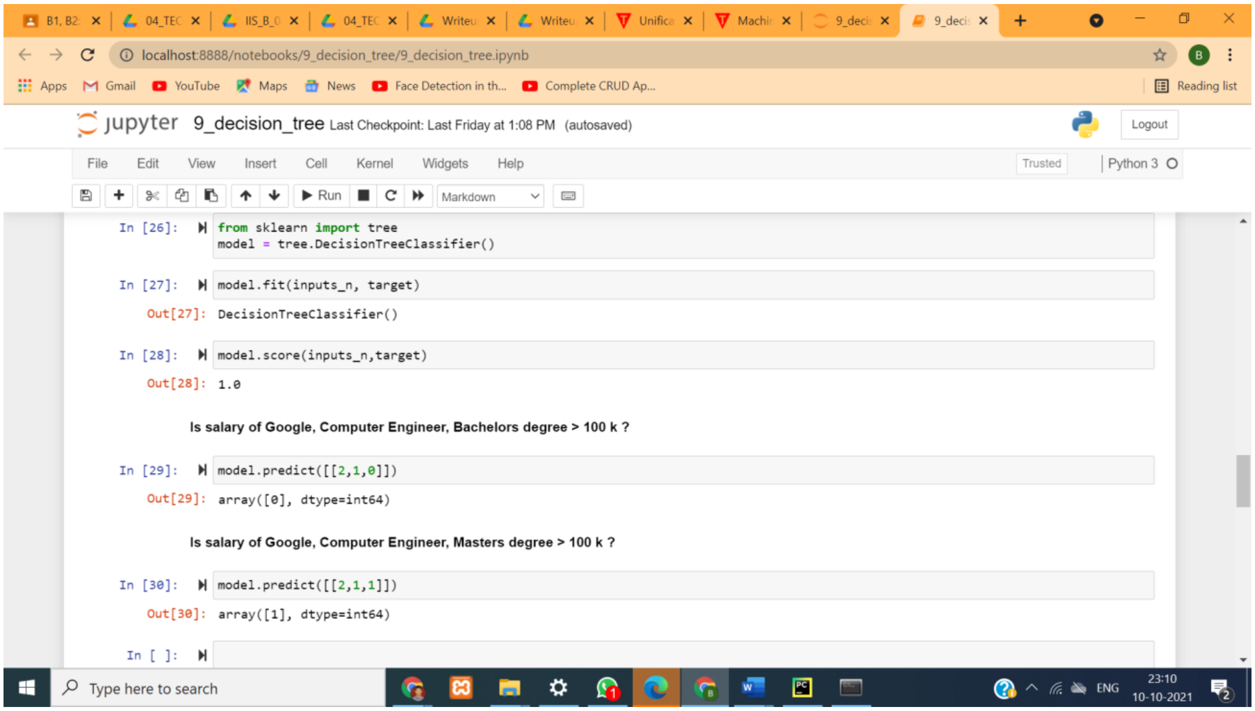

localhost:8888/notebooks/9_decision_tree/9_decision_tree.ipynb
jupyter 9_decision_tree Last Checkpoint: Last Friday at 1:08 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown
Type here to search
23:09
10-10-2021

```
In [23]:  inputs_n = inputs.drop(['company','job','degree'],axis='columns')

In [24]:  inputs_n

Out[24]:
```

	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


localhost:8888/notebooks/9_decision_tree/9_decision_tree.ipynb
jupyter 9_decision_tree Last Checkpoint: Last Friday at 1:08 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown
Type here to search
23:10
10-10-2021

```
In [25]:  target

Out[25]: 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 [26]:  from sklearn import tree
          model = tree.DecisionTreeClassifier()

In [27]:  model.fit(inputs_n, target)

Out[27]: DecisionTreeClassifier()
```


localhost:8888/notebooks/9_decision_tree/9_decision_tree.ipynb
jupyter 9_decision_tree Last Checkpoint: Last Friday at 1:08 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown
Type here to search
23:10
10-10-2021

```
In [26]:  from sklearn import tree
          model = tree.DecisionTreeClassifier()

In [27]:  model.fit(inputs_n, target)

Out[27]: DecisionTreeClassifier()

In [28]:  model.score(inputs_n,target)

Out[28]: 1.0
```

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

```
In [29]:  model.predict([[2,1,0]])

Out[29]: array([0], dtype=int64)
```

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

```
In [30]:  model.predict([[2,1,1]])

Out[30]: array([1], dtype=int64)

In [ ]:
```

Learning Outcomes: The student should have the ability to

LO9.1 Ability to formulate Problem

LO9.2 Ability to represent problem

LO9.3 Ability to apply Learning algorithms

LO9.4 Ability to verify the program output to check values at specified location in Decision Tree

Course Outcomes:

Understand various forms of learning. Apply decision tree learning to a given problems.

Conclusion: Successfully learned and applied Decision tree Algorithm.

Viva Questions:

Q.1 Is decision tree learning is inductive or deductive learning?

Q.2 What are the basic types of decision tree algorithm?

Q.3 List advantages and disadvantages of decision tree algorithms?

For Faculty Use

Correction Parameters	**Formative Assessment [40%]**	**Timely completion of Practical [40%]**	**Attendance / Learning Attitude [20%]**	
Marks Obtained				