

Decision Tree Classification

```
In [16]: import pandas as pd
df = pd.read_csv("C:/Users/prasa/Desktop/py codes/ds projects/ML/8 Decision Tree/salaries.csv")
df.head()
```

Out[16]:

	company	job	degree	salary_more_than_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_than_100k',axis='columns')
target = df ['salary_more_than_100k']
```

```
In [21]: from sklearn.preprocessing import LabelEncoder
```

```
In [22]: le_company=LabelEncoder()
le_job=LabelEncoder()
le_degree=LabelEncoder()
```

```
In [24]: 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'])
inputs.head()
```

Out[24]:

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

Out[25]:

	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

	company_n	job_n	degree_n
15	1	1	1

```
In [27]: from sklearn import tree
```

```
In [28]: model = tree.DecisionTreeClassifier()
```

```
In [29]: model.fit(inputs_n,target) #Gini entropy difference
```

```
Out[29]: DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                                max_depth=None, max_features=None, max_leaf_node
                                s=None,
                                min_impurity_decrease=0.0, min_impurity_split=No
                                ne,
                                min_samples_leaf=1, min_samples_split=2,
                                min_weight_fraction_leaf=0.0, presort='deprecate
                                d',
                                random_state=None, splitter='best')
```

```
In [30]: model.score(inputs_n,target) #one because we used same data set for tra
ining and test
```

```
Out[30]: 1.0
```

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

```
In [38]: model.predict([[2,1,0]]) #[1] = More then $100k
```

```
Out[38]: array([0], dtype=int64)
```

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

```
In [40]: model.predict([[2,1,1]])
```

Out[40]: array([1], dtype=int64)

Exercise: Build decision tree model to predict survival based on certain parameters



CSV file is available to download at

https://github.com/codebasics/py/blob/master/ML/9_decision_tree/Exercise/titanic.csv

In this file using following columns build a model to predict if person would survive or not,

1. Pclass
2. Sex
3. Age
4. Fare

Calculate score of your model