

```
In [1]: import pandas as pd
```

```
In [2]: df = pd.read_csv("Decision Tree.csv")  
df
```

```
Out[2]:
```

	company	job	degree	salary_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
5	google	computer programmer	masters	1
6	abc pharma	sales executive	masters	0
7	abc pharma	computer programmer	bachelors	0
8	abc pharma	business manager	bachelors	0
9	abc pharma	business manager	masters	1
10	facebook	sales executive	bachelors	1
11	facebook	sales executive	masters	1
12	facebook	business manager	bachelors	1
13	facebook	business manager	masters	1
14	facebook	computer programmer	bachelors	1
15	facebook	computer programmer	masters	1

```
In [4]: inputs = df.drop('salary_100k',axis='columns')
```

```
In [5]: target = df['salary_100k']
```

```
In [6]: from sklearn.preprocessing import LabelEncoder
```

```
In [10]: le_company = LabelEncoder()  
le_job = LabelEncoder()  
le_degree = LabelEncoder()
```

```
In [11]: 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 [12]: inputs
```

```
Out[12]:
```

	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
8	abc pharma	business manager	bachelors	0	0	0
9	abc pharma	business manager	masters	0	0	1
10	facebook	sales executive	bachelors	1	2	0
11	facebook	sales executive	masters	1	2	1
12	facebook	business manager	bachelors	1	0	0
13	facebook	business manager	masters	1	0	1
14	facebook	computer programmer	bachelors	1	1	0
15	facebook	computer programmer	masters	1	1	1

```
In [13]: inputs_n = inputs.drop(['company','job','degree'],axis='columns')
```

```
In [15]: target
```

```
Out[15]: 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_100k, dtype: int64
```

```
In [16]: from sklearn import tree
model = tree.DecisionTreeClassifier()
```

```
In [17]: model.fit(inputs_n, target)
```

```
Out[17]: DecisionTreeClassifier()
```

```
In [18]: model.score(inputs_n,target)
```

```
Out[18]: 1.0
```

```
In [19]: #Is salary of Google, Computer Engineer, Bachelors degree > 100 k ?
```

```
In [21]: model.predict([[2,1,0]])
```

```
C:\Users\prama.gupta\Anaconda3\lib\site-packages\sklearn\base.py:445: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

```
Out[21]: array([0], dtype=int64)
```

```
In [22]: #Survivial Analysis
#Exercise: Build decision tree model to predict survival based on certain paramet
```

```
In [36]: df = pd.read_csv("titanic.csv")
df
#df.head()
```

```
Out[36]:
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...)	female	38.0	1	0	PC 17599	71.2833	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	
...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	

891 rows × 12 columns

```
In [37]: df.drop(['PassengerId', 'Name', 'SibSp', 'Parch', 'Ticket', 'Cabin', 'Embarked'], axis=1)
```

```
In [38]: df.head()
```

```
Out[38]:
```

	Survived	Pclass	Sex	Age	Fare
0	0	3	male	22.0	7.2500
1	1	1	female	38.0	71.2833
2	1	3	female	26.0	7.9250
3	1	1	female	35.0	53.1000
4	0	3	male	35.0	8.0500

```
In [43]: inputs = df.drop('Survived',axis='columns')
target = df['Survived']
```

```
In [44]: inputs.Sex = inputs.Sex.map({'male': 1, 'female': 2})
```

```
In [45]: inputs.Age[:10]
inputs.Age = inputs.Age.fillna(inputs.Age.mean())
```

```
In [46]: inputs.head()
```

```
Out[46]:
```

	Pclass	Sex	Age	Fare
0	3	1	22.0	7.2500
1	1	2	38.0	71.2833
2	3	2	26.0	7.9250
3	1	2	35.0	53.1000
4	3	1	35.0	8.0500

```
In [48]: from sklearn.model_selection import train_test_split
```

```
In [49]: X_train, X_test, y_train, y_test = train_test_split(inputs,target,test_size=0.2)
```

```
In [50]: X_train, X_test, y_train, y_test = train_test_split(inputs, target, test_size = 0.2)
```

```
In [51]: from sklearn import tree
model = tree.DecisionTreeClassifier()
```

```
In [52]: model.fit(X_train,y_train)
```

```
Out[52]: DecisionTreeClassifier()
```

```
In [53]: model.score(X_test,y_test)
```

```
Out[53]: 0.7877094972067039
```

```
In [54]: model.score(X_train,y_train)
```

```
Out[54]: 0.9817415730337079
```

```
In [56]: model.predict([[1,2,36,8]])
```

```
C:\Users\prama.gupta\Anaconda3\lib\site-packages\sklearn\base.py:445: UserWarni  
ng: X does not have valid feature names, but DecisionTreeClassifier was fitted  
with feature names  
  warnings.warn(
```

```
Out[56]: array([1], dtype=int64)
```