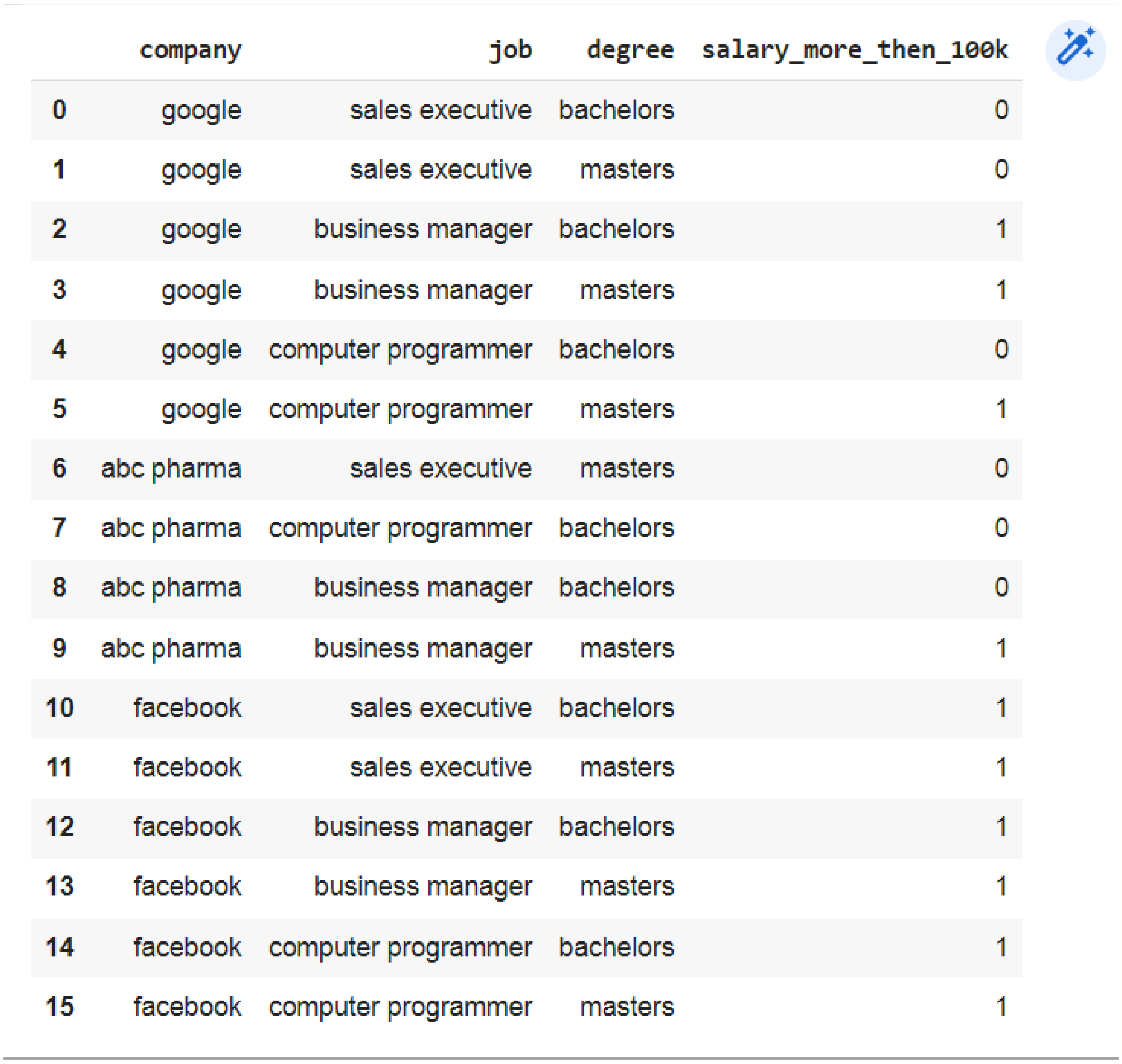
Practical No. 4

Exercise -

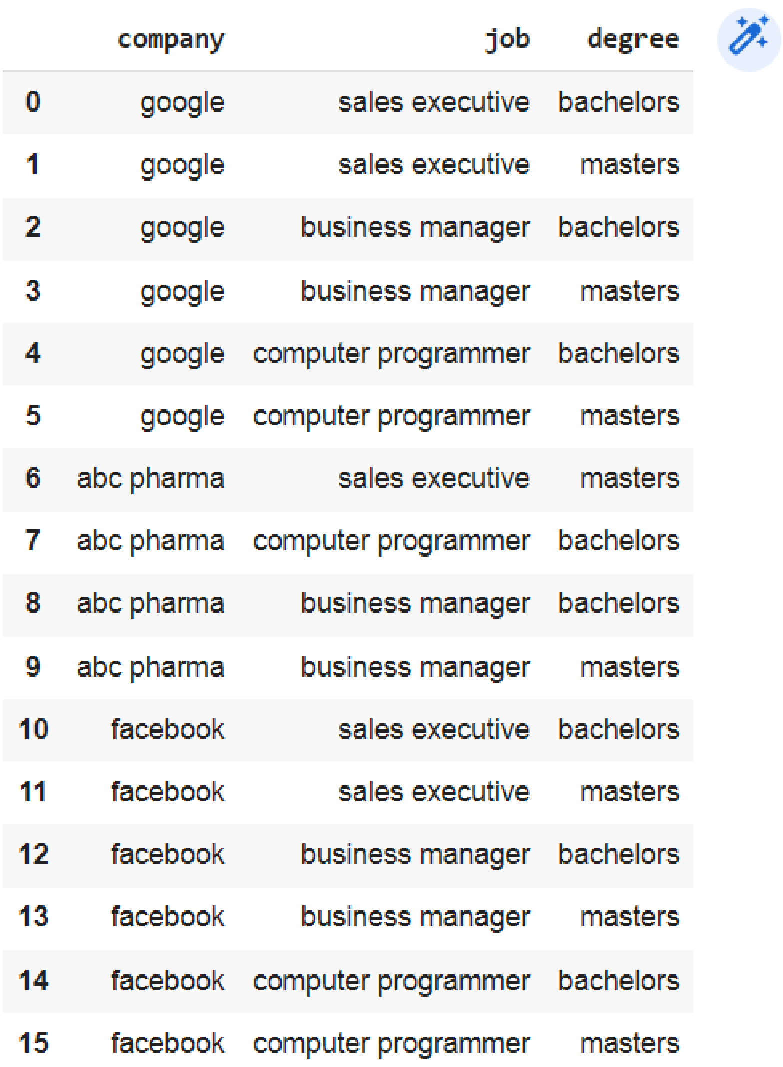
1. Implement given dataset to predict salary range of computer programmer from google having bachelor degree / master's degree

```
#Tree
import pandas as pd
url="https://raw.githubusercontent.com/codebasics/py/master/ML/9_decision_tree/salaries.csv"
df=pd.read_csv(url)
df
```



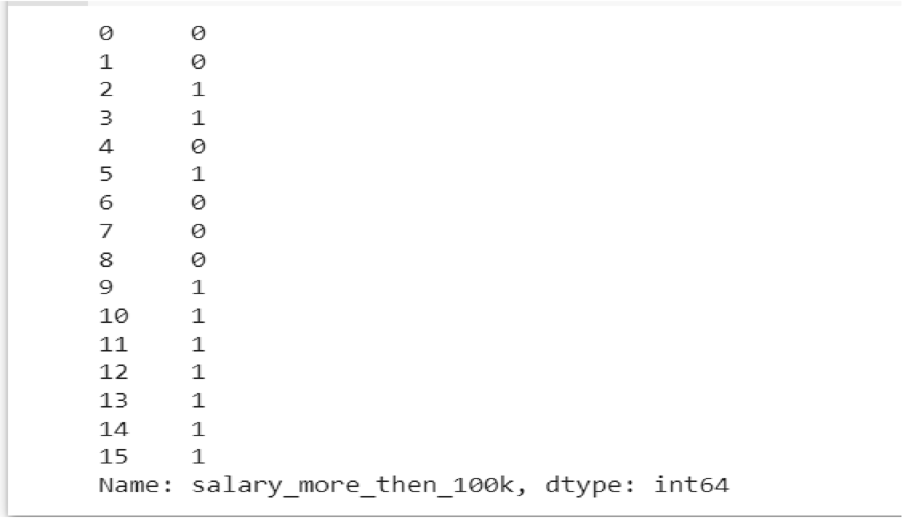
	company	job	degree	salary_more_then_100k
0	google	sales executive	bachelors	0
1	google	sales executive	masters	0
2	google	business manager	bachelors	1
3	google	business manager	masters	1
4	google	computer programmer	bachelors	0
5	google	computer programmer	masters	1
6	abc pharma	sales executive	masters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

```
inp=df.drop(['salary_more_then_100k'],axis="columns");
inp
```



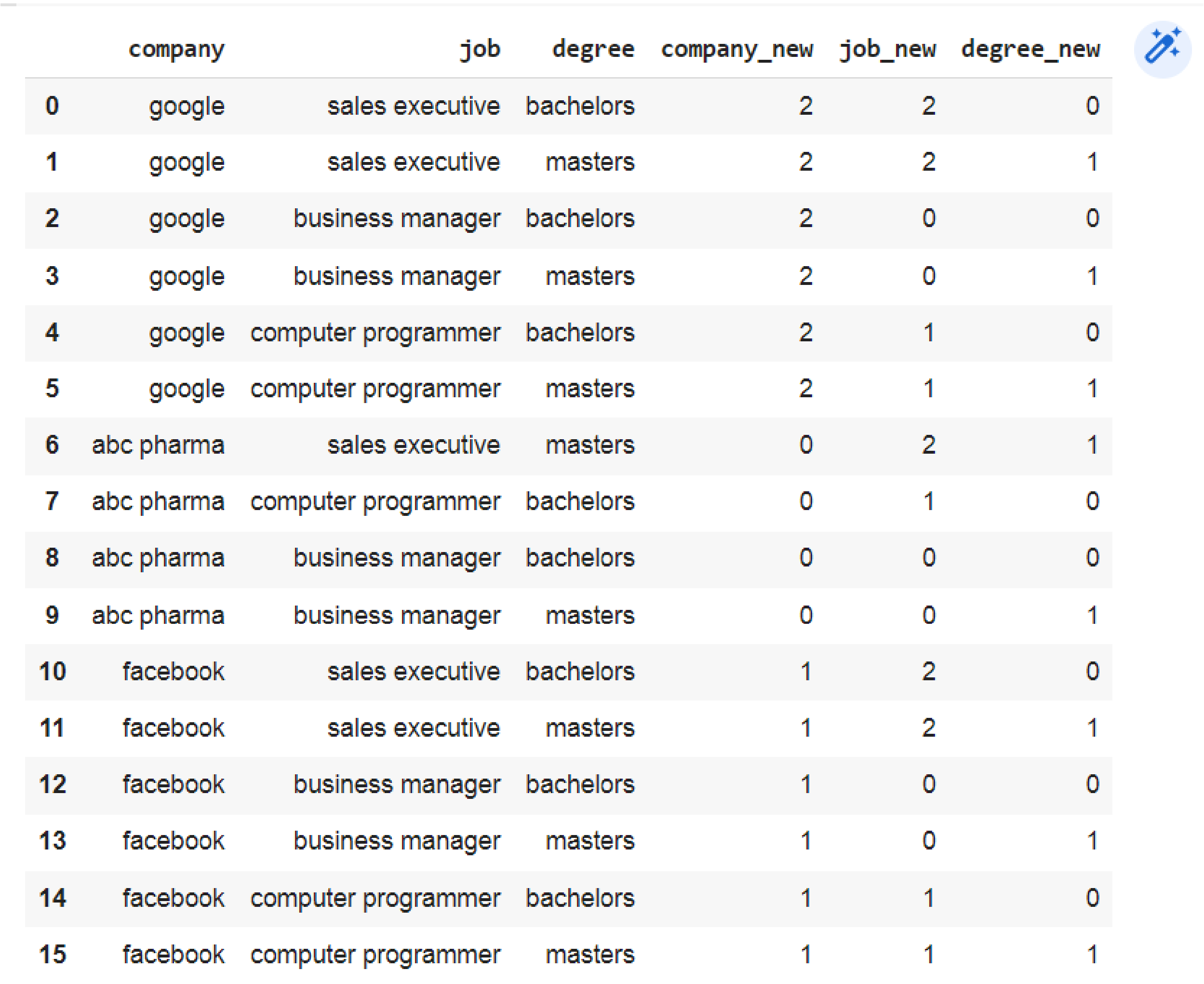
	company	job	degree
0	google	sales executive	bachelors
1	google	sales executive	masters
2	google	business manager	bachelors
3	google	business manager	masters
4	google	computer programmer	bachelors
5	google	computer programmer	masters
6	abc pharma	sales executive	masters
7	abc pharma	computer programmer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

```
trgt=df['salary_more_then_100k']
trgt
```



```
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
```

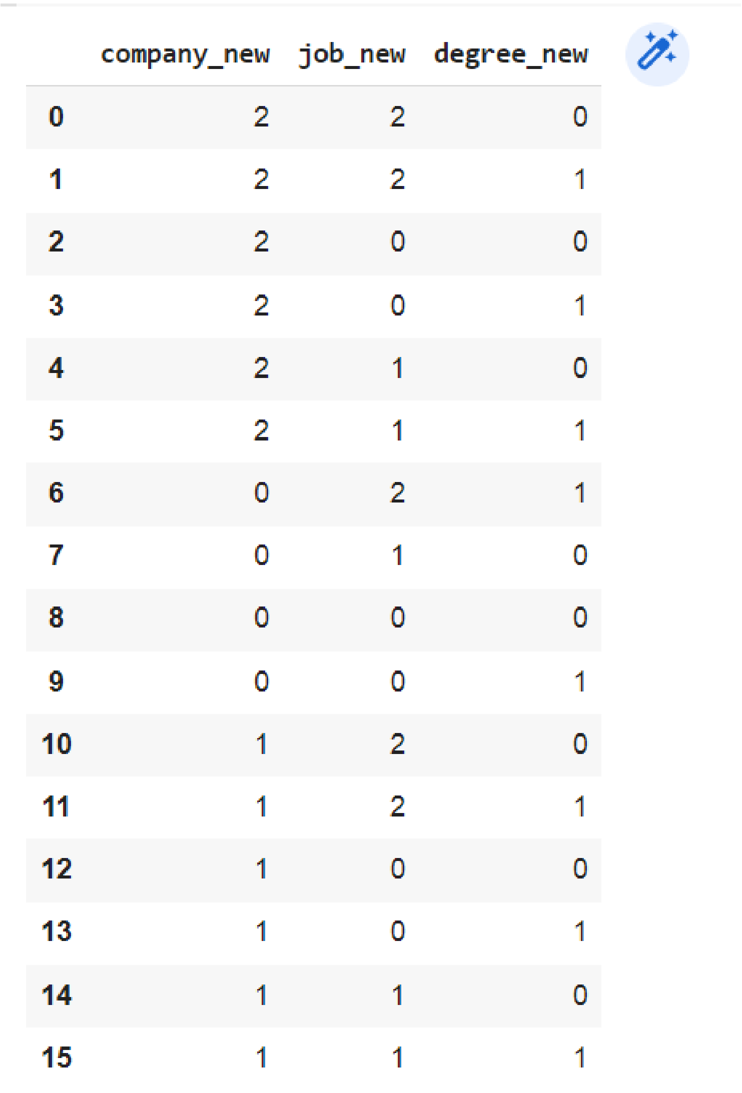
```
from sklearn.preprocessing import LabelEncoder
lbl_company = LabelEncoder()
lbl_job = LabelEncoder()
lbl_degree = LabelEncoder()


inp['company_new']=lbl_company.fit_transform(inp['company'])
inp['job_new']=lbl_company.fit_transform(inp['job'])
inp['degree_new']=lbl_company.fit_transform(inp['degree'])
inp
```



	company	job	degree	company_new	job_new	degree_new
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

```
inp_new=inp.drop(['company','job','degree'],axis="columns")
```

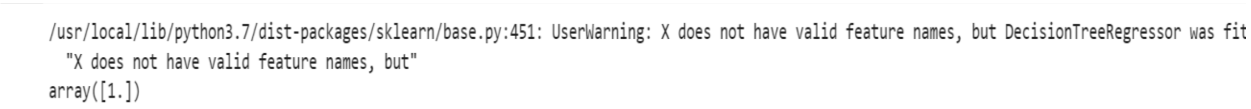
```
inp_new
```



	company_new	job_new	degree_new
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

```
from sklearn import tree
tree_model = tree.DecisionTreeRegressor()
tree_model.fit(inp_new,trgt)
```

```
tree_model.predict([[2,1,1]])
```



```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have valid feature names, but DecisionTreeRegressor was fit
  "X does not have valid feature names, but"
array([1.])
```