```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
import sklearn as sk
print(sk.__version__)
```

```
1.2.2
```

```
!pip install -U scikit-learn==1.2.2
```

```
Requirement already satisfied: scikit-learn==1.2.2 in /usr/local/lib/python3.10/dist-packages (1.2.2)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from scikit-learn==1
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from scikit-learn==1.
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn==1
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-l
```

```
#read hackathon data

data = pd.read_csv('/content/Employee2023.csv')
```

```
data.head()
```

	EmpID	FirstName	LastName	ADEmail	DepartmentType	Division	
0	3427	Uriah	Bridges	uriah.bridges@bilearner.com	Production	Finance	1
1	3428	Paula	Small	paula.small@bilearner.com	Production	Finance	1
2	3429	Edward	Buck	edward.buck@bilearner.com	Production	Finance	1
3	3430	Michael	Riordan	michael.riordan@bilearner.com	Production	Finance	1
4	3431	Jasmine	Onque	jasmine.onque@bilearner.com	Production	Finance	1

```
data.shape
```

```
(149, 9)
```

```
#check infor
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 149 entries, 0 to 148
Data columns (total 9 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   EmpID                  149 non-null    int64
 1   FirstName              149 non-null    object
 2   LastName               149 non-null    object
 3   ADEmail                149 non-null    object
 4   DepartmentType         149 non-null    object
 5   Division               149 non-null    object
 6   DOB                    149 non-null    object
```

```
 7   JobFunctionDescription  149 non-null    object
 8   GenderCode              149 non-null    object
dtypes: int64(1), object(8)
memory usage: 10.6+ KB
```

```
#check the null value
data.isnull().sum()
```

```
EmpID                   0
FirstName               0
LastName                0
ADEmail                 0
DepartmentType          0
Division                0
DOB                     0
JobFunctionDescription  0
GenderCode              0
dtype: int64
```

```
#check the number of counts of differnt deparment
data['Division'].value_counts()
```

```
Finance              30
Engineers            30
ProjectManagement    30
Sales&Marketing      30
Technology           28
Field Operations      1
Name: Division, dtype: int64
```

```
data['JobFunctionDescription'].value_counts()
```

```
Accounting          15
Ceo                 15
Field Technician    15
Program Manager     15
Intern              15
Specialist          15
Engineer            15
Manager             15
Director            15
Technician          14
Name: JobFunctionDescription, dtype: int64
```

```
data.head(30)
```

15	3442	Kaylah	Moon	kaylah.moon@bilearner.com	Production
16	3443	Kristen	Tate	kristen.tate@bilearner.com	Production
17	3444	Bobby	Rodgers	bobby.rodgers@bilearner.com	Production
18	3445	Reid	Park	reid.park@bilearner.com	Production
19	3446	Hector	Dalton	hector.dalton@bilearner.com	Production
20	3447	Mariela	Schultz	mariela.schultz@bilearner.com	Production
21	3448	Angela	Molina	angela.molina@bilearner.com	Production
22	3449	Gerald	Preston	gerald.preston@bilearner.com	Production
23	3450	Reilly	Moyer	reilly.moyer@bilearner.com	Production
24	3451	Carlee	French	carlee.french@bilearner.com	Production
25	3452	Jaydon	Blackburn	jaydon.blackburn@bilearner.com	Production
26	3453	Bridger	Carter	bridger.carter@bilearner.com	Production
27	3454	Leon	Beard	leon.beard@bilearner.com	Production
28	3455	Charity	Miranda	charity.miranda@bilearner.com	Production
29	3456	Axel	Howe	axel.howe@bilearner.com	IT/IS

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
label = le.fit_transform(data["Division"])
label
```

```
array([2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4,
       4, 4, 5, 5, 5, 5, 5, 5, 2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 3, 3,
       3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 2, 2, 2, 2, 2, 2,
       0, 0, 0, 0, 0, 0, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5,
       2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4,
       4, 4, 5, 5, 5, 5, 5, 5, 2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 3, 3,
       3, 3, 3, 3, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 1])
```

```
print(le.classes_)
```

```
['Engineers' 'Field Operations' 'Finance ' 'ProjectManagement'
 'Sales&Marketing' 'Technology']
```

```
data.drop("Division", axis=1, inplace=True)
```

```
data["Division"]=label
data.head(10)
```

	EmpID	FirstName	LastName	ADEmail	DepartmentType	DOB	JobFu
0	3427	Uriah	Bridges	uriah.bridges@bilearner.com	Production	07-10-1969	
1	3428	Paula	Small	paula.small@bilearner.com	Production	30-08-1965	
2	3429	Edward	Buck	edward.buck@bilearner.com	Production	06-10-1991	

```
job = LabelEncoder()
label = job.fit_transform(data["JobFunctionDescription"])
label
```

```
array([0, 1, 4, 7, 5, 8, 3, 6, 2, 9, 0, 1, 4, 7, 5, 8, 3, 6, 2, 9, 0, 1,
       4, 7, 5, 8, 3, 6, 2, 9, 0, 1, 4, 7, 5, 8, 3, 6, 2, 9, 0, 1, 4, 7,
       5, 8, 3, 6, 2, 9, 0, 1, 4, 7, 5, 8, 3, 6, 2, 9, 0, 1, 4, 7, 5, 8,
       3, 6, 2, 9, 0, 1, 4, 7, 5, 8, 3, 6, 2, 9, 0, 1, 4, 7, 5, 8, 3, 6,
       2, 9, 0, 1, 4, 7, 5, 8, 3, 6, 2, 9, 0, 1, 4, 7, 5, 8, 3, 6, 2, 9,
       0, 1, 4, 7, 5, 8, 3, 6, 2, 9, 0, 1, 4, 7, 5, 8, 3, 6, 2, 9, 0, 1,
       4, 7, 5, 8, 3, 6, 2, 9, 0, 1, 4, 7, 5, 8, 3, 6, 2])
```

```
print(job.classes_)
```

```
['Accounting' 'Ceo' 'Director' 'Engineer' 'Field Technician' 'Intern'
 'Manager' 'Program Manager' 'Specialist' 'Technician']
```

```
data.drop("JobFunctionDescription", axis=1, inplace=True)
```

```
data["JobFunctionDescription"]=label
data.head(10)
```

	EmpID	FirstName	LastName	ADEmail	DepartmentType	DOB	Gende
0	3427	Uriah	Bridges	uriah.bridges@bilearner.com	Production	07-10-1969	F
1	3428	Paula	Small	paula.small@bilearner.com	Production	30-08-	

```
dept= LabelEncoder()
label = dept.fit_transform(data["DepartmentType"])
label
```

```
array([2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
       3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
       4, 4, 4, 4, 4, 4, 4, 4, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

```
data.drop("DepartmentType", axis=1, inplace=True)
```

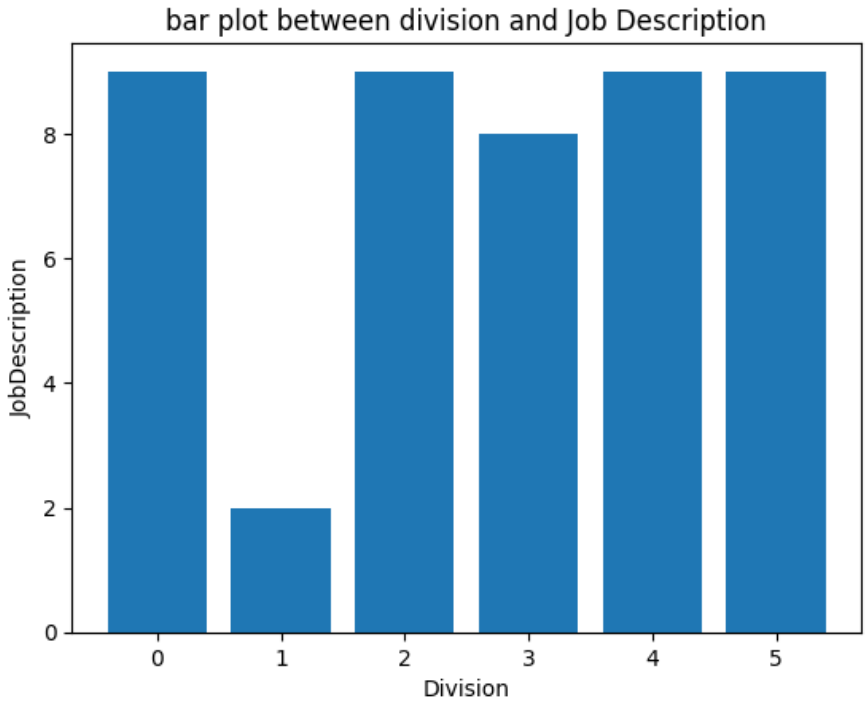
```
data["DepartmentType"]=label
data.head(10)
```

	EmpID	FirstName	LastName	ADEmail	DOB	GenderCode	Division
0	3427	Uriah	Bridges	uriah.bridges@bilearner.com	07-10-1969	Female	2
1	3428	Paula	Small	paula.small@bilearner.com	30-08-1965	Male	2
2	3429	Edward	Buck	edward.buck@bilearner.com	06-10-1991	Male	2
3	3430	Michael	Riordan	michael.riordan@bilearner.com	04-04-1998	Male	2
4	3431	Jasmine	Onque	jasmine.onque@bilearner.com	29-08-1969	Female	2
5	3432	Maruk	Fraval	maruk.fraval@bilearner.com	03-04-1949	Male	2
6	3433	Latia	Costa	latia.costa@bilearner.com	01-07-1942	Female	0
7	3434	Sharlene	Terry	sharlene.terry@bilearner.com	07-03-1957	Female	0
8	3435	Jac	McKinzie	jac.mckinzie@bilearner.com	15-05-1974	Male	0
9	3436	Joseph	Martins	joseph.martins@bilearner.com	11-11-1949	Male	0

```
# Div = list(le.inverse_transform(data['Division']))
Div =le.classes_
Div
```

```
array(['Engineers', 'Field Operations', 'Finance ', 'ProjectManagement',
       'Sales&Marketing', 'Technology'], dtype=object)
```

```
plt.bar(data['Division'],data['JobFunctionDescription'])
plt.title('bar plot between division and Job Description')
plt.xlabel('Division')
plt.ylabel('JobDescription')
plt.show()
```




```
data.columns
```

```
Index(['EmpID', 'FirstName', 'LastName', 'ADEmail', 'DOB', 'GenderCode',
       'Division', 'JobFunctionDescription', 'DepartmentType'],
      dtype='object')
```

```
data.drop('EmpID',axis=1,inplace=True)
```

```
data.head()
```

	FirstName	LastName	ADEmail	DOB	GenderCode	Division	JobFunctionDescription	Depart
0	Uriah	Bridges	uriah.bridges@bilearner.com	07-10-1969	Female	2	0	
1	Paula	Small	paula.small@bilearner.com	30-08-1965	Male	2	1	
2	Edward	Buck	edward.buck@bilearner.com	06-10-1991	Male	2	4	
3	Michael	Riordan	michael.riordan@bilearner.com	04-04-1998	Male	2	7	
4	Jasmine	Onque	jasmine.onque@bilearner.com	29-08-1969	Female	2	5	