Hr_Data_Preprocess

January 23, 2022

HR Dataset Preprocessing

<h4>by</h3>
<h1>Talend Team

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

[2]:
```
df = pd.read_csv(r'C:\Users\souran\Desktop\teland\HR_data.csv')
```

[3]:
```
df.head()
```

[3]:

	First name	Last name	Position	State	Zip	DOB	Sex
0	Brown	Mia	Accountant I	MA	1450	11/24/87	F
1	LaRotonda	William	Accountant I	MA	1460	04/26/84	M
2	Steans	Tyrone	Accountant I	MA	2703	1/9/1986	M
3	Howard	Estelle	Administrative Assistant	MA	2170	09/16/85	F
4	Singh	Nan	Administrative Assistant	MA	2330	05/19/88	F

	MaritalDesc	DateofHire	DateofTermination	TermReason
0	Married	10/27/2008	NaN	N/A - still employed
1	Divorced	6/1/2014	NaN	N/A - still employed
2	Single	9/29/2014	NaN	N/A - still employed
3	Married	2/16/2015	04/15/15	NaN
4	Single	1/5/2015	NaN	N/A - still employed

	EmploymentStatus	Department	ManagerName
0	Active	Admin Offices	Brandon R. LeBlanc
1	Active	Admin Offices	Brandon R. LeBlanc
2	Active	Admin Offices	Brandon R. LeBlanc
3	Terminated for Cause	Admin Offices	Brandon R. LeBlanc
4	Active	Admin Offices	Brandon R. LeBlanc

	RecruitmentSource	PerformanceScore	EngagementSurvey	EmpSatisfaction
0	Diversity Job Fair	Fully Meets	2.04	2
1	Website Banner Ads	Fully Meets	5.00	4

```
2         Internet Search      Fully Meets              3.90                  5
3  Pay Per Click - Google      Fully Meets              3.24                  3
4      Website Banner Ads      Fully Meets              5.00                  3
```

[4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 310 entries, 0 to 309
Data columns (total 18 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   First name          310 non-null    object
 1   Last name           310 non-null    object
 2   Position            310 non-null    object
 3   State               310 non-null    object
 4   Zip                 310 non-null    int64
 5   DOB                 310 non-null    object
 6   Sex                 310 non-null    object
 7   MaritalDesc         310 non-null    object
 8   DateofHire          310 non-null    object
 9   DateofTermination   103 non-null    object
 10  TermReason          309 non-null    object
 11  EmploymentStatus    310 non-null    object
 12  Department          310 non-null    object
 13  ManagerName         310 non-null    object
 14  RecruitmentSource   310 non-null    object
 15  PerformanceScore    310 non-null    object
 16  EngagementSurvey    310 non-null    float64
 17  EmpSatisfaction     310 non-null    int64
dtypes: float64(1), int64(2), object(15)
memory usage: 43.7+ KB
```

[5]:
```
print('Shape of the dataframe is: ' ,df.shape)
```

```
Shape of the dataframe is:  (310, 18)
```

[6]:
```
df.columns
```

[6]:
```
Index(['First name', 'Last name', 'Position', 'State', 'Zip', 'DOB', 'Sex',
       'MaritalDesc', 'DateofHire', 'DateofTermination', 'TermReason',
       'EmploymentStatus', 'Department', 'ManagerName', 'RecruitmentSource',
       'PerformanceScore', 'EngagementSurvey', 'EmpSatisfaction'],
      dtype='object')
```

Removing certain duplicate and unwanted columns from our dataset.

[7]:
```
duplicateRowsDF = df[df.duplicated()]
print("Duplicate Rows except first occurrence based on all columns are :")
print(duplicateRowsDF)
```

```
Duplicate Rows except first occurrence based on all columns are :
Empty DataFrame
Columns: [First name, Last name, Position, State, Zip, DOB, Sex, MaritalDesc,
DateofHire, DateofTermination, TermReason, EmploymentStatus, Department,
ManagerName, RecruitmentSource, PerformanceScore, EngagementSurvey,
EmpSatisfaction]
Index: []
```

There are not duplicate data

There are some attributes like " ZIP, and Date of Termination" we don't need them, I don't think it won't help in our analysis.

[8]:
```
df = df.drop(['Zip','DateofTermination'], axis=1)
```

[9]:
```
df.columns
```

[9]:
```
Index(['First name', 'Last name', 'Position', 'State', 'DOB', 'Sex',
       'MaritalDesc', 'DateofHire', 'TermReason', 'EmploymentStatus',
       'Department', 'ManagerName', 'RecruitmentSource', 'PerformanceScore',
       'EngagementSurvey', 'EmpSatisfaction'],
      dtype='object')
```

[10]:
```
print('Shape of the dataframe after removing certain unnecessary attributes: '␣
 ↪,df.shape)
```

```
Shape of the dataframe after removing certain unnecessary attributes:  (310, 16)
```

[11]:
```
df.describe()
```

[11]:

	EngagementSurvey	EmpSatisfaction
count	310.000000	310.000000
mean	3.332097	3.890323
std	1.290590	0.910690
min	1.030000	1.000000
25%	2.082500	3.000000
50%	3.470000	4.000000
75%	4.520000	5.000000
max	5.000000	5.000000

[ ]:
```
#Create a new col. Age

from datetime import datetime
now = datetime.now()

df['Age'] = (now - df['DOB']).astype('<m8[Y]')
```

[12]:
```
df.isna().sum()
```

[12]:
```
First name           0
Last name            0
Position             0
State                0
DOB                  0
Sex                  0
MaritalDesc          0
DateofHire           0
TermReason           1
EmploymentStatus     0
Department           0
ManagerName          0
RecruitmentSource    0
PerformanceScore     0
EngagementSurvey     0
EmpSatisfaction      0
dtype: int64
```

[13]:
```
df = df.dropna()
```

[14]:
```
df.isna().sum()
```

[14]:
```
First name           0
Last name            0
Position             0
State                0
DOB                  0
Sex                  0
MaritalDesc          0
DateofHire           0
TermReason           0
EmploymentStatus     0
Department           0
ManagerName          0
RecruitmentSource    0
PerformanceScore     0
EngagementSurvey     0
EmpSatisfaction      0
dtype: int64
```

[15]:
```
#Total Departments present in our dataset
department = df.Department.unique()
department
```

[15]:
```
array(['Admin Offices', 'Sales', 'IT/IS', 'Production       ',
       'Executive Office', 'Software Engineering'], dtype=object)
```

[16]:
```
#The record Production contains unnecessary spacing so I'm removing that.
df.Department.replace("Production       ", "Production",inplace=True)
```

[17]:
```
department = df.Department.unique()
department
```

[17]:
```
array(['Admin Offices', 'Sales', 'IT/IS', 'Production',
       'Executive Office', 'Software Engineering'], dtype=object)
```

[18]:
```
#Different Job Positions.
positions = df.Position.unique()
positions.sort()
positions
```

[18]:
```
array(['Accountant I', 'Administrative Assistant', 'Area Sales Manager',
       'BI Developer', 'BI Director', 'CIO', 'Data Analyst',
       'Data Analyst ', 'Data Architect', 'Database Administrator',
       'Director of Operations', 'Director of Sales',
       'Enterprise Architect', 'IT Director', 'IT Manager - DB',
       'IT Manager - Infra', 'IT Manager - Support', 'IT Support',
       'Network Engineer', 'President & CEO', 'Principal Data Architect',
       'Production Manager', 'Production Technician I',
       'Production Technician II', 'Sales Manager', 'Senior BI Developer',
       'Shared Services Manager', 'Software Engineer',
       'Software Engineering Manager', 'Sr. Accountant', 'Sr. DBA',
       'Sr. Network Engineer'], dtype=object)
```

Here is showing two time Data Analyst position in our dataset. Maybe Some of the record contains unnecessary spacing that's why It's showing two times so we can correct this error.

[19]:
```
df.Position.replace("Data Analyst ", "Data Analyst",inplace=True)
positions = df.Position.unique()
positions.sort()
positions
```

[19]:
```
array(['Accountant I', 'Administrative Assistant', 'Area Sales Manager',
       'BI Developer', 'BI Director', 'CIO', 'Data Analyst',
       'Data Architect', 'Database Administrator',
       'Director of Operations', 'Director of Sales',
       'Enterprise Architect', 'IT Director', 'IT Manager - DB',
       'IT Manager - Infra', 'IT Manager - Support', 'IT Support',
       'Network Engineer', 'President & CEO', 'Principal Data Architect',
       'Production Manager', 'Production Technician I',
       'Production Technician II', 'Sales Manager', 'Senior BI Developer',
       'Shared Services Manager', 'Software Engineer',
       'Software Engineering Manager', 'Sr. Accountant', 'Sr. DBA',
       'Sr. Network Engineer'], dtype=object)
```

[20]:
```
#Genrate new csv file after Data Preprocesssing
df.to_csv('preprocess_HR_dataset.csv', index=False)
```

[21]:
```
pwd
```

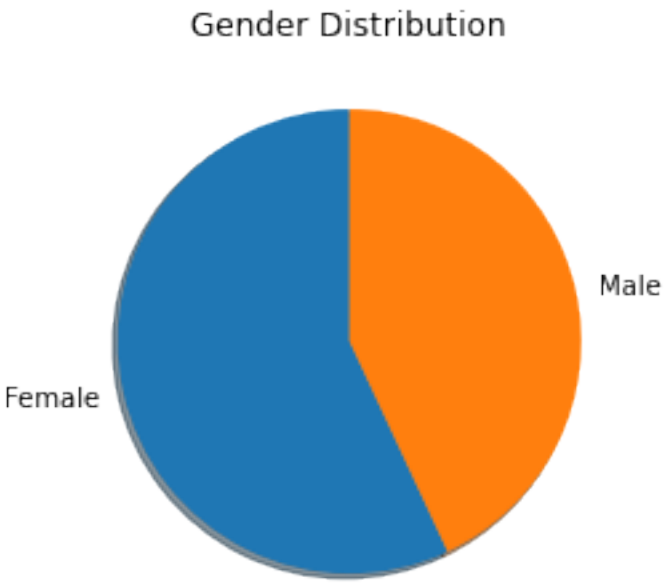
[21]:
```
'C:\\Users\\souran'
```

[22]:
```
recruitment = df.RecruitmentSource.unique()
recruitment
```

[22]:
```
array(['Diversity Job Fair', 'Website Banner Ads', 'Internet Search',
       'Social Networks - Facebook Twitter etc', 'Billboard',
       'Pay Per Click - Google', 'Monster.com', 'Newspager/Magazine',
       'Professional Society', 'Other', 'Employee Referral', 'Indeed',
       'Search Engine - Google Bing Yahoo', 'Glassdoor',
       'Vendor Referral', 'MBTA ads', 'Information Session',
       'Word of Mouth', 'Pay Per Click', 'On-campus Recruiting',
       'On-line Web application', 'Careerbuilder',
       'Company Intranet - Partner'], dtype=object)
```

[23]:
```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

[24]:
```
df.groupby('Sex').DOB.count()
gender = ["Female", "Male"]
plt.pie(df.groupby('Sex').DOB.count(), labels = gender, startangle = 90, shadow␣
↪= True)
plt.title('Gender Distribution');
```


Gender Distribution
Male
Female