import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import pandas as pd

df = pd.read\_csv('D:/Users/shammaab/Desktop/Bootcamp/data1.csv')

df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 19104 entries, 0 to 19103

Data columns (total 13 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 MMM-YY 19104 non-null object

1 Emp\_ID 19104 non-null int64

2 Age 19104 non-null int64

3 Gender 19104 non-null object

4 City 19104 non-null object

5 Education\_Level 19104 non-null object

6 Salary 19104 non-null int64

7 Dateofjoining 19104 non-null object

8 LastWorkingDate 1616 non-null object

9 Joining Designation 19104 non-null int64

10 Designation 19104 non-null int64

11 Total Business Value 19104 non-null int64

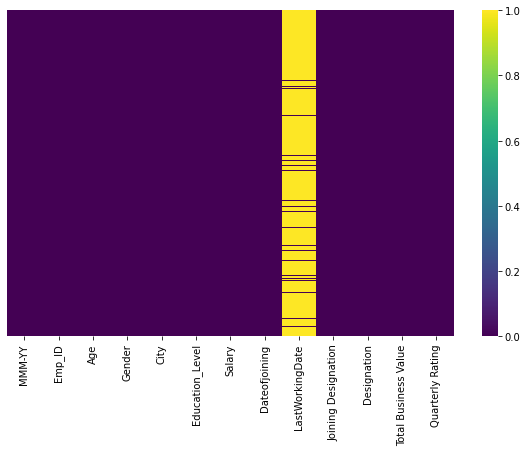
12 Quarterly Rating 19104 non-null int64

dtypes: int64(7), object(6)

memory usage: 1.9+ MB

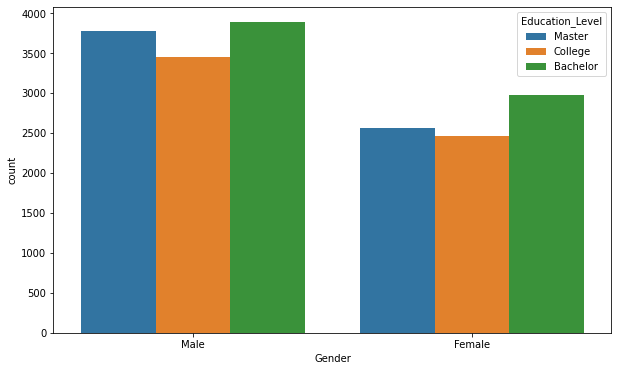
plt.figure(figsize=(10,6))

sns.heatmap(df.isnull(),yticklabels=False,cmap='viridis')

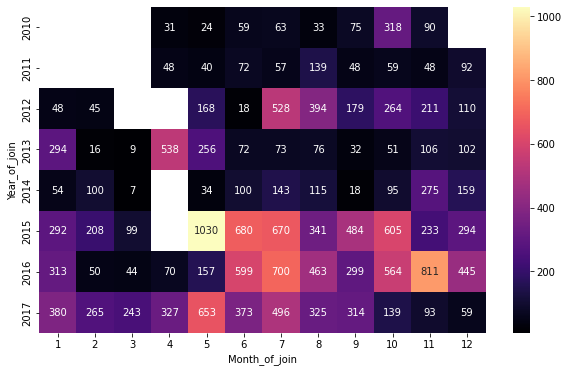


plt.figure(figsize=(10,6))

sns.countplot(data=df,x='Gender',hue='Education\_Level')



joiners = df.groupby(by=['Year\_of\_join','Month\_of\_join']).count()['Emp\_ID'].unstack() linkcode plt.figure(figsize=(10,6)) sns.heatmap(joiners,annot=True,fmt='.4g',cmap='magma')



train.drop(columns=['MMM-YY','Emp\_ID','Gender','City','Education\_Level','Joining Designation','Designation','Year\_of\_leave','Month\_of\_leave','Length\_of\_work'],inplace=True)

train = pd.concat([train,sex,city,edu],axis=1)

x = train.drop('Attrition',axis=1)

y = train['Attrition']

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x,y,test\_size=0.30)

from sklearn.linear\_model import LogisticRegression

logreg = LogisticRegression(solver='liblinear')

logreg.fit(x\_train,y\_train)

LogisticRegression(solver='liblinear')

logreg\_pred = logreg.predict(x\_test)

from sklearn.metrics import classification\_report, confusion\_matrix

viz\_str = '-'\* 20

print(viz\_str,'LOGISTIC REGRESSION',viz\_str)

print('Classification report:')

print(classification\_report(y\_test,logreg\_pred))

print('-'\*61)

plt.title('Confusion Matrix')

sns.heatmap(confusion\_matrix(y\_test,logreg\_pred),annot=True,fmt='g',cmap='cubehelix',cbar=False, yticklabels=False, xticklabels=False)

-------------------- LOGISTIC REGRESSION --------------------

Classification report:

precision recall f1-score support

0 0.92 1.00 0.96 8754

1 0.21 0.01 0.01 798

accuracy 0.91 9552

macro avg 0.56 0.50 0.48 9552

weighted avg 0.86 0.91 0.88 9552

-------------------------------------------------------------

<AxesSubplot:title={'center':'Confusion Matrix'}>

from sklearn.tree import DecisionTreeClassifier

dtree = DecisionTreeClassifier()

dtree.fit(x\_test,y\_test)

DecisionTreeClassifier()

dtree\_pred = dtree.predict(x\_test)

linkcode

viz\_str = '-'\* 20

print(viz\_str,'DECISION TREE',viz\_str)

print('Classification report:')

print(classification\_report(y\_test,dtree\_pred))

print('-'\*61)

plt.title('Confusion Matrix')

sns.heatmap(confusion\_matrix(y\_test,dtree\_pred),annot=True,fmt='g',cmap='cubehelix',cbar=False, yticklabels=False, xticklabels=False)

-------------------- DECISION TREE --------------------

Classification report:

precision recall f1-score support

0 0.97 1.00 0.98 8754

1 1.00 0.64 0.78 798

accuracy 0.97 9552

macro avg 0.98 0.82 0.88 9552

weighted avg 0.97 0.97 0.97 9552

-------------------------------------------------------------

<AxesSubplot:title={'center':'Confusion Matrix'}>

from sklearn.ensemble import RandomForestClassifier

rndfrst = RandomForestClassifier(n\_estimators=100)

rndfrst.fit(x\_train,y\_train)

RandomForestClassifier()

rndfrst\_pred = rndfrst.predict(x\_test)

linkcode

viz\_str = '-'\* 20

print(viz\_str,'RANDOM FOREST',viz\_str)

print('Classification report:')

print(classification\_report(y\_test,rndfrst\_pred))

print('-'\*61)

plt.title('Confusion Matrix')

sns.heatmap(confusion\_matrix(y\_test,rndfrst\_pred),annot=True,fmt='g',cmap='cubehelix',cbar=False, yticklabels=False, xticklabels=False)

-------------------- RANDOM FOREST --------------------

Classification report:

precision recall f1-score support

0 0.92 0.96 0.94 8754

1 0.13 0.06 0.08 798

accuracy 0.89 9552

macro avg 0.52 0.51 0.51 9552

weighted avg 0.85 0.89 0.87 9552

-------------------------------------------------------------

<AxesSubplot:title={'center':'Confusion Matri