retoempleados

May 12, 2024

1 Sección nueva

[192]:
```
!pip install panda
!pip install request
from google.colab import drive
drive.mount('/content/drive')
import numpy as np
import pandas as pd


EmpleadosAttrition0 = pd.read_csv("/content/drive/MyDrive/EmpleadosAttrition.
 ↪csv")

EmpleadosAttrition0

EmpleadosAttrition0= EmpleadosAttrition0.
 ↪drop(['EmployeeCount','EmployeeNumber','Over18','StandardHours'],axis=1)

EmpleadosAttrition0

#se suman los años del empleado
EmpleadosAttrition['HiringDate']
type(EmpleadosAttrition['HiringDate'])
EmpleadosAttrition0['año'] = EmpleadosAttrition0['HiringDate'].str[-4:].
 ↪astype(int)
EmpleadosAttrition0
EmpleadosAttrition0['año'] = 2024 - EmpleadosAttrition0['año']
EmpleadosAttrition0



#Se crea variable year
EmpleadosAttrition0 = EmpleadosAttrition0.rename(columns={'año':'Year'})
EmpleadosAttrition0

#Se crea variable YearsAtCompany
#No entendi como quitarle a 2018 el Year que esta en entero
```

```
EmpleadosAttrition0['YearsAtCompany']=  2018 -␣
 ↪EmpleadosAttrition0['HiringDate'].str[-4:].astype(int)
EmpleadosAttrition0


#Convertir variable DistanceFromHomea a entera
EmpleadosAttrition0['DistanceFromHome_km']=EmpleadosAttrition0['DistanceFromHome'].
 ↪str.split(pat=' ').str[0]
EmpleadosAttrition0



#Quitar variables Year, HiringDate, DistanceFromHome
EmpleadosAttrition0= EmpleadosAttrition0.
 ↪drop(['HiringDate','Year','DistanceFromHome'],axis=1)
EmpleadosAttrition0


#Generar un nuevo frame

SueldoPromedioDepto = EmpleadosAttrition0[['Department','MonthlyIncome']]
SueldoPromedioDepto
SueldoPromedio= SueldoPromedioDepto.groupby(['Department'])[['MonthlyIncome']].
 ↪mean()
SueldoPromedio

#Escalar la variable MonthlyIncome
EmpleadosAttrition0['MonthlyIncomeNorm']=(EmpleadosAttrition0['MonthlyIncome']-min(EmpleadosAt
 ↪(max(EmpleadosAttrition0['MonthlyIncome'])-min(EmpleadosAttrition0['MonthlyIncome']))
EmpleadosAttrition0



#Convertir categoricas en numericas
#No me convierte por el metodo get_dummies(solo muestra False y True)
#LO QUE ARROJA SON ARREGLOS Y AL METERLOS COMO UNA COLUMNA GENERA EN EL FRAME␣
 ↪ORIGINAL SOLO INCLUYE 1

from sklearn.preprocessing import LabelBinarizer
from sklearn.preprocessing import OneHotEncoder
BT=pd.get_dummies(EmpleadosAttrition0.BusinessTravel,prefix='BusinessTravel')
BT

BT1=LabelBinarizer().fit_transform(EmpleadosAttrition0.BusinessTravel)
BT1
```

```
BT2=OneHotEncoder().fit_transform(EmpleadosAttrition0[['BusinessTravel']]).
 ↪toarray()
BT2

DPT=pd.get_dummies(EmpleadosAttrition0.Department,prefix='Department')
DPT

DPT1=LabelBinarizer().fit_transform(EmpleadosAttrition0.Department)
DPT1
DPT2=OneHotEncoder().fit_transform(EmpleadosAttrition0[['BusinessTravel']]).
 ↪toarray()
DPT2


EDU=pd.get_dummies(EmpleadosAttrition0.EducationField,prefix='Education')
EDU

EDU1=LabelBinarizer().fit_transform(EmpleadosAttrition0.EducationField)
EDU1
EDU2=OneHotEncoder().fit_transform(EmpleadosAttrition0[['EducationField']]).
 ↪toarray()
EDU2

GEN=pd.get_dummies(EmpleadosAttrition0.Gender,prefix='Gender')
GEN

GEN1=LabelBinarizer().fit_transform(EmpleadosAttrition0.Gender)
GEN1
GEN2=OneHotEncoder().fit_transform(EmpleadosAttrition0[['Gender']]).toarray()
GEN2

job=pd.get_dummies(EmpleadosAttrition0.JobRole,prefix='Job')
job

job1=LabelBinarizer().fit_transform(EmpleadosAttrition0.JobRole)
job1
job2=OneHotEncoder().fit_transform(EmpleadosAttrition0[['JobRole']]).toarray()
job2

mar=pd.get_dummies(EmpleadosAttrition0.MaritalStatus,prefix='Mar')
mar

mar1=LabelBinarizer().fit_transform(EmpleadosAttrition0.MaritalStatus)
mar1
mar2=OneHotEncoder().fit_transform(EmpleadosAttrition0[['MaritalStatus']]).
 ↪toarray()
mar2
```

```
attri=pd.get_dummies(EmpleadosAttrition0.Attrition,prefix='Attri')
attri

attri1=LabelBinarizer().fit_transform(EmpleadosAttrition0.Attrition)
attri1
attri2=OneHotEncoder().fit_transform(EmpleadosAttrition0[['Attrition']]).
 ↪toarray()
attri2


EmpleadosAttrition0['BusinessTravel']=BT2
EmpleadosAttrition0['Department']=DPT2
EmpleadosAttrition0['EducationField']=EDU2
EmpleadosAttrition0['Gender']=GEN2
EmpleadosAttrition0['JobRole']=job2
EmpleadosAttrition0['MaritalStatus']=mar2
EmpleadosAttrition0['Attrition']=attri2
EmpleadosAttrition0


#calcular correlacion lineal


EmpleadosAttrition0['Age'].corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0['BusinessTravel'].corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0['Department'].corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0['Education'].corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0['EducationField'].corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0['EnvironmentSatisfaction'].
 ↪corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0['Gender'].corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0['JobInvolvement'].corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0['JobLevel'].corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0['JobRole'].corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0['JobSatisfaction'].corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0['MaritalStatus'].corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0['MonthlyIncome'].corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0['NumCompaniesWorked'].corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0['PerformanceRating'].corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0['RelationshipSatisfaction'].
 ↪corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0['StandardHours'].corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0['TotalWorkingYears'].corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0['TrainingTimesLastYear'].
 ↪corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0['WorkLifeBalance'].corr(EmpleadosAttrition0['Attrition'])
```

```
EmpleadosAttrition0['YearsInCurrentRole'].corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0['YearsSinceLastPromotion'].
 ↪corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0['YearsAtCompany'].corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0['DistanceFromHome_km'].
 ↪corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0['MonthlyIncomeNorm'].corr(EmpleadosAttrition0['Attrition'])
EmpleadosAttrition0.corr()


#Seleccionar solo las que tienen arriba de 0.1
#age, environment satisfaction, job involvement,  job level, job satisfaction,␣
 ↪marital status,
# monthly income, total working years, years in current role, years at company,␣
 ↪monthly income norm

EmpleadosAttritionFinal =␣
 ↪EmpleadosAttrition0[['Age','EnvironmentSatisfaction','JobInvolvement','JobLevel','JobSatisf
EmpleadosAttritionFinal

#Crear nueva variable EmpleadosAttritionPCA

from sklearn.decomposition import PCA

EmpleadosAttritionFinal
pca= PCA(4)
pca.fit(EmpleadosAttritionFinal)

print(pca.components_)

print(pca.explained_variance_)

print(pca.explained_variance_ratio_)

EmpleadosAttritionPCA=pca.transform(EmpleadosAttritionFinal)
EmpleadosAttritionPCA

#Agregar CP al frame EmpleadosAttritionFinal
#No entendi como hacerle, porfavor necesito un poco de ayuda aqui

EmpleadosAttritionPCA[:,0]
EmpleadosAttritionPCA[:,1]
EmpleadosAttritionPCA[:,2]
EmpleadosAttritionPCA[:,3]

EmpleadosAttritionFinal = EmpleadosAttritionPCA.assign(CP=['C0','C1','C2''C3'])
```

```
#Guardar datos

EmpleadosAttritionFinal.to_csv("EmpleadosAttritionFinal.csv", index = False,␣
↪sep = ' ')
```