# QUERYS USADAS EN POWER BI QUE INCLUYEN SCRIPTS DE PYTHON

# QUERY CON SCRIPT DF TRANSFORMADO

let

Origen = Python.Execute("import numpy as np#(lf)import pandas as pd#(lf)import matplotlib.pyplot as plt#(lf)#(lf)import os#(lf)import re#(lf)import seaborn as sns#(lf)sns.set\_style('darkgrid')#(lf)import warnings#(lf)warnings.filterwarnings('ignore')#(lf)from sklearn.feature\_extraction.text import CountVectorizer#(lf)from sklearn.feature\_extraction.text import TfidfVectorizer#(lf)#(lf)ruta\_proyecto = 'C:/Users/Usuario/miniconda3/SAI Dia FBS'#(lf)nombre\_fichero\_datos = 'trabajo.csv'#(lf)ruta\_completa = ruta\_proyecto + '/02\_Datos/03\_Trabajo/' + nombre\_fichero\_datos#(lf)df = pd.read\_csv(ruta\_completa)#(lf)df.rename(columns = {'CABECERA':'categoria', 'ID':'id', 'LINK':'link', 'CNO':'cno', 'TITULO OFERTA':'titulo\_oferta', 'DETALLE OFERTA':'detalle\_oferta','FIN OFERTA':'fin\_oferta','EMPRESA OFERTA':'empresa\_oferta' }, inplace=True)#(lf)df.drop(columns = {'Unnamed: 0','Unnamed: 0.1', 'link','fin\_oferta' }, inplace = True)#(lf)df.drop\_duplicates(inplace = True)#(lf)df = df.dropna(subset = ['detalle\_oferta'])#(lf)df = df[df.cno != 'CNO no informado']#(lf)CNO\_elegidos = ['3811', '3820', '2461', '5120'] #'2713',#(lf)df = df[df['cno'].isin(CNO\_elegidos)].reset\_index()#(lf)provincias\_lower = []#(lf)#(lf)provincias=[""Álava"", ""Albacete"", ""Alicante"", ""Almería"", ""Asturias"", ""Ávila"", ""Badajoz"", ""Barcelona"", ""Burgos"", ""Cáceres"", ""Cádiz"", ""Cantabria"", ""Castellón"", ""Ciudad Real"", ""Córdoba"", ""Cuenca"", ""Gerona"", ""Granada"", ""Guadalajara"", ""Guipúzcoa"", ""Huelva"", ""Huesca"", ""Islas Baleares"", ""Jaén"", ""La Coruña"", ""La Rioja"", ""Las Palmas"", ""León"", ""Lérida"", ""Lugo"", ""Madrid"", ""Málaga"", ""Murcia"", ""Navarra"", ""Orense"", ""Palencia"", ""Pontevedra"", ""Salamanca"", ""Santa Cruz de Tenerife"", ""Segovia"", ""Sevilla"", ""Soria"", ""Tarragona"", ""Teruel"", ""Toledo"", ""Valencia"", ""Valladolid"", ""Vizcaya"", ""Zamora"", ""Zaragoza""]#(lf)for i in provincias: #(lf) x = i.lower()#(lf) provincias\_lower.append(x)#(lf)stop\_words\_spanish = ['de',#(lf)'la',#(lf)'que',#(lf)'el',#(lf)'en',#(lf)'y',#(lf)'a',#(lf)'los',#(lf)'del',#(lf)'se',#(lf)'las',#(lf)'por',#(lf)'un',#(lf)'para',#(lf)'con',#(lf)'no',#(lf)'una',#(lf)'su',#(lf)'al',#(lf)'lo',#(lf)'como',#(lf)'más',#(lf)'pero',#(lf)'sus',#(lf)'le',#(lf)'ya',#(lf)'o',#(lf)'este',#(lf)'sí',#(lf)'porque',#(lf)'esta',#(lf)'entre',#(lf)'cuando',#(lf)'muy',#(lf)'sin',#(lf)'sobre',#(lf)'también',#(lf)'me',#(lf)'hasta',#(lf)'hay',#(lf)'donde',#(lf)'quien',#(lf)'desde',#(lf)'todo',#(lf)'nos',#(lf)'durante',#(lf)'todos',#(lf)'uno',#(lf)'les',#(lf)'ni',#(lf)'contra',#(lf)'otros',#(lf)'ese',#(lf)'eso',#(lf)'ante',#(lf)'ellos',#(lf)'e',#(lf)'esto',#(lf)'mí',#(lf)'antes',#(lf)'algunos',#(lf)'qué',#(lf)'unos',#(lf)'yo',#(lf)'otro',#(lf)'otras',#(lf)'otra',#(lf)'él',#(lf)'tanto',#(lf)'esa',#(lf)'estos',#(lf)'mucho',#(lf)'quienes',#(lf)'nada',#(lf)'muchos',#(lf)'cual',#(lf)'poco',#(lf)'ella',#(lf)'estar',#(lf)'estas',#(lf)'algunas',#(lf)'algo',#(lf)'nosotros',#(lf)'mi',#(lf)'mis',#(lf)'tú',#(lf)'te',#(lf)'ti',#(lf)'tu',#(lf)'tus',#(lf)'ellas',#(lf)'nosotras',#(lf)'vosostros',#(lf)'vosostras',#(lf)'os',#(lf)'mío',#(lf)'mía',#(lf)'míos',#(lf)'mías',#(lf)'tuyo',#(lf)'tuya',#(lf)'tuyos',#(lf)'tuyas',#(lf)'suyo',#(lf)'suya',#(lf)'suyos',#(lf)'suyas',#(lf)'nuestro',#(lf)'nuestra',#(lf)'nuestros',#(lf)'nuestras',#(lf)'vuestro',#(lf)'vuestra',#(lf)'vuestros',#(lf)'vuestras',#(lf)'esos',#(lf)'esas',#(lf)'estoy',#(lf)'estás',#(lf)'está',#(lf)'estamos',#(lf)'estáis',#(lf)'están',#(lf)'esté',#(lf)'estés',#(lf)'estemos',#(lf)'estéis',#(lf)'estén',#(lf)'estaré',#(lf)'estarás',#(lf)'estará',#(lf)'estaremos',#(lf)'estaréis',#(lf)'estarán',#(lf)'estaría',#(lf)'estarías',#(lf)'estaríamos',#(lf)'estaríais',#(lf)'estarían',#(lf)'estaba',#(lf)'estabas',#(lf)'estábamos',#(lf)'estabais',#(lf)'estaban',#(lf)'estuve',#(lf)'estuviste',#(lf)'estuvo',#(lf)'estuvimos',#(lf)'estuvisteis',#(lf)'estuvieron',#(lf)'estuviera',#(lf)'estuvieras',#(lf)'estuviéramos',#(lf)'estuvierais',#(lf)'estuvieran',#(lf)'estuviese',#(lf)'estuvieses',#(lf)'estuviésemos',#(lf)'estuvieseis',#(lf)'estuviesen',#(lf)'estando',#(lf)'estado',#(lf)'estada',#(lf)'estados',#(lf)'estadas',#(lf)'estad',#(lf)'he',#(lf)'has',#(lf)'ha',#(lf)'hemos',#(lf)'habéis',#(lf)'han',#(lf)'haya',#(lf)'hayas',#(lf)'hayamos',#(lf)'hayáis',#(lf)'hayan',#(lf)'habré',#(lf)'habrás',#(lf)'habrá',#(lf)'habremos',#(lf)'habréis',#(lf)'habrán',#(lf)'habría',#(lf)'habrías',#(lf)'habríamos',#(lf)'habríais',#(lf)'habrían',#(lf)'había',#(lf)'habías',#(lf)'habíamos',#(lf)'habíais',#(lf)'habían',#(lf)'hube',#(lf)'hubiste',#(lf)'hubo',#(lf)'hubimos',#(lf)'hubisteis',#(lf)'hubieron',#(lf)'hubiera',#(lf)'hubieras',#(lf)'hubiéramos',#(lf)'hubierais',#(lf)'hubieran',#(lf)'hubiese',#(lf)'hubieses',#(lf)'hubiésemos',#(lf)'hubieseis',#(lf)'hubiesen',#(lf)'habiendo',#(lf)'habido',#(lf)'habida',#(lf)'habidos',#(lf)'habidas',#(lf)'soy',#(lf)'eres',#(lf)'es',#(lf)'somos',#(lf)'sois',#(lf)'son',#(lf)'sea',#(lf)'seas',#(lf)'seamos',#(lf)'seáis',#(lf)'sean',#(lf)'seré',#(lf)'serás',#(lf)'será',#(lf)'seremos',#(lf)'seréis',#(lf)'serán',#(lf)'sería',#(lf)'serías',#(lf)'seríamos',#(lf)'seríais',#(lf)'serían',#(lf)'era',#(lf)'eras',#(lf)'éramos',#(lf)'erais',#(lf)'eran',#(lf)'fui',#(lf)'fuiste',#(lf)'fue',#(lf)'fuimos',#(lf)'fuisteis',#(lf)'fueron',#(lf)'fuera',#(lf)'fueras',#(lf)'fuéramos',#(lf)'fuerais',#(lf)'fueran',#(lf)'fuese',#(lf)'fueses',#(lf)'fuésemos',#(lf)'fueseis',#(lf)'fuesen',#(lf)'sintiendo',#(lf)'sentido',#(lf)'sentida',#(lf)'sentidos',#(lf)'sentidas',#(lf)'siente',#(lf)'sentid',#(lf)'tengo',#(lf)'tienes',#(lf)'tiene',#(lf)'tenemos',#(lf)'tenéis',#(lf)'tienen',#(lf)'tenga',#(lf)'tengas',#(lf)'tengamos',#(lf)'tengáis',#(lf)'tengan',#(lf)'tendré',#(lf)'tendrás',#(lf)'tendrá',#(lf)'tendremos',#(lf)'tendréis',#(lf)'tendrán',#(lf)'tendría',#(lf)'tendrías',#(lf)'tendríamos',#(lf)'tendríais',#(lf)'tendrían',#(lf)'tenía',#(lf)'tenías',#(lf)'teníamos',#(lf)'teníais',#(lf)'tenían',#(lf)'tuve',#(lf)'tuviste',#(lf)'tuvo',#(lf)'tuvimos',#(lf)'tuvisteis',#(lf)'tuvieron',#(lf)'tuviera',#(lf)'tuvieras',#(lf)'tuviéramos',#(lf)'tuvierais',#(lf)'tuvieran',#(lf)'tuviese',#(lf)'tuvieses',#(lf)'tuviésemos',#(lf)'tuvieseis',#(lf)'tuviesen',#(lf)'teniendo',#(lf)'tenido',#(lf)'tenida',#(lf)'tenidos',#(lf)'tenidas',#(lf)'tened']#(lf)# Creamos la función de limpoieza de textos#(lf)#(lf)#(lf)def clean(text): #(lf) # Quitar las expresiones entre corchetes #(lf) text = re.sub(r'\[.+?\]', ' ', str(text))#(lf) # Quita las aclaraciones entre paréntesis#(lf) text = re.sub(r'\(.+?\)', ' ', text)#(lf) # Quitar acentos#(lf) # text = unidecode.unidecode(text)#(lf) # Quitar aquello que no sean palabras o cosa que se le parezca#(lf) text = re.sub(r'\W', ' ', text)#(lf) # Quitar espacios extra en caso de haber#(lf) text = re.sub(r'\s+', ' ', text)#(lf) # Quitar un conjunto de carácteres que son intelegibles y se repiten a lo largo de los campos#(lf) text = re.sub('\r\n', ' ', text)#(lf) # Pasa el texto a minúsulas#(lf) text = text.lower()#(lf) # Quita palabras #(lf) text = re.sub(r'se busca', ' ', text)#(lf) # Quita palabras #(lf) text = re.sub(r'se necesita', ' ', text)#(lf) # Retira stopwords#(lf) text = text.split()#(lf) text = [ word for word in text if word not in stop\_words\_spanish] #stopwords.words('spanish')]#(lf) text = ' '.join(text)#(lf) return text#(lf)# Realizamos la limpieza#(lf)df['titulo\_oferta'] = df['titulo\_oferta'].apply(lambda x: clean(x))#(lf)# Creamos un listado de todas la palabras utilizadas para la oferta en una lista#(lf)words\_list = []#(lf)for i in range(0, len(df)):#(lf) text = df['titulo\_oferta'].reset\_index().iloc[i, 1]#(lf) for word in text.split():#(lf) words\_list.append(word)#(lf)words\_list#(lf)# Creamos un diccionario de las palabras y su frecuencia.#(lf)#(lf)def listaPalabrasDicFrec(listaPalabras):#(lf) frecuenciaPalab = [listaPalabras.count(p) for p in listaPalabras]#(lf) return dict(list(zip(listaPalabras,frecuenciaPalab)))#(lf)#(lf)# Ordemanos el diccionario por la frecuencia de la palabra#(lf)def ordenaDicFrec(dicfrec):#(lf) aux = [(dicfrec[key], key) for key in dicfrec]#(lf) aux.sort()#(lf) aux.reverse()#(lf) return aux#(lf)def remove\_provincia(text):#(lf) text = text.split()#(lf) text = [ word for word in text if word not in provincias\_lower]#(lf) text = ' '.join(text)#(lf) return text#(lf)# Realizamos la limpieza#(lf)#(lf)df['titulo\_oferta'] = df['titulo\_oferta'].apply(lambda x: remove\_provincia(x))#(lf)# Creamos un listado de todas la palabras utilizadas para la oferta en una lista#(lf)words\_list = []#(lf)for i in range(0, len(df)):#(lf) text = df['titulo\_oferta'].reset\_index().iloc[i, 1]#(lf) for word in text.split():#(lf) words\_list.append(word)#(lf)words\_list#(lf)df['detalle\_oferta'] = df['detalle\_oferta'].apply(lambda x: clean(x))#(lf)# Creamos el corpus de todas la palabras utilizadas para la oferta en una lista#(lf)words\_list = []#(lf)for i in range(0, len(df)):#(lf) text = df['detalle\_oferta'].reset\_index().iloc[i, 1]#(lf) for word in text.split():#(lf) words\_list.append(word)#(lf)df['categoria'] = df['categoria'].fillna(""0"")#(lf)df['categoria'] = df['categoria'].apply(lambda x: clean(x))#(lf)# Creamos el corpus de todas la palabras utilizadas para la oferta en una lista#(lf)words\_list = []#(lf)for i in range(0, len(df)):#(lf) text = df['categoria'].reset\_index().iloc[i, 1]#(lf) for word in text.split():#(lf) words\_list.append(word)#(lf)#(lf)"),

#"Se expandió Value" = Table.ExpandTableColumn(Origen, "Value", {"id", "cno", "titulo\_oferta", "detalle\_oferta", "empresa\_oferta", "categoria"}, {"id", "cno", "titulo\_oferta", "detalle\_oferta", "empresa\_oferta", "categoria"}),

#"Tipo cambiado" = Table.TransformColumnTypes(#"Se expandió Value",{{"Name", type text}, {"id", Int64.Type}, {"cno", Int64.Type}, {"titulo\_oferta", type text}, {"detalle\_oferta", type text}, {"empresa\_oferta", type text}, {"categoria", type text}}),

#"Columnas quitadas" = Table.RemoveColumns(#"Tipo cambiado",{"Name"}),

#"Tipo cambiado1" = Table.TransformColumnTypes(#"Columnas quitadas",{{"id", type text}, {"cno", type text}, {"titulo\_oferta", type text}, {"detalle\_oferta", type text}, {"empresa\_oferta", type text}, {"categoria", type text}})

in

#"Tipo cambiado1"

# QUERY CON SCRIPT MODELO

let

Origen1 = Excel.Workbook(Web.Contents("https://iddeas-my.sharepoint.com/personal/manuel\_ramon\_iddeass\_com/Documents/Formaci%C3%B3n/SAI/Input%20Perfil%20demandante%20empleo.xlsx"), null, true),

InputPerfil\_Table = Origen1{[Item="InputPerfil",Kind="Table"]}[Data],

#"Tipo cambiado" = Table.TransformColumnTypes(InputPerfil\_Table,{{"INTRODUCE TU PERFIL", type text}, {"\_\_PowerAppsId\_\_", type text}}),

#"Conservar filas superiores" = Table.FirstN(#"Tipo cambiado",1),

#"Columnas quitadas" = Table.RemoveColumns(#"Conservar filas superiores",{"\_\_PowerAppsId\_\_"}),

#"Columnas con nombre cambiado" = Table.RenameColumns(#"Columnas quitadas",{{"INTRODUCE TU PERFIL", "column1"}}),

PerfilTexto = #"Columnas con nombre cambiado"{0}[column1],

// Inicio Python

PythonScript =Python.Execute("import numpy as np#(lf)import pandas as pd#(lf)import matplotlib.pyplot as plt#(lf)#(lf)from sklearn.model\_selection import train\_test\_split#(lf)from sklearn.multiclass import OneVsRestClassifier#(lf)from sklearn.ensemble import GradientBoostingClassifier#(lf)from sklearn import metrics#(lf)from sklearn.metrics.pairwise import cosine\_similarity#(lf)#(lf)from sklearn.metrics import roc\_auc\_score#(lf)from sklearn.metrics import confusion\_matrix#(lf)from sklearn.metrics import plot\_confusion\_matrix#(lf)from sklearn.metrics import classification\_report#(lf)#(lf)from sklearn.feature\_extraction.text import TfidfVectorizer#(lf)from sklearn.metrics import plot\_precision\_recall\_curve#(lf)#(lf)import os#(lf)import re#(lf)import seaborn as sns#(lf)sns.set\_style('darkgrid')#(lf)import warnings#(lf)warnings.filterwarnings('ignore')#(lf)#(lf)ruta\_proyecto = 'C:/Users/Usuario/OneDrive/Documentos/Proyecto Saturdays AI/SAI Dia FBS/'#(lf)nombre = 'df\_transf.pickle'#(lf)df = pd.read\_pickle(ruta\_proyecto + '/02\_Datos/03\_Trabajo/' + nombre)#(lf)#(lf)# Vectorizar los textos. #(lf)#(lf)Text = df['detalle\_oferta'].values#(lf)Target = df['cno'].values#(lf)word\_vectorizer = TfidfVectorizer(#(lf) sublinear\_tf=True,#(lf) #stop\_words='english',#(lf) max\_features=1500)#(lf)word\_vectorizer.fit(Text)#(lf)WordFeatures = word\_vectorizer.transform(Text)#(lf)#(lf)# Realizamos el la separación nuestro conjunto de datos en Train (80%) y Test(20%)#(lf)train\_x, test\_x, train\_y, test\_y = train\_test\_split(WordFeatures, Target, random\_state=24, test\_size=0.2)#(lf)model\_GBM = OneVsRestClassifier(GradientBoostingClassifier())#(lf)model\_GBM.fit(train\_x, train\_y)#(lf)#(lf)# Realizamos la prediccón para los datos de prueba#(lf)pred\_y\_GBM = model\_GBM.predict(test\_x)#(lf)# Utilizamos la función de limpoieza de textos#(lf)stop\_words\_spanish = ['de',#(lf)'la',#(lf)'que',#(lf)'el',#(lf)'en',#(lf)'y',#(lf)'a',#(lf)'los',#(lf)'del',#(lf)'se',#(lf)'las',#(lf)'por',#(lf)'un',#(lf)'para',#(lf)'con',#(lf)'no',#(lf)'una',#(lf)'su',#(lf)'al',#(lf)'lo',#(lf)'como',#(lf)'más',#(lf)'pero',#(lf)'sus',#(lf)'le',#(lf)'ya',#(lf)'o',#(lf)'este',#(lf)'sí',#(lf)'porque',#(lf)'esta',#(lf)'entre',#(lf)'cuando',#(lf)'muy',#(lf)'sin',#(lf)'sobre',#(lf)'también',#(lf)'me',#(lf)'hasta',#(lf)'hay',#(lf)'donde',#(lf)'quien',#(lf)'desde',#(lf)'todo',#(lf)'nos',#(lf)'durante',#(lf)'todos',#(lf)'uno',#(lf)'les',#(lf)'ni',#(lf)'contra',#(lf)'otros',#(lf)'ese',#(lf)'eso',#(lf)'ante',#(lf)'ellos',#(lf)'e',#(lf)'esto',#(lf)'mí',#(lf)'antes',#(lf)'algunos',#(lf)'qué',#(lf)'unos',#(lf)'yo',#(lf)'otro',#(lf)'otras',#(lf)'otra',#(lf)'él',#(lf)'tanto',#(lf)'esa',#(lf)'estos',#(lf)'mucho',#(lf)'quienes',#(lf)'nada',#(lf)'muchos',#(lf)'cual',#(lf)'poco',#(lf)'ella',#(lf)'estar',#(lf)'estas',#(lf)'algunas',#(lf)'algo',#(lf)'nosotros',#(lf)'mi',#(lf)'mis',#(lf)'tú',#(lf)'te',#(lf)'ti',#(lf)'tu',#(lf)'tus',#(lf)'ellas',#(lf)'nosotras',#(lf)'vosostros',#(lf)'vosostras',#(lf)'os',#(lf)'mío',#(lf)'mía',#(lf)'míos',#(lf)'mías',#(lf)'tuyo',#(lf)'tuya',#(lf)'tuyos',#(lf)'tuyas',#(lf)'suyo',#(lf)'suya',#(lf)'suyos',#(lf)'suyas',#(lf)'nuestro',#(lf)'nuestra',#(lf)'nuestros',#(lf)'nuestras',#(lf)'vuestro',#(lf)'vuestra',#(lf)'vuestros',#(lf)'vuestras',#(lf)'esos',#(lf)'esas',#(lf)'estoy',#(lf)'estás',#(lf)'está',#(lf)'estamos',#(lf)'estáis',#(lf)'están',#(lf)'esté',#(lf)'estés',#(lf)'estemos',#(lf)'estéis',#(lf)'estén',#(lf)'estaré',#(lf)'estarás',#(lf)'estará',#(lf)'estaremos',#(lf)'estaréis',#(lf)'estarán',#(lf)'estaría',#(lf)'estarías',#(lf)'estaríamos',#(lf)'estaríais',#(lf)'estarían',#(lf)'estaba',#(lf)'estabas',#(lf)'estábamos',#(lf)'estabais',#(lf)'estaban',#(lf)'estuve',#(lf)'estuviste',#(lf)'estuvo',#(lf)'estuvimos',#(lf)'estuvisteis',#(lf)'estuvieron',#(lf)'estuviera',#(lf)'estuvieras',#(lf)'estuviéramos',#(lf)'estuvierais',#(lf)'estuvieran',#(lf)'estuviese',#(lf)'estuvieses',#(lf)'estuviésemos',#(lf)'estuvieseis',#(lf)'estuviesen',#(lf)'estando',#(lf)'estado',#(lf)'estada',#(lf)'estados',#(lf)'estadas',#(lf)'estad',#(lf)'he',#(lf)'has',#(lf)'ha',#(lf)'hemos',#(lf)'habéis',#(lf)'han',#(lf)'haya',#(lf)'hayas',#(lf)'hayamos',#(lf)'hayáis',#(lf)'hayan',#(lf)'habré',#(lf)'habrás',#(lf)'habrá',#(lf)'habremos',#(lf)'habréis',#(lf)'habrán',#(lf)'habría',#(lf)'habrías',#(lf)'habríamos',#(lf)'habríais',#(lf)'habrían',#(lf)'había',#(lf)'habías',#(lf)'habíamos',#(lf)'habíais',#(lf)'habían',#(lf)'hube',#(lf)'hubiste',#(lf)'hubo',#(lf)'hubimos',#(lf)'hubisteis',#(lf)'hubieron',#(lf)'hubiera',#(lf)'hubieras',#(lf)'hubiéramos',#(lf)'hubierais',#(lf)'hubieran',#(lf)'hubiese',#(lf)'hubieses',#(lf)'hubiésemos',#(lf)'hubieseis',#(lf)'hubiesen',#(lf)'habiendo',#(lf)'habido',#(lf)'habida',#(lf)'habidos',#(lf)'habidas',#(lf)'soy',#(lf)'eres',#(lf)'es',#(lf)'somos',#(lf)'sois',#(lf)'son',#(lf)'sea',#(lf)'seas',#(lf)'seamos',#(lf)'seáis',#(lf)'sean',#(lf)'seré',#(lf)'serás',#(lf)'será',#(lf)'seremos',#(lf)'seréis',#(lf)'serán',#(lf)'sería',#(lf)'serías',#(lf)'seríamos',#(lf)'seríais',#(lf)'serían',#(lf)'era',#(lf)'eras',#(lf)'éramos',#(lf)'erais',#(lf)'eran',#(lf)'fui',#(lf)'fuiste',#(lf)'fue',#(lf)'fuimos',#(lf)'fuisteis',#(lf)'fueron',#(lf)'fuera',#(lf)'fueras',#(lf)'fuéramos',#(lf)'fuerais',#(lf)'fueran',#(lf)'fuese',#(lf)'fueses',#(lf)'fuésemos',#(lf)'fueseis',#(lf)'fuesen',#(lf)'sintiendo',#(lf)'sentido',#(lf)'sentida',#(lf)'sentidos',#(lf)'sentidas',#(lf)'siente',#(lf)'sentid',#(lf)'tengo',#(lf)'tienes',#(lf)'tiene',#(lf)'tenemos',#(lf)'tenéis',#(lf)'tienen',#(lf)'tenga',#(lf)'tengas',#(lf)'tengamos',#(lf)'tengáis',#(lf)'tengan',#(lf)'tendré',#(lf)'tendrás',#(lf)'tendrá',#(lf)'tendremos',#(lf)'tendréis',#(lf)'tendrán',#(lf)'tendría',#(lf)'tendrías',#(lf)'tendríamos',#(lf)'tendríais',#(lf)'tendrían',#(lf)'tenía',#(lf)'tenías',#(lf)'teníamos',#(lf)'teníais',#(lf)'tenían',#(lf)'tuve',#(lf)'tuviste',#(lf)'tuvo',#(lf)'tuvimos',#(lf)'tuvisteis',#(lf)'tuvieron',#(lf)'tuviera',#(lf)'tuvieras',#(lf)'tuviéramos',#(lf)'tuvierais',#(lf)'tuvieran',#(lf)'tuviese',#(lf)'tuvieses',#(lf)'tuviésemos',#(lf)'tuvieseis',#(lf)'tuviesen',#(lf)'teniendo',#(lf)'tenido',#(lf)'tenida',#(lf)'tenidos',#(lf)'tenidas',#(lf)'tened']#(lf)#(lf)#(lf)def clean(text): #(lf) # Quitar las expresiones entre corchetes #(lf) text = re.sub(r'\[.+?\]', ' ', str(text))#(lf) # Quita las aclaraciones entre paréntesis#(lf) text = re.sub(r'\(.+?\)', ' ', text)#(lf) # Quitar acentos#(lf) # text = unidecode.unidecode(text)#(lf) # Quitar aquello que no sean palabras o cosa que se le parezca#(lf) text = re.sub(r'\W', ' ', text)#(lf) # Quitar espacios extra en caso de haber#(lf) text = re.sub(r'\s+', ' ', text)#(lf) # Quitar un conjunto de carácteres que son intelegibles y se repiten a lo largo de los campos#(lf) text = re.sub('\r\n', ' ', text)#(lf) # Pasa el texto a minúsulas#(lf) text = text.lower()#(lf) # Quita palabras #(lf) text = re.sub(r'se busca', ' ', text)#(lf) # Quita palabras #(lf) text = re.sub(r'se necesita', ' ', text)#(lf) # Retira stopwords#(lf) text = text.split()#(lf) text = [ word for word in text if word not in stop\_words\_spanish ]# stopwords.words('spanish')]#(lf) text = ' '.join(text)#(lf) return text#(lf)#(lf)text = """"""

"

& PerfilTexto &

"

""""""#(lf)text = [clean(text)]#(lf)WordF = word\_vectorizer.transform(text)#(lf)model\_GBM.predict(WordF)#(lf)prediccion = pd.DataFrame(model\_GBM.predict(WordF))#(lf)"),

prediccion = PythonScript{[Name="prediccion"]}[Value],

#"Tipo cambiado1" = Table.TransformColumnTypes(prediccion,{{"0", type text}}),

#"Columnas con nombre cambiado1" = Table.RenameColumns(#"Tipo cambiado1",{{"0", "Code"}})

in

#"Columnas con nombre cambiado1"