Top of Form

Machine Learning con Power BI Desktop...gratis!

[Marilin Vega León](https://blog.spainbs.com/autor/marilin-vega-leon)

jueves, 23 de julio de 2020

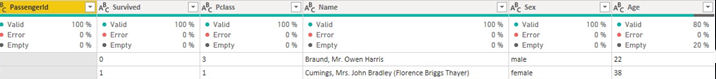
Son básicamente 3 tipos de transformaciones:

* **Análisis de texto (text analytics)**: que utiliza la API de servicios cognitivos.
* **Visión (vision)**: también utiliza la API de servicios cognitivos.
* **Aprendizaje automático (azure machine learning)**: expone sus funciones / modelos de Azure Machine Learning.

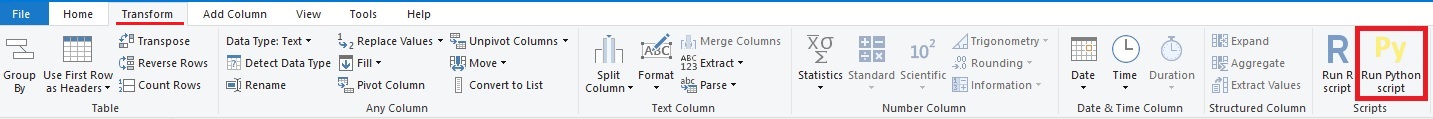
Sucede que un requisito súper importante para usar cualquiera de ellas es que necesitas tener una capacidad Premium en uno de sus espacios de trabajo para poder utilizarlas.  
  
Pero ¿qué tal si les digo que esta capacidad la podemos obtener de manera gratuita con Python?  
  
Sí señores, podemos hacer *Machine learning* de manera relativamente sencilla con Python en Power BI y sin necesidad de ser premium.  
  
Python con Power BI va mucho más allá del proceso de ETL y está creando una gran adición a la familia Power BI, al proporcionarle la capacidad de realizar importación, transformación de datos rápida o crear visualizaciones de datos interesantes. Pero va mucho más allá, e incluso puede expandir aún más sus informes al incorporar aprendizaje automático e IA.  
  
En este post veremos un ejemplo sencillo de cómo podemos hacer *Machine Learning* usando este lenguaje tan popular.  
  
Partiremos del uso de un Dataset muy conocido. El dataset del titanic. Intentaremos predecir la supervivencia y probabilidades de sobrevivir en este catastrófico suceso.  
   
Realizaremos un ejemplo sencillo de aprendizaje supervisado y clasificación binaria. (1 – pasajero sobrevive / 0 – pasajero muere)  
Antes de comenzar debemos asegurarnos de que el conjunto de prueba reúna las siguientes dos condiciones:

* Que sea lo suficientemente grande como para generar resultados significativos desde el punto de vista estadístico.
* Que sea representativo de todo el conjunto de datos. En otras palabras, no elijas un conjunto de prueba con características diferentes al del conjunto de entrenamiento.

Finalmente, si el conjunto de prueba reúne estas dos condiciones, tu objetivo es crear un modelo que generalice los datos nuevos de forma correcta.  
  
**Nota:** Nunca uses los datos de prueba para el entrenamiento. Si ves resultados sorpresivamente positivos en tus métricas de evaluación, es posible que estés usando los datos de prueba para el entrenamiento. Por ejemplo, tener una precisión muy alta puede ser un indicativo de que se filtraron datos de prueba en los de entrenamiento.  
Analicemos los datos, (ver Fig.1)

  
  
Tenemos 1309 registros y 12 características, incluyendo nuestra variable objetivo (Survived).  
Las variables que me servirán para predecir en el modelo.

* SibSp - # de hermanos / cónyuges a bordo del Titanic
* Parch - # de padres / hijos a bordo del Titanic
* Embarked - Puerto de embarque
* Fare - Tarifa de pasajero
* Age – Edad
* Sex – Sexo
* Pclass – Clase
* Survived - Sobrevivió (1 sobrevive - 0 no sobrevived

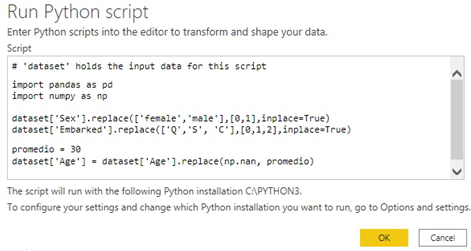
Vemos que nuestros datos son una combinación de categorías y números. Lo que significa, que deberemos codificar los datos categóricos para presentarlos como números, con el objetivo de que nuestro modelo de aprendizaje automático pueda trabajar correctamente con ellos.  
  
**Transformaciones**  
1- Elimino la columna Cabin porque tiene muchos datos vacíos y no me es significativa para el análisis.  
2- Las transformaciones que haremos a continuación son realizadas mediante script de Python (ver Fig.2)   
  
  
  
Se abrirá una ventana (ver Fig.3) donde copiaremos el siguiente código.

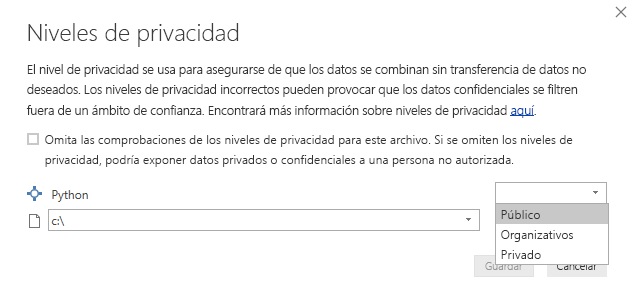
# se importan las librerias necesarias  
import pandas as pd  
import numpy as np  
   
# se reemplaza el sexo por femenino(0), masculino(1)  
dataset['Sex'].replace(['female','male'],[0,1],inplace=True)

# se reemplaza la puerta de embarque por números Q(0), S(1), C(2)  
dataset['Embarked'].replace(['Q','S', 'C'],[0,1,2],inplace=True)

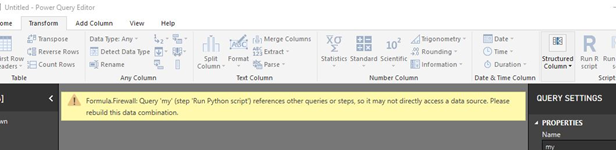
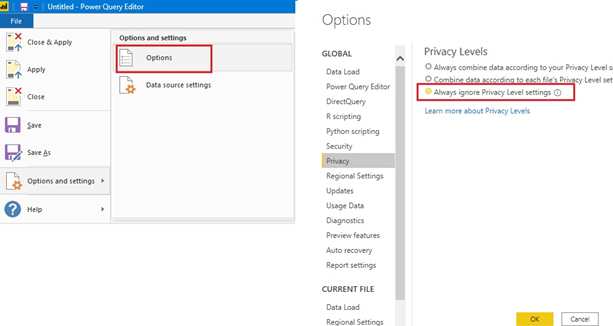
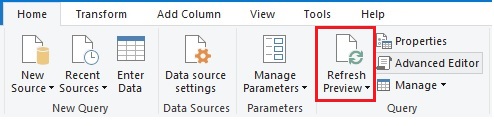
# se calcula la media de la edad  
promedio = dataset['Age'].mean()

# se inserta la media de la edad en los datos vacíos de la columna  
dataset['Age'] = dataset['Age'].replace(np.nan, promedio)

  
3- Después de seleccionar OK (Aceptar), el Editor de consultas muestra una advertencia sobre la privacidad de datos.

Para que los scripts de Python funcionen correctamente en el servicio Power BI, todos los orígenes de datos se deben establecer como públicos (ver Fig.4).  


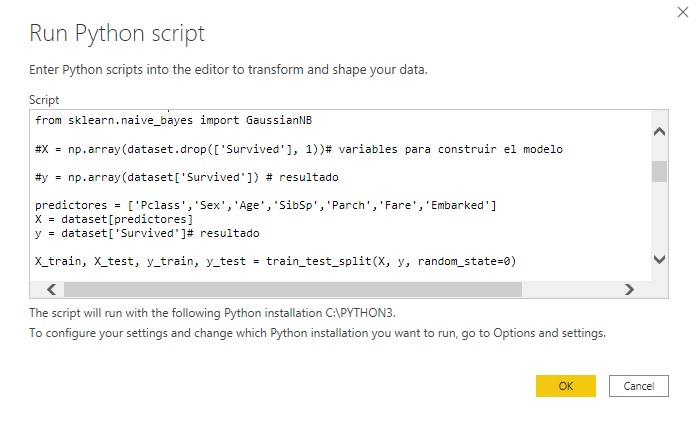
**Nota:** No en todos los casos funciona de igual manera. En el caso, por ejemplo, de que los datos cargados provengan de un conector de carpeta y se hayan fusionado al menos dos archivos es posible que no te pregunte por el nivel de privacidad. En ese caso te dará un error (Formula.Firewall error) (ver Fig.5). Si esto ocurre, debemos ir a la configuración e indicarlo manualmente (ver Fig.6) y por último refrescar (ver Fig.7).

  
  
  
  
  
  
4- Seguidamente, ejecutaremos otro script de Python para agrupar la edad por intervalos y que sea mejor el análisis.  
  
import pandas as pd  
import numpy as np

# bandas de edades  
bins = [0, 8, 15, 18, 25, 40, 60, 100]

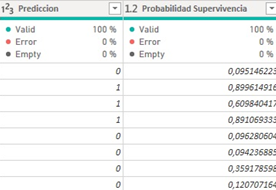
# Creo varios grupos de acuerdo a bandas de las edades  
names = ['1', '2', '3', '4', '5', '6', '7']

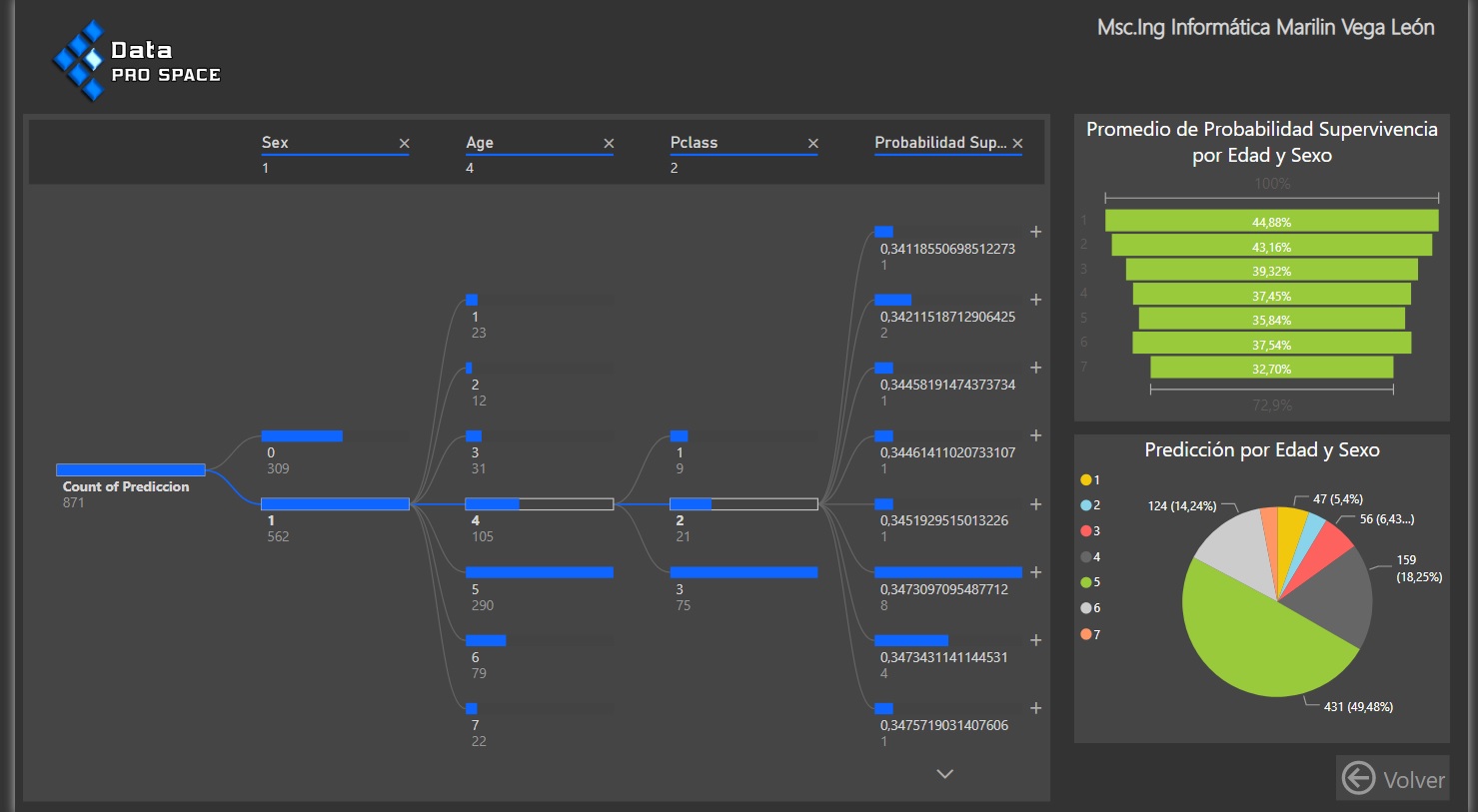
# Sustituyo la edad por el grupo al que pertenece  
dataset['Age'] = pd.cut(dataset['Age'], bins, labels = names)  
   
Nota: Después de ejecutar se cambia tipo de dato, verifique.  
  
**Aplicar algoritmos de Machine Learning**  
En el mundo del *Machine Learning* no existe un algoritmo único que funcione siempre mejor que los demás para resolver un problema. Por eso que he hecho pruebas con distintos algoritmos, tales como: Regresión logística, Vecinos más cercanos, Árboles de decisión, Naibe Bayes Gaussiano.  
  
# Se importan las librerias necesarias  
import matplotlib.pyplot as plt  
import numpy as np  
import pandas as pd  
from sklearn.model\_selection import train\_test\_split  
from sklearn.linear\_model import LogisticRegression  
from sklearn.svm import SVC  
from sklearn.neighbors import KNeighborsClassifier  
from sklearn.tree import DecisionTreeClassifier  
from sklearn.naive\_bayes import GaussianNB  
   
# variables para construir el modelo, sirven para predecir el resultado  
predictores = ['Pclass','Sex','Age','SibSp','Parch','Fare','Embarked']  
X = dataset[predictores]  
   
# resultado variable objetivo  
y = dataset['Survived']  
   
#Separo los datos de "train" en entrenamiento y prueba para probar los #algoritm#os  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, random\_state=0)  
   
#Regresión logística  
logreg = LogisticRegression()  
logreg.fit(X\_train, y\_train)  
print('Precisión Regresión Logística set train: {:.2f}'.format(logreg.score(X\_train, y\_train)))  
print('Precisión Regresión Logística set test: {:.2f}'.format(logreg.score(X\_test, y\_test)))  
   
#K neighbors  
knn = KNeighborsClassifier(n\_neighbors = 9)  
knn.fit(X\_train, y\_train)  
print('Precisión K-NN(Vecinos más cercanos) set train: {:.2f}'.format(knn.score(X\_train, y\_train)))  
print('Precisión K-NN(Vecinos más cercanos) set test: {:.2f}'.format(knn.score(X\_test, y\_test)))  
   
#Arboles de decisión  
arb = DecisionTreeClassifier()  
arb.fit(X\_train, y\_train)  
print('Precisión Arboles de decisión set train: {:.2f}'.format(arb.score(X\_train, y\_train)))  
print('Precisión Arboles de decisión set test: {:.2f}'.format(arb.score(X\_test, y\_test)))  
   
#Naibe Bayes Gaussiano  
gau = GaussianNB()  
gau.fit(X\_train, y\_train)  
print('Precisión Naibe Bayes Gaussiano set train: {:.2f}'.format(gau.score(X\_train, y\_train)))  
print('Precisión Naibe Bayes Gaussiano set test: {:.2f}'.format(gau.score(X\_test, y\_test)))  
   
#Ya teniendo nuestros modelos procedemos a realizar la predicción

#Selecciono el modelo de regresión logística  
   
y\_pred = logreg.predict(X) #se guarda la predicción  
y\_prob = logreg.predict\_proba(X) #se guarda la probabilidad  
   
#añadir los resultados en dos nuevas columnas (predicción, #probabilidad)  
dataset['Prediccion'] = y\_pred  
dataset['Probabilidad Supervivencia'] = y\_prob[:,1]  
   
Aunque no sea el algoritmo que más precisión nos otorga en la resolución de nuestro problema, he decidido usar el de la **Regresión Logística**. Este algoritmo es uno de los más utilizados en la resolución de problemas de clasificación pues nos devuelve la probabilidad como valor entre 0 y 1.  
  
Dicho algoritmo es extremadamente útil en casos como este, ya que no solo podremos saber la probabilidad de supervivencia, sino por el contrario, la probabilidad de que muera la persona.  
  
2-Una vez tenemos todo el código listo para ejecutar, lo pegamos en la ventana de Scripts de Python del editor de consultas y ejecutamos (ver Fig.8).  
  
**Nota:** Recomiendo crear scripts independientes del código que se va a utilizar para probar y depurar cualquier problema antes de usarlo en Power BI.

En mi caso uso Visual Studio Code, pero también podemos usar Notebook, Pysharms, Visual Studio

3-Finalmente obtendremos la información deseada como se muestra en la siguiente figura (ver Fig.9).



**Nota:** Cada vez que actualicemos nuestras consultas, el Script de Python se ejecutará, por lo que tendremos la capacidad de filtrar y segmentar estas columnas introducidas de la misma manera que cualquier otra información presente en nuestro modelo de datos.  
  
4- Finalmente, podemos visualizar dicha información en nuestros informes, dotándolos esta vez de aplicaciones de***Machine Learning*** capaces de resolver problemas de clasificación y regresión utilizando múltiples algoritmos, que podremos aplicar en una gran variedad de situaciones dentro del mundo empresarial.  
  
  
  
  
En futuros artículos profundizaré aún más en el tema del *Machine Learning*, así que atentos a futuros post :) .

**Artículos relacionados:**

* [IA y Machine Learning incrementan valor estratégico de los analistas de datos](https://blog.spainbs.com/2019/07/479/ia-y-machine-learning-incrementan-valor-estrategico-de-los-analistas-de-datos)

**Posts Relacionados:**

[**La voz del cliente en el ecosistema de la analítica web**](https://blog.spainbs.com/2021/11/1023/la-voz-del-cliente-en-el-ecosistema-de-la-analitica-web)

El Web Analytics Ecosystem consiste en agregar distintos sistemas de medición que nos permitan obtener una información cuantitativa y cualitativa de POR QUÉ, QUIÉN, QUÉ, CUÁNDO y CÓMO de lo que hacen, piensan, opinan y desean nuestros usuarios y clientes online.

[**¿Qué es el data-driven marketing y porqué se ha convertido en un estándar?**](https://blog.spainbs.com/2021/02/743/que-es-el-data-driven-marketing-y-porque-se-ha-convertido-en-un-estandar)

Conocido también como data-driven marketing, el marketing basado en datos aporta metodología renovada para el análisis del comportamiento del consumidor, agilizada por las herramientas de big data, para la toma de decisiones rápidas que impulsen la conversión en los negocios.

[**Infografía: principales KPIs en RRSS**](https://blog.spainbs.com/2021/02/28/infografia-principales-kpis-en-rrss)

No te pierdas esta infografía con los KPIs de las principales Redes Sociales

[**Técnicas cualitativas de analítica web**](https://blog.spainbs.com/2021/01/947/tecnicas-cualitativas-de-analitica-web)

En la analítica web no hablamos de vuestras opiniones, ni de las opiniones de terceros, sino de datos, y sobre éstos actuamos. La opinión recobra valor por el añadido factor cliente/consumidor. Aportando datos cuantitativos y cualitativos nuestros argumentos serán más valiosos.

Formación Relacionada

* [Máster en Analítica Web y Big Data](https://www.spainbs.com/master-analitica-web-big-data?utm_source=blog)
* [Principios Machine Learning (con Python)](https://www.spainbs.com/ficha.aspx?idc=428&utm_source=blog)

SBS en medios

* [Cómo evitar que un email marketing se vaya a spam](https://blog.spainbs.com/2021/09/1027/como-evitar-que-un-email-marketing-se-vaya-a-spam)
* [Escribir bien en el mundo profesional, condición necesaria para todos](https://blog.spainbs.com/2021/06/1000/escribir-bien-en-el-mundo-profesional-condicion-necesaria-para-todos)
* [El Máster en Emprendimiento de SpainBS recomendado por la revista Emprendedores](https://blog.spainbs.com/2021/06/999/el-master-en-emprendimiento-de-spainbs-recomendado-por-la-revista-emprendedores)