

¿Como comunicar de manera eficiente?

Del otro lado hay cerebros: sesgos, intereses, etc.

Entender quien es el interlocutor para armar la ppt "a medida".

Video miranda:

sugerir estrategia, no tanto explicar el modelo.

Breve (5') - ir a lo importante.

Armar gui n.

10% de la pantalla ocupada por el rostro

Qu  informaci n incluir/qu  an lisis hacer?

Presentar estructura de los clientes que se me est n dando de baja.

Perfilado de clientes(segmentaci n y entender la estructura de cada segmento): los clusters no suelen ser tan claros, por eso hago segmentos con alguna l gica de negocio.

PCA: no soporta nulos, problema con colinealidad, multinormal (no ser a la mejor herramienta, pero se podr a usar)

Kmeans: no soporta nulos. Maldici n de la dimensionalidad afecta al algoritmo.

Quiz s podr a imputar nulls, elegir unas pocas variables (15, x feature importance o variables importantes de negocio) y ajustar varios kmeans.

perfilado: tratar de entender qu  distingue a un cluster del resto de la poblaci n

 C mo hacerlo?

Primera opci n (estrategia gustavo):

Matriz de casos (matriz de similaridad) en la que computo cu ntas veces un registro aparece con otro registro en la misma hoja de cada  rbol de un random forest (hiperpar metros x defecto). Matriz supervisada.

Ciertos algoritmos, pueden tomar esta matriz y armar clusters. Ej. Clustering jer rquico

desventaja: no se banca matrices grandes. Quedarme solo con baja+1 y baja+2 en la matriz de similaridad.

Una vez hecho esto, hago un EDA de cada cluster.

Segunda opci n (estrategia Alejandro): SHAP Values (en python)

Shapley, teor a de juegos colaborativo: contribuci n de cada participante en un juego.

No necesita conocer el score que devuelve el modelo, pero funciona lento.

hay una aplicación más rápida con modelos basados en árboles.

Soporta más registros q el clustering jerárquico, podría trabajar con todo (igual, quizás convenga quedarse solo con los baja).

Cada variable va a tener su propia "importancia" para cada individuo y sirve mirar el signo para entender si la variable aportó score para que se vaya, o generó el efecto opuesto. (Como afecta cada variable a la "probabilidad" de irse del cliente).

Obtengo cuales son los features que más inciden en que un cliente se vaya.

Después, aplico clustering (ej: UMAP, versión superada de T-SNE)

En el ejemplo, el UMAP ya devuelve algo muy lindo. Le puedo aplicar kmeans o dbscan a eso para definir los clusters

En un mundo ideal, armaría esto tanto con continua y bajas, para entender el % de baja de cada grupo, etc.

En la práctica, quizás se puede hacer solo con los baja.