DATA DRIFT

En el siguiente documento, se muestra un resumen de los resultados obtenidos en el testeo de diferentes librerías y algoritmos para la detección de data drift. En él, se ha tenido en cuenta si el algoritmo es capaz de detectar el drift en los features que debe, y si muestra en qué momento aparece el drift.

LIBRERÍAS

Todas las librerías se han probado con el mismo dataset, perteneciente a la librería Menelaus. Contiene 10 features numéricas: a, b, c, d, e, f, g, h, i, j.

El drift debería ser detectado en b, c, d, e, f, h y j.

TABLA COMPARATIVA

- Criterios:
    - La casilla es verde si el algoritmo acierta. Es decir, si predice que no hay drift, y realmente no hay drift. Y lo mismo cuando hay drift, si predice que hay drift, y realmente hay drift, la casilla es verde.
    - Las casillas que estén marcadas por D (‘Detectado’) son los drift que ha detectado el algoritmo entre las features.
    - La casilla es amarilla y marca MD (‘Mal Detectado’) si el algoritmo detecta el drift más veces de lo que debía.

Nanny ML											
Método	**Where**										**When**
	a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	
Kolmogorov-Smirnov Test		D		D				D		D	Sí, muestra gráficamente los puntos en los que se sobrepasa el umbral.
Jensen-Shannon Distance		D		D				D		D	Sí, muestra gráficamente los puntos en los que se sobrepasa el umbral.
Porcentaje Drift detectado	57'14 % (No detecta drift en 'c', 'e' y 'f'. Aunque en 'c' está cerca del umbral)										
Características	- Los algoritmos dan la opción de dibujar los valores estadísticos de cada feature calculados en el tiempo, para poder compararlos con el umbral y determinar si hay drift o no. - NannyML proporciona la opción de trabajar con 'chunks', de esta manera se reduce considerablemente el coste computacional. - Su implementación es sencilla, se ha de llamar a la función Univariate DriftCalculator() y especificar el método tanto para variables numéricas como categóricas.										

Menelaus											
Método	**Where**										**When**
	a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	
Hellinger Distance DDM		D		D				D		D	Sí, muestra gráficamente en qué punto se da el drift en el tiempo.
Porcentaje Drift detectado	57'14 % (No detecta drift en 'c', 'e' y 'f')										
Características	- HDDDM presenta la opción de obtener las diferencias entre distribuciones mediante la instancia feature_epsilons en el tiempo. Esto permite dibujar el heatmap que muestra en qué features se está dando el drift en la línea temporal. - En este caso el dataset a analizar se agrupa según el año, y cada año se va actualizando el algoritmo HDDDM.										

Evidently											
Método	**Where**										**When**
	a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	
Kolmogorov-Smirnov Test		D		D				D		D	Sí, muestra gráficamente los puntos en los que se sobrepasa el umbral.
Wasserstein distance		D						D		D	Sí, muestra gráficamente los puntos en los que se sobrepasa el umbral.
Kullback-Leibler Divergence								D			Sí, muestra gráficamente los puntos en los que se sobrepasa el umbral.
Population Stability Index		D						D		D	Sí, muestra gráficamente los puntos en los que se sobrepasa el umbral.
Jensen-Shannon		D						D		D	Sí, muestra gráficamente los puntos en los que se sobrepasa el umbral.
Porcentaje Drift detectado	- **KS: 57,14 %** - Wasserstein: 42,85 % - KL: 14,28 % - PSI: 42,85 % - JS: 42,85 %										
Características	- La librería Evidently permite calcular el data drift de todos los features de forma instantánea, generando un dashboard que describe las distribuciones de cada una de las features. - Su implementación es muy sencilla y directa. Ya que no requiere una distribución en batch-es ni agrupaciones. Se introduce directamente todo el dataset.										

Frouros											
Algoritmo	**Where**										**When**
	a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	
Kolmogorov-Smirnov Test		D	D	D				D		MD	Sí, muestra gráficamente en qué punto se da el drift en el tiempo.
Cramer-Von Mises Test		D	D	D				D		MD	Sí, muestra gráficamente en qué punto se da el drift en el tiempo.
Anderson Darling Test		D	D	D				D		MD	Sí, muestra gráficamente en qué punto se da el drift en el tiempo.
Hellinger Distance		D	D	D				D		D	Sí, muestra gráficamente en qué punto se da el drift en el tiempo.
Jensen-Shannon Distance		D	D	D				D		D	Sí, muestra gráficamente en qué punto se da el drift en el tiempo.
Porcentaje Drift detectado	- Algoritmos basados en test estadísticos: 57,14 % - Algoritmos basados en la distancia: 71,43 %										
Características	- Para todos los algoritmos se emplea la estrategia de batch-s. Se crea un batch por cada año (20000 muestras), y se introducen los batch de forma secuencial al algoritmo. El proceso se repite para cada feature. - Los resultados se pueden ver en forma de graficas y mapa de calor o en forma de texto. - Su implementación es muy sencilla. Se toma una cantidad de muestras (año 2007) como referencia, y las nuevas muestras se van comparando con dicha referencia. - La librería Frouros también presenta el algoritmo **IncrementalKSTest**, que pertenece al grupo de algoritmos que trabajan con entradas de datos en tiempo real. En este caso, cada instancia se va introduciendo uno por uno al algoritmo.										

Alibi-Detect											
Algoritmo	**Where**										**When**
	a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	
Kolmogorov-Smirnov Test		D	D	D				D		MD	Sí, muestra gráficamente en qué punto se da el drift en el tiempo.
Cramer-Von Mises Test		D	D	D				D		MD	Sí, muestra gráficamente en qué punto se da el drift en el tiempo.
Porcentaje Drift detectado	- Algoritmos basados en test estadísticos: 57,14 %										
Características	- Al igual que en Frouros, se emplea el método de batch-s en estos algoritmos. - Su implementación es muy sencilla. Se toma una cantidad de muestras (año 2007) como referencia, y las nuevas muestras se van comparando con dicha referencia. - La librería Frouros también presenta el algoritmo **MMDDriftOnline**, que pertenece al grupo de algoritmos que trabajan con entradas de datos en tiempo real. En este caso, cada instancia se va introduciendo uno por uno al algoritmo.										

CONCLUSIÓN

Una vez conocida la documentación de la librería, y ver como están compuestas las funciones, se concluye en que la implementación de todas las librerías estudiadas es sencilla.

Es cierto que **Evidently** es una librería que es muy diferente a las demás, ya que todos los resultados se muestran mediante dashboards, y por lo tanto es muy visual. Sin embargo, los resultados obtenidos no son muy satisfactorios.

Por otro lado, se observa que los mejores resultados se han obtenido utilizando los métodos de distancia pertenecientes a la librería **Frouros**. Hellinger Distance y Jensen-Shannon Distance proporcionan un nivel de acierto elevado. Además, el tipo de dato a la salida del algoritmo permite dibujar gráficos mediante Seaborn y ver los resultados de forma muy visual e inmediata.

Teniendo en cuenta que se ha trabajado con 10 features y 300.000 muestras, se espera que Frouros tenga buen rendimiento con datasets grandes.