



FACULDADE DE MATEMÁTICAS

Trabajo Fin de Grado

Aspectos matemáticos del *Concept Drift*

Fernando Fraile Mulas

Junio, 2025

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

GRADO DE MATEMÁTICAS

Trabajo Fin de Grado

Aspectos matemáticos del *Concept
Drift*

Fernando Fraile Mulas

Junio, 2025

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

*A mis padres y hermana, por ser la razón y el motor de mi formación académica.
A Ana, mi pareja, por acompañarme en todo momento con su apoyo incondicional.*

A mis amigos, por hacer mucho más llevadera esta etapa formativa.

*A Rosa y David, mis tutores, por sus valiosos conocimientos, su apoyo indispensable y su
coordinación durante la elaboración de ambos trabajos.*

Trabajo propuesto

Área de Conocimiento	Estadística e Investigación Operativa
Título:	Aspectos matemáticos del <i>Concept Drift</i>
Breve descripción del contenido:	<p>El procesamiento de datos en línea surge de la creciente necesidad de analizar y manipular grandes volúmenes de flujos de datos en tiempo real, que superan las capacidades del procesamiento de datos tradicional. En estos entornos dinámicamente cambiantes y no estacionarios, se produce el denominado <i>Concept Drift</i>, referido al cambio en la relación entre los datos de entrada y la salida (variable objetivo) a lo largo del tiempo, lo que obliga a los modelos predictivos a adaptarse a esta situación, aunque en algunos casos el cambio puede afectar únicamente a la relación entre las propias variables de entrada.</p> <p>En este TFG se abordarán diversos aspectos matemáticos necesarios para comprender y modelar este fenómeno: modelos de probabilidad, distribuciones para técnicas de detección de cambios, contrastes de hipótesis, inferencia no paramétrica, etc. Se analizarán distintas formas de abordar el fenómeno y cómo tratar este problema. Además, los contenidos teóricos irán acompañados de ilustraciones prácticas con datos reales y/o simulados. Este trabajo se desarrolla de forma complementaria a otro TFG del Grado en Ingeniería Informática, centrado en la revisión y evaluación de varios algoritmos presentes en la literatura para la detección de <i>Concept Drift</i>. Ambos estudios comparten una base experimental común, y la comparativa realizada en el trabajo informático se incluye en el desarrollo del presente estudio.</p>

Índice

Índice de figuras	IX
Índice de tablas	XI
Resumen	XIII
Introducción	XV
1. Metodología	1
1.1. Distribución conjunta, condicionada y marginal. Teorema de Bayes	1
1.2. Contrastes de hipótesis	3
1.3. Estimación no paramétrica mediante Regresión Local Lineal	13
1.4. Enfoques y técnicas del aprendizaje automático	16
1.5. Concept Drift: Conceptualización y detección	17
1.5.1. Tipos de Concept Drift	18
1.5.2. Detección de Concept Drift	21
1.6. Métricas utilizadas en la experimentación	22
1.6.1. Mediciones en la modificación para datos de entrada continuos en el detector	22
1.6.2. Mediciones en la modificación para métricas como datos de entrada en el detector	23
2. Algoritmo y modificaciones	27

2.1. Modificación para datos procedentes de distribuciones continuas	30
2.2. Modificación para métricas de rendimiento continuas como datos de entrada en el detector	33
3. Pruebas y resultados	41
3.1. Entornos de pruebas	41
3.2. Resultados y discusión	43
3.2.1. Modificaciones para datos procedentes de distribuciones continuas	43
3.2.2. Modificación para métricas de rendimiento continuas como datos de entrada en el detector	49
4. Conclusiones y posibles ampliaciones	55
I. Tabla para el test de KS	59
II. Resultados de las pruebas	61
Bibliografía	65

Índice de figuras

1.1. Evolución de la métrica <i>Accuracy</i>	6
1.2. Funciones de distribución acumuladas del <i>Accuracy</i>	6
1.3. Tipos de <i>Concept Drift</i> según la probabilidad afectada en un modelo de clasificación, reproducida de [2]. En la fila superior se representa la distribución original de los datos (a), junto con los casos de <i>Real Drift</i> (b) y <i>Virtual Drift</i> (c). En la fila inferior se ilustran distintas situaciones del <i>Class Prior Shift</i>	19
1.4. Tipos de <i>Concept Drift</i> según la velocidad del cambio, reproducida de [9]	20
1.5. <i>Accuracy</i> del modelo	25
1.6. <i>Accuracy</i> Modificado del modelo. Línea azul: <i>Accuracy</i> original. Línea naranja: <i>Accuracy</i> modificado	26
2.1. Funcionamiento de la configuración original del KSWIN	28
2.2. Funcionamiento del KSWIN Modificado (configuración 2)	32
2.3. Funcionamiento del KSWIN Modificado (configuración 3)	34
2.4. Ventana de confirmación del KSWIN	37
2.5. Comportamiento de la métrica de rendimiento en <i>Abrupt e Incremental Drift</i>	39
2.6. Comportamiento de la métrica de rendimiento en <i>Gradual Drift</i>	39
3.1. FDR - <i>Abrupt Drift</i> - Datos de distribuciones continuas simuladas	44
3.2. FDR - <i>Incremental Drift</i> - Datos de distribuciones continuas simuladas	44
3.3. FDR - <i>Gradual Drift</i> - Datos de distribuciones continuas simuladas	44
3.4. MDR - <i>Abrupt Drift</i> - Datos de distribuciones continuas simuladas	44

3.5. MDR - <i>Incremental Drift</i> - Datos de distribuciones continuas simuladas	44
3.6. MDR - <i>Gradual Drift</i> - Datos de distribuciones continuas simuladas	45
3.7. Tiempos de ejecución - <i>Abrupt Drift</i> - Datos de distribuciones continuas simuladas	46
3.8. Tiempos de ejecución - <i>Incremental Drift</i> - Datos de distribuciones continuas simuladas	46
3.9. Tiempos de ejecución - <i>Gradual Drift</i> - Datos de distribuciones continuas simuladas	46
3.10. Monitorización de datos procedentes de una distribución continua usando KSWIN Modificado (configuración 2)	47
3.11. FDR - Monitorización del error en modelo de predicción	47
3.12. MDR - Monitorización del error en modelo de predicción	47
3.13. Media de los errores con <i>Abrupt Drift</i> en modelo de predicción	48
3.14. Media de los errores con <i>Gradual Drift</i> en modelo de predicción	48
3.15. Monitorización del modelo de predicción - KSWIN Modificado (configuración 2)	48
3.16. IR de KSWIN Modificado (configuración 3)	50
3.17. Monitorización del modelo de clasificación - KSWIN Modificado (configuración 3)	50
3.18. Media del FDR para todos los tipos de <i>drift</i> - Comparativa entre algoritmos no optimizados y modificación. Los resultados se muestran ordenados de menor a mayor según el valor del ratio	51
3.19. Media de las métricas - Comparativa entre algoritmos no optimizados y modifica- ción. Resultados ordenados de mayor a menor <i>CohenKappa</i>	51
3.20. Media del FDR para todos los tipos de <i>drift</i> - Comparativa entre algoritmos opti- mizados y modificación. Los resultados se muestran ordenados de menor a mayor según el valor del ratio	52
3.21. Media de las métricas - Comparativa entre algoritmos optimizados y modificación. Resultados ordenados de mayor a menor <i>CohenKappa</i>	53

Índice de tablas

1.1. Errores en el contraste de hipótesis	4
1.2. Errores cometidos al contrastar m hipótesis nulas, reproducida de [3]	8
3.1. Resultados de la experimentación de KSWIN Modificado (configuración 3)	49
I.1. Tabla para test Kolmogorov-Smirnov [19]	60
II.1. Media de los ratios obtenidos para la prueba del KSWIN Modificado (configuración 3)	61
II.2. Resultados obtenidos para la prueba del KSWIN Modificado (configuración 3) en el caso No Optimizado	62
II.3. Resultados obtenidos para la prueba del KSWIN Modificado (configuración 3) en el caso Optimizado	63

Resumen

En este trabajo se aborda el fenómeno del Concept Drift, que se manifiesta en entornos dinámicos y no estacionarios, donde las relaciones estadísticas entre las variables del modelo varían con el tiempo, afectando al rendimiento de los algoritmos de aprendizaje automático. Como objetivo principal, se desarrolla una modificación del algoritmo KSWIN, perteneciente a la librería RiverML, basado en el contraste de Kolmogorov-Smirnov. La propuesta incorpora contrastes múltiples y la corrección de Benjamini-Hochberg, con el fin de mejorar la robustez estadística del test y reducir la tasa de falsos positivos.

Se proponen diversas configuraciones del detector, orientadas tanto a la monitorización de datos procedentes de distribuciones continuas como a la evaluación de métricas de rendimiento. Para este último enfoque, se introduce un mecanismo para la identificación del tipo de *drift*, aplicando técnicas de inferencia no paramétrica. En el primer caso, se diseña un entorno de pruebas con datos generados artificialmente. En el segundo, se integra la comparativa desarrollada en un Trabajo de Fin de Grado del Grado en Ingeniería Informática, centrado en la evaluación de múltiples detectores presentes en la literatura.

Los experimentos realizados muestran una reducción significativa en la tasa de falsos positivos sin comprometer la potencia del contraste, mejorando la eficacia del algoritmo original y otros detectores clásicos. Además, la capacidad de identificación del tipo de *drift* aporta un valor añadido a una de las configuraciones propuestas.

Abstract

This work addresses the phenomenon of Concept Drift, which arises in dynamic and non-stationary environments where the statistical relationships between model variables change over time, thus affecting the performance of machine learning algorithms. The main objective is to

develop a modification of the KSWIN algorithm, part of the RiverML library, which is based on the Kolmogorov-Smirnov test. The proposed modification incorporates multiple hypothesis tests and the Benjamini-Hochberg correction in order to enhance the statistical robustness of the test and reduce the false positive rate.

Several configurations of the detector are proposed, targeting both the monitoring of data drawn from continuous distributions and the evaluation of performance metrics. For the latter approach, a mechanism is introduced to identify the type of drift, using non-parametric inference techniques. For the first case, a testing environment with artificially generated data is designed. In the second, the work integrates a comparative study developed in a Bachelor's Thesis in Computer Engineering, focused on the empirical evaluation of several drift detection algorithms from the literature.

The experiments show a significant reduction in the false positive rate without compromising test power, improving the effectiveness of both the original algorithm and other classical detectors. Furthermore, the ability to identify the type of drift adds practical value to one of the proposed configurations.

Introducción

Los modelos de aprendizaje automático se despliegan cada vez con mayor frecuencia en múltiples ámbitos con el propósito de automatizar tareas, optimizar procesos y apoyar la toma de decisiones. A medida que el volumen de datos procesados crece año tras año, el papel de estos modelos adquiere una relevancia cada vez mayor para organizaciones y empresas.

En muchos casos, se emplean flujos de datos como entrada, con una característica temporal presente, y en entornos dinámicos y no estacionarios. A modo de ejemplo, puede ocurrir que un modelo de aprendizaje automático se despliegue en una fábrica con el objetivo de analizar y predecir la calidad del material producido. Con el paso del tiempo, es posible que los sensores encargados de recolectar los datos se degraden, que cambien las condiciones ambientales o que la materia prima presente ligeras variaciones. Este tipo de alteraciones puede provocar una disminución en el rendimiento del modelo, dando lugar al fenómeno conocido como *Concept Drift*, considerado una de las principales problemáticas en el aprendizaje automático y una de las grandes causas de degradación en modelos desplegados.

El *Concept Drift* se puede definir como un cambio en la relación entre las variables de entrada y/o la variable objetivo o *target*, o únicamente entre las variables de entrada [4]. A lo largo de este trabajo, los términos *Concept Drift*, *drift* y cambio de concepto se utilizarán indistintamente para referirse a este fenómeno. Este puede presentarse tanto en el aprendizaje supervisado, donde los datos están etiquetados y se cuenta con una variable objetivo, como en el aprendizaje no supervisado, en el que no se dispone de dichas etiquetas. Asimismo, puede darse en situaciones de aprendizaje semi supervisado, en las que solo una parte del conjunto de datos está etiquetada. En este trabajo se profundizará en la detección de *Concept Drift* en conjuntos de datos continuo, simulando la monitorización de las variables de entrada de un modelo, independientemente del tipo de algoritmo de aprendizaje automático utilizado. De igual forma, se indagará en la aparición del *drift* en el contexto del aprendizaje supervisado, con un enfoque particular en modelos de clasificación.

Resulta fundamental abordar este problema, ya que el rendimiento del modelo tiende a degradarse tan pronto como se manifiesta, lo que puede volverlo inservible para su propósito original.

La detección de este fenómeno es crucial; no obstante, a menudo representa un desafío debido a la naturaleza desconocida y cambiante de los datos de entrada. La mayoría de los detectores existentes se basa en la realización de contrastes de hipótesis, analizando distintos aspectos de las muestras obtenidas. En este contexto, el presente estudio se centrará en el análisis de uno de los detectores disponibles en la librería `RiverML` [13], el cual emplea un contraste de homogeneidad mediante el test clásico de Kolmogorov-Smirnov para evaluar la similitud entre las distribuciones poblacionales de dos muestras distintas del conjunto de datos.

Cuando se realiza un contraste de hipótesis, es habitual focalizarse en el Error Tipo I, que, extrapolado a la problemática estudiada, equivale a la probabilidad de detectar un *drift* cuando en realidad la distribución de los datos no ha cambiado. Sin embargo, en este contexto resulta fundamental obtener un sistema que emita alertas verdaderamente confiables, maximizando la proporción de alarmas válidas frente al total de alertas generadas. La presencia de numerosos falsos positivos podría derivar en reentrenamientos innecesarios del modelo, lo que puede provocar sobreajuste a los datos más recientes, pérdida de la información contenida en datos históricos o, incluso, una interpretación errónea del estado real del entorno. Todo ello debe lograrse sin comprometer la capacidad del sistema para detectar los *drift* reales, por lo que es necesario recurrir a un enfoque que equilibre de manera efectiva la sensibilidad del sistema con la fiabilidad de sus detecciones.

A este inconveniente se suma la complejidad inherente al análisis de un flujo de datos. En efecto, estudiar la evolución de las distribuciones implica realizar contrastes no solo entre las últimas muestras recogidas, sino también frente a muestras anteriores en el tiempo. Puede suceder que los datos más recientes presenten una distribución similar a los inmediatamente anteriores, pero distinta a la de muestras más antiguas. Para abordar esta situación, surge la necesidad de realizar contrastes entre varias muestras, lo que conduce al denominado problema de los contrastes múltiples. Cuando se realizan múltiples contrastes con el objetivo de tomar una decisión, en este caso detectar la aparición de *drift*, la probabilidad de cometer un Error Tipo I en un contraste individual entre todas las pruebas realizadas se incrementa, condicionando así el error global del contraste múltiple. Por consiguiente, resulta fundamental tener en cuenta este factor y aplicar una corrección adecuada sobre los resultados obtenidos, con el fin de controlar, en la medida de lo posible, el error cometido en el contraste múltiple.

Por otro lado, dentro de la detección del *Concept Drift* se encuentra la capacidad para diferenciar el tipo de *drift* que está ocurriendo, específicamente, según la velocidad de cambio en la distribución de los datos. Actualmente, no existen métodos definidos y contrastados adecuadamente en la literatura, por lo que se trata de un campo que permanece en investigación y desarrollo.

El objetivo principal de este trabajo es realizar una modificación sobre el algoritmo `KSWIN`,

el cual realiza un contraste de Kolmogorov-Smirnov. Se busca adaptar su funcionamiento para realizar los contrastes necesarios sobre el flujo de datos y aplicar una corrección para el problema de los contrastes múltiples que surge de esta situación. Esta modificación debe adaptarse bien a la problemática y ofrecer resultados consistentes y generalizables, aplicables a un caso real. Además, se aportará una solución propia para resolver el problema de la identificación del tipo de *drift* que está ocurriendo. De igual modo, se hará uso de la comparativa elaborada en el TFG del Grado de Informática, trabajo desarrollado de forma complementaria a este estudio, en el que se analiza el comportamiento de la modificación frente a diversos algoritmos presentes en la literatura, monitorizando el rendimiento de un modelo de clasificación utilizando estos algoritmos para detectar y adaptarse al *Concept Drift*. A lo largo del documento, este trabajo podrá referirse también de forma abreviada como trabajo informático o estudio informático. Este trabajo se puede dividir en los siguientes objetivos específicos:

1. Revisión y análisis de las herramientas y conceptos tanto matemáticos como informáticos necesarios.
2. Elaboración de las modificaciones pertinentes del algoritmo KSWIN para la detección de *Concept Drift* en datos procedentes de una distribución continua, acompañada de una justificación teórica de su funcionamiento.
3. Diseño e implementación de un entorno de pruebas adecuado para las modificaciones propuestas, incluyendo conjuntos de datos artificiales para simular distintos tipos de *drift*, métricas y sistemas de visualización adecuados para la interpretación de los resultados obtenidos.
4. Integración de la modificación en la comparativa llevada a cabo en el TFG del Grado de Informática. Análisis de la naturaleza de los datos obtenidos en este estudio y adaptación necesaria para un correcto funcionamiento de la modificación.
5. Desarrollo de un sistema de identificación del tipo de *drift* según su velocidad de aparición o severidad.
6. Discusión y conclusión acerca de los resultados obtenidos durante la experimentación.

Este trabajo se organiza como sigue. En el Capítulo 1 se presentan los fundamentos matemáticos e informáticos que soportan el estudio realizado. Asimismo, se incluyen las diversas métricas utilizadas en la experimentación, así como la adaptación del *Accuracy* para poder utilizarla como indicador actualizado del rendimiento del modelo y como conjunto de datos de entrada para la modificación del algoritmo. A continuación, en el Capítulo 2 se presenta la configuración original del algoritmo KSWIN de la librería RiverML [13], analizando sus debilidades y errores. Se aportarán dos configuraciones distintas para la modificación propuesta en el caso de monitorizar datos

procedentes de distribuciones continuas, y una configuración de la modificación propuesta para el caso de la monitorización de métricas de rendimiento continuas, siendo esta última la integrada en el TFG de Informática, y que requiere un tratamiento especial debido al comportamiento de las métricas. Todas las modificaciones contienen una justificación teórica con los conceptos presentados en el Capítulo 1. Posteriormente, en el Capítulo 3, se presenta el entorno de pruebas elaborado, así como las pruebas y los resultados obtenidos para las modificaciones propuestas. Se aportará tanto una comparativa con la configuración original del *KSWIN* como un contraste con el resto de algoritmos evaluados en el estudio informático. Finalmente, en el Capítulo 4 se aporta una discusión de los resultados obtenidos en el Capítulo 3, analizando el impacto de las modificaciones propuestas y diversas ampliaciones que se consideran de interés en futuros trabajos, pero que exceden el alcance de este estudio. Por último, es necesario señalar que se ha utilizado la herramienta de inteligencia artificial generativa *ChatGPT* para la revisión y mejora de la redacción de diversos aspectos del trabajo.

Capítulo 1

Metodología

A lo largo de este capítulo se desarrollarán herramientas y conceptos tanto matemáticos como informáticos. Estos son necesarios para comprender la naturaleza de la problemática estudiada, así como para justificar las decisiones tomadas en la experimentación. Además, se aportará una explicación de las diversas métricas utilizadas para evaluar el rendimiento y calidad de los resultados obtenidos.

1.1. Distribución conjunta, condicionada y marginal. Teorema de Bayes¹

En esta sección se muestran los conceptos matemáticos necesarios para entender la definición formal del *Concept Drift*.

Sea (Ω, \mathcal{F}, P) un espacio de probabilidad, un vector aleatorio es una colección de variables aleatorias medidas simultáneamente sobre un experimento aleatorio. Se puede representar como

$$X = (X_1, \dots, X_m)' : \Omega \rightarrow \mathbb{R}^m.$$

Se dice que un vector aleatorio X es discreto si sus posibles valores son un conjunto de puntos $\{x_1, x_2, \dots\}$ finito o infinito numerable de \mathbb{R}^m , de modo que $P(X = x_k) = p_k \quad \forall k \in \{1, 2, \dots\}$ es la función de masa de probabilidad discreta, con $p_k > 0 \quad \forall k \in \{1, 2, \dots\}$ y $\sum_k p_k = 1$.

Por otro lado, se dice que un vector aleatorio X es absolutamente continuo si existe una función de densidad de probabilidad $f : \mathbb{R}^m \rightarrow \mathbb{R}$ tal que $P(X \in A) = \int_A f(x)dx \quad A \subset \mathcal{B}(\mathbb{R}^m)$.

¹Los conceptos presentados en esta sección fueron tratados en *Probabilidad y Estadística*, asignatura del tercer curso del Grado en Matemáticas.

Cabe destacar que es habitual emplear el σ -álgebra de Borel, pero se puede definir sobre cualquier conjunto que pertenezca a un σ -álgebra sobre la que esté definida la probabilidad, cumpliendo la axiomática de Kolmogorov. Se define una probabilidad o medida de probabilidad sobre un espacio medible (Ω, \mathcal{F}) a una aplicación $P : \mathcal{F} \rightarrow \mathbb{R}$ que cumple las siguientes tres condiciones:

1. $P(A) \geq 0 \quad \forall A \in \mathcal{F}$
2. Sea $\{A_n\} \subset \mathcal{F}$ una colección numerable de sucesos disjuntos entre sí, se cumple que:

$$P(\cup_n A_n) = \sum_n P(A_n)$$

3. $P(\Omega) = 1$

Para estudiar la relación entre las componentes de un vector aleatorio se hace uso de la distribución conjunta. Esta se define como una aplicación $F : \mathbb{R}^m \rightarrow \mathbb{R}$ tal que

$$F(x) = P(X \leq x), \quad \text{esto es,} \quad F(x_1, \dots, x_m) = P(X_1 \leq x_1, \dots, X_m \leq x_m).$$

La distribución conjunta es por lo tanto un reparto de probabilidad en el espacio \mathbb{R}^m . Cada componente individual del vector aleatorio multivariante tiene una propia distribución, denominadas distribuciones marginales $F_k \quad \forall k \in \{1, \dots, m\}$. Se tiene entonces que, en el caso discreto, cada componente del vector aleatorio X son variables aleatorias discretas, y sumando las probabilidades en la distribución conjunta se obtiene la función de masa de probabilidad, como se podía ver anteriormente. Por otro lado, en el caso continuo, las componentes del vector son absolutamente continua, e integrando cada componente con respecto a las demás variables se obtiene la función de densidad. Además, la distribución condicionada aporta una información más específica de la relación entre las componentes del vector. Dependiendo de si el vector es discreto o absolutamente continuo, esta distribución se puede definir de dos formas diferentes.

Sea $X = (X_1, X_2)'$ un vector aleatorio en \mathbb{R}^m , donde X_1 es el vector aleatorio formado por sus primeras r componentes, y X_2 es el vector aleatorio formado por sus $m - r$ últimas componentes. Si $P(X_2 = x_2) > 0$ entonces la distribución de X_1 condicionada por X_2 es:

$$P\{X_1 \leq x_1 \mid X_2 = x_2\} = \frac{P\{X_1 \leq x_1, X_2 = x_2\}}{P\{X_2 = x_2\}}.$$

En el caso discreto, si $x_2 \in \mathbb{R}^{m-r}$ tal que $P(X_2 = x_2) > 0$, se define la distribución de X_1 condicionada a $X_2 = x_2$ como la distribución de probabilidad discreta con función de masa

$$P(X_1 = x_1 \mid X_2 = x_2) = \frac{P(X_1 = x_1, X_2 = x_2)}{P(X_2 = x_2)}, \quad x_1 \in \mathbb{R}^r.$$

En el caso absolutamente continuo, en las mismas condiciones, con densidad $f_2(x_2) > 0$, se define la distribución de X_1 condicionada a $X_2 = x_2$ como la distribución de probabilidad absolutamente continua cuya densidad viene dada por

$$f_{X_1|X_2}(x_1 | x_2) = \frac{f_{X_1, X_2}(x_1, x_2)}{f_{X_2}(x_2)}, \quad x_1 \in \mathbb{R}^r.$$

Por último, tras definir las distribuciones condicionadas se presenta la Regla o Teorema de Bayes. Dados un sistema completo de sucesos $\{A_i\}$ $i \in \{1, \dots, n\}$ y B un suceso tal que $P(B) > 0$, entonces se cumple que:

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)}.$$

1.2. Contrastes de hipótesis²

En esta sección se introducen los conceptos matemáticos para comprender el proceso de detección e identificación del *Concept Drift*. Asimismo, se explica la corrección aplicada frente al problema de los contrastes múltiples que mejor se ajusta a las necesidades del fenómeno tratado, la corrección de Benjamini-Hochberg. Se incluyen las demostraciones pertinentes para justificar el control del *False Discovery Rate* mediante el procedimiento de Benjamini-Hochberg [3].

Posteriormente se verá como la gran mayoría de algoritmos utilizados en la detección del *Concept Drift* realizan un contraste de hipótesis para determinar la aparición de este fenómeno, por lo que es necesario definir este concepto. El objetivo de un contraste de hipótesis es decidir si la conjetura sobre una distribución poblacional estudiada se puede confirmar con suficiencia estadística a partir de la muestra aleatoria simple observada. Por lo tanto, se tiene una hipótesis acerca de la distribución poblacional, denominada hipótesis nula H_0 que se contrasta frente a una hipótesis alternativa H_1 , que contiene todas las distribuciones poblacionales posibles en las que no se cumple la hipótesis nula.

Por lo tanto, los dos únicos resultados posibles son aceptar o rechazar la hipótesis nula, lo que hace que el espacio muestral se divida en dos conjuntos. Sea el espacio muestral \mathcal{X} , si la muestra pertenece a la región crítica C , se rechaza H_0 en favor de H_1 , y si la muestra pertenece a la región de aceptación C^c , no existen evidencias para rechazar H_0 , con una relación inversa entre ambas regiones $C^c = \mathcal{X} - C$. Un test aleatorizado es una función medible $\varphi : \mathcal{X} \rightarrow [0, 1]$ denominada función crítica del test que expresa la probabilidad de rechazar la hipótesis nula con

²Varios de los conceptos presentados en esta sección que fueron tratados en *Inferencia estadística*, asignatura del tercer curso del Grado en Matemáticas.

$\varphi(x_1, \dots, x_n)$, y de aceptarla con $1 - \varphi(x_1, \dots, x_n)$. El caso determinista y no aleatorizado es un subcaso del anterior, con la siguiente función crítica:

$$\varphi(x) = \begin{cases} 1 & \text{si } (x_1, \dots, x_n) \in C \\ 0 & \text{si } (x_1, \dots, x_n) \in C^c. \end{cases}$$

Se tiene entonces que al realizar el test se tienen las siguientes posibilidades:

Realidad \ Decisión	No rechazar H_0	Rechazar H_0
H_0 es cierta	Correcto $P_{H_0}(C^c) = 1 - \alpha$	Error tipo I $P_{H_0}(C) = \alpha$
H_0 es falsa	Error tipo II $P_{H_1}(C^c) = \beta$	Correcto $P_{H_1}(C) = 1 - \beta$

Tabla 1.1: Errores en el contraste de hipótesis

Lo ideal sería minimizar ambos errores por igual, pero por lo general no es posible, ya que al disminuir la probabilidad de que se de uno de ellos aumenta la probabilidad de que se de el otro. Por lo tanto, habitualmente se le da más importancia al Error Tipo I, y se suele realizar en contraste bajo un nivel de significación α , siendo este la probabilidad de cometer el error mencionado. Del mismo modo se puede definir la sensibilidad y la especificidad de un contraste de acorde a las decisiones correctas tomadas, mostradas en la Tabla 1.1. La sensibilidad o potencia del contraste es la probabilidad de rechazar correctamente la hipótesis nula H_0 cuando es falsa, es decir, $P_{H_1}(C) = 1 - \beta$. Por otra parte, la especificidad se define como la probabilidad de no rechazar correctamente la hipótesis nula H_0 cuando es cierta, es decir, $P_{H_0}(C^c) = 1 - \alpha$

Los contrastes de hipótesis se pueden utilizar para verificar distintos tipos de información. Uno de los tipos son los denominados contrastes paramétricos, en lo que se verifica información acerca uno o más parámetros de la distribución poblacional de una muestra. Sin embargo, en este estudio se van a analizar muestras de las cuáles no se conoce la distribución de la población, caso en el que utilizan los contrastes no paramétricos. Los más habituales son los test de bondad de ajuste, en el que a partir de una muestra de la población se determina si esta sigue una distribución dada, y el de homogeneidad, en el que se analiza si varias muestras provienen de un poblaciones con la misma distribución teórica. Como se ha examinado un algoritmo existente en la literatura que utiliza el test de Kolmogorov-Smirnov como prueba para determinar la aparición de *Concept Drift*, se va a profundizar en este. Más específicamente, en el contraste de homogeneidad de Kolmogorov-Smirnov, al no conocer de antemano la distribución poblacional de las muestras y comparar por lo tanto si los datos más recientes siguen la misma distribución que lo anteriores.

Dadas dos muestras aleatorias simples X_1, \dots, X_n e Y_1, \dots, Y_n , independientes entre si, el test

de homogeneidad consiste en comparar las distribuciones poblacionales de ambas muestras, F_x y F_y . El test se puede construir como:

$$\begin{aligned} \text{Hipótesis nula (H}_0\text{): } & F_X = F_Y \\ \text{Hipótesis alternativa (H}_1\text{): } & F_X \neq F_Y. \end{aligned} \tag{1.1}$$

Se recuerda que un estadístico es una función medible que sólo depende de la muestra. Cuando la distribución poblacional es desconocida, el estimador más natural para la distribución de la primera muestra aleatoria X es la función de distribución empírica, definida como $F_n^*(x) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}_{\{X_i \leq x\}}$, donde $\mathbb{I}_{\{\cdot\}}$. Es lógico utilizar el mismo estimador para la segunda muestra Y , teniendo entonces también $G_m^*(x) = \frac{1}{m} \sum_{i=1}^m \mathbb{I}_{\{Y_i \leq x\}}$. El estimador cumple varias propiedades que reafirman su utilización:

- Es un estimador insesgado: $\mathbb{E}[F_n^*(x)] = F(x) \quad \forall x \in \mathbb{R}$.
- Es un estimador consistente: $\text{Var}(F_n^*(x)) = \frac{F(x)(1-F(x))}{n} \quad \forall x \in \mathbb{R}$.
- Consistencia universal (Teorema de Glivenko-Cantelli): $\Delta_n = \sup_{x \in \mathbb{R}} |F_n^*(x) - F(x)| = L(F_n^*, F) = \|F_n^* - F\|_\infty \xrightarrow{c.s.} 0$ cuando $n \rightarrow \infty$. Es decir, dado el espacio de medida (Ω, \mathcal{A}, P) , $P(\omega \in \Omega : \lim_{n \rightarrow \infty} F_n^*(\omega) = F) = P(\lim_{n \rightarrow \infty} \Delta_n = 0) = 1$.

El teorema de Glivenko-Cantelli permite formalizar el estadístico utilizando las dos funciones de distribución empíricas $\Delta_{n,m} = \sup_{x \in \mathbb{R}} |F_n^*(x) - G_m^*(x)|$, ya que si H_0 es cierta, el supremo de la diferencia será muy pequeño al provenir ambas de la misma distribución poblacional. Por otro lado, si no provienen de la misma, la diferencia será mayor, por lo que la región crítica vendrá dada por $\{\Delta_{n,m} > k\}$, siendo k el punto crítico asociado a cada nivel de significación.

La distribución del estadístico $\Delta_{n,m}$ no depende de la distribución F , si y solo si F es continua, siendo así de distribución libre. Esto implica que para aplicar el contraste de homogeneidad de KS, es necesario evaluar muestras continuas. Por otro lado, se puede asegurar que el test no va a depender de la distribución poblacional de los datos, obteniendo unos resultados válidos sea cual sea esta, algo que es muy interesante ya que no se conoce la distribución de los datos de origen, y esta puede ir variando a lo largo del tiempo.

Cabe destacar que el estadístico $\Delta_{n,m}$ se utiliza para contrastes bilaterales, en los que se compara si la distribución $F_x = F_y$, pero hay casos en los que es interesante realizar un contraste unilateral, en el que se compare si una distribución es "mayor" que otra. Se puede definir por lo tanto el estadístico $\Delta_{n,m}^+ = \sup_{x \in \mathbb{R}} (F_n^*(x) - G_m^*(x))$ para la hipótesis nula $F_X \leq F_Y$, y el estadístico $\Delta_{n,m}^- = \sup_{x \in \mathbb{R}} (G_m^*(x) - F_n^*(x))$ para la hipótesis nula $F_X \geq F_Y$. De esta forma, si se realiza el test utilizando $\Delta_{n,m}^+$, rechazar la hipótesis nula significará afirmar que la función

de distribución acumulada de los datos más recientes es más alta que la función de distribución acumulada de los datos menos recientes. Con $\Delta_{n,m}^-$ se tiene el caso contrario, en la que la función de distribución acumulada de los datos más recientes es más baja que la función de distribución acumulada de los datos menos recientes. Esto es importante a la hora de evaluar tendencias en una muestra, por ejemplo, si se desea evaluar si una métrica tiende a degradarse con el tiempo.

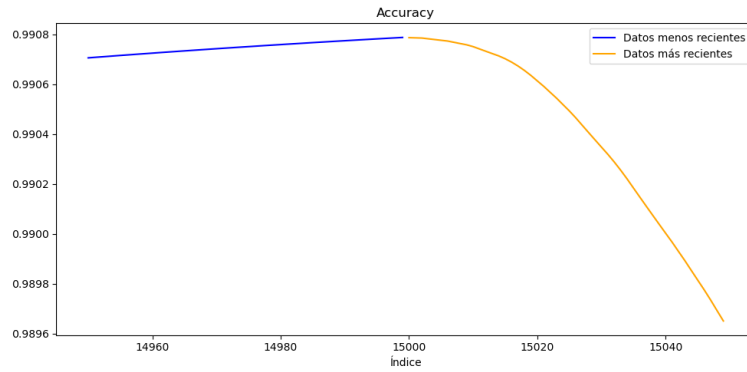


Figura 1.1: Evolución de la métrica *Accuracy*

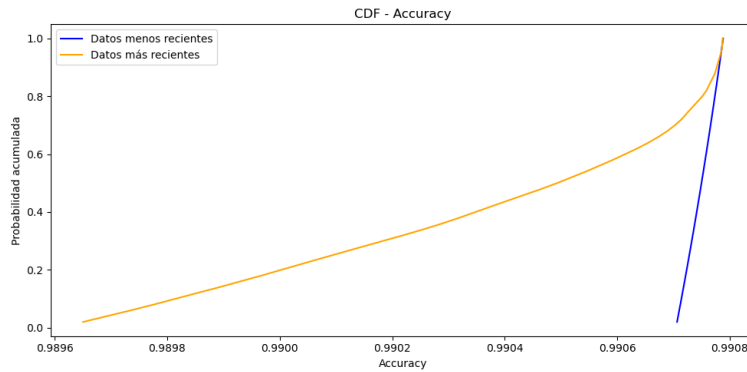


Figura 1.2: Funciones de distribución acumuladas del *Accuracy*

En la Figura 1.1 se puede observar la evolución de una métrica, la cual sufre un decaimiento considerable en el punto 15000. Por lo tanto, la función de distribución acumulada de los datos más recientes va a ser más alta que la de los datos más antiguos, como se aprecia en la Figura 1.2. Por lo tanto, para comprobar un decaimiento en la métrica habría que utilizar el estadístico $\Delta_{n,m}^+$. Esto se ha utilizado en una las modificaciones aplicadas en el estudio.

Es importante señalar que la sensibilidad del test aumenta con el tamaño muestral. En la Tabla I.1 (Anexo I) se muestran los puntos críticos k según el tamaño de la muestra n y el nivel de significación α . Como se puede observar, fijado un nivel α , el punto crítico disminuye conforme aumenta el n , lo que supone que para muestras mayores es necesario un estadístico menor para

rechazar la hipótesis nula en favor de la hipótesis alternativa. Esto es un aspecto relevante a la hora de calibrar el contraste.

Las pruebas se han realizado con la función `ks_2samp`, perteneciente a la librería `Scipy` [18]. La implementación es similar a la que se encuentra en [21]. Cuando el conjunto de datos es muy grande el algoritmo utiliza la ley límite de Kolmogorov-Smirnov, la cual afirma que

$$\lim_{n,m \rightarrow \infty} \Pr \left(\sqrt{\frac{mn}{m+n}} \Delta_{m,n} \leq z \right) = \sum_{i=-\infty}^{\infty} (-1)^i e^{-2i^2 z^2}.$$

Por lo tanto, se puede realizar un test de homogeneidad para comprobar si dos muestras siguen la misma distribución. Sin embargo, hay casos en los que una de las muestras no se quiere comparar solo frente a otra, sino frente a un conjunto diverso de ellas. En este contexto, en el que es necesario realiza múltiples contrastes entre varios conjuntos muestrales, surge el denominado problema de la hipótesis múltiple.

Supongamos que para un test de homogeneidad de Kolmogorov-Smirnov se fija el nivel de significación $\alpha = 0,05$. Este nivel marca la probabilidad de cometer un Error Tipo I o Falso Positivo, es decir, de rechazar la hipótesis nula H_0 siendo esta cierta. Si se realizan múltiples test, por ejemplo 100, en promedio se rechazará 5 veces la hipótesis nula H_0 en favor de la hipótesis alternativa H_1 siendo cierta la hipótesis nula, por lo que, de media, aparecerá el Error Tipo I en 5 experimentos. En el ejemplo utilizado previamente para mostrar la evolución de la métrica, si se una muestra con los n últimos datos y se comparan con 100 combinaciones de conjuntos o muestras de tamaño n previos para verificar si siguen la misma distribución, en promedio se obtendrá un falso positivo en 5 de ellos, lo que puede no reflejar la realidad de lo que ocurre en la tendencia de la métrica.

La probabilidad de obtener un falso positivo cuando se realizan k comparaciones múltiples aumenta con respecto a realizar una única prueba. A esta se le conoce como *family-wise error rate (FWER)*, cuya ecuación viene dada por:

$$\text{FWER} = \bar{\alpha} = 1 - (1 - \alpha)^k.$$

Hay diversos métodos para tratar de solventar esta problemática. Uno de los más conocidos es el método de Bonferroni, en el que se ajusta el nivel de significación dividiéndolo entre el número de pruebas que se van realizar, garantizando que el FWER no supere el valor de α . Sin embargo, tener en cuenta únicamente el FWER para los múltiples contrastes tiene diversos inconvenientes. El primero de ellos es que en casos en los que se realizan un gran número de combinaciones, el resultado de la corrección es un nivel de significación excesivamente pequeño, que hará aumentar drásticamente la aparición del Error Tipo II o Falso Negativo, lo que puede ser una situación no

deseada en muchos de los experimentos. Por otro lado, existen varias correcciones, como Tukey HSD o Dunnett, que suponen que el estadístico de prueba sigue una distribución normal o t de Student. Sin embargo, esto no siempre se cumple, como ocurre en el caso de los estadísticos del test de Kolmogorov-Smirnov. Otros métodos, como la corrección de Bonferroni, presentan una limitación importante en la interpretación de los resultados. Aunque permiten determinar si se rechaza o no la hipótesis nula al realizar múltiples comparaciones, no ofrecen información detallada sobre qué contrastes individuales contribuyen a esa decisión ni sobre la importancia de cada uno dentro del conjunto total de pruebas. Esto puede dificultar la comprensión de los resultados cuando se realizan muchas comparaciones, ya que no se especifica la relevancia de cada diferencia específica. Por ejemplo, al comparar varios tratamientos frente a un control, Bonferroni puede indicar que hay diferencias significativas en general con todos, pero no facilita entender qué tratamientos en particular son los que realmente presentan diferencias significativas y con qué grado de confianza.

De este modo, existe una forma alternativa de tratar el problema de la hipótesis múltiple, aplicando la denominada corrección de Benjamini-Hochberg. Se hará referencia a la corrección con las siglas BH a lo largo de este trabajo. Este enfoque se centra en limitar el número esperado de errores de tipo I entre los contrastes considerados significativos. Se define por lo tanto el FDR (*False Discovery Rate*), el cual mide la proporción de test en los que la hipótesis nula H_0 es cierta entre los que se han considerado significativos en favor de la hipótesis alternativa H_1 , o lo que es lo mismo, la probabilidad de que una hipótesis nula sea cierta habiendo sido rechazada por el test estadístico. De esta forma, se puede fijar un nivel de significación para controlar que el FDR no supere el valor definido. Por ejemplo, si se fija el valor $\bar{\alpha} = 0,1$, se espera que como máximo un 10 % de los resultados considerados como significativos sean realmente falsos positivos.

Realidad\Decisión	H_0 no rechazadas	H_0 rechazadas	Total
Hipótesis nulas H_0 ciertas	U	V	m_0
Hipótesis nulas H_0 falsas	T	S	$m - m_0$
Total	$m - R$	R	m

Tabla 1.2: Errores cometidos al contrastar m hipótesis nulas, reproducida de [3]

En la Tabla 1.2 se pueden ver las diferentes situaciones a la hora de cometer un error en un contraste de hipótesis. El FWER presentado anteriormente se puede definir en este caso como $P(V \geq 1)$. Sea V el número de hipótesis nulas ciertas rechazadas en favor de la hipótesis alternativa, S el número de hipótesis nulas falsas rechazadas en favor de la hipótesis alternativa, $Q = V/(V + S)$ la variable aleatoria discreta que mide la proporción de hipótesis nulas rechazadas de forma errónea, se define $FDR = \mathbb{E}[V/(V + S)] = \mathbb{E}[V/R] = \mathbb{E}[Q]$. Cuando $V + S = 0$, es decir, cuando no se rechaza la hipótesis nula en ninguno de los casos, el FDR se considera nulo.

Tiene dos propiedades muy importantes que permiten relacionarlo con el FWER:

1. Si $m_0 = m$, es decir, si todas las hipótesis nulas son ciertas, se cumple $S = 0$ y $V = R$ y Q solo puede tomar los valores $q_1 = 0$ y $q_2 = 1$. En este caso, es equivalente al FWER:
 - Si $(V + S) = 0 \implies FDR = 0$ y $FWER = 0$.
 - Si $V \geq 1 \implies FDR = \mathbb{E}[Q] = \sum_i q_i P(Q = q_i) = P(Q = 1) = P(V \geq 1) = FWER$.
2. Si $m_0 < m$, es decir, si el número de hipótesis nulas ciertas es menor que el total, en este caso $V > 0 \implies Q = V/R \leq 1 \implies \mathbb{I}_{V \geq 1} \geq 1 \geq Q \implies FWER = P(V \geq 1) = \mathbb{E}[\mathbb{I}_{V \geq 1}] \geq \mathbb{E}[V/R] = FDR$. Esto tiene diversas implicaciones. En primer lugar, cualquier estudio en el que se controle el FWER se controlará también el FDR, al ser este segundo más pequeño. Sin embargo, si en un estudio se controla el FDR, no se garantiza el control sobre el FWER. De este modo, el FDR permite tener un contraste más potente, con una tasa de Error Tipo II más reducida. La diferencia se acrecienta cuando aumentan el número de hipótesis nulas, por eso se recomienda la utilización de esta tasa cuando el número de experimentos es muy elevado, aunque no es excluyente para los casos con una experimentación menor.

Así, el FDR permite solventar algunos de los problemas que surgen al tener solo en cuenta el FWER cuando se realizan múltiples contrastes. Realizar un control sobre este ratio parece además la opción más sensata al tratar el problema del *Concept Drift*. Como se ha visto previamente, es necesario controlar la aparición de falsos positivos en la detección, pero sin perder la capacidad de detección, lo que se consigue controlando este ratio. Para controlar este ratio, Benjamini y Hochberg propusieron el método que se va a presentar a continuación.

Supongamos que en un estudio con m múltiples comparaciones se desea que el FDR no supere el porcentaje q^* . Sean $H_{0,1}, H_{0,2}, \dots, H_{0,m}$ las hipótesis nulas contrastadas sobre una muestra y sus correspondientes p-valores P_1, P_2, \dots, P_m . Tras ordenar estos últimos de forma que $P_{(1)} \leq P_{(2)} \leq \dots \leq P_{(m)}$. Sea k la última posición para la cual se cumple que:

$$P_{(i)} \leq \frac{i}{m} q^*. \quad \text{Entonces, se rechazan todas las hipótesis nulas} \quad (1.2)$$

$$H_{0(1)}, H_{0(2)}, \dots, H_{0(m)} \text{ correspondientes a los } P_{(i)} \text{ ordenados.}$$

Este criterio se respalda en los dos siguientes resultados.

Lema 1.1. *Para cualquier cantidad $0 \leq m_0 \leq m$ de p-valores que correspondan a hipótesis nulas verdaderas, y para cualquier conjunto de valores que puedan tomar los $m_1 = m - m_0$ p-valores correspondientes a las hipótesis nulas falsas, entonces el procedimiento definido en (1.2) cumple que:*

$$\mathbb{E}[Q | P_{m_0+1} = p_1, \dots, P_m = p_{m_1}] \leq \frac{m_0}{m} q^*. \quad (1.3)$$

Demostración. ³ Se prueba el lema mediante inducción sobre m .

- $m = 1$: Cuando el número de contrastes es 1, se tiene lo siguiente.

Si $m_0 = 0$, es decir, si no hay hipótesis nulas falsas $\implies Q$ es idénticamente nula $\implies \mathbb{E}[Q | P_1 = p_1] = 0 \leq 0 = \frac{m_0}{m} q^*$.

Si $m_0 = 1 \implies m_1 = 0 \implies$ No hay p-valores correspondientes a hipótesis nulas falsas. De este modo, dado un umbral q^* , se tiene que

$$\left. \begin{array}{l} P_0 \leq q^* \implies Q = 1 \\ P_0 > q^* \implies Q = 0 \end{array} \right\} \implies \mathbb{E}[Q] = P(P_0 \leq q^*) \underset{P_i \in \overline{U}(0,1)}{\leq} q^* = \frac{m_0}{m} q^*.$$

- $m > 1$. Se supone cierto para m y se prueba para $m + 1$ Si $m_0 = 0$, es decir, si no hay hipótesis nulas falsas $\implies Q$ es idénticamente nula $\implies \mathbb{E}[Q | P_1 = p_1, \dots, P_m = p_m] = 0 \leq \frac{m_0}{m+1} q^*$.

Si $m_0 > 0$, sean $P'_i \in U(0, 1), i \in \{1, \dots, m_0\}$ los p-valores independientes entre si correspondientes a las hipótesis nulas verdaderas, y $P'_{(m_0)}$ es más alto de ellos tras ordenarlos. Sean $p_{(1)} \leq p_{(2)} \leq \dots \leq p_{(m_1)}$ los p-valores ordenados ya calculados correspondientes a las hipótesis nulas falsas (se denotarán mediante $p_1 \leq p_2 \leq \dots \leq p_{m_1}$ para simplificar la notación) y j_0 el valor más alto que cumple que $0 \leq j \leq m_1$ tal que

$$p_j \leq \frac{m_0 + j}{m + 1} q^*.$$

Sea $p'' = \frac{m_0 + j_0}{m + 1} q^*$. Si $P'_{(m_0)} = p$ se tiene que

$$\begin{aligned} \mathbb{E}[Q | P_{m_0+1} = p_1, \dots, P_m = p_{m_1}] &= \int_0^{p''} \mathbb{E}[Q | P'_{(m_0)} = p, P_{m_0+1} = p_1, \dots, P_m = p_{m_1}] \\ &\quad f_{P'_{(m_0)}}(p) dp \\ &\quad + \int_{p''}^1 \mathbb{E}[Q | P'_{(m_0)} = p, P_{m_0+1} = p_1, \dots, P_m = p_{m_1}] \\ &\quad f_{P'_{(m_0)}}(p) dp. \end{aligned} \quad (1.4)$$

Ahora, se tiene que $F_{P'_{(m_0)}}(p) = P(P'_{(m_0)} \leq p) = P(P'_1 \leq p, P'_2 \leq p, \dots, P'_{m_0} \leq p) \underset{P'_i \text{ indep.}}{=} P(P'_1 \leq p)(P'_2 \leq p) \dots (P'_{m_0} \leq p) \underset{P'_i \in U(0,1)}{=} p^{m_0}$. Por lo tanto, la función de densidad viene dada por $f_{P'_{(m_0)}}(p) = \frac{d}{dp} p^{m_0} = m_0 p^{(m_0-1)}$.

³Demostración recogida en [3], ampliada para facilitar su comprensión.

En la primera integral se tiene que $p < p''$. Por lo tanto, cogiendo el p'' como nivel de significación, todas las hipótesis $m_0 + j_0$ se rechazan, y por lo tanto $Q = V/(V + S) = m_0/(m_0 + j_0)$. Evaluando la integral así la primera integral de la ecuación (1.4)

$$\begin{aligned} & \int_0^{p''} \mathbb{E}[Q | P'_{(m_0)} = p, P_{m_0+1} = p_1, \dots, P_m = p_{m_1}] f_{P'_{(m_0)}}(p) dp \\ &= \int_0^{p''} \frac{m_0}{m_0 + j_0} m_0 p^{(m_0-1)} dp = \left[\frac{m_0}{m_0 + j_0} p^{m_0} dp \right]_0^{p''} = \frac{m_0}{m_0 + j_0} (p'')^{m_0} \\ & \stackrel{(1.14)}{\leq} \frac{m_0}{m_0 + j_0} \left(\frac{m_0 + j_0}{m_0 + 1} q^* \right)^{m_0} = \frac{m_0}{m_0 + j_0} \frac{m_0 + j_0}{m_0 + 1} q^* (p'')^{m_0-1} = \\ & \frac{m_0}{m + 1} q^* (p'')^{m_0-1}. \end{aligned}$$

Para la segunda integral de (1.4), se consideran todas las hipótesis juntas, tanto verdaderas como falsas. Sean por tanto las hipótesis $H_{(i)}$ ordenadas por sus p-valores, cada una se podrá rechazar si $\exists k/i \leq k \leq m_0 + j - 1$ cumpliendo que $p_{(k)} \leq \left\{ \frac{k}{m+1} \right\} q^*$, o equivalentemente

$$\frac{p_{(k)}}{p} \leq \frac{k}{m_0 + j - 1} \frac{m_0 + j - 1}{(m + 1)p} q^*.$$

Bajo la condición de que $P'_{(m_0)} = p$, los $p_{(i)}/p = P'_{(i)}/p \in U(0, 1)$ independientes entre sí para $i = 1, \dots, m_0 - 1$, y los $p_{(i)}/p = p_i/p$ para $i = 1, \dots, j$ son valores correspondientes a la hipótesis nulas falsas entre 0 y 1. Usando la hipótesis de inducción, como se supone cierto para todo $m_0 + j - 1 = m' \leq m$, se tiene que que es un proceso similar al definido en (1.2) pero cogiendo como constante $\frac{m_0+j-1}{(m+1)p} q^* = \hat{q}^*$. De este modo, se tiene que

$$\begin{aligned} \mathbb{E}[Q | P'_{(m_0)} = p, P_{m_0+1} = p_1, \dots, P_m = p_{m_1}] & \stackrel{\text{inducción}}{\leq} \frac{m_0 - 1}{m_0 + j - 1} \cdot \frac{m_0 + j - 1}{(m + 1)p} \\ & = \frac{m_0 - 1}{(m + 1)p} q^*. \end{aligned}$$

Ahora, se tiene que

$$\begin{aligned} & \int_{p''}^1 \mathbb{E}[Q | P'_{(m_0)} = p, P_{m_0+1} = p_1, \dots, P_m = p_{m_1}] f_{P'_{(m_0)}}(p) dp \\ &= \int_{p''}^1 \frac{m_0 - 1}{(m + 1)p} q^* \cdot m_0 p^{m_0-1} dp = \frac{m_0}{m + 1} q^* \int_{p''}^1 (m_0 - 1) p^{m_0-2} dp \\ & \quad = \frac{m_0}{m + 1} q^* \left(1 - (p'')^{m_0-1} \right). \end{aligned}$$

Por lo tanto, se tienen los dos desigualdades de la integral (1.4), y de este modo

$$\mathbb{E}[Q \mid P_{m_0+1} = p_1, \dots, P_m = p_{m_1}] \leq \frac{m_0}{m+1} q^* (p'')^{m_0-1} + \frac{m_0}{m+1} q^* (1 - (p'')^{m_0-1}) = \frac{m_0}{m+1} q^*.$$

Probado de este modo para $m+1$ se demuestra mediante inducción el lema.

□

Teorema 1.2. *Para múltiples contrastes independientes entre si y para cualquier configuración de hipótesis nulas falsas, el procedimiento definido en (1.2) permite controlar el FDR en q^**

Demostración. ⁴

Supongamos que las hipótesis $m_1 = m - m_0$ son falsas. Sea cual sea la distribución conjunta de P_1'', \dots, P_{m_1}'' correspondientes a las hipótesis falsas, entonces se puede deducir lo siguiente. Aplicando la esperanza a ambos lados de la desigualdad (1.3), haciendo uso de la ley de esperanzas iteradas o ley de esperanzas totales [25], se obtiene lo siguiente:

$$\mathbb{E}[\mathbb{E}[Q \mid P_{m_0+1} = p_1, \dots, P_m = p_{m_1}]] \underset{\text{ley esp. iter.}}{=} \mathbb{E}(Q) \leq \mathbb{E}\left[\frac{m_0}{m} q^*\right] = \frac{m_0}{m} q^* \leq q^*.$$

Obteniendo por lo tanto un control del FDR mediante q^*

□

Se tendrá entonces un control sobre el FDR cometido cuando se realizan múltiples contrastes, haciendo que tras aplicar la corrección, entre todos los test considerados como significativos en favor de la hipótesis alternativa, la proporción de hipótesis nulas verdaderas (Error Tipo I) no supere el umbral prefijado.

El procedimiento de Benjamini-Hochberg controla el FDR, y en el caso extremo en que todas las hipótesis nulas son verdaderas, el FDR se reduce a la probabilidad de cometer al menos un falso positivo, lo que lo hace más conservador en ese escenario particular. Sin embargo, BH no es tan conservador como métodos como Bonferroni, y es más potente cuando hay hipótesis alternativas verdaderas. Esto no tiene por qué ser una desventaja, sino una característica a tener en cuenta para aplicar correctamente la corrección en un contexto determinado. Se tiene por lo tanto un método que elimina las desventajas de controlar únicamente el FWER, con una tasa de falsos negativos más reducida que en correcciones como la de Bonferroni, y que además aporta

⁴Demostración recogida en [3], ampliada para facilitar su comprensión.

una información relevante sobre cada test individual garantizando que el FDR no supere un valor prefijado por el usuario, que puede servir como umbral para considerar múltiples test como positivo o negativo en función de uno de los resultados individuales tras aplicar la corrección, o incluso mediante la combinación de varios de ellos.

1.3. Estimación no paramétrica mediante Regresión Local Lineal

En este trabajo se ha hecho uso de la estimación no paramétrica mediante regresión *kernel* local lineal en el proceso de la identificación del tipo de *drift*, haciendo uso de la función `KernelReg`, disponible en la librería `statsmodels` [20]. Específicamente, se ha utilizado para estimar la derivada de la función de interés, con el objetivo de utilizar el valor obtenido como criterio diferencial a la hora de interpretar el *drift* que se detecta. Los conceptos teóricos que aparecen esta función son muy extensos, por lo que se va a mostrar un resumen de la base fundamental para el funcionamiento de esta función de aproximación.

El método de regresión local no paramétrico se basa en estimar la esperanza condicional $\mathbb{E}[Y/X = x]$, donde $Y = r(x) + \epsilon$, sin asumir una forma funcional paramétrica para $r(x)$ [20]. En primer lugar es necesario introducir el estimador *kernel* de Nadayara-Watson [23]. Partiendo de la introducción de lo que es una función *kernel*, el cual hace referencia a cualquier función suave (al menos una vez derivable) $K(x) \geq 0$ tal que

$$\int K(x) dx = 1, \quad \int xK(x) dx = 0, \quad \sigma_K^2 \equiv \int x^2K(x) dx < \infty.$$

Existen varios tipos de kernels utilizados en la literatura; sin embargo, en este trabajo se hará especial énfasis en el *kernel* Gaussiano, ya que es el que se emplea en la función utilizada, y que garantiza suavidad, diferenciabilidad y un soporte infinito.

Dados los n pares de observaciones $(X_1, Y_1), \dots, (X_n, Y_n)$, siendo $r(x) = \mathbb{E}[Y/X = x]$. Fijado un $h > 0$, llamado ancho de banda, el estimador *kernel* de Nadayara-Watson, basado en un promedio ponderado de las observaciones Y_i dando pesos a los puntos cercano a X , se define como:

$$\hat{r}_n(x) = \sum_{i=1}^n l_i(x) Y_i.$$

Los pesos $l_i(x)$ son dados por la siguiente fórmula

$$l_i(x) = \frac{\left(\frac{Kx - X_i}{h}\right)}{\sum_{j=1}^n K\left(\frac{x - X_j}{h}\right)}.$$

Sin embargo, este estimador tiene diversos puntos débiles. En primer lugar, cuando se usa para estimar la función en puntos cercanos al límite de los valores posibles para x , se produce un error en la estimación al no haber valores en uno de los lados del intervalo de puntos cercanos. Por ejemplo, si $x \in [0, 1]$, cuando se usa para estimar en $x = 0$, el intervalo de puntos cercano se puede definir como $u \in (x - h, x + h)$ $h > 0$, pero en el lado izquierdo del intervalo no hay ningún valor posible, lo que conduce al denominado sesgo en los extremos. Además existe también el sesgo de diseño, provocado cuando la distribución de los datos de X es una función compleja. Con Nadayara-Watson se asume que la función estimada es constante alrededor de x , lo que es bastante inexacto en la realidad. Para solucionar estos inconvenientes, se define la regresión local polinómica.

Sea u los valores cercanos a x , se define el polinomio

$$P_x(u; a) = a_0 + a_1(u - x) + a_2 \frac{(u - x)^2}{2!} + \dots + a_p \frac{(u - x)^p}{p!}. \quad (1.5)$$

Así, la regresión local polinómica consiste en aproximar la función $r(x)$ en los puntos cercanos a x como

$$r(u) \approx P_x(u; a).$$

Estimando los $a = (a_0, \dots, a_p)^T$ eligiendo a los $\hat{a} = (\hat{a}_0, \dots, \hat{a}_p)^T$ que minimicen la suma del cuadrado de los errores ponderada como $\sum_{i=1}^n w_i(x) (Y_i - P_x(X_i; a))^2$, se obtiene que la estimación local de $r(u)$ es

$$\hat{r}_n(u) = P_x(u; \hat{a}).$$

En particular, si $u = x$, se tiene que

$$\hat{r}_n(x) = P_x(x; \hat{a}) = \hat{a}_0(x).$$

Para el caso de la regresión lineal en el que $p = 1$, y teniendo en cuenta que los pesos se definen de forma que tengan más importancia los puntos cercanos al estimado, se tiene que la regresión local lineal consiste en estimar $r(x)$ como

$$\hat{r}_n(u) = \hat{a}_0 + \hat{a}_1(u - x)$$

donde \hat{a}_0 y \hat{a}_1 se obtienen minimizando la suma ponderada de cuadrados

$$\sum_{i=1}^n K\left(\frac{X_i - x}{h}\right) (Y_i - a_0 - a_1(X_i - x))^2.$$

Como la regresión local lineal se ajusta a una recta local, se mejora la estimación en los extremos del dominio, mejorando el problema del sesgo en estos valores en Nadayara-Watson. Además, puede captar pendientes de forma local en la función $r(x)$, lo que hace que en funciones de distribuciones complejas se reduzca de forma significativa el error cometido en la estimación. Esto es bastante intuitivo, ya que en vez de aproximar una constante alrededor del punto u se estima un polinomio de grado uno. Por lo que se tiene una pendiente y la estimación se adapta mucho mejor a la función aproximada, especialmente cuando su pendiente cambia de forma rápida, como en funciones suavizadas.

Además, otro de los grandes beneficios frente a Nadayara-Watson es que permite obtener una estimación de la derivada de la función alrededor del punto x . Si se observa la ecuación (1.5), se ve que la estructura coincide con el desarrollo de un polinomio de Taylor de orden p . Por lo que el valor estimado \hat{a}_1 de la regresión local lineal representa la estimación de la derivada de $r(x)$ en ese punto. Esto, sumado a que se utiliza un *kernel* Gaussiano, que al ser diferenciable permite que la función $\hat{r}_n(x)$ varíe "suavemente", la estimación de la derivada es más exacta.

Por último, es necesario mencionar el método de validación cruzada por mínimos cuadrados para la selección automática del ancho de banda h , método que se utiliza en la función `KernelReg`. Para cada observación de X_i , esta se quita del conjunto de entrenamiento. Posteriormente, se estiman todos los $\hat{r}_n^{(-i)}(X_i)$ si el dato eliminado, y se calcula su error cuadrático $(Y_i - \hat{r}_n^{(-i)}(X_i))$. Se realiza el mismo proceso para todas las muestras con $i = 1, \dots, n$, para después sumar todos los errores y calcular así el ancho de banda que minimiza esta suma

$$\hat{h} = \arg \min_h \sum_{i=1}^n (Y_i - \hat{r}_n^{(-i)}(X_i))^2.$$

1.4. Enfoques y técnicas del aprendizaje automático

En esta sección se aportará una breve explicación acerca de los conceptos relevantes con el fin de contextualizar la problemática, ya que a lo largo de este estudio se ha profundizado en distintos aspectos del aprendizaje automático.

En primer lugar, se presentan los enfoques de aprendizaje supervisado y no supervisado, los cuales constituyen las dos principales metodologías dentro de la inteligencia artificial, y más concretamente del aprendizaje automático. Cabe destacar que el aprendizaje automático o *machine learning* es un subconjunto de la inteligencia artificial, enfocado en sistemas que aprenden patrones para tomar decisiones basadas en datos. Para ello, es necesario entrenar estos modelos, y la forma en que se realiza este entrenamiento es lo que diferencia el aprendizaje supervisado del no supervisado.

El aprendizaje supervisado emplea datos de entrenamiento etiquetados, es decir, datos en los que se conoce el valor real o deseado de la variable objetivo del modelo, lo cual permite guiar el proceso de aprendizaje. El objetivo principal es minimizar el error cometido frente a los valores esperados, encontrando relaciones entre las variables de entrada y salida. Por otro lado, el aprendizaje no supervisado se caracteriza por el uso de datos no etiquetados. En este caso, no se dispone de información sobre la variable objetivo, por lo que el modelo busca identificar patrones en los propios datos, sin supervisión externa. Por ejemplo, el aprendizaje no supervisado se utiliza para la exploración de datos, así como en tareas de agrupación en clústeres o detección de anomalías. El aprendizaje supervisado se centra en las tareas de clasificación y regresión, dando pie a los dos conceptos presentados a continuación.

Los algoritmos de clasificación se utilizan cuando el conjunto de datos pertenece a dos o más clases o categorías, y se necesita clasificar cada dato en una de ellas. Dentro de ellos se pueden encontrar múltiples tipos, como la regresión logística, los árboles de decisión o las redes neuronales para clasificación, entre otros. Por otro lado, los algoritmos de regresión tienen como objetivo predecir un valor continuo a partir del conjunto de datos. Ejemplos comunes de este tipo de algoritmos incluyen la regresión lineal, la regresión polinómica o las redes neuronales aplicadas a tareas de regresión.

Existen otras clasificaciones dentro del *machine learning*, pero no son objeto de estudio en este trabajo. No obstante, es necesario incluir dos enfoques muy específicos: el denominado *online machine learning* y el *incremental learning*. Mientras que en un modelo *batch* tradicional se realiza un entrenamiento previo al despliegue, quedando por tanto estático a lo largo del tiempo, el *online learning* cambia por completo esta filosofía. En este caso, el modelo se entrena con cada dato entrante del flujo continuo, a la vez que realiza una predicción o clasificación, adaptándose así en tiempo real a los datos. El caso más común se da en el aprendizaje supervisado, en el que el

modelo utiliza la información real para ajustarse con cada instancia. No obstante, también existen modelos *online* para aprendizaje no supervisado, cuyo funcionamiento resulta más complejo.

Relacionado con este enfoque se encuentra el *incremental learning*, en el que el modelo realiza reentrenamientos para actualizar sus parámetros a medida que recibe nuevos datos. En este caso, dicho reentrenamiento se efectúa por lotes de mayor tamaño que en el *online learning* y no en tiempo real. Se trata de un enfoque menos agresivo, que busca adaptar el modelo sin perder la información o el conocimiento previo.

El *Concept Drift* aparece de forma recurrente en estos escenarios, por lo que es necesario detectarlo y adaptarse correctamente según las necesidades del modelo. Los algoritmos utilizados en este estudio pertenecen a una librería diseñada para el *online learning*, aunque aquí se emplean únicamente para la detección del *drift*, sin aplicar el enfoque de aprendizaje *online*. Asimismo, aunque el trabajo se centra principalmente en la detección, uno de los entornos de prueba incluye un método de adaptación similar al *incremental learning*, mediante reentrenamiento con *mini batch*, por lo que resulta relevante conocer estos conceptos.

1.5. Concept Drift: Conceptualización y detección

En esta sección se presenta el fenómeno estudiado en este trabajo, el *Concept Drift*, a través de una explicación teórico-formal de cuáles son sus características principales y qué métodos de detección existen en la literatura.

En el contexto en el que se presenta un entorno dinámico, cambiante o no estacionario, puede ocurrir que las propiedades estadísticas de la variable objetivo o *target* cambien a lo largo del tiempo. Más concretamente, cuando se produce un cambio en la relación de las variables de entrada y/o la variable *target*, o dicho de otro modo, cuando la distribución de los datos varía a lo largo del tiempo [9] [4]. Sea t un instante temporal del intervalo $[0, T]$, X vector aleatorio de características de entrada e Y variable aleatoria objetivo, que puede ser un valor o un conjunto de clases. Sea (X_t, Y_t) una observación y $P_t(X_t = x, Y_t = y) = P_t(X, Y)$ la función de distribución de probabilidad conjunta de los datos de entrada y variable objetivo en el instante t , el *Concept Drift* se define formalmente como un cambio en la función de distribución de probabilidad conjunta:

$$\exists t \in [0, T) : P_t(X, Y) \neq P_{t+1}(X, Y).$$

La probabilidad conjunta se calcula como $P_t(X, Y) = P_t(Y|X)P_t(X)$, siguiendo las definiciones proporcionadas en la Sección 1.1, dependiendo de si se da el caso continuo, en cuyo caso se usa la función de densidad de probabilidad, o caso discreto, en el que se usa la función de masa.

Esta definición es válida tanto para problemas de regresión como para problemas de clasificación. Además, se puede hacer uso del Teorema de Bayes para completar la definición del *Concept Drift*. En general, se define:

$$P_t(Y|X) = \frac{P_t(X|Y)P_t(Y)}{P_t(X)} \quad (1.6)$$

en donde

- $P_t(Y)$: la probabilidad *a priori* de la variable aleatoria objetivo Y .
- $P_t(X|Y)$: la función de verosimilitud o la probabilidad de X en la hipótesis de Y .
- $P_t(X)$: la probabilidad marginal de X .
- $P_t(Y|X)$: la probabilidad *a posteriori*.

En el problema concreto de clasificación, en las que la variable objetivo toma valores discretos en el vector de clases (c_1, \dots, c_n) . Sea por lo tanto $P_t(c_i|X = x) = P_t(c_i|X)$, se define:

$$P_t(c_i|X) = \frac{P_t(X|c_i)P_t(c_i)}{P_t(X)}, \quad (1.7)$$

$$P_t(X) = \sum_{j=1}^n P_t(X|c_j)P_t(c_j)$$

en donde

- $P_t(c_i)$: la probabilidad *a priori* de la clase c_i .
- $P_t(X|c_i)$: la función de verosimilitud o la probabilidad de X en la hipótesis de c_i .
- $P_t(X)$: la probabilidad marginal de X .
- $P_t(c_i|X)$: la probabilidad *a posteriori*.

1.5.1. Tipos de Concept Drift

Existen diversas clasificaciones atendiendo a características distintas del fenómeno y a las probabilidades afectadas en las ecuaciones (1.6) y (1.7):

- **Según la probabilidad afectada:**

- **Real Concept Drift:** En este caso la frontera de decisión del modelo, o lo que es lo mismo, $P_t(Y|X)$ sufre un cambio sin que lo haga la $P_t(X)$. Para el caso concreto de clasificación, se da lugar cuando $P_t(c_i|X)$ cambia.
- **Virtual Concept Drift:** cuando se produce un cambio sin afectar a la frontera de decisión del modelo, es decir, cuando $P_t(X)$ o $P_t(X|Y)$ se ven afectadas. En el caso del problema de clasificación, se tiene que la función de verosimilitud $P_t(X|c_i)$ varía, además de la probabilidad marginal de X . Cabe destacar que en algunos artículos [17] se hace referencia a este fenómeno como *Data drift*, aunque de aquí en adelante se utilizará el término *Virtual*.
- **Class prior/ Target shift:** aunque podría ser considerado un caso concreto de *virtual drift*, en [8] lo sitúa como un escenario diferente. De forma general, se produce cuando la $P_t(Y)$ cambia. En el contexto de clasificación, se da cuando la probabilidad $P_t(c_i)$ se ve afectada. Puede dar lugar a un desbalanceo de clases, a la fusión de clases existentes o incluso a la aparición de una nueva clase.

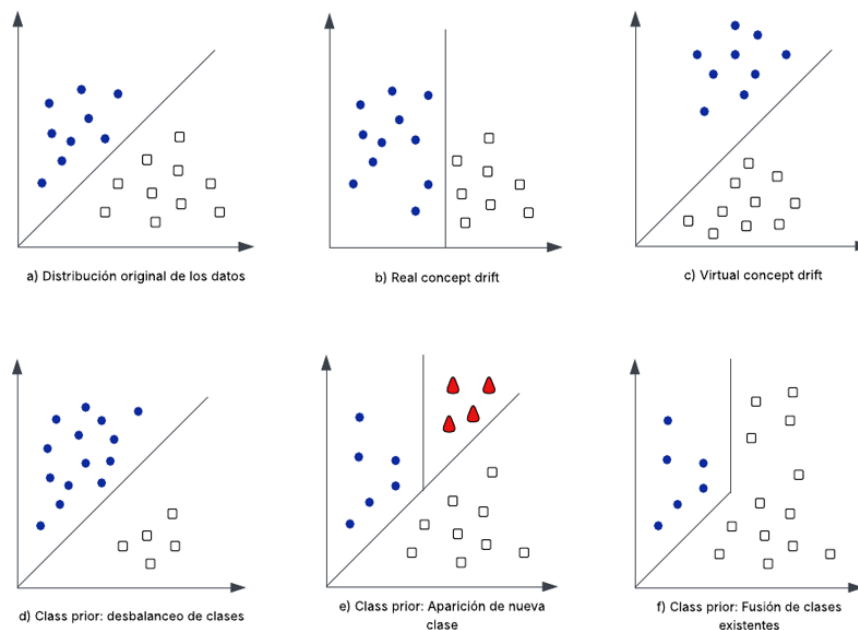


Figura 1.3: Tipos de *Concept Drift* según la probabilidad afectada en un modelo de clasificación, reproducida de [2]. En la fila superior se representa la distribución original de los datos (a), junto con los casos de *Real Drift* (b) y *Virtual Drift* (c). En la fila inferior se ilustran distintas situaciones del *Class Prior Shift*

- Según la velocidad del cambio:

- ***Abrupt Drift***: Se da lugar cuando el cambio de concepto ocurre de forma brusca o repentina. A partir del punto de *drift*, el viejo concepto desaparece y solo se presenta el nuevo.
- ***Incremental Drift***: En este caso, se produce un cambio de concepto durante un periodo de tiempo, hasta que se considera que el nuevo concepto está presente y estabilizado. Se puede considerar que el modelo pasa por múltiples conceptos intermedios antes de alcanzar el final.
- ***Gradual Drift***: El cambio de concepto no ocurre de forma repentina, sino que se producen pequeños cambios entre el viejo y el nuevo, hasta que a partir de un punto concreto el nuevo se manifiesta de forma constante. La interpretación y detección de este tipo de drift es compleja, ya que podrían manifestarse anomalías o *outliers* durante la evaluación del modelo, sin llegar a materializarse un cambio de concepto.
- ***Recurrent Drift***: Se da lugar cuando se produce un cambio de concepto, pero en un espacio temporal lo suficientemente pequeño vuelve a cambiar al antiguo concepto. La frontera de decisión entre considerar dos *abrupt drift* o un *recurrent drift* no está muy clara, y habría que adaptarlo a cada estudio en concreto. Incluso podría darse lugar un *recurrent drift* combinado con un *gradual* o *incremental drift*, lo que hace este caso el más difícil de tratar y detectar.

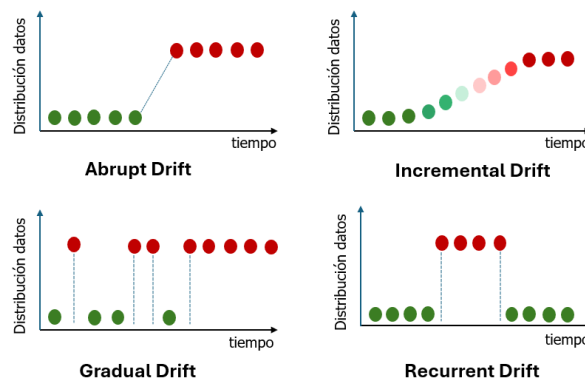


Figura 1.4: Tipos de *Concept Drift* según la velocidad del cambio, reproducida de [9]

■ Según la región de afectación:

- ***Local Drift* o *Regional Drift***: cuando el cambio ocurre únicamente en una región o en algunas instancias del conjunto de datos. Suele ser difícil de detectar, ya que es habitual confundirlo con *outliers*. En este estudio no se profundiza en esta tipología.
- ***Global Drift***: cambio que afecta a todo el conjunto de datos, y por lo tanto, suele ser más sencillo de detectar.

1.5.2. Detección de Concept Drift

Teniendo en cuenta la clasificación anterior, uno de los pasos más importantes a la hora de monitorizar el modelo para detectar el fenómeno es seleccionar adecuadamente los datos de entrada para el algoritmo que se va a encargar de la detección de *drift*. Las posibilidades en el aprendizaje supervisado, en el que se conoce el resultado real de la predicción en cada iteración, las opciones son más amplias que en aprendizaje no supervisado. En función de las variables analizadas se estará evaluando la aparición de *drift* según la probabilidad afectada en las ecuaciones (1.6) y (1.7):

- **Virtual Drift:** En el caso de monitorizar los *datos de entrada* del modelo, se detectarán cambios en $P_t(X)$. Para el caso del aprendizaje supervisado, conociendo la predicción o la clase perteneciente, se podría percibir el cambio en $P_t(X|Y)$ o $P_t(X|c_i)$ respectivamente.
- **Class Prior / Target Shift:** evaluando la salida del modelo se percibirían variaciones en $P_t(Y)$ o $P_t(c_i)$ en clasificación.
- **Real Drift:** aplicable solo al aprendizaje supervisado, evaluando métricas de rendimiento del modelo, se detectarían cambios en $P_t(Y/X)$ o $P_t(c_i/X)$

La diferenciación de *local* y *global drift* es algo más compleja, disponible solo en algunos algoritmos de detección, como los árboles de decisión, por lo que no será objeto de estudio en este trabajo. Existen sistemas capaces de monitorizar distintas variables para identificar diferentes variables para determinar cuando ocurre un tipo de *drift* según la probabilidad y no otro, aunque su revisión queda exento también queda exenta de los objetivos de este trabajo.

Una vez determinadas las variables a monitorizar, se procede al cálculo del estadístico de prueba y al contraste de hipótesis para evaluar si se ha producido un drift. Según la clasificación presentada en [9], existen diferentes tipos de detectores: algunos se basan en la tasa de error generada por el modelo de aprendizaje automático, utilizando este valor para realizar el contraste; otros se apoyan en la comparación de las distribuciones de dos ventanas temporales diferentes para detectar cambios; y también existen detectores que emplean varios contrastes de hipótesis, llevando a cabo pruebas diferentes sobre varios parámetros, ya sea en una única etapa de detección o distribuidos entre una capa de detección y otra de validación. Para el problema de la detección de *Concept Drift*, de forma general se define el siguiente contraste, atendiendo a la ecuación (1.1):

Hipótesis nula (H_0): no hay *drift*

Hipótesis alternativa (H_1): hay *drift*.

En el contexto de la detección del *Concept Drift*, la aparición del Error Tipo I conllevaría diversos problemas, como un reentrenamiento del modelo innecesario y sobreajuste a los últimos datos con la pérdida de importancia de los anteriores, fenómeno denominado *catastrophic forgetting*, además de una bajada de rendimiento del modelo con un gasto de recursos computacionales extra. Estos conceptos se han presentado previamente, pero se recuerdan nuevamente debido a su relevancia en el tratamiento de este fenómeno. Todos estos argumentos hacen que focalizarse en reducir la aparición de los falsos positivos sea un aspecto clave, aunque sin descuidar la capacidad de detección del algoritmo. Como se ha visto en la Sección 1.2, al reducir la tasa de aparición del Error Tipo I, aumenta la probabilidad de aparición del Error Tipo II, teniendo por lo tanto un contraste menos potente y así menos capacidad de detección. La corrección de BH mencionada, la cual realiza un control sobre el FDR, se adapta adecuadamente a los requisitos de este problema, reduciendo la tasa de falsos positivos sobre el total de positivos detectados, pero sin perder mucha capacidad de detección.

Por último, es importante señalar que no hay una metodología definida para detectar los diferentes tipos de *drift* según su velocidad, siendo un aspecto en desarrollo en muchos de los algoritmos existentes. En [9] se mencionan estrategias como el *clustering* para caracterizar diferentes patrones y así identificar distintos tipos de *drift*. En [24] se menciona la posibilidad de utilizar la distancia de Kullback-Leibler para diferenciar la severidad con la que ocurre el *drift*, identificando entre *abrupt* y *gradual*, aunque no se aporta un método definido. Por lo tanto, el desarrollo de metodologías claras y efectivas para la caracterización del *drift* sigue siendo un desafío y está en constante evolución.

1.6. Métricas utilizadas en la experimentación

En esta sección se presentan las métricas utilizadas en las pruebas llevadas a cabo en la experimentación en función de las configuraciones que se presentarán posteriormente del algoritmo. Se tiene tanto el caso tanto de datos procedentes de distribuciones continuas como el caso de métricas continuas como datos de entrada en el detector de *drift*. Son utilizadas para evaluar la eficacia de la utilización de la modificación propuesta en el algoritmo de detección, además de evaluar el rendimiento en modelos de clasificación y predicción utilizando el detector.

1.6.1. Mediciones en la modificación para datos de entrada continuos en el detector

Estas métricas se han utilizado en la monitorización un conjunto de datos, que sigue una distribución, continua generado artificialmente, simulando la detección del *Virtual Concept Drift*,

y en el seguimiento del error absoluto cometido en la predicción de un modelo, simulando la detección de *Real Concept Drift*. La notación para las fórmulas se sigue de la presentada en la Tabla 1.2.

- **FDR (*False Discovery rate*) basado en el número total de detecciones:** Evalúa la proporción de falsas alarmas frente a todos los cambios detectados. Es la medida más importante en el estudio, siendo relevante para determinar si las modificaciones funcionan según lo esperado. $FDR = \frac{V}{V+S} = Q$.⁵
- **MDR (*Missed Detection Rate*):** indica la probabilidad de que el algoritmo no detecte un cambio cuando realmente está sucediendo, el Error Tipo II. $MDR = \frac{T}{T+U}$.
- **Tiempo_ejecucion:** con el objetivo de evaluar la eficiencia computacional en cuanto a tiempo, se ha medido el tiempo que tarda la ejecución de las pruebas, viendo así la rapidez del algoritmo para procesar los datos de entrada.
- **Error absoluto:** error absoluto cometido en la predicción del modelo, calculado con el resultado de la predicción y el valor real. Error absoluto = $|y_i - \hat{y}_i|$, donde y_i es el valor real y \hat{y}_i es la predicción.
- **MAE:** el error medio absoluto. $MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$.

1.6.2. Mediciones en la modificación para métricas como datos de entrada en el detector

En este caso se utilizan las mismas métricas que en el TFG del Grado en Informática. La modificación propuesta para monitorizar una métrica continua está pensada para integrarla en el estudio mencionado y ampliar la comparativa realizada. Resulta, por tanto, razonable emplear las mismas métricas.

- **FDR y MDR:** mismos ratios que en el caso anterior.
- **MTD (*Mean Time to Detect*):** en segundos, evalúa lo rápido que reacciona el sistema ante el cambio producido. Está condicionada a la rapidez del algoritmo, pero puede ser interesante para evaluar qué detector es más rápido también en términos de eficiencia computacional

⁵Aunque se utilice la misma notación que en la Tabla 1.2, el lector de este trabajo no debe confundir los valores dentro de un contraste múltiple con los valores de todos los contrastes múltiples realizados. A modo de ejemplificación, para cada muestra del flujo de datos entrante, se le aplica un contraste múltiple con las muestras previas. En este caso V hace referencia a todos los falsos positivos tras aplicar todos los contrastes múltiples en el flujo de datos.

- **IR (*Identification Rate*)**: ratio para medir el número de identificaciones de *drift* que se realizan correctamente con respecto al total de los cambios de conceptos verdaderos detectados. No se incluye la detección de los falsos positivos, ya que a efectos de medir la efectividad en la identificación no aporta información de utilidad.

Por otro lado, las métricas de rendimiento utilizadas en el modelo de clasificación, las cuales pueden servir como datos de entrada para el algoritmo de detección y como medida de seguimiento de la utilización del algoritmo de detección de *Concept Drift*, son las siguientes.

- **Accuracy**: medida muy utilizada para modelos de clasificación. Mide la proporción de las predicciones correctas entre el total de predicciones realizadas. No es muy adecuada para el caso del desbalanceo de clases, aunque sí una buena referencia para casos generales. $Accuracy = \frac{S+U}{S+U+V+T}$.
- **Cohen Kappa**: mide el nivel de acuerdo entre el modelo de predicción y el resultado real, corrigiendo el acuerdo esperado por azar. $\kappa = \frac{P_o - P_e}{1 - P_e}$. Con P_o proporción de acuerdo y P_e probabilidad de acuerdo por azar. Es una medida
- **Precision**: mide la proporción de positivos reales del modelo. $Precision = \frac{S}{V+S}$.
- **F1**: media armónica entre la *Precision* y *Recall*. Es más adecuada cuando hay clases desbalanceadas, aportando más información que el *Accuracy* en esos casos. $F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$. Destacar que el *Recall* se puede calcular como $Recall = \frac{S}{S+T}$.

En principio cualquiera de las métricas de rendimiento presentadas anteriormente se puede utilizar para evaluar la tendencia de un modelo de clasificación a lo largo del tiempo. Para este estudio, específicamente en el caso de la integración de la modificación propuesta en el trabajo informático, se ha elegido el *Accuracy* como referencia, utilizando sus valores como datos de entrada de una de las modificaciones propuestas. Es una medida muy reactiva ante cambios, disminuyendo sus valores cuando el modelo deja de clasificar o predecir correctamente. Sin embargo, se ha observado que puede no tener un comportamiento significativo y útil para este caso, analizando los resultados obtenidos en el estudio informático en la medida del rendimiento del modelo de clasificación.

En la Figura 1.5 se puede observar cómo se comporta el *Accuracy* del modelo a lo largo del procesamiento del flujo de datos. Puede ocurrir, como en este caso, que la métrica sea muy alta al principio y que decaiga muy lentamente durante un tiempo prolongado, hasta estabilizarse en torno a un valor. En este caso no está ocurriendo realmente un cambio de concepto, sino que los primeros valores elevados condicionan esa pequeña caída durante los próximos ejemplos. Por

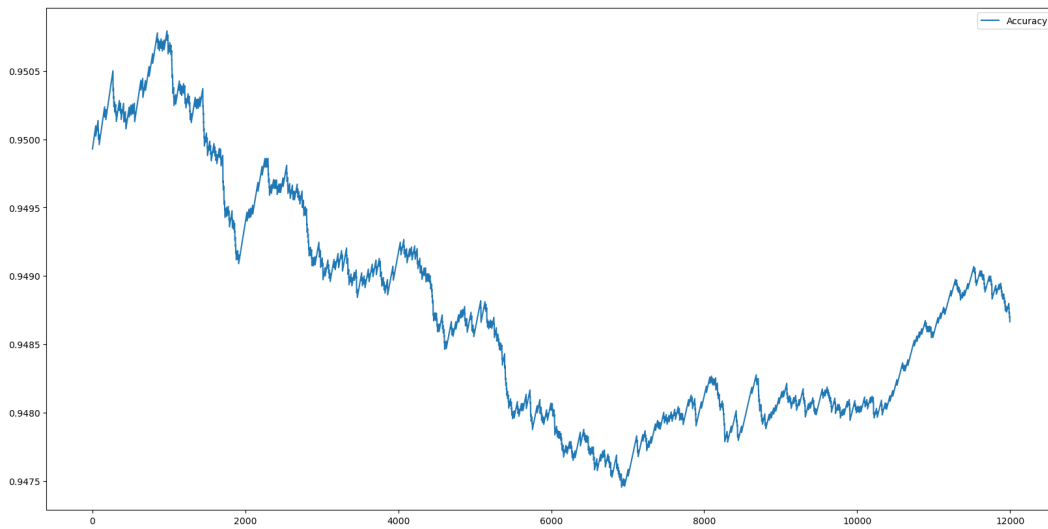


Figura 1.5: Accuracy del modelo

ello, se ha decidido realizar una modificación en esta métrica, añadiendo una ventana deslizante con un tamaño indicado por el usuario, manteniendo solo los últimos resultados del modelo. De esta forma, se obtiene un resultado que refleja con más clarividencia la tendencia del modelo.

Otro de los problemas detectados de usar esta métrica, y en general de la gran mayoría de ellas, es que aunque sobre la teoría sean métricas con valores continuos, cuando la ventana de datos es pequeña se obtienen unas mediciones discretizadas, especialmente al usar el método de la ventana deslizante, con muchas mediciones que se repiten. Para solucionar este problema se ha aplicado un suavizado exponencial *EWMA* (*Exponentially Weighted Moving Average*), específicamente utilizando la siguiente fórmula:

$$S_t = \lambda X_t + (1 - \lambda)S_{t-1} \quad (1.8)$$

donde:

- S_t es el valor suavizado en el tiempo t .
- X_t es el valor actual (sin suavizar).
- S_{t-1} es el valor suavizado anterior.
- λ es el factor de suavizado, definido como: $\alpha = \frac{2}{\text{span}+1}$

El usuario puede indicar el *span* deseado, el cual indica de forma aproximada el número de muestras previas que influyen en el cálculo del suavizado, consiguiendo un suavizado que se

adapta más lentamente a los datos cuando se indica un *span* grande, y viceversa. Con estos dos cambios se consiguen las condiciones necesarias para que la test *KS* funcione correctamente. Se hará referencia a lo largo de este estudio a la modificación de esta métrica como *Accuracy* Modificado. En la Figura 1.6 se muestra una comparativa entre la métrica original y la modificación, teniendo este último un comportamiento más oscilante pero que refleja de mejor forma los resultados recientes.

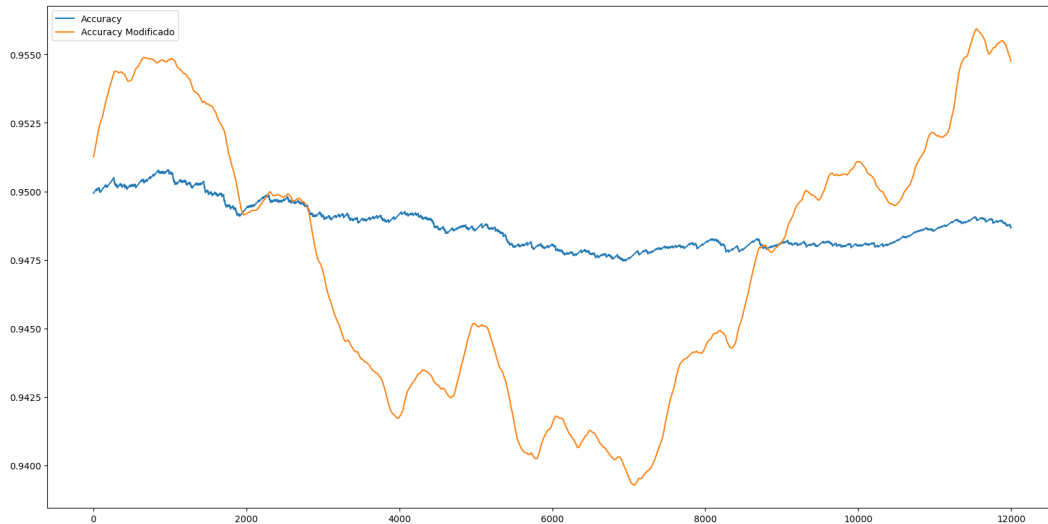


Figura 1.6: *Accuracy* Modificado del modelo. Línea azul: *Accuracy* original. Línea naranja: *Accuracy* modificado

Tanto la utilización de una ventana deslizando como el suavizado se puede aplicar a cualquiera de las métricas utilizadas en la materia. La posibilidad del suavizado se ha incluido en el propio algoritmo, que lo aplica si el usuario indica que los datos no son continuos. Realizar una ventana deslizando queda en responsabilidad del usuario, al no poder adaptar un cálculo que se adapte a todas las métricas.

Capítulo 2

Algoritmo y modificaciones

Para la detección y el tratamiento del *Concept Drift* existe múltiples librerías e implementaciones de algoritmos con los que se pueden abordar esta problemática. En este caso se ha hecho uso de la librería [13], diseñada en código Python. La librería tiene como uso principal el *online machine learning*, aunque como se ha comentado previamente, únicamente se va a hacer uso de sus algoritmos para la detección del *Concept Drift*, y no para aplicar una metodología *online* en el modelo de aprendizaje automático.

De este modo, se ha analizado uno de los detectores presentes en la librería, el KSWIN, que utiliza como base teórica el test de *Kolmogorov Smirnov* presentado en la Sección 1.2, específicamente la prueba de homogeneidad entre dos conjuntos de datos. El algoritmo está implementado para monitorizar datos de entrada continuos. De acuerdo con las clasificaciones presentadas en la Sección 1.5.1, la utilización de este algoritmo de forma individual permitiría detectar *Global Drift*, en función de la región afectada; *Real Drift* o *Virtual Drift*, dependiendo de si se ve alterada la distribución conjunta o marginal (lo que puede identificarse mediante la monitorización continua del rendimiento del modelo o de los datos de entrada, respectivamente), y cualquier tipo de *drift* en función de su velocidad de aparición.

Revisando el código fuente, disponible en el repositorio de *Github* [14], se puede observar cómo hace uso de la función `ks_2samp` de la librería *Scipy* [18]. El usuario puede elegir el nivel de significación del contraste de hipótesis, además del tamaño de la ventana de datos sobre el que se va a realizar el test y una ventana estadística¹ para realizar la comparación. Fijado un tamaño de ventana, el test se realiza con los últimos datos determinados por el tamaño de la ventana estadística frente al mismo número de datos elegidos aleatoriamente de los anteriores, siempre y cuando estén en la ventana de datos fijada. En la Figura 2.1 se muestra una representación gráfica de cómo se realiza el test en esta configuración original del detector, proseguido del pseudocódigo

¹Se emplea la notación de la variable disponible en [13].

1 del algoritmo.

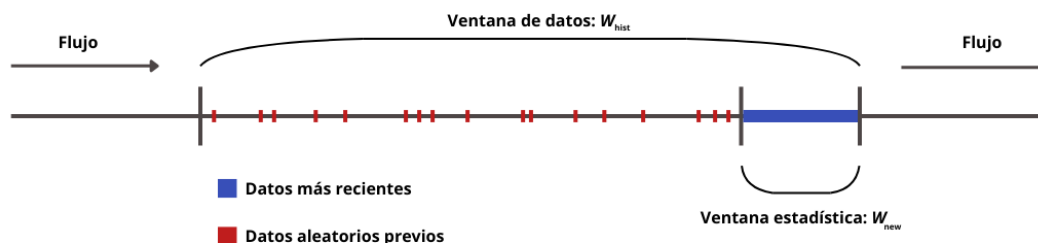


Figura 2.1: Funcionamiento de la configuración original del KSWIN

Con esta metodología hay un aspecto que no se tiene en cuenta. Supongamos que el detector de *drift* se actualiza dato a dato, modificando un valor tanto en W_{hist} como en W_{new} . Cuando se realiza el test, se obtiene un resultado para determinar si hay evidencias estadísticamente significativas de que la distribución de ambas ventanas es distinta. Sin embargo, no se tiene en cuenta los resultados obtenidos previamente, que pueden influir en que realmente haya un cambio de concepto o sea un resultado positivo puntual. Esta situación es el denominado contraste de hipótesis múltiple, explicado en la Sección 1.2.

Además, para que se cumpla la condición, y por lo tanto se detecte un *drift*, se tiene que producir que el estadístico calculado en el contraste sea mayor a 0,1. Esto hace que el procedimiento no esté bien calibrado, ya que no depende únicamente del nivel de significación elegido por el usuario. Aunque no se aporta ninguna explicación en la documentación de la implementación, es probable que se definiese este umbral para tratar la sensibilidad del contraste. Tal y como se vio en la Sección 1.2, cuando el tamaño muestral aumenta, el test se vuelve más sensible, y por lo tanto es necesario una distancia menor entre las distribuciones para obtener pruebas estadísticamente significativas en favor de la hipótesis alternativa, es decir, que las muestras siguen distribuciones diferentes. Por ejemplo, haciendo uso de la Tabla I.1, fijado un nivel de significación $\alpha = 0,01$ y con un tamaño muestral de 1000, el p-valor obtenido será menor que 0,01 cuando el estadístico obtenido sea mayor que aproximadamente a 0.051, límite mucho menor que para un tamaño muestral de 50, para el cual el p-valor será menor a 0.01 cuando el estadístico sea mayor que 0.226. Fijando el umbral para el estadístico a 0.1 se consigue corregir la sensibilidad del test para muestras de tamaños muy grandes. Sin embargo, esta metodología conlleva a la obtención de un test mal calibrado. Aceptar o rechazar la hipótesis nula fijando un nivel de significación α y asumir un límite de $k = 0,1$ para el estadístico, cuando este último varía en función del tamaño muestral n y del α escogido, no se sustenta en ninguna justificación teórica

Algoritmo 1 KSWIN*Estado 1 – Inicializacion*

Input: α : nivel de significación para la prueba Kolmogorov-Smirnov, `window_size`: tamaño de la ventana deslizante, `stat_size`: tamaño de la ventana estadística, `seed`: semilla para garantizar la reproducibilidad de los resultados, `window`: datos pre-colectados opcionales para evitar el arranque en frío

Output: $h_0(\mathbf{x}; \alpha, \text{window_size}, \text{stat_size}, \text{seed}, \text{window})$: modelo inicializado (detector)

Estado 2 – Actualizacion

Input: x : dato individual de entrada

Output: detector actualizado

- 1: Añadir x a la ventana deslizante
- 2: **if** tamaño de `window` \geq `window_size` **then**
- 3: `rnd_window`: seleccionar r muestras aleatorias de los primeros `window_size` – `stat_size` elementos en la ventana
- 4: `most_recent`: obtener las `stat_size` muestras más recientes de la ventana
- 5: Realizar la prueba bilateral Kolmogorov-Smirnov con `rnd_window` y `most_recent`, obteniendo `p_value` y `st` (distancia máxima entre distribuciones).
- 6: **if** `p_value` $\leq \alpha$ y `st` $> 0,1$ **then**
- 7: Detectar cambio: `drift_detected` \leftarrow True
- 8: Actualizar la ventana con las muestras más recientes
- 9: **else**
- 10: No detectar cambio: `drift_detected` \leftarrow False
- 11: **end if**
- 12: **else**
- 13: No hay suficientes muestras para realizar una prueba válida: `drift_detected` \leftarrow False
- 14: **end if**

y por lo tanto se ha decidido prescindir de esta condición. Por otro lado, en muchas situaciones se producen cambios sutiles en la distribución de los datos cuya detección resulta especialmente relevante en el contexto de esta experimentación. Establecer un umbral mínimo para el estadístico de Kolmogorov-Smirnov podría llevar a descartar precisamente estos cambios, limitando así la sensibilidad del análisis. Por esta razón, en las modificaciones propuestas se ha optado por eliminar dicho umbral.

A partir de este punto, las modificaciones aplicadas varían según el tipo de problema en el que se utilice el detector. En un primer lugar se elaboró una serie de cambios para un caso general, en el que los datos de entrada siguen una distribución continua, condición necesaria para el test de Kolmogorov-Smirnov. En segundo lugar se ha desarrollado una modificación

orientada específicamente a detectar el *Real Concept Drift* presentado en la Sección 1.4 en un modelo de clasificación, monitorizando una métrica de rendimiento. El código correspondiente a los algoritmos de detección se puede encontrar en el repositorio de [GitHub KSWIN Modification REPO](#).

2.1. Modificación para datos procedentes de distribuciones continuas

La utilidad de esta modificación propuesta es similar a la del algoritmo original. Según la región afectada que se monitorice se puede identificar *Global Drift*. Por otro lado, es posible detectar *Real Drift* si se utiliza sobre una métrica de rendimiento continua del modelo o *Virtual Drift* si se despliega sobre los datos de entrada. Asimismo, está preparado para detectar cualquier cambio de concepto según su velocidad de aparición, aunque no se va a proporcionar un método para la identificación del tipo de *drift*. En la siguiente modificación se presentará el método propuesto, el cual depende en gran medida del conocimiento previo sobre el funcionamiento de la métrica de rendimiento empleada. Por ello, es un aspecto susceptible de mejora e investigar en futuros estudios que puedan ampliar esta metodología.

En primera instancia se ha introducido el concepto de los contrastes múltiples, manteniendo en cierto modo el funcionamiento del algoritmo original. Es necesario recordar que el contraste entre la muestra de datos más reciente y una muestra anterior puede no ser suficiente para determinar si la distribución de datos que siguen ambas es similar o diferente. Por ello, en lugar de realizar un único test entre W_{new} y datos aleatorios anteriores pertenecientes a W_{hist} , se ha realizado tantas pruebas como el tamaño del conjunto W_{new} . De esta forma se pretende reducir la obtención de un falso positivo provocado por el conjunto aleatorio seleccionado, abarcando así la gran mayoría de la ventana W_{hist} .

Sin embargo, en este contexto surge el problema de los contrastes múltiples presentado anteriormente en la Sección 1.2. Al realizar diversas pruebas entre W_{new} y varias muestras pertenecientes a W_{hist} , la probabilidad de obtener un falso positivo en una prueba individual de entre todas aumenta. Se recuerda además que es crucial reducir la tasa de falsos positivos sin perder potencia en el contraste para no reducir la capacidad de detección de los *drift* cuando ocurren, por lo que la corrección de BH controlando el FDR es especialmente adecuada para este caso. El criterio para determinar un positivo, tras realizar todos los contrastes pertinentes, consistirá en verificar si alguno de los p-valores corregidos mediante BH es menor que el nivel de significación fijado por el usuario. Destacar que únicamente es necesario que uno de ellos sea menor que el nivel, por lo que puede llegar a ser muy sensible si una de las muestras empleada en el contraste es diferente al resto de ellas. Esta situación está controlada mediante el FDR para

evitar múltiples falsos positivos, y no se pierde la capacidad de detección del contraste. Ampliar el criterio para considerar un positivo, exigiendo la obtención de varios p-valores corregidos por debajo del nivel prefijado, puede llevar a una situación en la que no se detecte ningún cambio de concepto, especialmente si la naturaleza de los datos es muy oscilante.

Algoritmo 2 KSWIN Modificado (configuración 1)

Estado 1 – Inicialización

Input: mismos parámetros que el KSWIN (en este caso el nivel de significación para el FDR $\bar{\alpha}$), **window_start:** índice a partir del cual se empieza a hacer la detección, **alternative:** tipo de contraste KS

Output: $h_0(\mathbf{x}; \bar{\alpha}, \text{window_size}, \text{stat_size}, \text{seed}, \text{window}, \text{window_start}, \text{alternative})$: modelo inicializado (detector)

Estado 2 – Actualización

Input: \mathbf{x} lote de datos de entrada

Output: Detector actualizado

```

1: if window_start  $\leq$  0 then
2:   Añadir  $\mathbf{x}$  a la ventana deslizante
3:   if tamaño de window  $\geq$  window_size then
4:     for cada  $i$  en el rango de stat_size do
5:       Seleccionar less_recent como una muestra aleatoria de stat_size elementos de
       los primeros window_size - stat_size elementos de la ventana
6:       Seleccionar most_recent como las stat_size muestras más recientes de la ventana
7:       Realizar la prueba de Kolmogorov-Smirnov entre most_recent y less_recent
8:       Calcular p_value y añadir a la lista p_values
9:     end for
10:    Aplicar corrección de Benjamini-Hochberg a p_values y obtener corrected_p_values
11:    if alguna  $p$  en corrected_p_values  $\leq$   $\bar{\alpha}$  then
12:      Detectar cambio: drift_detected  $\leftarrow$  True
13:    end if
14:  else
15:    No hay suficientes muestras para realizar una prueba válida: drift_detected  $\leftarrow$  False
16:  end if
17: else
18:   window_start = window_start - 1
19: end if

```

Es necesario señalar que se mantiene la notación de $\bar{\alpha}$ para determinar el nivel de significación

para el FDR, evitando la confusión con el α de un contraste de hipótesis individual. Se hará referencia a este detector como **KSWIN Modificado (configuración 1)**, con el pseudocódigo 2.

Por otro lado, se ha estudiado una alternativa a esta primera configuración. Entendiendo la naturaleza tanto del test realizado como de los datos de entrada, la posibilidad de aparición de ruido (valores atípicos) en estos últimos puede influir en los resultados del contraste realizado. Recordemos que se comparan las funciones de distribución acumuladas empíricas de ambos conjuntos. De este modo, al introducir un componente aleatorio en la selección de la ventana de datos menos recientes, puede condicionar negativamente los resultados obtenidos. Por ello, se ha probado una segunda configuración, comparando lotes de datos previos secuenciales pertenecientes a W_{hist} , del tamaño de la ventana de datos más recientes, con la propia ventana W_{new} , realizando de nuevo múltiples contrastes entre la muestra de datos más reciente y muestras previas.

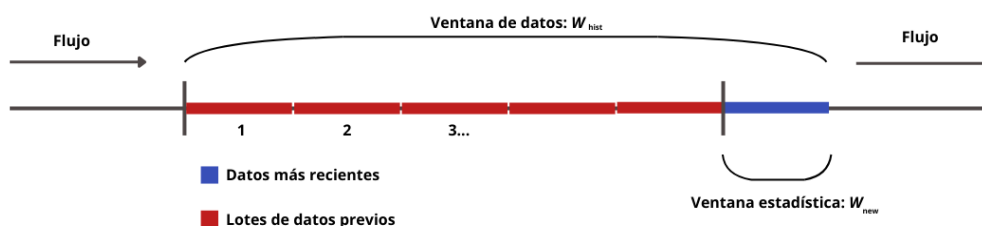


Figura 2.2: Funcionamiento del KSWIN Modificado (configuración 2)

En la Figura 2.2 se ve ilustrado el comportamiento del detector. Se van comparando en el orden indicado los lotes de datos previos con la ventana estadística, observando así si la tendencia de esta última ha variado con respecto a los datos anteriores. Además, esta configuración es mucho más eficiente en tiempo que la primera, ya que el número de comparaciones realizadas se reduce drásticamente, aumentando la rapidez de evaluación del algoritmo. Cabe destacar que la corrección de BH se mantiene para controlar el FDR cometido. En adelante se hará referencia a esta modificación como **KSWIN Modificado (configuración 2)**, con el pseudocódigo 3 mostrado a continuación.

Ambas configuraciones son únicamente compatibles con datos de entrada continuos, debido a la naturaleza del test de Kolmogorov-Smirnov. Puede servir para monitorizar tanto flujos de datos para detectar un posible *Virtual Drift* como para el error cometido en un modelo predictivo y así visualizar un posible *Real Concept Drift*, como se ha comentado previamente.

Algoritmo 3 KSWIN Modificado (configuración 2)

Estado 1 – Inicialización

Input: mismos parámetros que el KSWIN Modificado (configuración 1)

Output: $h_0(\mathbf{x}; \bar{\alpha}, \text{window_size}, \text{stat_size}, \text{seed}, \text{window}, \text{window_start}, \text{alternative})$:
modelo inicializado (detector)

Estado 2 – Actualización

Input: \mathbf{x} lote de datos de entrada

Output: Detector actualizado

```
1: if window_start  $\leq$  0 then
2:   Añadir  $\mathbf{x}$  a la ventana deslizante
3:   if tamaño de window  $\geq$  window_size then
4:     for cada  $i$  en el rango de  $\left(\frac{\text{window\_size}}{\text{stat\_size}}\right) - 1$  do
5:       Seleccionar less_recent como una sublista de la ventana desde  $i \times \text{stat\_size}$ 
        hasta  $(i + 1) \times \text{stat\_size}$ 
6:       Seleccionar most_recent como las stat_size muestras más recientes de la ventana
7:       Realizar la prueba de Kolmogorov-Smirnov entre most_recent y less_recent
8:       Calcular p_value y añadir a la lista p_values
9:     end for
10:    Aplicar corrección de BH a p_values y obtener corrected_p_values
11:    if alguna  $p$  en corrected_p_values  $\leq \bar{\alpha}$  then
12:      Drift detectado: drift_detected  $\leftarrow$  True
13:    end if
14:  else
15:    No hay suficientes muestras para realizar una prueba válida: drift_detected  $\leftarrow$  False
16:  end if
17: else
18:   window_start = window_start - 1
19: end if
```

2.2. Modificación para métricas de rendimiento continuas como datos de entrada en el detector

Esta modificación está orientada a integrarla en una comparativa realizada en el estudio informático, como se ha mencionado previamente, en la que se evalúa el rendimiento de un modelo de clasificación. Está pensada exclusivamente para detectar *Global Drift* y *Real Drift*, al focali-

zarse únicamente sobre una métrica de rendimiento y evaluando así únicamente la probabilidad $P_t(c_i/X)$ al tratarse de un modelo de clasificación. Además, podrá identificar tanto *abrupt* como *gradual* e *incremental drift*, excluyendo el caso del *recurrent*, debido al sistema de confirmación del cambio de concepto que se presentará posteriormente.

Es importante destacar que en la comparativa del estudio mencionado se utiliza directamente la clasificación del modelo y su etiqueta real como datos de entrada para los detectores, ya que la mayoría están pensados para soportar datos discretos. Esto es un inconveniente a la hora de probar una de las modificaciones propuestas anteriormente, ya que la continuidad de los datos es un requisito indispensable para el contraste de Kolmogorov-Smirnov. Lo cierto es que en la librería RiverML no se especifica este detalle y genera una confusión sobre el tipo de muestras que se pueden utilizar en el algoritmo. Para solventar esta problemática se ha decidido hacer uso del *Accuracy Modificado* presentado en la Sección 1.6.2, a través del cual se podrá detectar el *Real Concept Drift* con muestras de datos continuas, cumpliendo así los requisitos de la comparativa y de el contraste utilizado.

Se han investigado diversas formas de evaluar la ventana de datos W_{new} frente a los valores de W_{hist} . Tras realizar distintas pruebas con el KSWIN Modificado (configuración 1) y el KSWIN Modificado (configuración 2), utilizando el *Accuracy Modificado* como datos de entrada, no se han obtenido resultados satisfactorios. En la Figura 1.6 se observa lo oscilante que resulta esta métrica, lo cual dificulta la elección del tamaño adecuado para W_{hist} y W_{new} . Considerando un tamaño suficientemente grande para ambas ventanas, y siendo W_{new} mayor o igual que la ventana deslizante indicada por la métrica, podría contemplarse el uso de la segunda configuración. Sin embargo, en la práctica no se puede asegurar su funcionamiento en todos los casos, ya que dependería de que los cambios de concepto se produjeran en intervalos superiores al tamaño de la ventana W_{hist} , algo que, a priori, no se puede conocer.

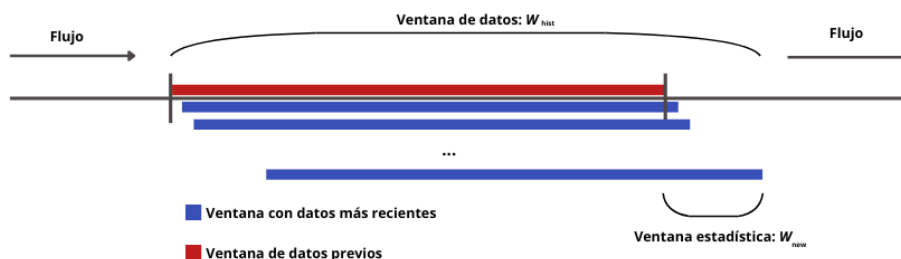


Figura 2.3: Funcionamiento del KSWIN Modificado (configuración 3)

Observando varias muestras de la métrica ante la aparición de distintos tipos de *drift*, se ha concluido que en todos los casos se produce un decaimiento, en mayor o menor medida, de la métrica utilizada. Esto se cumple siempre que no se trate de un *drift* de tipo recurrente, en el

cual se retorna al concepto original. Aprovechando este comportamiento, se ha configurado el sistema para que se realicen comparaciones entre una ventana de datos antiguos y otra ventana generada eliminando el valor más antiguo y añadiendo uno de los valores de la ventana W_{new} , realizando este proceso hasta llegar a la muestra más reciente. En la Figura 2.3 se ilustra este sistema.

Este método, junto a la realización de un test de *Kolmogorov Smirnov* unilateral comparando si las ventanas con los datos más recientes tienen una función de distribución acumulada más alta, permite detectar cuando la tendencia de la métrica decae. Se utiliza el contraste unilateral de Kolmogorov-Smirnov visto en la Sección 1.2. Se realizan por lo tanto un número de pruebas igual al tamaño de W_{new} por cada lote de datos entrante, al que también se le aplica la corrección de Benjamini-Hochberg para reducir la tasa de falsos positivos y controlar el FDR. La lógica correspondiente a este sistema se puede visualizar en el pseudocódigo 4.

Es importante señalar que, aunque no se refleja en el pseudocódigo de la modificación con el fin de facilitar su comprensión, se aplica un proceso de suavizado sobre la métrica calculada a partir de los datos de entrada, tal como se detalla en el *Accuracy* Modificado (ver Sección 1.6). Dado que el algoritmo realiza contrastes múltiples para detectar posibles cambios en la ventana W_{new} , resulta coherente utilizar el tamaño de dicha ventana como valor del parámetro *span* en el suavizado, fijándolo así a este valor. Sin embargo, no se ha elaborado la adaptación para implementar una ventana deslizante en función del dato de la métrica que llegue como dato de entrada, ya que se necesita información específica sobre la métrica utilizada para aplicar esta característica, sin poder realizar un caso general. Por lo tanto, es responsabilidad del usuario utilizar una métrica con ventana deslizante que refleje bien los cambios que se producen durante los datos más recientes. Se recomienda utilizar un tamaño de ventana deslizante similar al tamaño de la ventana W_{hist} , para mantener una consistencia con el funcionamiento de la configuración propuesta.

Algoritmo 4 KSWIN Modificado (configuración 3)

*Estado 1 – Inicialización***Input:** mismos parámetros que KSWIN, `window_start`: índice a partir del cual se empieza a hacer la detección, `metric`: métrica utilizada en la detección de *drift***Output:** $h_0(\mathbf{x}; \alpha, \text{window_size}, \text{stat_size}, \text{seed}, \text{window}, \text{window_start}, \text{metric})$: modelo inicializado (detector)*Estado 2 – Actualización (Detección)***Input:** \mathbf{x} lote de datos de entrada**Output:** Detector actualizado

```

1: if window_start ≤ 0 then
2:   Añadir x a la window deslizante
3:   if tamaño de window ≥ window_size then
4:     for cada i en el rango de stat_size do
5:       Seleccionar less_recent como una sublista de window desde 0 hasta window_size
       – stat_size
6:       Seleccionar most_recent como una sublista de window desde i hasta window_size
       – stat_size + i
7:       Realizar la prueba de KSv entre most_recent y less_recent
8:       Calcular st y p_value y añadir p_value a la lista p_values
9:     end for
10:    Aplicar corrección de BH a p_values y obtener corrected_p_values
11:    if alguna p en corrected_p_values ≤  $\bar{\alpha}$  then
12:      drift_detected, drift_confirmed, y analisisPrevio ← True
13:    end if
14:  else
15:    No hay suficientes muestras para realizar una prueba válida: drift_detected ← False
16:  end if
17: else
18:   window_start ← window_start – 1
19: end if

```

El criterio para considerar un positivo, tras aplicar la corrección, sigue siendo la obtención de un p-valor corregido menor o igual al nivel prefijado por el usuario. Anteriormente se ha mencionado el escenario en el que los datos de entrada pueden presentar una alta oscilación, lo cual ocurre en este caso, como se puede observar en el análisis de la métrica mostrado en la Sección 1.6. Si se fijara el criterio en la obtención de múltiples p-valores corregidos por debajo

del nivel establecido, podría suceder que no se detectara el *drift*. Esto se debe a que, aunque la muestra más reciente no sea homogénea respecto a la inmediatamente anterior, podría sí serlo con varias muestras anteriores, debido a la naturaleza oscilante de los datos. Como consecuencia, se perdería capacidad de detección. Por esta razón, en esta modificación se mantiene el criterio individual. No obstante, dicha naturaleza oscilante puede incrementar la tasa de falsos positivos, incluso aplicando un control sobre el FDR. Por ello, se ha desarrollado un sistema de confirmación de *drift* con el fin de aportar mayor robustez a los resultados obtenidos.

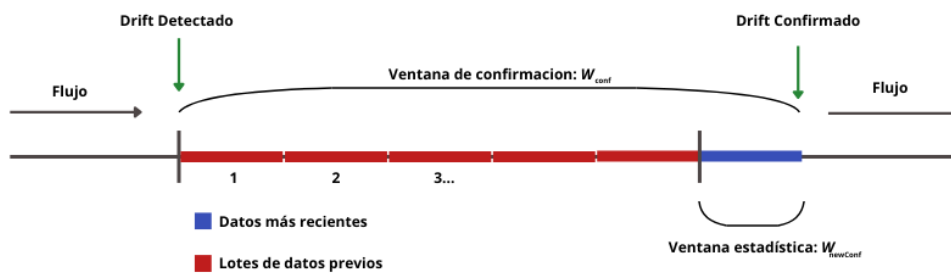


Figura 2.4: Ventana de confirmación del KSWIN

Para ello, se ha implementado una ventana de confirmación, durante la cual se verifica si efectivamente la métrica sigue reduciendo sus valores tras detectar el *drift*. En este caso, se ha seguido la política del KSWIN Modificado (configuración 2), aunque de nuevo con el contraste unilateral para verificar si la ventana de datos más recientes perteneciente a la ventana de confirmación no cambia su tendencia y vuelve a subir. Se le añade un pequeño análisis previo para verificar si el cambio de concepto detectado viene dado por un repunte de la métrica previo, lo cual descartaría que ha ocurrido un *drift* anteriormente. Para que tenga una consistencia con la propia detección, el tamaño de la ventana de confirmación W_{conf} es igual al tamaño de W_{hist} , y el tamaño de la ventana de datos recientes de confirmación $W_{newConf}$ es igual a W_{new} . Al evaluar toda esta ventana, que empieza a rellenarse al detectar un cambio de concepto, el usuario podrá conocer si el *drift* está o no confirmado. En la Figura 2.4 y en el pseudocódigo 5 se puede visualizar el funcionamiento de la ventana de confirmación.

Destacar que esta versión del algoritmo está pensada para realizar un reentrenamiento con datos posteriores a la detección del cambio de concepto, utilizando una metodología de *incremental learning* con *mini batch*. No tendría sentido utilizar este sistema de confirmación en un modelo de *Online Machine Learning*, en el que el modelo aprende de cada dato que le llega. Aún así, se podría utilizar exclusivamente para la detección del *drift* sin utilizar ninguna metodología

de adaptación posterior, siempre y cuando no interfiera en el sistema de confirmación. Por otro lado, no está pensado para la detección de *Recurrent Drift*, caso especial en el que habría que considerar la posible vuelta al concepto original, quedando como posible ampliación para esta modificación. Se hará referencia a esta versión como **KSWIN Modificado** (configuración 3).

Algoritmo 5 KSWIN Modificado (configuración 3)

Estado 3 – Actualización (Confirmación)

Input: x lote de datos de entrada

Output: Detector actualizado

```

1: Análisis previo
2: Seleccionar datos previos al drift de tamaño stat_size y los datos más recientes de tamaño
   stat_size
3: Prueba de Kolmogorov-Smirnov entre most_recent y less_recent con alternativa 'less'
4: Calcular st y p_value
5: if p_value  $\leq$  alpha then
6:   Establecer drift_confirmed y drift_detected como False
7:   Vaciar confirm_window
8: end if
9: Confirmación
10: if la longitud de confirm_window es menor que window_size then
11:   Añadir las últimas stat_size muestras de la ventana a confirm_window
12:   if la longitud de confirm_window es al menos  $2 \times$  stat_size then
13:     for cada i en el rango de  $\left(\frac{\text{longitud de confirm\_window}}{\text{stat\_size}}\right) - 1$  do
14:       Seleccionar less_recent como una sublista de confirm_window desde  $i \times$ 
         stat_size hasta  $(i + 1) \times$  stat_size
15:       Seleccionar most_recent como las últimas stat_size muestras de
         confirm_window
16:       Test Kolmogorov-Smirnov entre most_recent y less_recent (alternativa 'less')
17:       Calcular st y p_value y añadir p_value a la lista p_values
18:     end for
19:     Aplicar corrección de BH a p_values y obtener corrected_p_values
20:     if alguna p en corrected_p_values es menor o igual a  $\leq \bar{\alpha}$  then
21:       Establecer drift_confirmed y drift_detected  $\leftarrow$  False
22:     end if
23:   end if
24: else
25:   Identificar tipo de drift si se ha confirmado
26: end if

```

Identificación del tipo de drift

Como se ha mencionado previamente, una de las principales incógnitas en la detección de *Concept Drift* es diferenciar el tipo de *drift* que está ocurriendo en el momento de la detección, pudiendo elegir la política de reentrenamiento o tratamiento del modelo en cada caso. Haciendo uso de la estimación no paramétrica desarrollada en la Sección 1.3, con la función `KernelReg` de la librería `statsmodel` [20], se ha elaborado un método para diferenciar entre *Gradual*, *Incremental* y *Abrupt Drift* en el caso del uso del algoritmo para monitorizar métricas de rendimiento (recordemos que el caso del *Recurrent Drift* no se ha tenido en cuenta y queda pendiente de estudio).

La regresión local lineal permite estimar la derivada de la función en base a los valores suavizados recogidos de la métrica. Puede ser un método redundante, precisamente por utilizar los valores suavizados para estimar la derivada, pero se ha implementado de esta forma para obtener un método generalizable que se pueda utilizar en casos posteriores de estudio. Además, se combina con un uso del test de *Kolmogorov Smirnov*. Se han simulado varios cambios de conceptos de los tres tipos, y junto con la estimación teórica de lo que debería pasar, se ha observado que las métricas se comportan del siguiente modo en función del tipo que ocurre.

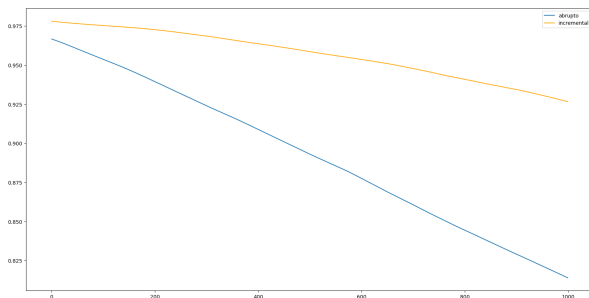


Figura 2.5: Comportamiento de la métrica de rendimiento en *Abrupt e Incremental Drift*

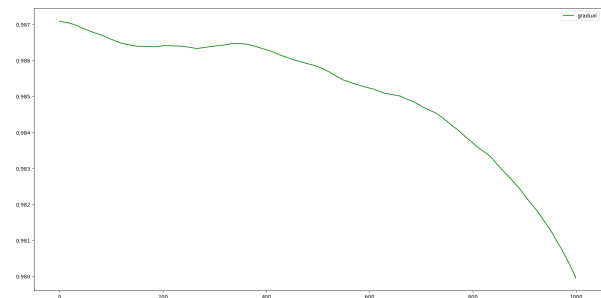


Figura 2.6: Comportamiento de la métrica de rendimiento en *Gradual Drift*

Como se puede observar en la Figura 2.6, cuando se produce un *Gradual Drift*, la métrica no baja de forma constante, sino que como se intercalan muestras del viejo y del nuevo concepto, sufre oscilaciones positivas y negativas. Si en la ventana de confirmación se detecta una derivada estimada positiva de la función estimada, se concluirá que es un *Gradual Drift*.

Si no se cumple esta condición, se procede a evaluar los primeros datos de la ventana de confirmación, específicamente del tamaño de $W_{newConf}$. Conociendo los datos reales de la métrica y la fórmula del suavizado (1.8), se puede deducir cuál debería ser el comportamiento de la métrica si se produjese un *Abrupt Drift*. Por lo tanto, teniendo esta estimación teórica, se realiza una

comparación con la función real, y mediante el contraste unilateral de Kolmogorov-Smirnov se deduce si es abrupto o por el contrario incremental.

En la Figura 2.5 se deduce que en un *Incremental Drift* la métrica va a bajar más lentamente que en un abrupto, por lo que se basa en comparar si la función de distribución acumulada es mayor para afirmar que está ocurriendo un *Abrupt Drift*. Se fija un nivel de significación en $\alpha = 0,05$, considerado suficientemente sensible para afirmar que existen evidencias significativas de que se está produciendo un *Abrupt Drift*. La elección de este nivel se adecúa a la prioridad de detectar este caso, minimizando la posibilidad de falsos negativos e identificando así correctamente los cambios bruscos en la métrica. El sistema tiene un punto débil, y es que es necesario presuponer la magnitud del cambio, ya que se fija que va a acertar en torno a un 60% de los casos. Si se produce un cambio abrupto a un concepto que provoca que el modelo acierte un porcentaje mayor de los casos, lo detectará como incremental. El objetivo de la diferenciación obtener una idea sobre cuando está ocurriendo un gran cambio abrupto, para que el usuario tome una decisión de cómo tratar dicha situación, por lo que no se considera relevante cuando se produzca la situación mencionada. La lógica de la identificación del cambio de concepto se presenta en el pseudocódigo 6.

Algoritmo 6 KSWIN Modificado (configuración 3) - Identificación del tipo de *drift*

```

1: Crear un modelo de regresión de kernel kr con la mitad de los valores de confirm_window
2: Obtener y_pred y dy_dx del ajuste del modelo
3: if algún valor en dy_dx es mayor que 0 then
4:   Establecer tipo_drift como Gradual
5: else
6:   for cada i en el rango de 1 a stat_size do
7:     Simular que el modelo acierte con una probabilidad de 0.6
8:     Añadir el valor de la métrica a valores_en_drift, con datos previos al drift
9:   end for
10:  Calcular smoothed_accuracies de los valores obtenidos usando el suavizado exponencial
11:  Realizar la prueba de Kolmogorov-Smirnov entre los valores teóricos
    (smoothed_accuracies) y los valores correspondientes reales (confirm_window), con
    alternativa 'greater'
12:  Calcular st y p_valor
13:  if p_valor es menor que 0.05 then
14:    Establecer tipo_drift como Abrupt
15:  else
16:    Establecer tipo_drift como Incremental
17:  end if
18: end if

```

Capítulo 3

Pruebas y resultados

En este capítulo se explicará el entorno sobre el que se realizaron las pruebas y los resultados obtenidos en la experimentación.

3.1. Entornos de pruebas

El primer entorno de pruebas, o escenario de simulación, para evaluar modificaciones en presencia de Concept Drift con datos procedentes de distribuciones continuas se ha construido mediante la generación artificial de varios conjuntos de datos. El objetivo es simular la detección de *Virtual Drift* en un modelo, analizando directamente $P_t(X)$. Para ello, se ha empleado la biblioteca NumPy [11], que ofrece funciones para generar datos a partir de distribuciones normales, exponenciales y gamma. El *drift* se ha simulado concatenando distribuciones de una misma familia pero con distintos parámetros, asumiendo que los cambios entre familias de distribuciones diferentes no son habituales en escenarios reales. Destacar que para la realización de pruebas se ha tenido en cuenta únicamente el *Abrupt*, el *Incremental* y el *Gradual Drift*, ya que el *Recurrent Drift* no deja de ser una sucesión de cambios de conceptos abruptos, y para evaluar únicamente la capacidad de detección se pueden sacar conclusiones teniendo en cuenta el abrupto como individual. Los tres tipos de cambio de concepto se han generado por separado, para estudiar cómo funcionan los algoritmos en cada caso. Además, se le ha añadido un nivel de ruido para simular una situación real en la que aparecen estos valores. Estos conjuntos de datos están orientados a probar el KSWIN, KSWIN Modificado (configuración 1) y KSWIN Modificado (configuración 2).

En segundo lugar se ha generado un *dataset* para utilizar en un modelo de predicción, utilizando la función *FriedmanDrift* [15], presente en la librería RiverML. El objetivo en este caso es monitorizar la aparición de *Real Drift*, analizando en este caso la $P_t(Y|Z)$. Para ello se utiliza

error cometido por el modelo como dato de entrada en el algoritmo, definido como la diferencia entre el valor real y la predicción, un valor continuo al tratarse de un modelo de regresión. En este caso solo se simularán dos tipos de *drift*: *Abrupt Drift* y *Gradual Drift*, siendo estas las dos únicas posibilidades que ofrece la función de creación. Además, se seguirá una política de reentrenamiento del modelo en cuanto se detecte un cambio de concepto, con el fin de evaluar la utilidad de aplicar estos algoritmos en un caso real, observando si las modificaciones suponen una mejora respecto a la configuración original. Específicamente, se almacenarán en una pila los 10.000 datos posteriores a la detección. Durante este periodo de almacenamiento, la detección permanecerá desactivada, cobrando especial relevancia la necesidad de evitar falsos positivos. Por lo tanto, será crucial que el FDR (*False Discovery Rate*) sea mínimo, ya que una falsa alarma no solo implicaría una detección errónea, sino que también provocaría la desactivación del mecanismo de detección, impidiendo identificar un cambio real durante el tiempo en que se realiza la recolección de datos para la pila. Esta situación se contabilizará como un falso negativo, con el objetivo de reflejar el impacto que este comportamiento puede tener en un entorno real. Las configuraciones que se probarán serán de nuevo KSWIN, KSWIN Modificado (configuración 1) y KSWIN Modificado (configuración 2).

En ambos escenarios se realizaron 1000 simulaciones por cada caso, considerándolo un número suficiente para extraer conclusiones. Los parámetros utilizados en los detectores en estos dos escenarios son los que se indican a continuación. Destacar que los tamaños de las ventanas de datos se eligieron de acuerdo a las necesidades del conjunto de datos evaluado.

- KSWIN: $\alpha=0.001$, `window_size=3000`, `stat_size=300`, `seed=42`
- KSWIN Modificado (configuración 1 y 2): $\bar{\alpha}=0.001$, `window_size=3000`, `stat_size=300`, `window_start=0`, `alternative='two-sided'`,

Por último, se hará uso de un entorno de pruebas elaborado en trabajo informático para evaluar la modificación del KSWIN Modificado (configuración 3) como métrica de datos de entrada. En este caso se monitoriza la aparición de *Real Drift* en un modelo de clasificación, analizando la $P_t(c_i|X)$ a través del seguimiento del *Accuracy* Modificado del modelo. Es importante señalar que en la comparativa del estudio informático se utiliza la etiqueta binaria para detectar el *drift*, pero en este caso hubo que utilizar la métrica suavizada para aportar al algoritmo un conjunto de datos de entrada continuo. El estudio compara diversos detectores presentes en la literatura para monitorizar un modelo de clasificación binaria, con el objetivo de determinar cuál es la opción más adecuada de forma global. Se sigue una política de reentrenamiento similar a la mencionada en el modelo de predicción, almacenando los 10.000 datos posteriores a la detección. Para ello, se ha utilizado la clase *Agrawal* [16] de *RiverML*, sobre la cual se ha realizado una modificación que permite simular distintos tipos de *drift*.

Las pruebas se limitarán a *Abrupt*, *Gradual* e *Incremental Drift*, ya que la modificación no está preparada para manejar el *Recurrent Drift*, como se explicó previamente. Además de analizar la eficacia individual del algoritmo, se presentará una comparativa global con el resto de algoritmos evaluados en el estudio mencionado, con el fin de valorar su efectividad relativa. Cabe destacar que en el estudio informático se realizó una comparación tanto con parámetros por defecto como con parámetros optimizados para el caso analizado; en este trabajo se evaluará el rendimiento en ambos escenarios. Finalmente, se analizará la eficacia en la identificación del tipo de *drift*, un aspecto que, aunque no se utilice en el reentrenamiento del modelo para garantizar una comparación justa, proporciona información útil para su aplicación práctica y constituye una base para posibles investigaciones futuras.

Para cada tipo de *drift*, se ejecutaron 25 iteraciones, con 8 *drifts* en cada una. Al tratarse de una comparativa realizada en el estudio informático, solo se presentan los parámetros utilizados en la experimentación para la configuración propuesta. El tamaño de la ventana de datos se eligió de acorde a las necesidades de la problemática, en consonancia con el tamaño de la ventana móvil de la modificación del *Accuracy*, siguiendo las recomendaciones presentadas anteriormente.

- KSWIN Modificado (configuración 3): $\bar{\alpha}=0.000001$, `window_size=3000`, `stat_size=300`, `window_start=1500`
- *Accuracy* Modificado: `window_size` (ventana móvil) = 3000, `span` = 300.

3.2. Resultados y discusión

Se han separado las pruebas en función de las modificaciones evaluadas, con el caso diferenciado de la tercera configuración. Se mostrarán los resultados de la configuración original del KSWIN junto con las modificaciones propuestas pertinentes. Para la visualización se ha utilizado la herramienta *Microsoft Power BI* [10].

3.2.1. Modificaciones para datos procedentes de distribuciones continuas

Dataset con datos de distribuciones continuas simuladas

A continuación se muestran los gráficos correspondientes a la medición del FDR según el tipo de *drift* y la distribución de datos evaluada, en las Figuras 3.1, 3.2 y 3.3. Como se puede observar, la tasa de falsos positivos se reduce drásticamente al comparar la configuración original del KSWIN y las modificaciones. Específicamente, el KSWIN Modificado (configuración 2) obtiene unos mejores resultados, independientemente de la distribución probada.

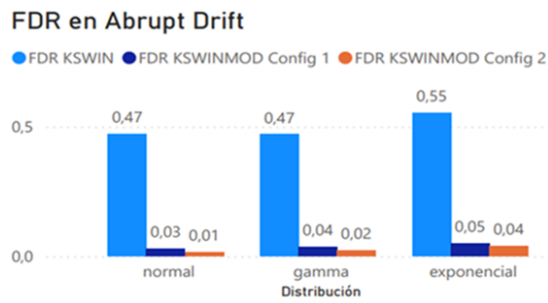


Figura 3.1: FDR - *Abrupt Drift* - Datos de distribuciones continuas simuladas

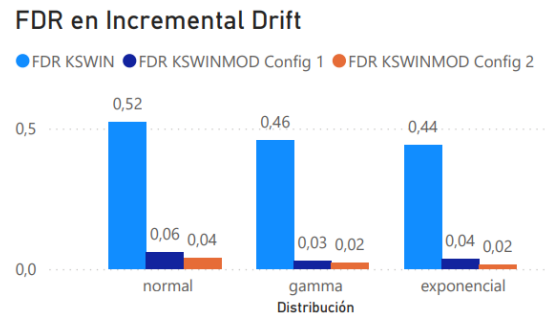


Figura 3.2: FDR - *Incremental Drift* - Datos de distribuciones continuas simuladas

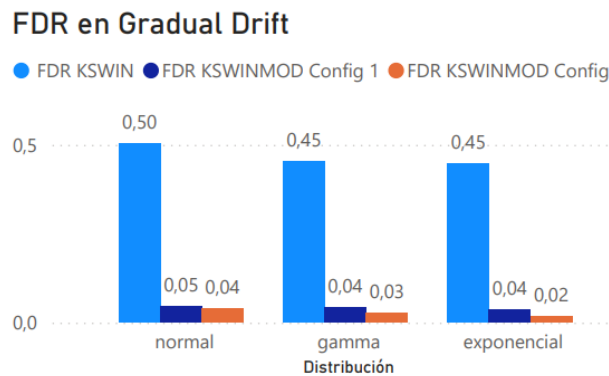


Figura 3.3: FDR - *Gradual Drift* - Datos de distribuciones continuas simuladas

En el caso del *Gradual Drift* se obtienen resultados similares, siendo la segunda configuración de las modificaciones propuestas la que mejor se comporta. En ambos casos se reduce la tasa de falsos positivos de forma significativa, por lo que las modificaciones aplicando la corrección de hipótesis múltiple de Benjamini-Hochberg funcionan.

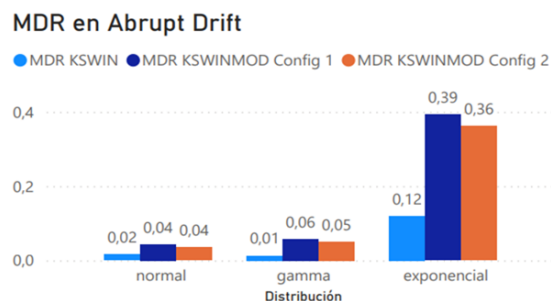


Figura 3.4: MDR - *Abrupt Drift* - Datos de distribuciones continuas simuladas

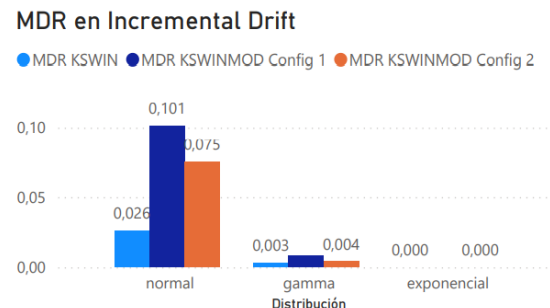


Figura 3.5: MDR - *Incremental Drift* - Datos de distribuciones continuas simuladas

En cuanto a los falsos negativos, vemos como la tasa en este caso aumenta en las modificaciones, especialmente en el caso de la distribución exponencial o la gamma, en el *Abrupt Drift*, como se puede observar en la Figura 3.4. Esto es probablemente consecuencia de la asimetría de la distribución exponencial y la gamma, dependiendo de sus parámetros. Es posible que un cambio en los parámetros de ambas distribuciones no se traduzca en una transformación de su distribución muy clara, por lo que el test de Kolmogorov-Smirnov pierde sensibilidad en esos casos. Sin embargo, se ve un caso algo anómalo, ya que como se muestra en la Figura 3.5, en el *Incremental Drift* no se detectan falsos negativos. Como el cambio ocurre más lentamente, la diferencia entre las funciones de distribución acumuladas se va acrecentando con el tiempo, haciendo que, al hacer el contraste múltiple, acabe detectando el *drift*, incluso cuando la diferencia no es muy grande.

Un comportamiento parecido se aprecia en el *Gradual Drift* en la Figura 3.6, subiendo la tasa de falsos negativos con las modificaciones, tanto en la distribución normal como gamma, y con unos valores nulos en el caso de la exponencial. Aunque aumente la tasa de falsos negativos, el incremento no es relevante, en una proporción mucho menor a la reducción de los falsos positivos, por lo que se puede ver que la corrección no penaliza mucho a este aspecto. En el MDR se aprecia también cómo la segunda configuración tiene unos mejores resultados, con ambos ratios más bajos que la configuración 1, en todos los casos.

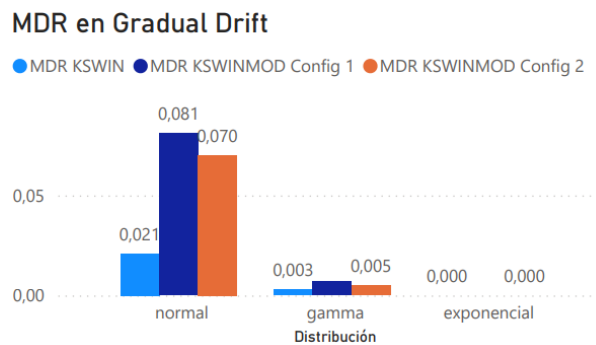


Figura 3.6: MDR - *Gradual Drift* - Datos de distribuciones continuas simuladas

El hecho anterior unido a la gran diferencia en tiempo de ejecución entre ambas configuraciones, como se puede apreciar en las Figuras 3.7, 3.8 y 3.8, se puede concluir que en términos de mejoría con respecto al algoritmo original y en eficiencia computacional, el **KSWIN Modificado (configuración 2)** es la mejor opción para este caso. En la Figura 3.10 se muestra una imagen de la monitorización de una detección de *Abrupt Drift* utilizando este algoritmo.

Tiempos de ejecución de las pruebas (en segundos) - Abrupt Drift



Figura 3.7: Tiempos de ejecución - *Abrupt Drift* - Datos de distribuciones continuas simuladas

Tiempos de ejecución de las pruebas (en segundos) - Gradual Drift



Figura 3.8: Tiempos de ejecución - *Incremental Drift* - Datos de distribuciones continuas simuladas

Tiempos de ejecución de las pruebas (en segundos) - Gradual Drift



Figura 3.9: Tiempos de ejecución - *Gradual Drift* - Datos de distribuciones continuas simuladas

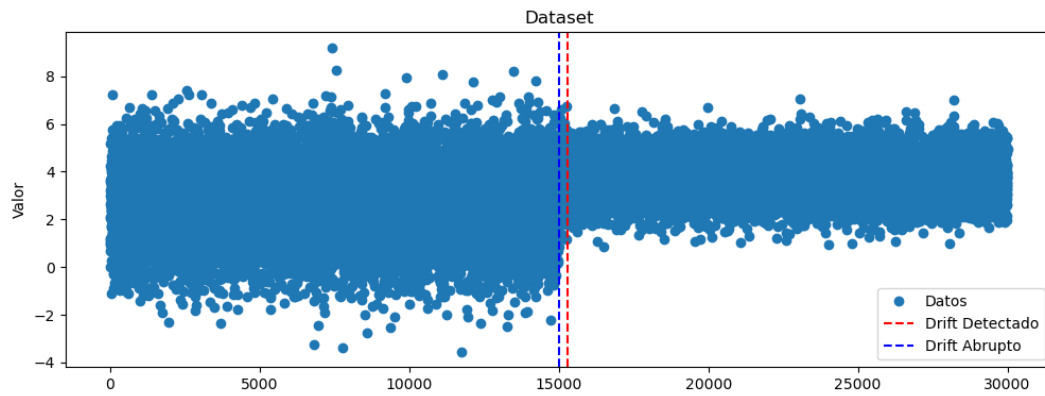


Figura 3.10: Monitorización de datos procedentes de una distribución continua usando KSWIN Modificado (configuración 2)

Monitorización del error cometido en modelo de predicción

Para el análisis de la monitorización del error cometido en el modelo de predicción, se han obtenido los siguiente resultados.

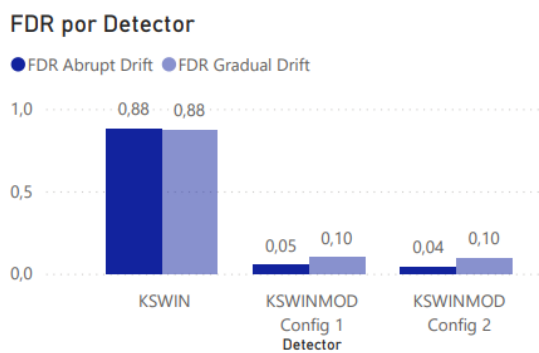


Figura 3.11: FDR - Monitorización del error en modelo de predicción

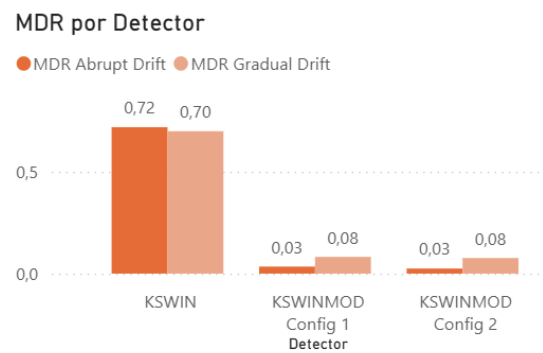
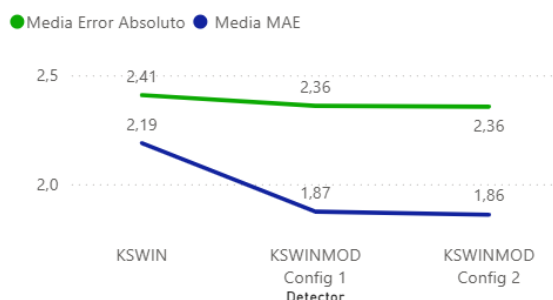


Figura 3.12: MDR - Monitorización del error en modelo de predicción

Como se puede observar en las Figuras 3.11 y 3.12, la tasa de falsos positivos se reduce tanto en el caso abrupto como en el caso gradual, con las dos configuraciones modificadas. Además, la tasa de falsos negativos también se reduce drásticamente, algo que no ocurría en el análisis anterior. Puede parecer confuso, pero hay que tener en cuenta que cuando se detecta un cambio de concepto falso, si inmediatamente después ocurre uno verdadero, no se va a detectar. Por lo tanto, es posible que la métrica del KSWIN sea muy elevada por esta misma razón, en comparación

con las mediciones de las modificaciones. En cualquier caso, es algo positivo y a tener en cuenta en cuanto a la robustez de las modificaciones propuestas, dando así resultados confiables con lo que está ocurriendo realmente.

Medias de errores - Abrupt Drift



Media de errores - Gradual Drift

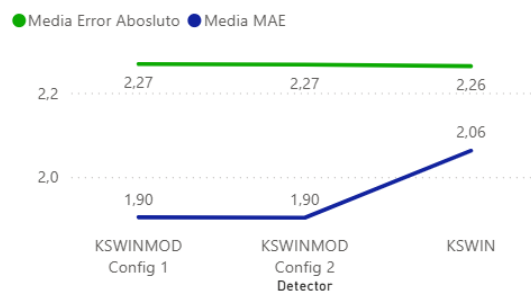


Figura 3.13: Media de los errores con *Abrupt Drift* en modelo de predicción

Figura 3.14: Media de los errores con *Gradual Drift* en modelo de predicción

Analizando la media del error cometido en las Figuras 3.13 y 3.14, se puede observar como no hay una gran diferencia al utilizar el algoritmo original o las modificaciones. En efecto, el error cometido en el *Gradual Drift* es incluso mayor en las modificaciones. Esto puede deberse a que en el KSWIN se realiza un sobreajuste a los datos más recientes, en concordancia con la alta tasa de falsos positivos, por lo que el rendimiento del modelo no se ve sustancialmente afectado. Aunque esto no es una situación deseada, por motivos obvios, ya que no se está adaptando el modelo a los nuevos conceptos cuando se debe, produciéndose un sobreajuste que en otros casos puede ser perjudicial.

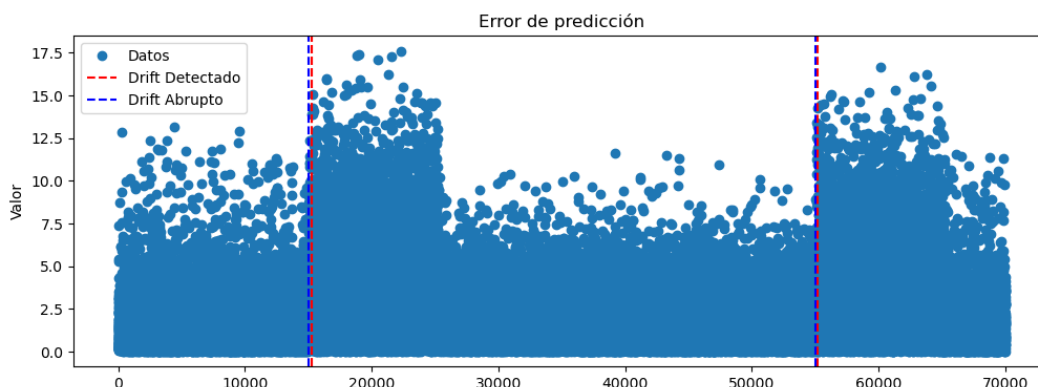


Figura 3.15: Monitorización del modelo de predicción - KSWIN Modificado (configuración 2)

Por lo tanto, analizando todos estos aspectos, y teniendo en cuenta que la segunda con-

figuración obtiene mejores resultados en el FDR, se puede concluir que el KSWIN Modificado (configuración 2) es también la mejor opción para monitorizar el error en un modelo de predicción. A continuación, en la Figura 3.1, se muestra la monitorización de una de las iteraciones en la detección de *Abrupt Drift*.

3.2.2. Modificación para métricas de rendimiento continuas como datos de entrada en el detector

En primer lugar se muestran los resultados de las mediciones presentadas en la Sección 1.6.2 en formato de tabla.

Resultados del KSWIN Modificado (configuración 3)

Tipo de drift	FDR	MDR	MTD	IR	Media CohenKappa	Media Accuracy	Media F1	Media Precision
abrupt	0,28	0,33	5,93	0,52	0,75	0,95	0,79	0,78
gradual	0,10	0,33	21,83	0,61	0,69	0,93	0,73	0,75
incremental	0,25	0,25	14,66	0,53	0,71	0,93	0,75	0,76

Tabla 3.1: Resultados de la experimentación de KSWIN Modificado (configuración 3)

Como se puede observar, el FDR es bastante bajo, especialmente si se compara con el MDR, que se mantiene en unos niveles similares. Las medias de las métricas obtenidas también son bastante elevadas, teniendo unos resultados positivos en cuanto a mantenimiento del rendimiento del modelo.

Analizando el IR en la Figura 3.16, se puede observar cómo el *Gradual Drift* es el que mejor resultados tiene, con una tasa de acierto de 0,6. Este resultado es esperable, ya que la forma de diferenciar el tipo de *drift* excluye primero este caso, y si no se para a analizar si puede ser abrupto o incremental. En cuanto a la diferenciación del *Incremental* y *Abrupt Drift*, se obtiene una tasa de en torno a 0,5. No son unos resultados excelentes, aunque hay que tener en cuenta que el método de diferenciación para un abrupto presupone que va a reducirse hasta que el modelo acierte un 60% de las veces, lo cual descarta ya los cambios de concepto en los que no se produce una caída tan elevada. Los errores en la identificación del incremental también puede venir dados por este hecho, y es que si la caída es hacia un umbral de acierto de un 50%, puede ser que la métrica evolucione más como un cambio de concepto abrupto.

La monitorización sobre 5 cambios de concepto de la Figura 3.17 refuerza la teoría previa. Solamente hay un cambio de concepto detectado que se clasifica erróneamente, en torno a la instancia 55.000, en el cual el *Accuracy* no decae hasta el rango de acierto del 60%, por lo que los casos en los que realmente hay un cambio abrupto significativos sí que pueden estar

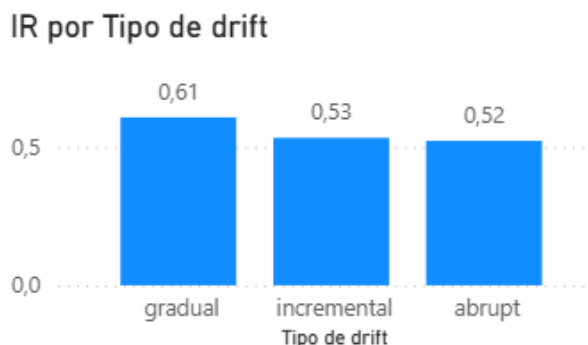


Figura 3.16: IR de KSWIN Modificado (configuración 3)

percibiéndose en su mayoría. En cuanto a la comparativa con el resto de algoritmos, se han obtenido los siguientes resultados. Para facilitar la visualización, se ha hecho referencia al KSWIN Modificado (configuración 3) como KSWIN MODIFICADO.

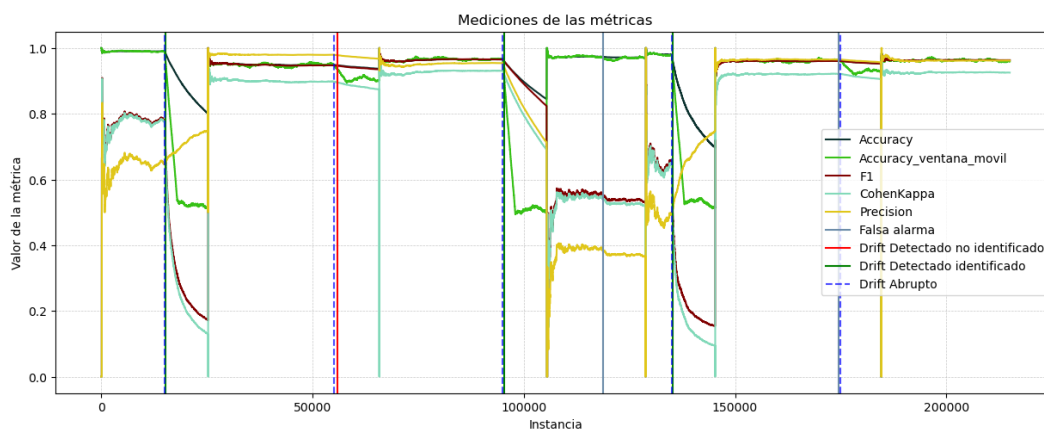


Figura 3.17: Monitorización del modelo de clasificación - KSWIN Modificado (configuración 3)

En la Figura 3.18 se muestra la media de los ratios tanto en las pruebas realizadas para *Abrupt*, *Gradual* e *Incremental Drift*³. En el caso del no optimizado, la modificación propuesta presenta un FDR superior a las dos versiones del algoritmo *HDDM_W*. Este resultado es lógico, ya que como se ha mencionado, el *HDDM_W* es un caso especialmente preparado para la clasificación binaria. Además, en la Figura 3.17 se puede observar cómo una de las falsas alarmas detectadas, en torno a la instancia 175.000, se confirma debido a que justo después se produce el *drift*. A efectos teóricos es una falsa alarma, aunque no afectará significativamente al rendimiento

³Para ver los resultados exactos, consultar las tablas en el Anexo II

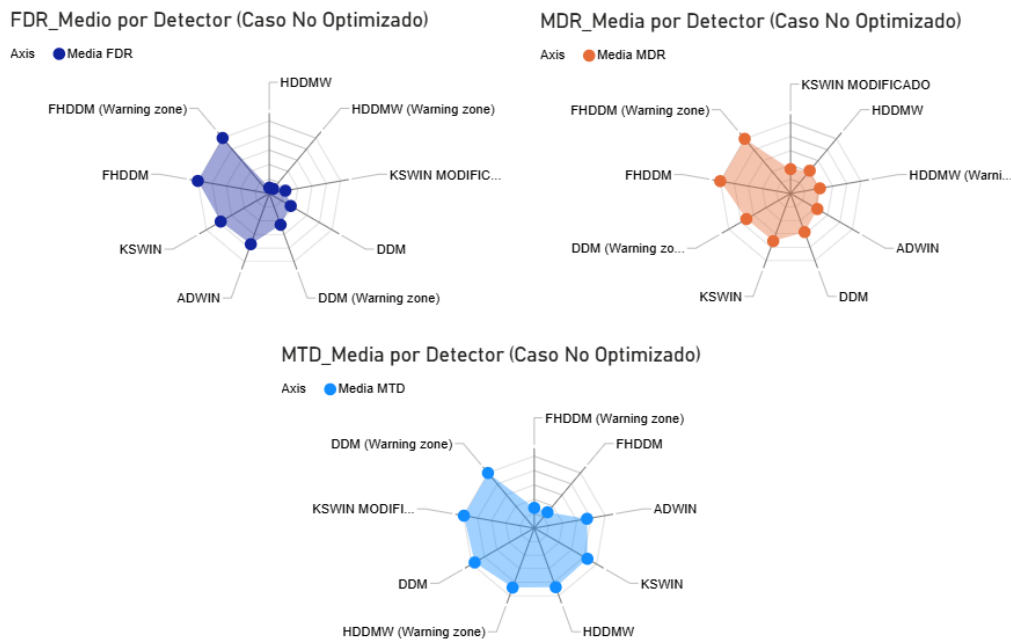


Figura 3.18: Media del FDR para todos los tipos de *drift* - Comparativa entre algoritmos no optimizados y modificación. Los resultados se muestran ordenados de menor a mayor según el valor del ratio

del modelo, ya que inmediatamente después sí que habría que realizar el reentrenamiento. Sin embargo, sí que se obtienen mejores resultados en el MDR. En el MTD se puede observar como la modificación propuesta es el algoritmo más lento de todos, aunque este dato está condicionado a que la rapidez de evaluación es mucho menor al realizar más pruebas que los otro detectores.

CohenKappa_Media, Accuracy_Media, F1_Media y Precision_Media por Detector (Caso No Optimizado)

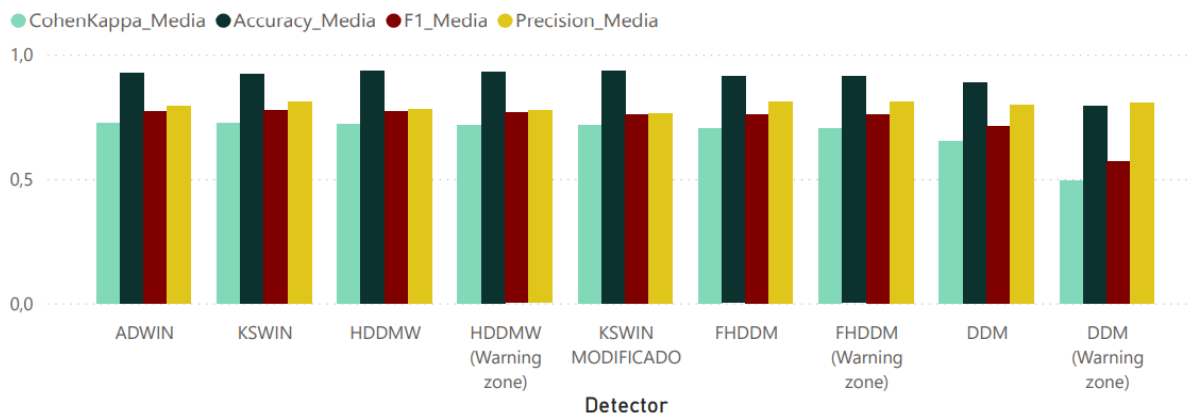


Figura 3.19: Media de las métricas - Comparativa entre algoritmos no optimizados y modificación. Resultados ordenados de mayor a menor *CohenKappa*

En el análisis de las métricas obtenidas en la Figura 3.19 puede observar cómo la modificación no es de las mejores opciones, aunque tampoco lo es el mencionado $HDDM_W$. Estos resultados vienen condicionados por el fenómeno del sobreajuste a los datos más recientes provocado por las falsas alarmas, como ocurría en el modelo de predicción.

Para el caso optimizado la modificación propuesta tiene el mejor FDR sin sacrificar el MDR, en el que se encuentra en segunda posición, como se puede observar en la Figura 3.20. El resultado es muy positivo, superando incluso a los detectores diseñados específicamente para este propósito ⁴. La optimización de hiperparámetros busca la combinación óptima para equilibrar la detección entre todos los tipos de *drift* y la rapidez en esta, por lo que nos da una idea de que el KSWIN Modificado (configuración 3) se comporta adecuadamente en todos los casos. Vuelve a aparecer un valor muy elevado en el MTD con respecto al resto, pero de nuevo provocado por el elevado coste computacional que conlleva la utilización de la modificación, y no por reaccionar tarde ante un cambio de concepto. La interpretación de las métricas, presentadas en la Figura 3.21, es similar al caso no optimizado, condicionadas por un sobreajuste propiciado por los falsos positivos.

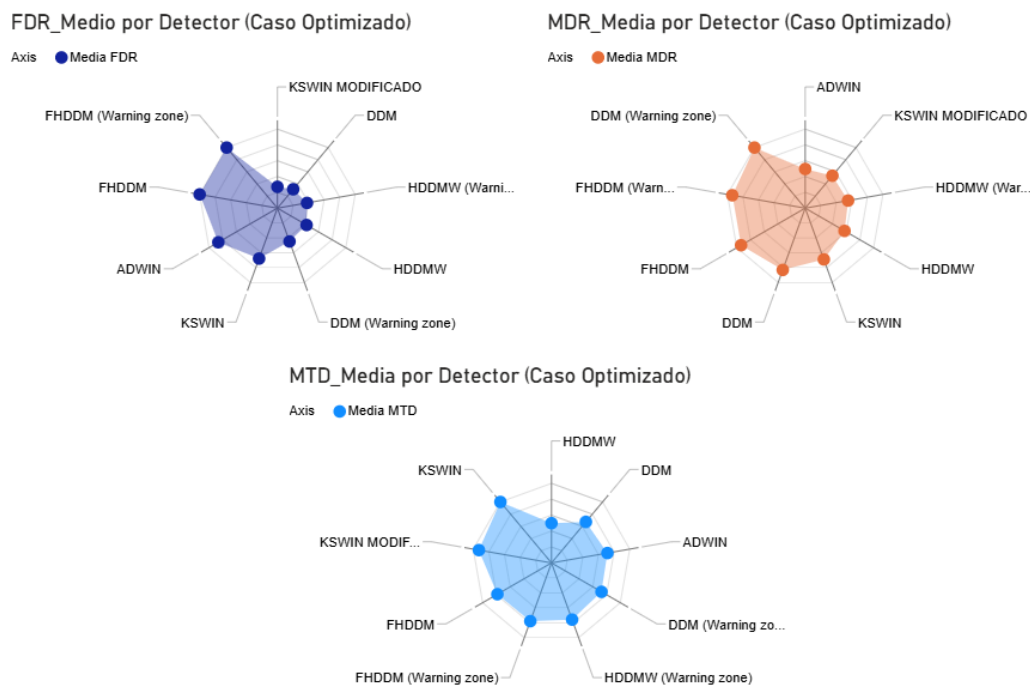


Figura 3.20: Media del FDR para todos los tipos de *drift* - Comparativa entre algoritmos optimizados y modificación. Los resultados se muestran ordenados de menor a mayor según el valor del ratio

⁴Para ver los resultados exactos, consultar las tablas en el Anexo II

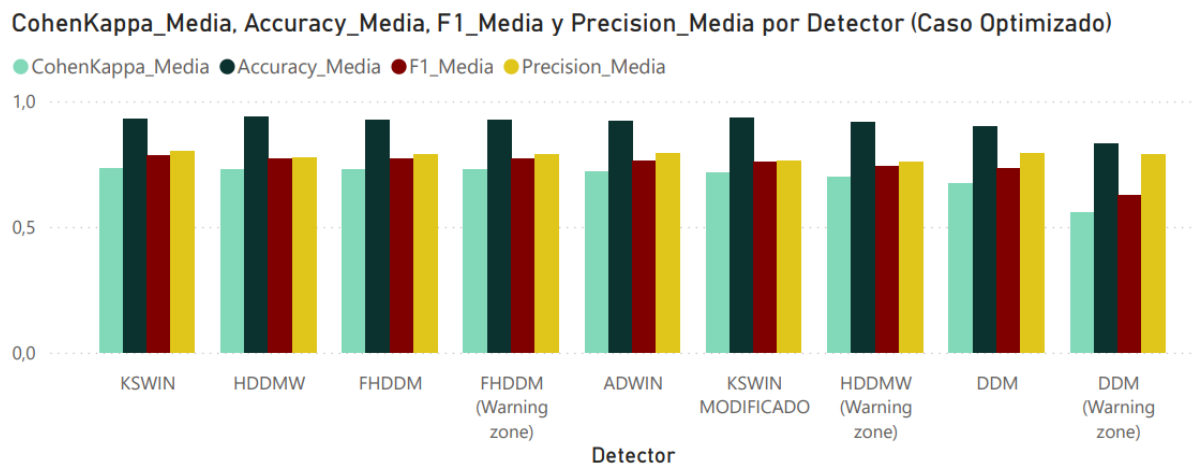


Figura 3.21: Media de las métricas - Comparativa entre algoritmos optimizados y modificación. Resultados ordenados de mayor a menor *CohenKappa*

Teniendo en cuenta los resultados obtenidos en la comparativa, sumado a la información que aporta la identificación del tipo de *drift*, muy útil para el usuario cuando quiere programar la política de reentrenamiento en su modelo, se puede concluir que el **KSWIN Modificado** (configuración 3) iguala e incluso supera el rendimiento de algoritmos. Asimismo, posee el valor añadido de poder utilizarse en cualquier caso en el que se quiera monitorizar una métrica de rendimiento.

Capítulo 4

Conclusiones y posibles ampliaciones

En este trabajo se ha implementado una modificación del algoritmo KSWIN de la librería RiverML, el cual emplea el contraste de Kolmogorov-Smirnov. Dicha modificación adapta su funcionamiento para abordar el problema de los contrastes múltiples mediante la corrección de Benjamini-Hochberg, además de corregir deficiencias presentes en la implementación original del algoritmo.

En primer lugar, se han presentado los conceptos teóricos, tanto matemáticos como informáticos, necesarios para comprender el fenómeno del *Concept Drift* y el contexto en el que se produce. Asimismo, se ha expuesto la base matemática que justifica el funcionamiento del algoritmo y las decisiones adoptadas durante su desarrollo. Entre los temas tratados se incluyen alguno de los contenidos abordados en el Grado, como la distribución conjunta y condicionada o los contrastes de hipótesis, junto con otros nuevos, como el problema de los contrastes múltiples, la corrección de Benjamini-Hochberg y la estimación no paramétrica mediante regresión local lineal. También se han introducido las métricas de rendimiento empleadas para evaluar los resultados obtenidos, tanto en las modificaciones orientadas a datos procedentes de distribuciones continuas como en las adaptadas a la comparativa realizada en el TFG del Grado en Informática.

A continuación, se implementaron las modificaciones sobre el algoritmo, con una justificación basada en los conceptos teóricos previamente expuestos y adaptadas al caso de monitorización de datos continuos. Las dos configuraciones propuestas comparten el mismo fundamento teórico y solo difieren en el tipo de muestras utilizadas para realizar los contrastes múltiples.

En este punto se realizó la modificación específica para su integración en la comparativa del estudio informático, aplicando las mismas adaptaciones, incluida la corrección de Benjamini-Hochberg, y analizando las particularidades del problema. Ante las dificultades del uso de una métrica de rendimiento para detectar *Real Concept Drift*, se suavizó dicha métrica y se generó un conjunto de datos de entrada acorde a los requisitos establecidos. Además, se ha desarrollado

un sistema de identificación del tipo de *drift*, basado en la estimación no paramétrica mediante regresión local lineal y el test unilateral de Kolmogorov-Smirnov.

Una vez implementadas todas las modificaciones, se ha diseñado un entorno de pruebas para evaluar los resultados obtenidos, tanto frente a la versión original del algoritmo KSWIN como frente al resto de detectores incluidos en la comparativa del trabajo informático. Para ello, se han generado múltiples *datasets* artificiales con datos extraídos de diversas distribuciones continuas, se ha simulado la monitorización del error cometido por un modelo de predicción, y se ha utilizado el conjunto de datos del estudio previamente citado. La integración de la modificación se ha realizado sin alterar el contexto original, lo que ha permitido una comparativa justa entre los algoritmos. Tras visualizar y analizar los resultados mediante *PowerBI*, se han extraído las siguientes conclusiones:

Es necesario distinguir entre los diferentes enfoques en función del tipo de datos de entrada. En primer lugar, se presentan los resultados correspondientes a las modificaciones orientadas a la monitorización de datos provenientes de distribuciones continuas, recogidos en la Sección 3.2.1. En concreto, sobre el conjunto de datos con distribuciones continuas simuladas, se observa que aplicar contrastes múltiples entre la muestra más reciente y varias muestras anteriores, junto con la corrección de Benjamini-Hochberg, mejora notablemente el rendimiento del algoritmo KSWIN original. Esta estrategia ofrece resultados más robustos y consistentes, eliminando la aleatoriedad derivada de comparar con una única muestra aleatoria del pasado. Además, se consigue un control eficaz del FDR, evidenciando una mejora clara respecto al comportamiento de la versión sin modificar. Esta mejora no implica una pérdida apreciable de sensibilidad del test, confirmando así los beneficios esperados de la corrección aplicada. En cuanto a la elección de la configuración más adecuada, los valores obtenidos para las métricas FDR y MDR, así como las diferencias en tiempo de ejecución, permiten concluir que el KSWIN Modificado (configuración 2) es la variante con mejor rendimiento. Esto confirma que realizar contrastes entre la muestra más reciente y muestras secuenciales previas resulta más eficaz que utilizar muestras aleatorias del pasado.

En segundo lugar, se analizan los resultados obtenidos al monitorizar el error cometido por un modelo de predicción, también recogidos en la Sección 3.2.1. Las métricas muestran una reducción significativa tanto del FDR como del MDR. En general, la media del error absoluto y el MAE se reducen con las modificaciones propuestas. De nuevo, el KSWIN Modificado (configuración 2) es la opción con mejores resultados, con una interpretación análoga al caso anterior. Así, se concluye que las modificaciones propuestas son útiles para detectar tanto *Virtual Drift* en modelos con datos de entrada continuos como *Real Drift* en modelos de predicción. No obstante, aunque los resultados son positivos, existen posibles ampliaciones aplicables a esta casuística o que se pueden integrar en futuras versiones del KSWIN Modificado (configuración 2):

- Evaluar la efectividad de los algoritmos sobre un conjunto de datos reales, con el objetivo de analizar su comportamiento desde una perspectiva aplicada.
- Estudiar en mayor profundidad la tasa de aparición de falsos negativos en las distribuciones gamma y exponencial.
- Diseñar un método de identificación de *drift* específico para este escenario, ya sea adaptando el previamente propuesto o desarrollando uno nuevo.

Por último, se presentan los resultados obtenidos con la modificación para métricas de rendimiento continuas como datos de entrada, expuestos en la Sección 3.2.2. Tanto en la comparativa con detectores optimizados como no optimizados, se observa que el **KSWIN Modificado (configuración 3)** no solo mejora a su versión original, sino que es el algoritmo más equilibrado entre falsos positivos y falsos negativos. La media del FDR es muy baja, incluso la menor en el caso optimizado, y el MDR es inferior al del resto de detectores. No obstante, el MTD presenta un valor considerablemente alto en ambos casos, probablemente debido a la mayor carga computacional que requiere esta modificación, y no a una baja capacidad de reacción ante el cambio. Precisamente, el criterio seleccionado para la consideración de una detección positiva debería permitir que el algoritmo identifique el *drift* en las primeras muestras tras su aparición. Aun así, puede concluirse que, bajo el requisito principal de mantener tasas bajas de falsos positivos y falsos negativos, la modificación propuesta supera al resto de algoritmos disponibles en la librería *RiverML* para el caso de monitorización de *Real Drift* en un modelo de clasificación binaria.

Respecto a la identificación del tipo de *drift*, la información aportada por el algoritmo podría permitir al usuario decidir el método de adaptación más adecuado tras una detección. Los casos extremos, como el *Abrupt Drift*, caracterizados por una caída brusca en la métrica de rendimiento, son identificados correctamente. Sin embargo, los resultados aún no permiten confiar plenamente en esta información sin margen de error. Los métodos de adaptación difieren considerablemente entre un *Abrupt* y un *Gradual* o *Incremental Drift*, por lo que se requiere una tasa de acierto mayor para aplicar este sistema en escenarios reales. Aun así, se establece una base sólida al proponer un método novedoso en la literatura para abordar esta problemática. Aunque se han cumplido los objetivos y los resultados son nuevamente positivos, se proponen algunas ampliaciones aplicables a la última de las modificaciones, el **KSWIN Modificado (configuración 3)**:

- Adaptar el algoritmo para monitorizar el *Recurrent Drift*.
- Analizar y proporcionar una mejora para el sistema de identificación del tipo de *drift*, buscando la obtención de una tasa de acierto mayor.
- Mejorar la eficiencia computacional del algoritmo en términos temporales.

Anexo I

Tabla para el test de KS

n	Test de Kolmogorov-Smirnov							
	Nivel de significación α							
	0.20	0.10	0.05	0.02	0.01	0.005	0.002	0.001
1	0.90000	0.95000	0.97500	0.99000	0.99500	0.99750	0.99900	0.99950
2	0.68337	0.77639	0.84189	0.90000	0.92929	0.95000	0.96838	0.97764
3	0.56481	0.63604	0.70760	0.78456	0.82900	0.86428	0.90000	0.92065
4	0.49265	0.56522	0.62394	0.68887	0.73424	0.77639	0.82217	0.85047
5	0.44698	0.50945	0.56328	0.62718	0.66853	0.70543	0.75000	0.78137
6	0.41037	0.46799	0.51926	0.57741	0.61661	0.65287	0.69571	0.72479
7	0.38148	0.43607	0.48342	0.53844	0.57581	0.60975	0.65071	0.67930
8	0.35831	0.40962	0.45427	0.50654	0.54179	0.57429	0.61368	0.64098
9	0.33910	0.38746	0.43001	0.47960	0.51332	0.54443	0.58210	0.60846
10	0.32260	0.36866	0.40925	0.45562	0.48893	0.51872	0.55500	0.58042
11	0.30829	0.35242	0.39122	0.43670	0.46770	0.49539	0.53135	0.55588
12	0.29577	0.33815	0.37543	0.41918	0.44905	0.47672	0.51047	0.53422
13	0.28470	0.32549	0.36143	0.40362	0.43247	0.45921	0.49189	0.51490
14	0.27481	0.31417	0.34890	0.38970	0.41762	0.44352	0.47520	0.49753
15	0.26589	0.30397	0.33750	0.37713	0.40420	0.42934	0.45611	0.48182
16	0.25778	0.29472	0.32733	0.36571	0.39201	0.41644	0.44637	0.46750
17	0.25039	0.28627	0.31796	0.35528	0.38086	0.40464	0.43380	0.45540
18	0.24360	0.27851	0.30936	0.34569	0.37062	0.39380	0.42224	0.44234
19	0.23735	0.27136	0.30143	0.33685	0.36117	0.38379	0.41156	0.43119
20	0.23156	0.26473	0.29408	0.32866	0.35241	0.37451	0.40165	0.42085
21	0.22517	0.25858	0.28724	0.32104	0.34426	0.36588	0.39243	0.41122
22	0.22115	0.25283	0.28087	0.31394	0.33666	0.35782	0.38382	0.40223
23	0.21646	0.24746	0.27491	0.30728	0.32954	0.35027	0.37575	0.39380
24	0.21205	0.24242	0.26931	0.30104	0.32286	0.34318	0.36787	0.38588
25	0.20790	0.23768	0.26404	0.29518	0.31657	0.33651	0.36104	0.37743
26	0.20399	0.23320	0.25908	0.28962	0.30963	0.33022	0.35431	0.37139
27	0.20030	0.22898	0.25438	0.28438	0.30502	0.32425	0.34794	0.36473
28	0.19680	0.22497	0.24993	0.27942	0.29971	0.31862	0.34190	0.35842
29	0.19348	0.22117	0.24571	0.27471	0.29466	0.31327	0.33617	0.35242
30	0.19032	0.21756	0.24170	0.27023	0.28986	0.30818	0.33072	0.34672
31	0.18732	0.21412	0.23788	0.26596	0.28529	0.30333	0.32553	0.34129
32	0.18445	0.21085	0.23424	0.26189	0.28094	0.29870	0.32058	0.33611
33	0.18171	0.20771	0.23076	0.25801	0.27577	0.29428	0.31584	0.33115
34	0.17909	0.21472	0.22743	0.25429	0.27271	0.29005	0.31131	0.32641
35	0.17659	0.20185	0.22425	0.25073	0.26897	0.28600	0.30597	0.32187
36	0.17418	0.19910	0.22119	0.24732	0.26532	0.28211	0.30281	0.31751
37	0.17188	0.19646	0.21826	0.24404	0.26180	0.27838	0.29882	0.31333
38	0.16966	0.19392	0.21544	0.24089	0.25843	0.27483	0.29498	0.30931
39	0.16753	0.19148	0.21273	0.23785	0.25518	0.27135	0.29125	0.30544
40	0.16547	0.18913	0.21012	0.23494	0.25205	0.26803	0.28772	0.30171
41	0.16349	0.18687	0.20760	0.23213	0.24904	0.26482	0.28429	0.29811
42	0.16158	0.18468	0.20517	0.22941	0.24613	0.26173	0.28097	0.29465
43	0.15974	0.18257	0.20283	0.22679	0.24332	0.25875	0.27778	0.29130
44	0.15795	0.18051	0.20056	0.22426	0.24060	0.25587	0.27468	0.28806
45	0.15623	0.17856	0.19837	0.22181	0.23798	0.25308	0.27169	0.28493
46	0.15457	0.17665	0.19625	0.21944	0.23544	0.25038	0.26880	0.28190
47	0.15295	0.17481	0.19420	0.21715	0.23298	0.24776	0.26600	0.27896
48	0.15139	0.17301	0.19221	0.21493	0.23059	0.24523	0.26328	0.27611
49	0.14987	0.17128	0.19028	0.21281	0.22832	0.24281	0.26069	0.27339
50	0.14840	0.16959	0.18841	0.21068	0.22604	0.24039	0.25809	0.27067
n>50	1.07	1.22	1.36	1.52	1.63	1.73	1.85	1.95
	$\frac{1.07}{\sqrt{n}}$	$\frac{1.22}{\sqrt{n}}$	$\frac{1.36}{\sqrt{n}}$	$\frac{1.52}{\sqrt{n}}$	$\frac{1.63}{\sqrt{n}}$	$\frac{1.73}{\sqrt{n}}$	$\frac{1.85}{\sqrt{n}}$	$\frac{1.95}{\sqrt{n}}$

Tabla I.1: Tabla para test Kolmogorov-Smirnov [19]

Anexo II

Resultados de las pruebas

Media de los ratios obtenidos - Caso Optimizado

Detector	Media del FDR	Media del MDR	Media del MTD (s)
KSWIN MODIFICADO	0,21	0,30	14,14
DDM	0,24	0,47	10,27
HDDMW (Warning zone)	0,29	0,31	11,56
HDDMW	0,33	0,33	7,61
DDM (Warning zone)	0,34	0,57	11,08
KSWIN	0,52	0,39	15,24
ADWIN	0,66	0,28	10,90
FHDDM	0,77	0,53	11,99
FHDDM (Warning zone)	0,77	0,53	11,87

Media de los ratios obtenidos - Caso No Optimizado

Detector	Media del FDR	Media del MDR	Media del MTD (s)
HDDMW	0,07	0,37	12,44
HDDMW (Warning zone)	0,08	0,37	12,53
KSWIN MODIFICADO	0,21	0,30	14,14
DDM	0,31	0,51	13,61
DDM (Warning zone)	0,42	0,63	14,24
ADWIN	0,68	0,38	10,60
KSWIN	0,71	0,63	12,15
FHDDM	0,91	0,89	4,08
FHDDM (Warning zone)	0,91	0,89	4,01

Tabla II.1: Media de los ratios obtenidos para la prueba del KSWIN Modificado (configuración 3)

Resultados obtenidos para Abrupt Drift - Caso No Optimizado

Detector	FDR	MDR	MTD (s)	Media del CohenKappa	Media del Accuracy	Media del F1	Media del Precision
HDDMW (Warning zone)	0,02	0,36	0,32	0,76	0,94	0,80	0,79
HDDMW	0,02	0,36	0,29	0,76	0,94	0,80	0,79
KSWIN MODIFICADO	0,28	0,33	5,93	0,75	0,95	0,79	0,78
DDM	0,29	0,59	2,58	0,65	0,88	0,73	0,82
DDM (Warning zone)	0,44	0,73	2,37	0,48	0,78	0,57	0,84
KSWIN	0,49	0,38	0,85	0,77	0,94	0,81	0,81
ADWIN	0,71	0,39	1,66	0,77	0,94	0,80	0,81
FHDDM	0,86	0,82	3,03	0,73	0,93	0,78	0,80
FHDDM (Warning zone)	0,86	0,82	3,03	0,73	0,93	0,78	0,80

Resultados obtenidos para Incremental Drift - Caso No Optimizado

Detector	FDR	MDR	MTD (s)	Media del CohenKappa	Media del Accuracy	Media del F1	Media del Precision
HDDMW	0,20	0,50	10,55	0,70	0,92	0,76	0,80
HDDMW (Warning zone)	0,21	0,50	11,52	0,69	0,92	0,75	0,78
KSWIN MODIFICADO	0,25	0,25	14,66	0,71	0,93	0,75	0,76
DDM	0,50	0,66	7,17	0,62	0,87	0,69	0,81
DDM (Warning zone)	0,65	0,78	5,01	0,49	0,79	0,57	0,81
ADWIN	0,68	0,50	7,32	0,72	0,92	0,77	0,81
KSWIN	0,91	0,91	3,44	0,72	0,92	0,78	0,84
FHDDM	0,98	0,98	0,22	0,71	0,91	0,77	0,84
FHDDM (Warning zone)	0,98	0,98	0,23	0,71	0,91	0,77	0,84

Resultados obtenidos para Gradual Drift - Caso No Optimizado

Detector	FDR	MDR	MTD (s)	Media del CohenKappa	Media del Accuracy	Media del F1	Media del Precision
HDDMW (Warning zone)	0,01	0,25	25,73	0,70	0,93	0,75	0,75
HDDMW	0,01	0,25	26,49	0,71	0,93	0,75	0,75
KSWIN MODIFICADO	0,10	0,33	21,83	0,69	0,93	0,73	0,75
DDM	0,14	0,29	31,09	0,68	0,91	0,73	0,77
DDM (Warning zone)	0,16	0,40	35,33	0,52	0,82	0,57	0,78
ADWIN	0,65	0,27	22,81	0,70	0,92	0,74	0,76
KSWIN	0,72	0,61	32,17	0,69	0,91	0,74	0,79
FHDDM	0,90	0,86	8,99	0,67	0,89	0,73	0,80
FHDDM (Warning zone)	0,90	0,86	8,76	0,67	0,89	0,73	0,80

Tabla II.2: Resultados obtenidos para la prueba del KSWIN Modificado (configuración 3) en el caso No Optimizado

Resultados obtenidos para Abrupt Drift - Caso Optimizado

Detector	FDR	MDR	MTD	Media del CohenKappa	Media del Accuracy	Media del F1	Media del Precision
DDM	0,17	0,48	2,41	0,71	0,91	0,77	0,81
HDDMW (Warning zone)	0,19	0,33	1,89	0,72	0,92	0,76	0,76
KSWIN MODIFICADO	0,28	0,33	5,93	0,75	0,95	0,79	0,78
DDM (Warning zone)	0,28	0,61	2,50	0,57	0,82	0,65	0,78
HDDMW	0,28	0,35	0,60	0,75	0,94	0,79	0,78
KSWIN	0,47	0,35	0,36	0,77	0,95	0,81	0,81
ADWIN	0,64	0,20	4,82	0,75	0,93	0,78	0,80
FHDDM	0,67	0,33	6,46	0,77	0,94	0,80	0,80
FHDDM (Warning zone)	0,67	0,33	6,16	0,77	0,94	0,80	0,80

Resultados obtenidos para Incremental Drift - Caso Optimizado

Detector	FDR	MDR	MTD	Media del CohenKappa	Media del Accuracy	Media del F1	Media del Precision
KSWIN MODIFICADO	0,25	0,25	14,66	0,71	0,93	0,75	0,76
DDM	0,39	0,58	6,75	0,67	0,90	0,74	0,80
HDDMW	0,48	0,43	5,59	0,73	0,93	0,77	0,79
HDDMW (Warning zone)	0,48	0,42	7,13	0,70	0,92	0,75	0,78
DDM (Warning zone)	0,58	0,70	4,66	0,57	0,84	0,64	0,79
KSWIN	0,60	0,57	10,70	0,73	0,92	0,78	0,83
ADWIN	0,71	0,46	6,66	0,73	0,92	0,78	0,81
FHDDM	0,94	0,90	2,45	0,73	0,92	0,78	0,81
FHDDM (Warning zone)	0,94	0,90	2,45	0,73	0,92	0,78	0,81

Resultados obtenidos para Gradual Drift - Caso Optimizado

Detector	FDR	MDR	MTD	Media del CohenKappa	Media del Accuracy	Media del F1	Media del Precision
ADWIN	0,63	0,19	21,21	0,69	0,92	0,74	0,77
DDM	0,16	0,35	21,64	0,64	0,89	0,69	0,78
DDM (Warning zone)	0,17	0,39	26,08	0,54	0,83	0,60	0,79
FHDDM	0,68	0,37	27,05	0,69	0,91	0,74	0,77
FHDDM (Warning zone)	0,68	0,37	27,01	0,69	0,91	0,74	0,77
HDDMW	0,21	0,21	16,65	0,71	0,94	0,75	0,75
HDDMW (Warning zone)	0,20	0,20	25,65	0,67	0,91	0,72	0,74
KSWIN	0,49	0,25	34,66	0,71	0,93	0,76	0,78
KSWIN MODIFICADO	0,10	0,33	21,83	0,69	0,93	0,73	0,75

Tabla II.3: Resultados obtenidos para la prueba del KSWIN Modificado (configuración 3) en el caso Optimizado

Bibliografía

- [1] Amat Rodrigo, J. (2016). *Comparaciones múltiples: corrección de p-value y FDR*. RPubs. https://rpubs.com/Joaquin_AR/236898
- [2] Agrahari, S., & Singh, A. K. (2022). Concept Drift detection in data stream mining: A literature review. *Journal of King Saud University-Computer and Information Sciences*, 34(10), 9523-9540.
- [3] Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1), 289-300.
- [4] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on Concept Drift adaptation. *ACM Computing Surveys (CSUR)*, 46(4), 1-37.
- [5] Garavito, D. (2018, octubre 18). *Introducción a las pruebas de hipótesis: hipótesis nula, alternativa, sistema de hipótesis, error tipo I, error tipo II, nivel de significancia, valor p*. RPubs. <https://rpubs.com/bogotan/PruebasHipo>
- [6] Google Cloud. (s.f.). *Aprendizaje supervisado frente a aprendizaje no supervisado: ¿Cuál es la diferencia?* Google. <https://cloud.google.com/discover/supervised-vs-unsupervised-learning?hl=es>
- [7] Hodges Jr, J. L. (1958). The significance probability of the Smirnov two-sample test. *Arkiv för matematik*, 3(5), 469-486.
- [8] Khamassi, I., Sayed-Mouchaweh, M., Hammami, M., & Ghédira, K. (2018). Discussion and review on evolving data streams and Concept Drift adapting. *Evolving Systems*, 9, 1-23.
- [9] Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2018). Learning under Concept Drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12), 2346-2363.
- [10] Microsoft. (s.f.). *Power BI Desktop*. <https://powerbi.microsoft.com/es-es/desktop/>

- [11] NumPy. (s.f.). *NumPy documentation*. Documentación de NumPy. <https://numpy.org/doc/stable/>
- [12] Raab, C., Heusinger, M., & Schleif, F. M. (2020). Reactive soft prototype computing for Concept Drift streams. *Neurocomputing*, 416, 340-351.
- [13] River. (s.f.). *River: A Python library for online machine learning* (versión 0.21.2). <https://riverml.xyz/0.21.2/>
- [14] River. (s.f.). *KSWIN – Kolmogorov-Smirnov Windowing* [Código fuente]. GitHub. <https://github.com/online-ml/river/blob/main/river/drift/kswin.py>
- [15] River. (s.f.). *FriedmanDrift dataset*. <https://riverml.xyz/0.21.2/api/datasets/synth/FriedmanDrift/>
- [16] River. (s.f.). *agrawal.py*. En *River: A Python library for online machine learning*. GitHub. <https://github.com/online-ml/river/blob/main/river/datasets/synth/agrawal.py>
- [17] Sahiner, B., Chen, W., Samala, R. K., & Petrick, N. (2023). Data drift in medical machine learning: implications and potential remedies. *The British Journal of Radiology*, 96(1150).
- [18] SciPy. (s.f.). *Two-sample Kolmogorov-Smirnov test*. Documentación de SciPy. https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ks_2samp.html
- [19] Silva Palomino, R. (2015, diciembre 15). *Tablas Kolmogorov-Smirnov* [Presentación de diapositivas]. SlideShare. <https://es.slideshare.net/slideshow/tablas-kolmogorovsmirnov/58083881>
- [20] StatsModels. (s.f.). *StatsModels documentation*. Documentación de StatsModels. https://www.statsmodels.org/dev/generated/statsmodels.nonparametric.kernel_regression.KernelReg.html
- [21] Vélez Ibarrola, R. & García Pérez, A. (1997). *Principios de Inferencia Estadística*. Madrid: Universidad Nacional de Educación a Distancia.
- [22] Vélez Ibarrola, R. (2019). *Cálculo de Probabilidades 2*. Universidad Nacional de Educación a Distancia.
- [23] Wasserman, L. (2006). *All of Nonparametric statistics*. Springer Science & Business Media.
- [24] Webb, G. I., Hyde, R., Cao, H., Nguyen, H. L., & Petitjean, F. (2016). Characterizing Concept Drift. *Data Mining and Knowledge Discovery*, 30(4), 964-994.
- [25] Weiss, Neil A. (2005). *A Course in Probability*. Boston: Addison–Wesley. ISBN 0-321-18954-X.