Universidad de las Ciencias Informáticas

Facultad 2

Extensión de algoritmos de estadística multivariada para problemas de predicción con salidas compuestas como parte de la herramienta Mulan

Trabajo de Diploma

Autores: Eduardo Hernández García

Antonio Darío Blanco Rodríguez

Tutor: MSc. Hector Raúl González Diez

Ciudad de la Habana, 2015

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la UCI los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

_____ _____

Eduardo Hernández García Antonio Darío Blanco Rodríguez

_____

MSc. Hector Raúl González Diez

Agradecimientos

.

Eduardo Hernández García

- A todas las personas que de una forma u otra han tenido que ver con mi formación y con la investigación.
- A mi familia, en especial a mis padres y abuelos, que nada de esto hubiera sido posible si no fuera por su dedicación e interés en mí superación, espero que se sientan orgullosos.
- A mi tutor por su apoyo, por todas las horas dedicadas, ojalá algún día llegue a ser como él.

Antonio Darío Blanco Rodríguez

- A mis padres, por la educación y la dedicación constante, esta tesis es fruto de ello.
- A mi hermana, por ser un pilar fundamental en mi vida.
- A mis abuelos, por sus consejos ante la vida.
- A mi tutor, por la confianza que depositó en nosotros y brindarnos un conocimiento infinito.
- A mis amigos, porque el haber llegado hasta aquí habría carecido de la fuerza y los momentos inolvidables que brinda la amistad.
- A todos aquellos que han aportado a mi formación profesional, sirva esta tesis como premio a su esfuerzo.

Dedicatoria

Eduardo Hernández García

A mis padres y abuelos por todo el amor y apoyo que me han dado. Gracias por existir.

Antonio Darío Blanco Rodríguez

A mi familia, en especial a mis padres, mi hermana y abuelos, no existe mayor amor que el que emana de ustedes.

A mis hermanos Carlitos, Jp y Reydel.

A la memoria de Rafael.

Resumen

La presente investigación, titulada Extensión de algoritmos de estadística multivariada para problemas de predicción con salidas compuestas como parte de la herramienta Mulan, tiene como objetivo implementar, como extensión de Mulan, los algoritmos basados en modelos de predicción con salidas compuestas, que explotan la interdependencia entre las variables de salida a través de la estadística multivariada. Además mejorar el error predictivo con respecto a los algoritmos del estado del arte para determinados problemas. En la investigación se partió del estudio de los diferentes algoritmos de estadística multivariada, que explotan la interdependencia entre las variables de salida, para tratar problemas de predicción con salidas compuestas, así como un estado actual de los algoritmos que solucionan problemas de predicción con salidas compuestas. La estadística multivariada proporciona métodos objetivos para conocer la relación que existe entre dos conjuntos de variables y cuántas variables son necesarias para describir una realidad compleja y determinar su estructura. Los problemas de predicción con salidas compuestas son aquellos que permiten predecir un conjunto de variables de salida de manera simultánea a partir de un conjunto de entrenamiento dado, estos se dividen en dos grupos, los que explotan la interdependencia entre las variables de salida (explota la relación entre las variables de salida) y los que no (cada variable de salida se determina de forma independiente). Se emplearon 12 bases de datos reales para evaluar los algoritmos mediante el promedio del error cuadrático medio entre lo predicho por los algoritmos y lo medido en cada base de datos para las variables de salida. Se aplicó la prueba no paramétrica de Friedman para comparar estadísticamente los resultados de los algoritmos y la prueba *post-hoc* de Nemenyi para determinar si existen diferencias significativas. Como conclusiones, se pudo determinar que los algoritmos que tratan la interdependencia entra las variables de salida en los problemas de predicción con salidas compuestas muestran mejores resultados en cuanto al error de la predicción con respecto a los que no la tratan, además haciendo uso de la estadística multivariada para el tratamiento de la interdependencia entre las variables de salida se mejoró el error predictivo en varias de las bases de datos evaluadas.

Palabras claves: Problemas de predicción con salidas compuestas, estadística multivariada, interdependencia, predicción.

Índice General

Introducción

Dentro de la inteligencia artificial existe la rama de aprendizaje automático [1], que permite a las computadoras aprender de forma automática modelos para resolver nuevos problemas o mejorar el comportamiento en problemas ya observados. El objetivo principal del proceso de aprendizaje es utilizar la evidencia conocida para poder crear una hipótesis y poder dar una respuesta a nuevas situaciones no conocidas. En función de la manera en que se utilizan los datos y el conocimiento que se tiene a priori de la información histórica se identifican varios paradigmas. Los paradigmas son: el aprendizaje supervisado, que se caracteriza por utilizar el conocimiento de cuáles son las salidas esperadas para cierto conjunto de datos de entrada (también conocidos como datos etiquetados) y el aprendizaje no supervisado, en el que no se tiene ningún conocimiento a priori sobre el etiquetado de los datos iniciales.

El aprendizaje automático supervisado tiene diferentes tareas dentro de las cuales las más comunes son: los problemas de clasificación y predicción. Los mismos son modelados a partir de un conjunto de variables predictoras y una variable de salida la cual se desea predecir o clasificar a partir de un modelo matemático aprendido. Si los posibles valores de la salida se restringen a un conjunto binario, entonces se trata de un problema de clasificación. Por otra parte, cuando la salida toma valores dentro de un conjunto infinito de la recta de números reales, entonces es un problema de predicción.

En los últimos años, en la rama de aprendizaje automático han aparecido problemas de la vida práctica donde se aplican las tareas de clasificación o predicción, que no pueden ser modelados según estos problemas clásicos. Estos problemas están relacionados con la necesidad de predecir valores de salidas, modelados a través de diferentes estructuras complejas. Las salidas complejas pueden ser: grafos, jerarquías o vectores con valores reales o binarios. Como término general este tipo de tarea es conocida como problemas de predicción con salidas estructuradas[2].

Los problemas de predicción con salidas estructuradas están estrechamente relacionados con muchos problemas reales de la vida práctica que por su importancia tienen en cuenta la relación que existe entre los valores de salida. Un ejemplo práctico tiene que ver con los problemas para la clasificación de texto [3]: la ambigüedad en este caso surge debido a que es razonable que un documento esté en dos o más categorías simultáneamente, y por tanto debiera estar asociado a varias etiquetas. No es difícil encontrar un artículo científico en el área de Ciencias de la Computación que pueda ser categorizado como de aprendizaje automático y bases de datos simultáneamente.

Otros dominios donde se han resuelto los problemas con salidas estructuradas son la predicción del ruido de componentes de vehículos [4], el modelado ecológico [5], monitoreo de la calidad del agua [6] y la vigilancia de los bosques [7].

Un caso particular de los problemas con salidas estructuradas es la predicción con salidas compuestas donde la variable de salida es un vector de valores reales. De manera general podemos afirmar que, un problema de predicción con salidas compuestas intenta predecir simultáneamente y con un único modelo todos los atributos(reales) del espacio de salida expresados en forma vectorial.

En este contexto, hay varios enfoques: un primer enfoque es descomponer cada variable de salida de conjunto con las variables predictoras para conformar múltiples problemas de salidas simples, lo cual consiste en construir un modelo independiente para cada salida, mientras se ignoran las restantes. Un segundo enfoque se basa en construir un único modelo para predecir todas las variables de salida simultáneamente, a su vez existen trabajos donde se mejora el poder predictivo de los algoritmos explotando la interdependencia entre las variables de salida.

Los algoritmos de predicción con salidas compuestas obtienen mejores resultados en cuanto al error predictivo cuando se explota la interdependencia entre las variables de salida. En este sentido, los primeros resultados fueron propuestos en [8], cuyos resultados de predicción pueden ser mejorados en algunas de las bases de datos de dominio específico donde han sido evaluados.

Otros problemas de salidas estructuradas son los problemas de clasificación multi-etiqueta, que no es más que la tarea de clasificar vectores de salida binario; y el aprendizaje multi-tarea, que tiene como objetivo mejorar el desempeño de varias tareas de aprendizaje a través de modelos compartidos. Los modelos multi-tarea

normalmente predicen cada atributo de destino de forma individual, pero con modelos al menos parcialmente distintos.

Una forma de tratar la interdependencia entre las variables de salida para problemas de predicción con salidas compuestas es a través de la correlación que pueda existir entre dichas variables. En este contexto los algoritmos de estadística multivariada permiten determinar la relación entre dos conjuntos de variables, reducir el conjunto de variables en nuevas variables, construidas como transformaciones de las originales, con la mínima pérdida de información, así como clasificar nuevas observaciones en grupos definidos.

Una de las técnicas de estadística multivariada es el análisis de correlación canónica que mide el nivel de relación lineal entre dos conjuntos de variables, introducido por Hotelling en 1936 [9]. En su trabajo pretendía investigar las relaciones entre ambos conjuntos de variables y conocer cuántas dimensiones independientes tenía la relación existente entre ellas. Este trabajo marco el inicio en la historia del estudio de la estadística multivariada. Los modelos de estadística multivariada han sido estudiados durante muchos años con un enfoque de evaluación estadística para el análisis de correlación canónica. Estos modelos han arrojado resultados satisfactorios en la forma en que se tratan dos conjuntos de vectores y los transforman a un espacio donde su correlación es máxima. Un enfoque de estadística multivariada para problemas de predicción con salidas compuestas, donde los vectores de entrada y salida sean transformados al espacio canónico, no ha sido tratado.

Para el aprendizaje automático existen varias herramientas como son: R [10], Matlab [11], Weka [12], entre otras. R es un conjunto integrado de servicios de software para la manipulación de datos, cálculo y representación gráfica; Matlab, es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M), disponible para las plataformas Unix, Windows, Mac OS X y GNU/Linux; y Weka, es una plataforma de software para el aprendizaje automático y la minería de datos escrito en Java y desarrollado en la Universidad de Waikato. Todas estas herramientas son de propósito general y permiten la experimentación en el ámbito del aprendizaje automático. Los algoritmos de aprendizaje automático implementados en ellas se centran en las tareas clásicas de predicción y clasificación. La extensión en estas herramientas a otras tareas de predicción estructuradas no resultaría costoso por las facilidades que brindan, en el caso de Matlab el trabajo con matrices es una de sus potencialidades mientras que en Weka al emplearse como librería es posible utilizar sus algoritmos de aprendizaje automático como

base para otras herramientas que dan solución a las tareas de predicción con salidas compuestas.

Los problemas con salidas estructuradas son tratados en diferentes herramientas como Mulan[8], Clus[13] y Meka[14]. Mulan es una plataforma de software libre para problemas de salidas múltiples (clasificación multi-etiqueta, clasificación de salidas compuestas); Clus , es un sistema que implementa el marco de trabajo *predictive clustering (PC)* basado en árboles de decisión y reglas de inducción. Este marco de trabajo unifica los métodos no supervisados y los modelos predictivos permitiendo de forma natural la extensión a modelos predictivos mas complejos tales como aprendizaje multitarea y clasificación multi-etiqueta. Meka proporciona una implementación de código abierto de los algoritmos de clasificación y evaluación multi-etiqueta. Las tres herramientas anteriormente descritas emplean a Weka como librería para el aprendizaje automático. La herramienta Mulan contiene la mayor parte de los algoritmos de predicción con salidas compuestas propuestos en la literatura, solo el PCR y el PCT se encuentran en Clus los cuales pueden ser ejecutados e implementados en la herramienta Mulan a través del paquete *mulan.regressor.clus* contenido en ella.

Tomando en cuenta los elementos anteriormente planteados se afirma que:

- Los algoritmos de predicción con salidas compuestas que explotan la interdependencia entre las variables de salida muestran mejores resultados en cuanto al error de predicción que aquellos que predicen independientemente las variables de salida.

- Los algoritmos del estado del arte de predicción con salidas compuestas que explotan la interdependencia son mejorables en cuanto al error predictivo para dominios de aplicación específicos.

- El tratamiento de la interdependencia a través de la correlación entre las variables predictoras y de salida empleando la teoría conocida como estadística multivariada no ha sido utilizada con anterioridad para resolver problemas de predicción con salidas compuestas.

- La herramienta Mulan contiene una base experimental y un paquete de algoritmos de problemas de predicción con salida compuestas que facilita el diseño de experimentos en este contexto sin embargo no cuenta con algoritmos de estadística multivariada.

A partir de la problemática existente, se formula como **Problema a resolver** en la investigación: ¿Cómo mejorar el error predictivo en los problemas de predicción con salidas compuestas para dominios de aplicación específicos, que explotan la interdependencia entre las variables de salida en la herramienta Mulan?

A partir del problema científico se define como **Objeto de estudio** para esta investigación: Algoritmos de predicción con salidas compuestas, y como **Campo de acción**: Algoritmos de predicción con salidas compuestas que explotan la interdependencia a través de la estadística multivariada, integrado en la herramienta Mulan.

La presente investigación tiene como **Objetivo General**: Implementar, como extensión de Mulan, los algoritmos basados en modelos de predicción con salidas compuestas, que explotan la interdependencia entre las variables de salida a través de la estadística multivariada.

Como **Objetivos Específicos**:

- Caracterizar el marco teórico-conceptual de los algoritmos de estadística multivariada así como los de predicción con salidas compuestas haciendo énfasis en aquellos que explotan la interdependencia entre las variables de salida.
- Implementar los algoritmos basados en estadística multivariada para problemas de predicción con salidas compuestas en la herramienta Mulan.
- Validar la solución implementada mediante el diseño de experimentos para evaluar el error de predicción de estos algoritmos y compararlos con los del estado del arte, haciendo énfasis en los que explotan la interdependencia entre las variables de salida.

Como **idea a defender** en la presente investigación se plantea que: Con la implementación en Mulan de los algoritmos basados en modelos de predicción con salidas compuestas, que explotan la interdependencia entre las variables de salida a través de la estadística multivariada se reducirá el error predictivo en dominios de aplicación específicos, con respecto a los algoritmos de predicción con salidas compuestas del estado del arte que explotan la interdependencia.

Se emplean los siguientes **Métodos Científicos:**

Analítico-Sintético

Se emplea para conformar un estado del arte sobre los algoritmos de predicción con salidas compuestas y estadística multivariada, así como la métrica y los métodos para evaluar los resultados de los experimentos.

Inductivo-Deductivo

Se usa para llegar a conclusiones sobre qué algoritmos de estadística multivariada implementar para tratar la interdependencia entre las variables de salida.

Modelación

Se emplea en el diseño del diagrama de clases para la integración con la herramienta Mulan y en la modelación de las variantes del algoritmo de estadística multivariada utilizado.

Medición

En la experimentación se utiliza el promedio del error cuadrático medio para el cálculo del error predictivo, así como la prueba de Friedman y el *post-hoc* de Nemenyi para comparar los resultados de los algoritmos propuestos con los del estado del arte.

Aportes de la investigación

Siguiendo los conceptos de ciencia y tecnología, la ciencia se proyecta en la tesis mediante la investigación realizada sobre los distintos algoritmos de estadística multivariada, que se proponen incluir en la herramienta Mulan, y así aplicar el nuevo conocimiento en el desarrollo de proyectos futuros, para lograr avances en la economía, la biotecnología, la bioinformática, entre otros. Mulan posee una comunidad de desarrollo, es de código abierto y cuenta con los requerimientos necesarios para generar experimentos e investigaciones en el ámbito de la predicción con salidas compuestas. Con los nuevos algoritmos implementados en la herramienta Mulan, se beneficiará la comunidad científica internacional, debido a que estos estarán a su disposición para que puedan valorarlos, utilizarlos, así como poder compararse con los resultados obtenidos en los experimentos realizados durante la investigación.

Capítulo 1

Fundamentación teórica

1.1. Introducción

En este capítulo se realizará un estudio sobre los modelos de regresión lineal, la estadística multivariada y los algoritmos de predicción con salidas compuestas haciendo énfasis en aquellos que explotan la interdependencia entre las variables de salida. Un estado del arte a partir del año 2007 de los diferentes algoritmos de predicción con salidas compuestas. Además se expresa la relación entre la estadística multivariada y la predicción con salidas compuestas. Se describen las herramientas a utilizar para el desarrollo e implementación de la investigación y los métodos y metodologías para la evaluación de los algoritmos haciendo énfasis en las pruebas estadísticas no paramétricas.

1.2. Estadística Multivariada

Una generalización de la regresión lineal es cuando se tienen múltiples variables predictoras p y se desea predecir múltiples variables de salida q. La solución a este tipo de problemas lo constituye la estadística multivariada o modelo de regresión multivariado.

Como se expresa en [15] el análisis de datos multivariado constituye el estudio estadístico de varias variables medidas en elementos de una población y tiene como objetivos:

1. **Relacionar dos conjuntos de variables.**

2. **Reducir el conjunto de variables en nuevas variables conocidas como indicadores, construidas como transformaciones de las originales, con la mínima pérdida de información.** Disponer de estos indicadores tiene varias ventajas: (1) si son pocas se pueden representar gráficamente y comparar distintos conjuntos de datos o instantes en el tiempo; (2) simplifican el análisis al permitir trabajar con un número menor de variables; (3) si las variables indicadoras pueden interpretarse, se puede mejorar el conocimiento de la realidad estudiada.

3. **Encontrar grupos en los datos si existen.** En muchas situaciones los grupos son desconocidos a priori y se quiere disponer de un procedimiento objetivo para obtener los grupos existentes y clasificar las observaciones.

4. **Clasificar nuevas observaciones en grupos definidos.**

En la actualidad son muchas las esferas del desarrollo científico donde se emplean técnicas de estadística multivariada como son: en la detección de anomalías de precipitación en Centroamérica[16], el reconocimiento de tumores malignos mediantes imágenes digitales[17], así como en la interpretación comparada de la génesis y evolución de las gnammas en macizos antiguos[18].

En los métodos multivariados, se supone que las variables están correlacionadas, pero las observaciones sobre los individuos son independientes. Generalmente se supone también que el conjunto de variables que intervienen en el análisis poseen una distribución normal multivariada. Esta suposición permite que el análisis multivariado se desarrolle paralelamente al correspondiente análisis univariado basado en una distribución normal [19].

Las variables observadas pueden ser cuantitativas, cuando su valor se exprese numéricamente, o cualitativas, cuando su valor sea un atributo o categoría. Las variables cuantitativas pueden a su vez clasificarse en continuas, cuando pueden tomar cualquier valor real en un intervalo, o discretas, cuando sólo toman un valor entero. Las variables cualitativas pueden clasificarse en binarias, cuando toman únicamente dos valores posibles, o generales, cuando toman más de dos valores.

Según el objetivo del análisis y el tipo de datos obtenidos se sugiere un tipo de tratamiento de la información. En ese sentido existen distintas metodologías o técnicas multivariadas, las cuales han sido clasificadas o

agrupadas en dos métodos: el primero es el método de dependencia, enfocado en las relaciones entre los individuos, entre ellos tenemos: Análisis Discriminante, Regresión Múltiple, Análisis de Varianza Multivariado, entre otras; el segundo grupo corresponde a los llamados métodos de interdependencia, este grupo contrario al anterior se enfoca a la relación entre variables, entre ellas se pueden mencionar: El Análisis de Componentes Principales, Análisis de Conglomerados, Análisis de Factores, Análisis de Correlación Canónica, entre otros [19].

La información de partida en el análisis multivariante puede ser representada mediante una tabla de datos correspondiente a distintas variables medidas en los elementos de un conjunto. La manipulación de estos datos se simplifica mucho utilizando los conceptos de vector, matriz y sus propiedades.

Un conjunto de n datos numéricos de una variable puede representarse geométricamente asociando cada valor de la variable a una dimensión del espacio n dimensional, obteniendo un punto en ese espacio, y también el vector que une el origen con dicho punto. Esta analogía entre variables y vectores es útil, porque los métodos de descripción estadística de una variable tienen una correspondencia clara con las operaciones básicas que se realizan con vectores.

Un conjunto de n números reales x puede representarse como un punto en el espacio de n dimensiones, R^n. El segmento orientado que une el origen de coordenadas con el punto x se conoce como vector x. En Estadística se asocian los valores de una variable en n elementos de un vector en R^n, cuyo componente i-ésimo es el valor de la variable en el elemento i.

Por otra parte para trabajar conjuntamente con p variables o vectores definimos el concepto de matriz. Una matriz es un conjunto de números dispuestos en filas y columnas y puede verse como un conjunto de vectores columna o un conjunto de vectores fila. Se dice que una matriz tiene dimensiones $(n \times p)$ si tiene n filas y p columnas.

Un elemento fundamental para el análisis de los datos lo constituye en si la matriz de datos, $\boldsymbol{X}$, de dimensiones $(n \times p)$ donde se observan p variables en un conjunto de n elementos. Cada una de estas p variables se denomina variable escalar o univariante y el conjunto de las p variables forman una variable vectorial o multivariante. Se denota por x_{ij} al elemento genérico de esta matriz, que representa el valor de la variable escalar j sobre el individuo i.

La comprensión de la estructura de dependencia entre las variables es un elemento imprescindible en el análisis de los datos, pues permite valorar cuán relacionados se encuentran las variables en la matriz de datos que se analiza.

1.2.1. Medidas de dependencia lineal

Las medidas de dependencia lineal pueden estudiarse: (1) entre pares de variables, (2) entre una variable y todas las demás, (3) entre pares de variables pero eliminando el efecto de las demás variables, (4) entre el conjunto de todas las variables. Por la relevancia que representa para la investigación solo se analizará la dependencia entre pares de variables.

1.2.2. La matriz de correlación. Dependencia a pares

La dependencia lineal entre dos variables se estudia mediante el coeficiente de correlación lineal o simple. Este coeficiente para las variables x_j , x_k es:

$$r_{jk} = \frac{S_{jk}}{S_j S_k}$$

donde S_{jk} representa la relación lineal entre las variables x_j y x_k conocida como **covarianza** y se calcula como:

$$S_{jk} = \frac{1}{n} \sum_{i=1}^{n} (x_{ij} - \overline{x}_j)(x_{ik} - \overline{x}_k)$$

y tiene las propiedades siguientes:

1. $0 \leq |r_{jk}| \leq 1$.
2. Si existe una relación lineal exacta entre las variables, $x_{ij} = a + bx_{ik}$, entonces $|r_{jk}| = 1$.
3. r_{jk} es invariante ante transformaciones lineales de las variables.

La dependencia por pares entre las variables se mide por la matriz de correlación, $\boldsymbol{R}$, matriz cuadrada y simétrica que tiene unos en la diagonal principal y fuera de ella los coeficientes de correlación lineal entre pares de variables.

$$\boldsymbol{R} = \begin{bmatrix} 1 & r_{12} & \dots & r_{1p} \\ \vdots & \vdots & \dots & \vdots \\ r_{p1} & r_{p2} & \dots & 1 \end{bmatrix}$$

Para medir el nivel de relación lineal entre dos conjuntos de variables se emplea el análisis de correlación canónica.

1.2.3. Análisis de Correlación Canónica (ACC)

Introducido por Hotelling en 1936 [9], en su trabajo pretendía investigar las relaciones entre dos conjuntos de variables y conocer cuántas dimensiones independientes tenía la relación existente entre ellas[15].

Dado dos conjuntos de variables $y = (y_1, y_2, \dots, y_p)$ y $x = (x_1, x_2, \dots, x_q)$ tomados de la misma fuente de datos se puede medir la relación entre ambos mediante la correlación canónica empleando la matriz de covarianza, $\boldsymbol{S}$, y la matriz de correlación, $\boldsymbol{R}$.

En [20] se expresa que la correlación canónica es una extensión de la regresión múltiple, la cual mide la relación entre una variable de salida y y múltiples variables predictoras x.

La covarianza y la correlación existente entre $y, x_1, x_2, \dots, x_q$ de la muestra se expresa como:

$$S = \begin{pmatrix} \boldsymbol{s}_y^2 & \boldsymbol{s}_{yx} \\ \boldsymbol{s}_{yx} & \boldsymbol{S}_{xx} \end{pmatrix} \tag{1.1}$$

$$R = \begin{pmatrix} 1 & \boldsymbol{r}_{yx} \\ \boldsymbol{r}_{yx} & \boldsymbol{R}_{xx} \end{pmatrix} \tag{1.2}$$

donde $\boldsymbol{s}_{yx} = (\boldsymbol{s}_{y1}, \boldsymbol{s}_{y2}, \ldots, \boldsymbol{s}_{yq})$ contiene la covarianza de y con $x_1, x_2, \ldots, x_q$ y $\boldsymbol{S_{xx}}$ es la matriz de covarianza de las x. Igualmente la matriz $\boldsymbol{R}$ contiene la correlación entre y con $x_1, x_2, \ldots, x_q$ denotado por $r_{yx} = (r_{y1}, r_{y2}, \ldots, r_{yq})$ y $\boldsymbol{R_{xx}}$ es la matriz de correlación de las x.

A partir de la matriz de correlación y la matriz de covarianza es posible determinar el coeficiente de correlación múltiple al cuadrado, $\boldsymbol{R^2}$:

$$R^2 = \frac{\boldsymbol{s}_{yx}\boldsymbol{S}_{xx}^{-1}\boldsymbol{s}_{yx}}{\boldsymbol{s}_y^2} = \boldsymbol{r}_{yx}\boldsymbol{R}_{xx}^{-1}\boldsymbol{r}_{yx} \tag{1.3}$$

En R^2 las q covarianzas entre y y las x en S_{yx} o las q correlaciones entre y y las x en r_{yx} son canalizadas en una relación lineal entre y y las x. Así como la correlación múltiple, R, se define como la máxima correlación entre y y una combinación lineal de las x, $R = \max br_y, b^{'}x$.

En el caso de que se necesiten analizar varias variables de salida, y, y varias variables predcitivas, x, la estructura de la covarianza está asociada a dos subvectores $\boldsymbol{y}$ y $\boldsymbol{x}$, denotados por $(y_1, \ldots, y_p, x_1, \ldots, x_q)$ los cuales tienen asociados una matriz de covarianza:

$$S = \begin{pmatrix} \boldsymbol{S}_{yy} & \boldsymbol{S}_{yx} \\ \boldsymbol{S}_{xy} & \boldsymbol{S}_{xx} \end{pmatrix}$$

donde $\boldsymbol{S}_{yy}$ es la $(p \times p)$ matriz de covarianza de las y, $\boldsymbol{S}_{yx}$ es la $(p \times q)$ matriz de covarianza entre las y y las x y $\boldsymbol{S}_x x$ es la $(q \times q)$ matriz de covarianza entre las x. Partiendo de estas matrices se puede calcular el coeficiente de correlación al cuadrado como:

$$R_M^2 = |\boldsymbol{S}_{yy}^{-1}\boldsymbol{S}_{yx}\boldsymbol{S}_{xx}^{-1}\boldsymbol{S}_{xy}| = \Pi_{i=1}^{s} r_i^2$$

donde $s = \min(p, q)$ y $r_1^2, r_2^2, \ldots, r_s^2$ son los autovalores que se obtienen de $\boldsymbol{S}_{yy}^{-1}\boldsymbol{S}_{yx}\boldsymbol{S}_{xx}^{-1}\boldsymbol{S}_{xy}$ La raíz cuadrada de los autovalores, $r_1, r_2, \ldots, r_s$ es lo que se conoce como la correlación canónica.

El mejor valor de la relación es el que se obtiene a partir del máximo autovalor de la correlación canónica al cuadrado, r_1^2 de $\boldsymbol{S}_{yy}^{-1}\boldsymbol{S}_{yx}\boldsymbol{S}_{xx}^{-1}\boldsymbol{S}_{xy}$, el resto de los autovalores proveen la medida de las dimensiones existentes en la relación lineal entre y y x.

El autovalor r_1^2 representa además la máxima correlación entre la combinación lineal de las y, $u = a^{'}y$, y la combinación lineal de las x, $v = b^{'}x$, representado por:

$$r_1 = \max_{a,b} r_{a^{'}y, b^{'}x} \tag{1.4}$$

Se denota como los vectores de coeficientes que producen la correlación máxima a b_1 y a_1. De tal manera r_1, la raíz positiva de r_1^2, es la correlación entre $u_1 = a_1^{'}y$ y $v_1 = b_1^{'}x$. Los vectores de coeficientes a_1 y b_1 se establecen como autovectores. La función lineal entre u_1 y v_1 es lo que se conoce como la primera variable canónica [15]. El resto de las variables canónicas $r_2, r_3, \ldots, r_s$ corresponde a $u_i = a_i^{'}y$ y $v_i = b_i^{'}x$.

Este método es empleado en la visión por computadoras para el reconocimiento de objetos, específicamente estimación de posturas, predicción de superficie facial y el reconocimiento de coincidencias en modelos de apariencia con nuevas imágenes [21]. Además, se utiliza en la meteorología en aspectos como la predicción de eventos extremos y días con precipitaciones, atendiendo a la temperatura superficial del mar [22], y en climatología para la comprensión de la variabilidad climática dada la ocurrencia de ciertos fenómenos climáticos de macro escala [23]. En Cuba se ha aplicado en la medicina en estudios de la Aterosclerosis coronaria y daño miocárdico por [24].

1.2.4. Análisis de Correlación Canónica y la regresión lineal

Un problema de múltiples variables de salida $y_1, y_2, \ldots, y_q$ y múltiples variables de entrada $x_1, x_2, \ldots, x_p$ con n observaciones se puede expresar de forma matricial como:

$$Y = XC + E \tag{1.5}$$

donde Y es las $n \times q$ matriz de variables de salida, X es la $n \times p$ matriz de variables predictoras, C es la $p \times q$ matriz de los coeficientes de la regresión y E es la $n \times q$ matriz de error, donde cada fila de X y Y coincide con una observación. La generalización del criterio de mínimos cuadrados para múltiples salidas se expresa como:

$$RSS(C) = \sum_{j=1}^{q}\sum_{i=1}^{n}(y_{ij} - \sum_{k=1}^{p} x_{ik}c_{kj})^2 \tag{1.6}$$

$$= tr[(Y - XC)^T(Y - XC)] \tag{1.7}$$

$$= \|Y - XC\|^2 \tag{1.8}$$

donde $|\cdot|$ denota la norma Frobenius [25] para una matriz, el estimador para mínimos cuadrados ordinarios (OLS) para C es:

$$\hat{C}_{OLS} = (X^T X)^{-1} X^T Y \tag{1.9}$$

La estimación de las variables dependientes se dificultan para la regresión lineal cuando los espacios de entrada y salida son muy grandes y el tamaño del conjunto de datos de entrenamiento es mayor que el número de parámetros a estimar. El análisis de correlación canónica es una herramienta versátil para darle solución a los problemas antes expuestos pues permite la reducción de la dimensión, el reconocimiento de los subespacios relevantes dentro de la regresión y la extracción de características [21].

A continuación se exponen varios algoritmos de estadística multivariada que ofrecen mejoras en la predicción de las variables de salida para la regresión lineal a través de la estimación de la matriz de coeficientes C.

1.2.5. *Ordinal Least Squares y Redge Regression (RR)*

El método OLS [26], es una técnica estadística que utiliza datos de la muestra para estimar la verdadera relación de la población entre dos variables, además minimiza la suma de las distancias verticales entre las salidas observadas en la muestra y las salidas del modelo.

No es inusual ver que el número de variables de entrada exceda en gran medida el número de observaciones, y que cuando existen muchos predictores, ajustar el modelo completo y sin penalización dará lugar a intervalos de predicción grandes y que el estimador de regresión puede no existir de forma única.

En [27], asumiendo que los datos están centrados, se propuso que el potencial de inestabilidad de OLS,

$$\hat{y} = (X^T X)^{-1} X^T Y \tag{1.10}$$

donde $\hat{y}$ son los valores predichos intermedios, también denotado como la matriz OLS A, podría mejorarse mediante la adición de un valor constante λ por una matriz identidad I_p a las entradas de la diagonal de la matriz $X^T X$ antes de tomar su inversa.

$$\hat{y} = (X^T X + \lambda I_p)^{-1} X^T y \tag{1.11}$$

Ridge regression coloca una forma particular de restricción en los parámetros $\hat{y}$ que se elige para minimizar la suma de cuadrados penalizados. Por lo tanto, *Ridge regression* pone restricciones adicionales sobre los parámetros, en el modelo lineal. En este caso, lo que se hace es que en vez de minimizar la suma de los cuadrados residuales, también se tiene un término de penalidad en el $\hat{y}$. Este término de penalización es λ (un pre-elegido constante) veces la norma al cuadrado del vector $\hat{y}$. Esto significa que si los $\hat{y}$ toman valores grandes, la función de optimización es penalizada.

1.2.6. *Reduced Rank Regresion (RRR)*

RRR es un método de estimación explícita en la regresión multivariante, que tiene en cuenta la restricción de rango reducido en la matriz de coeficientes. Una incidencia directa del RRR sobre la regresión lineal es el número de restricciones lineales sobre la matriz de coeficientes C(1.5) de la regresión, permitiendo que el número de parámetros efectivos se reduzca y la eficiencia de la estimación aumente [28].

1.2.7. *Factor Estimation and Selection (FES)*

En [29] se propone un procedimiento que impone menos restricciones que la selección de variables y el factor lineal de la regresión al permitir que se determinen simultáneamente el factor de estimación, η y el factor de selección A. Similar a la selección de variables, la selección del factor es más potente si todas

las variables de salida pueden ser predichas por pequeños subconjuntos de factores comunes. Idealmente los $\eta_1, \ldots, \eta_p$ debe contener un conjunto base del espacio lineal de la matriz de contracción B que permita la representación del esparcimiento de B en el espacio de los factores.

1.2.8. *Multivariate Regression with Covariance Estimation (MRCE)*

El método Regresion Multivariada con Estimación de Covarianza, (*Multivariate Regression with Covariance Estimation (MRCE)*), propuesto en [30] reduce el número de parámetros usando la penalización *Lasso* [31] en los coeficientes de la matiz $C(1.5)$. Además asume que los predictores no son aleatorios, sin embargo la fórmula resultante para la estimación puede ser la misma con predictores aleatorios. Tiene en cuenta la distribución condicionada de las variables de salidas producto de las variables de entrada.

1.2.9. *Curds and Whey (CW)*

La idea general del algoritmo *Curds and Whey* es tomar las regresiones de mínimos cuadrados, y luego de modificar los valores previstos de esas regresiones por la contracción de ellos, utilizando las correlaciones canónicas entre las variables de salida y las variables predictoras. Breiman y Friedman en su trabajo [32] propone la contracción simultánea tanto en el espacio de entrada como en el de salida.

El enfoque estándar de determinación de la matriz de coeficientes $C(1.5)$ es a través del OLS, donde el resultado que se obtiene es equivalente a la regresión de cada una de las variables de salida a través de las variables predictoras de forma separada. Para este enfoque el error predictivo es pobre en presencia de una correlación alta entre las variables redictivas o cuando existen una cantidad significativa de este tipo de variables.

1.2.10. Análisis de los algoritmos

Los algoritmos expresados anteriormente a excepción del *Curds and Whey* resuelven estos problemas mediante la reducción de dimensionalidad tal es el caso del: FES, RRR y MRCE. En estos algoritmos las

variables de salida son estimadas a partir de un grupo de transformaciones lineales que reducen la dimensión de la matriz de coeficientes de las variables predictoras. Estos algoritmos difieren en la forma en que se determina la matriz de coeficientes. Los coeficientes de la matriz son determinados basados en un número $r \leq \min(p, q)$ de factores seleccionados para una matriz $(n \times p)$ de variables predictoras y una matriz $(n \times q)$ de variables de salida.

Comparado con el RRR, el FES suaviza mejor la restricción de los valores singulares y es más estable debido a que la estimación del RRR difiere solo de la estimación de mínimos cuadrados en los valores singulares. Sin embargo no tiene en cuenta para la determinación de los coeficientes de las variables predictoras el error de la correlación. En el MRCE la correlación de las variables de salida se expresa solo desde la correlación en los errores de la regresión. Este tipo de proceso según Breiman [32] puede ser inestable, es decir que pequeñas modificaciones en los datos pueden ocasionar diferencias en las estimaciones.

A diferencia de la reducción de la dimensionalidad, el algoritmo *Curds and Whey* propone una forma de elevar el poder predictivo teniendo en cuenta la correlación entre las variables de salida. Este método explota la correlación en las variables de salida a través de compartir aleatoriamente con las variables predictivas el error de la predicción.

Por lo antes expuesto la aplicación de técnicas de estadística multivariada en los problemas de predicción con salidas compuestas, particularmente donde se explota la correlación entre las variables, podría ofrecer mejoras en cuanto al error predictivo, en determinados dominios de aplicación. A continuación se explica en que consisten los problemas de predicción con salidas compuestas.

1.3. Problemas de predicción con salidas compuestas

Sean x, y dos vectores aleatorios donde $x = [x_1, \ldots x_d]$ del espacio de entrada $\Re^d$ y $y = [y_1, \ldots y_m]$ en el espacio de salida $\Re^m$ para una instancia determinada. Dado el conjunto $D = \left\{(x^1, y^1), \ldots (x^n, y^n) \subset \Re^d X \Re^m\right\}$ de n instancias de entrenamiento, el objetivo en un problema de predicción de salidas compuestas es aprender un modelo $h : \Re^d \rightarrow \Re^m$ de modo que dada una instancia con entrada conocida x^q es posible predecir, a través de un único modelo, todas las salidas $\hat{y}^q = h(x^q)$ de forma simultánea. La interdependencia para la

$j-$ésima variable de salida queda expresada en un modelo $\tilde{h}(x, \tilde{y})$ con $\tilde{y} = y \backslash y_j$.

Un enfoque del aprendizaje con salidas compuestas es el aprendizaje multitarea [33], donde para cada salida $t \in T$ se tiene un conjunto de ejemplos $D = (\vec{x}_1^t, {y_1}^t), ..., (\vec{x}_m^t, {y_m}^t) \in \Re^d \times \Re$ y define un conjunto de modelos $h_t : \Re^d \longrightarrow \Re$ relativo a cada salida.

La principal diferencia entre éstos es el número de modelos entrenados: un modelo separado para cada una de las tareas frente a un único modelo entrenado durante todo el problema. El aprendizaje con salidas compuestas tiene como meta predecir las características de las salidas y describir explícitamente su relación con las características descriptivas. Además, se describe implícitamente las relaciones entre características de las salidas. El modelo multitarea, por otra parte, no apunta específicamente para describir las relaciones entre características de las salidas.

El aprendizaje con salidas compuestas capta implícitamente las dependencias entre las salidas y los representa en un modelo único generado. A través de este modelo, se determina el efecto de las características descriptivas de todas las salidas, y analizar las relaciones, ya sea lineal o no lineal, entre las salidas (o grupos de salidas). En caso de que las salidas estén relacionadas, se obtiene información sobre estas relaciones.

Varios métodos estándar de aprendizaje han sido extendidos a problemas de predicción con salidas compuestas en los últimos años como (Clusbag, MORF) [34], PCR [35], KNN-SP [36], FIRE [37],(ST, MTSC, MTS, ERC, ERCC) [8].

1.3.1. *Multi-objetive Bagging (Clusbag) y Multi-objetive Random Forest (MORF)*

Clusbag y MORF surgen al aplicar métodos de ensamble *Bagging* [38] y *Random Forests* [39] respectivamente al método *Multi-objective decision trees* (MODTs) [40]. Los métodos de ensamble no son más que un conjunto de clasificadores construidos con un algoritmo dado. Cada nuevo ejemplo se clasifica mediante la combinación de las predicciones de cada clasificador desde el conjunto. Éstas se pueden combinar tomando el promedio (para las tareas de regresión) o el voto de la mayoría (para las tareas de clasificación), como se describe por Breiman [38], o tomando combinaciones más complejas [41, 42].

Para aplicar *Bagging* a MODTs, el procedimiento de *Predictive Clustering Trees* PCTs(E_i) [40] se utiliza

como un clasificador base. Para la aplicación de *Random Forests*, se sigue el mismo enfoque, cambiando el procedimiento *BestTest* de tomar un subconjunto de tamaño aleatorio $f(x)$ de todos los atributos posibles.

Con el fin de combinar la salida de las predicciones por los clasificadores base, se toma la media de la regresión, y se aplica un voto distribución de probabilidad en lugar de una simple mayoría de votos para la clasificación, según lo sugerido por Bauer y Kohavi [43]. Esta combinación de funciones se generaliza trivialmente para el caso en que existan varias variables de salida. Cada conjunto consta de 100 árboles, que son sin podar [43]. Para la construcción de los bosques al azar, el parámetro $f(x)$ se establece en $\log_2(x) + 1$ como en Breiman [39].

1.3.2. *Predictive clustering rules (PCRs)*

Incluyen ideas de aprendizaje de reglas de agrupamiento. El propio algoritmo es una generalización de los enfoques de reglas de aprendizaje existentes. La regla de la función de evaluación, la cual sirve como una heurística de búsqueda, sin embargo, emplea técnicas comúnmente utilizadas en agrupamiento.

1.3.3. *k-Nearest Neighbors for Structured Prediction (KNN-SP)*

Para predecir la estructura de las salidas con el método KNN, se toma el k vecino más cercano de los ejemplos de entrenamiento y sus valores de salida. KNN-SP necesita calcular el prototipo de los valores de salida estructurados de los vecinos para hacer la predicción. Los prototipos se calculan atendiendo a cada una de las distintas salidas estructuradas, que son *multi-target regression, Hierarchical multi-label classification, Predicting (short) time series*[36].

1.3.4. *FIRE*

El algoritmo FIRE se emplea para resolver los problemas de predicción con salidas compuestas, que emplea el enfoque de conjuntos de reglas. FIRE optimiza los pesos de las reglas y términos lineales con un gradiente dirigido al algoritmo de optimización. Este procedimiento de optimización depende de un gradiente

del parámetro de umbral t; se repite la optimización para diferentes valores de t con el fin de encontrar un conjunto de pesos con el error de validación más pequeño. Al final, se eliminan todas las reglas y términos lineales cuya ponderación es cero.

1.3.5. *Single Target (ST)*

En el método ST [8], un modelo h con salidas compuestas es convertido a m modelos de salidas simples $h_j : X \rightarrow R$ donde cada modelo h_j es entrenado en una transformación del conjunto de entrenamiento $D = (x^1, y_j^1), .., (x^n, y_j^n)$ para predecir el valor de una variable de salida simple Y_j. De esta manera, las variables de salida son predichas independientemente y no se explota la relación entre ellas.

1.3.6. *Multi-Target Stacking (MTS)*

En [8] se adapta el método de *stacking* para regresión con salidas compuestas, el cual se denota como *Multi-Target Stacking* (MTS). El entrenamiento de MTS consta de dos etapas. En la primera etapa, *m* modelos independientes de salidas simples $h_j : X \rightarrow R$ son aprendidas como un *Single Transformation* (ST). Sin embargo, en lugar de usar directamente estos modelos para predicción, MTS tiene una segunda etapa de entrenamiento adicional donde un segundo conjunto de *m* modelos objetivos $h_j^* : X \times R^{m-1} \rightarrow R$ son aprendidos, uno para cada salida Y_j.

1.3.7. *Regressor Chains (RC)*

El método de *Regressor Chains* (RC) se basa en la idea de modelos de encadenamiento de salidas simples. El entrenamiento del RC consiste en seleccionar una cadena aleatoria (permutación) del conjunto de variables de salida y luego construir un modelo de regresión separado para cada salida. La diferencia con respecto a MTS es que primeramente construye un modelo igual al construido por el método *Single Transformation* (ST) para esta salida, pero los modelos subsiguientes son entrenados en un conjunto de datos transformados.

Una propiedad notable de RC es que es sensible en el ordenamiento de la cadena seleccionada. El principal

problema que surge de la utilización de una sola cadena al azar, es que las salidas que aparecen antes en una cadena no pueden modelar posibles relaciones estadísticas con las salidas que aparecen más adelante en la cadena. Además el error de predicción es probable que sea propagado y amplificado a lo largo de la cadena cuando hacemos predicciones para una nueva instancia de prueba. Para mitigar estos efectos, se propone un esquema de conjunto llamado *Ensemble of Classifier Chains* donde una prueba de k (típicamente k=10) modelos *Classifier Chains* (CC) con cadenas de diferente orden son construido sobre muestras de arranque del conjunto de entrenamiento y las predicciones finales provienen de la votación por mayoría. Este esquema ha demostrado mejorar consistentemente la exactitud de un solo CC en el dominio de clasificación. Se aplica la misma idea (sin muestreo) en RC y se calcula las predicciones finales tomando la media de las estimaciones de k para cada objetivo. El método resultante se llama *Ensemble of Regressor Chains* (ERC).

1.3.8. *MTS Corrected (MTSC) y ERC Corrected (ERCC)*

Una modificación de los algoritmos MTS y ERC son *MTS Corrected* (MTSC) y *ERC Corrected* (ERCC), en [8] se plantea que tanto MTS y ERC se basan en la misma idea central de tratamiento de las otras salidas de predicción como variables de entrada adicionales que aumentan el espacio de entrada original. Estas meta-variables difieren de las variables de entrada común en el sentido de que mientras que sus valores reales se encuentran disponibles en el tiempo de formación, se están perdiendo en la predicción.

Con el fin de enfrentar el problema antes mencionado se propone en [8] unas modificaciones mediante el empleo de un procedimiento que se asemeja a la de *f-fold cross-validation* con el fin de obtener estimaciones imparciales fuera de la muestra de las meta-variables. El enfoque de validación cruzada evita el problema del enfoque *hold-out* como todos los ejemplos de entrenamientos son usados en el segundo estado ST de los modelos MTS y el RC en los modelos ERC. Se espera que en comparación con el uso de los valores reales o estimados en la muestra, las estimaciones de validación cruzada se aproximarán a la distribución de las estimaciones obtenidas en el momento de la predicción de forma más precisa y por lo tanto será más útil para el modelo.

Después de realizar un estudio de los diferentes algoritmos de predicción con salidas compuestas, se

identifica que los algoritmos MTS, MTSC, ERC y ERCC explotan la interdependencia entre las variables de salida y muestran los mejores resultados en cuanto al error predictivo, según [8], con respecto a los que tratan cada variable de salida de forma independiente.

1.4. Metodología

En esta sección se exponen las diferentes metodologías empleadas para la evaluación y comparación de los resultados en cuanto al error de predicción.

1.4.1. Validación cruzada

La validación cruzada es un método de evaluación de modelos que se emplea para mejorar el error predictivo. El problema de la evaluación del error es que no se conoce que tan bien el modelo ha aprendido para predecir datos desconocidos. Una forma de solucionar este problema es no emplear todo el conjunto de datos para entrenar el modelo. Una parte del conjunto de datos es removido del conjunto de entrenamiento y empleado para la prueba, una vez que el modelo halla sido entrenado.

El método *hold-out* es el mas simple de los tipos de validación cruzada. Este consiste en dividir en dos conjuntos complementarios el conjunto datos, realizar el análisis de un subconjunto (denominado datos de entrenamiento), y validar el análisis en el otro subconjunto (denominado datos de prueba), de forma que la función de aproximación sólo se ajusta con el conjunto de datos de entrenamiento y a partir de aquí calcula los valores de salida para el conjunto de datos de prueba (valores que no ha analizado antes). La ventaja de este método es que es muy rápido a la hora de computar. Sin embargo, este método no es demasiado preciso debido a la variación del resultados obtenidos para diferentes datos de entrenamiento. La evaluación puede depender en gran medida de cómo es la división entre datos de entrenamiento y de prueba, y por lo tanto puede ser significativamente diferente en función de cómo se realice esta división. Debido a estas carencias aparece el concepto de validación cruzada [44].

En la validación cruzada de k iteraciones o *k-fold cross-validation* los datos de muestra se dividen en

k subconjuntos. Uno de los subconjuntos se utiliza como datos de prueba y el resto $(k-1)$ como datos de entrenamiento. El proceso de validación cruzada es repetido durante k iteraciones, con cada uno de los posibles subconjuntos de datos de prueba. Finalmente se realiza la media aritmética de los resultados de cada iteración para obtener un único resultado. Este método es muy preciso puesto que evaluamos a partir de k combinaciones de datos de entrenamiento y de prueba, pero aun así tiene una desventaja, y es que, a diferencia del método de retención, es lento desde el punto de vista computacional. En la práctica, la elección del número de iteraciones depende de la medida del conjunto de datos. Lo más común es utlizar la validación cruzada de 10 iteraciones (*10-fold cross-validation*) [45].

La validación cruzada dejando uno fuera o *Leave-one-out cross-validation* (LOOCV) implica separar los datos de forma que para cada iteración se tiene una sola muestra para los datos de prueba y todo el resto conformando los datos de entrenamiento. La evaluación viene dada por el error, y en este tipo de validación cruzada el error es muy bajo, pero en cambio, a nivel computacional es muy costoso, puesto que se tienen que realizar un elevado número de iteraciones, tantas como n muestras tengamos y para cada una analizar los datos tanto de entrenamiento como de prueba [44].

De los métodos de validación cruzada antes expuesto se emplea para evaluar los modelos predictivos la validación cruzada de k iteraciones con $k = 10$, teniendo en cuenta que es el más empleado en la evaluación de los algoritmos del estado del arte. Además este método es más preciso que el *hold-out*, puesto que evalúa a partir de k combinaciones de datos de entrenamiento y de prueba, todo el conjunto de datos y computacionalmente es menor el costo que el *Leave-one-out*.

1.4.2. Pruebas estadísticas

Según el artículo de [46], para la comparación de múltiples clasificadores se encuentra la prueba de Friedman [47, 48], la cual jerarquiza los algoritmos por cada colección de datos separadamente y en caso de empate asigna un rango promedio. Tomando r_i^j como el rango del algoritmo j de k algoritmos en la i-*ésima* colección de datos de N colecciones de datos, esta prueba compara los rasgos promedios de los algoritmos $R_j = 1/N \sum_i r_i^j$. En esta, la hipótesis nula afirma que todos los algoritmos son equivalentes y por eso sus

rangos R_j deberían ser iguales. Bajo esta hipótesis nula, la estadística de la Prueba de Friedman se calcula:

$$X_F^2 = \frac{12N}{k(k+1)} \left[\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right]$$

y es distribuida de acuerdo a X_F^2 con $k-1$ grados de libertad cuando k y N son lo suficientemente grandes ($k > 5, N > 10$) y en caso contrario [49, 50] se calculan valores críticos exactos. En [51] se demostró que la X_F^2 de Freidman no es lo conservativa y se derivó una mejor estadística:

$$F_F = \frac{(N-1)X_F^2}{N(k-1) - X_F^2}$$

la cual es distribuida de acuerdo a la distribución F con $k-1$ y $(k-1)(N-1)$ grados de libertad. Teóricamente la Prueba de Friedman no paramétrica tiene menos poder que el ANOVA [52] paramétrico cuando las asunciones de ANOVA se encuentran, en caso contrario no necesariamente. En [48] experimentalmente se compara en 56 problemas independientes y se demostró que los dos coincidían generalmente. Cuando uno encontraba significancia en $p < 0{,}01$ el otro la muestra en al menos $p < 0{,}05$ respectivamente. Solamente en dos casos ANOVA encontró significancia que era insignificante para Friedman, mientras que lo contrario ocurrió en 4 casos.

Si la hipótesis nula es rechazada se puede proceder con una Prueba de Post-hoc. La Prueba de Nemenyi [53] es similar a la de Tukey [54] para ANOVA y es usado cuando los clasificadores son comparados entre sí. El desempeño de los mismos es significantemente diferente si el rango ordinario correspondiente difiere al menos la diferencia crítica (CD) donde los valores críticos q_a están basados en la estadística del rango *Studentized* dividida por $\sqrt{2}$. Esta diferencia crítica se calcula de la siguiente manera:

$$CD = q_a \sqrt{\frac{k(k+1)}{6N}} \tag{1.12}$$

Cuando todos los clasificadores se comparan con un clasificador de control, se puede hacer la prueba de Nemenyi para controlar el error *Family-Wise* (FWE) en la prueba de múltiples hipótesis como la corrección de Bonferroni u otros similares. El FWE es la probabilidad de hacer al menos un tipo de error 1 en cualquiera de las comparaciones. Aunque estos métodos se pueden considerar generalmente conservativos, en este caso

son más efectivos que el de Nemenyi, ya que este ajusta el valor crítico para realizar $k(k1)/2$ comparaciones mientras que al comparar con un clasificador de control solo se hacen $k1$ comparaciones. La estadística para comparar los clasificadores i y j utilizando estos métodos es:

$$z = (R_i - R_j)/\sqrt{\frac{k(k+1)}{6N}}$$

Este valor es usado para encontrar la probabilidad correspondiente para compararla con un a apropiada. Las pruebas difieren en la forma que ajustan el valor de a para compensar por comparaciones múltiples.

El test de Bonferroni-Dunn [55] controla la tasa del FWE al dividir a por la cantidad de comparaciones realizadas, o calcular la CD usando la ecuación de Nemenyi pero usando los valores críticos para $a/(k1)$. Como contraste del test de Bonferroni-Dunn, los procedimientos de paso-adelante y paso-atrás prueban secuencialmente las hipótesis ordenadas por su significancia, denotando los valores ordenados p por $p_1, p_2, ..., p_{k1}$ de manera que $p_1 < p_2 < p_{k1}$. Ambos comparan cada p_i con $a/(ki)$, pero difieren en el orden de las pruebas. El procedimiento paso-atrás de Holm [56] comienza con el valor p más significante. Si p_1 es menor que $a/(k1)$ la hipótesis correspondiente es rechazada y se permite la comparación de p_2 y así sucesivamente. En el momento en que alguna hipótesis nula no pueda ser rechazada, las restantes son retenidas también. El procedimiento de paso-adelante de Hochberg [57] trabaja de manera opuesta, comparando el valor más grande de p con a, el segundo con $a/2$, y así consecutivamente hasta que encuentre una hipótesis que pueda rechazar. Todas las hipótesis con valores p más pequeños son rechazadas también.

El procedimiento de Hommel [58] encuentra el valor más grande j para el cual $p_{nj+k} > ka/j$ para todas las $k = 1..j$. Si j no existe se rechazan todas las hipótesis, en caso contrario se rechazan todas para las cuales $pi <= a/j$.

El procedimiento de Holm es más potente que el de Bonferroni-Dunn y no realiza asunciones adicionales sobre la hipótesis probada. La única ventaja de la prueba de Bonferroni-Dunn es que es más fácil de describir y visualizar porque utiliza la misma CD para todas las comparaciones. Los métodos de Hochberg y Hommel rechazan más hipótesis que el de Holm pero en algunas circunstancias exceden el FWE prescrito ya que se basan en las conjeturas de Simes las cuales todavía se encuentran bajo investigación. Las diferencias entre

todos estos métodos en la práctica son pequeñas, así que el método más complejo de Hommel no ofrece gran ventaja sobre el más sencillo de Holm. Aunque estos procedimientos se proponen como Pruebas *Post-hoc* para la Prueba de Friedman, también pueden usarse para controlar el FWE cuando múltiples hipótesis, de posiblemente varios tipos, son probadas.

Se selecciona la prueba de Friedman para determinar si existen diferencias entre los algoritmos evaluados debido a que para utilizar ANOVA se deben cumplir las asunciones, las cuales casi siempre son violadas en el aprendizaje automático. Una primera asunción es que las muestras son extraídas de distribuciones normalizadas cuando no hay garantía de ello, la segunda y más importante asunción de ANOVA es la esfericidad, que es una propiedad que requiere que las variables aleatorias tengan igual varianza, debido a la naturaleza de los algoritmos de aprendizaje y los conjuntos de datos no se puede dar por hecho. Las violaciones de estas asunciones tiene un efecto aún mayor en las pruebas *Post-hoc*, por lo que ANOVA no puede considerarse adecuada para estudios típicos de aprendizaje automático. Además las diferencias entre los resultados de ambas no es significativa. Así como se emplea el *post-hoc* de Nemenyi para determinar si existen diferencias significativas entre los algoritmos evaluados, debido a que muestra una representación gráfica, lo que facilita el análisis y comprensión de los resultados.

1.5. Lenguaje de Programación

Como lenguaje de programación se utilizó:

Java 7

Java (el lenguaje) es un lenguaje de programación de alto nivel orientado a objetos, que está influenciado de varias maneras por C, C++ y Smalltalk, y también por ideas que ha tomado prestadas de otros lenguajes. Su sintaxis fue diseñada para ser familiar a aquellos que estaban familiarizados con los lenguajes que descienden más directamente de C, pero tiene los principios de la orientación a objetos más asumidos que C++, objetos fuertemente tipados y un sistema más justo e inflexible de excepciones que requiere que cada método que se llama trate cualquier tipo de excepción o especifique que puede lanzarlas. La recogida de basura es automática, de esta manera se evita al programador tener que liberar la memoria usada por los objetos que ya no se van a

emplear [59].

1.6. Herramientas

Como herramientas se utilizaron:

Weka 7.3.10

Weka es una extensa colección de algoritmos de máquina de aprendizaje para tareas de minería de datos implementados en Java y desarrollado por la universidad de Waikato, Nueva Zelanda. Incluye herramientas para el pre-procesamiento de los datos, clasificación, regresión, agrupamiento, reglas de asociación y visualización. Weka es un software de código abierto publicado bajo la licencia GNU. Los algoritmos pueden ser aplicados directamente a un conjunto de datos desde la interfaz gráfica del programa (Java Swing), cargarlos desde el shell o utilizar los códigos independientes que se proporcionan mandándolos a llamar desde el propio programa en Java, utilizándolos de la forma en que indica la documentación. Es posible agregar códigos para extender la funcionalidad del paquete ya que proporcionan el código fuente completo [60].

Mulan 1.5.0

Mulan es un software de código abierto para máquinas de aprendizaje cuyo objetivo principal es el trabajo con datos multi-etiquetas. Ofrece los algoritmos que existen sobre clasificación multi-etiqueta y ranking de etiquetas, así como un marco de trabajo que contiene un compendio de medidas para evaluar la clasificación multi-etiqueta mediante la validación *hold-out* y la validación cruzada. Mulan está escrita en Java y se construye encima de Weka para emplear todos los recursos que contiene sobre algoritmos de aprendizaje supervisado aunque esta es independiente. Una característica exclusiva de la librería Mulan es la introducción reciente de un paquete de experimentos que tiene como objetivo reproducir los resultados de los experimentos en un documento de aprendizaje multi-etiqueta. Además, da soporte a la clasificación multi-etiqueta siendo un aporte significativo debido a la no existencia en la mayoría de las plataformas de aprendizaje actuales [61].

Netbeans 7.4

NetBeans es un proyecto exitoso de código abierto con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun MicroSystems fundó el proyecto de

código abierto NetBeans en junio 2000 y continúa siendo el patrocinador principal de los proyectos [62].

Exsisten dos productos: el NetBeans IDE y NetBeans Platform.

NetBeans IDE es un entorno de desarrollo - una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso [62].

También está disponible NetBeans Platform, una base modular y extensible usada como estructura de integración para crear grandes aplicaciones de escritorio. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones [62].

Visual Paradigm 8.0

Visual Paradigm para UML es una herramienta para desarrollo de aplicaciones utilizando modelado UML (*Unified Modeling Language*) ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de sistemas que están interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos [63]. Soporta UML versión 2.1.

1.7. Conclusiones del Capítulo

Luego del análisis del marco teórico de la investigación se puede concluir que:

- Los algoritmos plateados en [8], son los que mejores resultados muestran en cuanto al error predictivo.
- Se escoge el algoritmo *Curds and Whey* para el desarrollo de la investigación, pues propone una forma de elevar el poder predictivo teniendo en cuenta la correlación entre las variables dependientes. Este método explota la correlación en las variables de salida a través de compartir aleatoriamente con las variables predictoras el error de la predicción.
- Se emplea para evaluar los modelos predictivos la validación cruzada de k iteraciones con $k = 10$, teniendo en cuenta que es el más empleado en la evaluación de los algoritmos del estado del arte.

Además este método es más preciso que el *hold-out*, puesto que evalúa a partir de k combinaciones de datos de entrenamiento y de prueba, todo el conjunto de datos y computacionalmente es menor el costo que el *Leave-one-out*.

- Para realizar la comparación de los algoritmos se selecciona la prueba no paramétrica de Friedman para determinar si existen diferencias entre los algoritmos evaluados debido a que para utilizar ANOVA se deben cumplir las asunciones, las cuales casi siempre son violadas en el aprendizaje automático.

- Se utiliza el análisis *post-hoc* de Nemenyi, con el fin de identificar si existen diferencias significativas entre los algoritmos evaluados, debido a que muestra una representación gráfica, lo que facilita el análisis y comprensión de los resultados.

Capítulo 2

Propuesta de Solución

2.1. Introducción

En el siguiente capítulo se explicará en detalle los diferentes algoritmos de predicción con salidas compuestas que explotan la interdependencia entre las variables de salida, así como la implementación del algoritmo *Curds and Whey* y su variante de validación cruzada para el tratamiento de la interdependencia mediante la estadística multivariada. Además se ejemplifican los patrones de diseño utilizados en la implementación del *Curds and Whey*.

2.2. Descripción e Implementación

2.2.1. *Multi Target Stacking (MTS)*

En [8] se adapta el método de *stacking* para regresión con salidas compuestas, el cual se denota como *Multi-Target Stacking* (MTS). El entrenamiento de MTS consta de dos etapas. En la primera etapa, *m* modelos independientes de salidas simples $h_j : X \rightarrow R$ son aprendidas como un *Single Transformation* (ST). Sin embargo, en lugar de usar directamente estos modelos para predicción, MTS tiene una segunda etapa de entrenamiento adicional donde un segundo conjunto de *m* modelos objetivos $h_j^* : X \times R^{m-1} \rightarrow R$ son aprendidos, uno para cada salida Y_j.

En MTS cada modelo objetivo h_j^* está aprendiendo en un conjunto de entrenamiento transformado $D_j^* =$

$(x^{*1}, y_j^1), \ldots, (x^{*n}, y_j^n)$ donde $x_j^{*i} = [x_1^i, \ldots, x_n^i, \hat{y}_1^i, \ldots, \hat{y}_{j-1}^i, \hat{y}_{j+1}^i, \ldots, \hat{y}_m^i]$ son expandidos en vectores de entradas consistente del vector de entrada original de los ejemplos de entrenamiento argumentado por *m*-1 predicciones de las restantes variables de salidas obtenidas por las primeras etapas de los modelos. Intencionalmente se diferencia levemente este método del multi-etiqueta por el punto de no incluir las predicciones de la primera etapa del modelo para la variable de salida Y_j en el espacio de entrada de la segunda etapa del modelo para estas variables, desde este se debería añadir información redundante para la segunda etapa de los modelos. Obtener los predictores para una instancia desconocida x^q, las primeras etapas de los modelos son primero aplicados y luego se obtiene un vector de salida, $\hat{y}^q = [\hat{y}_1^q, \ldots, \hat{y}^m] = [h_1(x^q), \ldots, h_m(x^q)]$. Posteriormente en la segunda etapa de los modelos se aplica una transformación en el vector de salida,
$x_j^{*q} = [x_1^q, \ldots, x_n^q, \hat{y}_1^q, \ldots, \hat{y}_j^q, \ldots, \hat{y}_{j-1}^q, \hat{y}_{j+1}^q, \ldots, \hat{y}_m^q]$ para producir el de salida final,
$\hat{y}^q = [h_1^*(x^{*q}, \ldots, h_m^*(x^{*q})]$.

Algoritmo 1: Entrenamiento MTSC

Data: Conjunto de entrenamiento D, número de *cross-validation folds* interna f

Result: Primero y Segundo estado de modelos h_j y $h_j^*, j = 1..m$

Construir primer estado de modelos

for $j = 1, ..., m$ **do**

    $D_j = (x^1, y_j^1), ..., (x^n, y_j^n)$ // ejecutar la transformación ST a D

    $h_j : D_j \rightarrow R$ // construir un modelo con ST para Y_j usando todo D_j

    $D_j^* \leftarrow \emptyset$ // se inicializa el segundo estado del conjunto de entrenamiento para la salida Y_j

    Dividir D_j en f partes $D_j^i, i = 1..f$

    for $i = 1, ..., f$ **do**

        $h_j^i : D_j \; D_j^i \rightarrow R$ // construir un modelo con ST para Y_j usando $D_j \; D_j^i$

Generando Segundo estado de los conjuntos de entrenamientos

for $i = 1, ..., f$ **do**

    $D_j^{*i} \leftarrow \emptyset$ // Inicializar segundo estado del conjunto de entrenamiento para la salida Y_j y división i

    for $j = 1, ..., m$ **do**

        for $x_j^i \epsilon D_j^i$ **do**

            $x{*}i_j = x_j^i$

            for $l = 1, ..., l, l \neq j$ **do**

                $\hat{y}_l^i = h_l^i(x_j^i)$

                $x{*}i_j = [x^{*i}, \hat{y}_l^i]$ // añadir $x{*}i_j$ con la estimación de $\hat{y}_l^i$

                $D_j^{*i} = D_j^{*i} \cup x{*}i_j$

    $D_j^* = D_j^* \cup D_j^{*i}$

Construir segundo estado de modelos

for $j = 1, ..., m$ **do**

    $h_j^* : D_j^* \rightarrow R$

Algoritmo 2: Predicción MTS

Data: Instancia desconocida x^q, primero y segundo estado de modelos h_j y $h_j^*, j = 1..m$

Result: Vactor de Salida $\hat{\hat{y}}^q$ del segundo estado de modelos

$\hat{y}^q = \hat{\hat{y}}^q = 0$ // Inicializar el primero y segundo vector de salida ;

Aplicar primer estado de modelos ;

for $j = 1, ..., m$ **do**

  $\hat{y}^q = h_j(x^q)$

$x^{*q} = [x^q, \hat{y}^q]$ // Concatenar x^q y el vector de salida $\hat{y}^q$;

Aplicar segundo estado de modelos ;

for $j = 1, ..., m$ **do**

  $\hat{\hat{y}}^q = h_j^*(x^{*q})$

2.2.2. *Regressor Chains (RC)*

El método de *Regressor Chains* (RC) se basa en la idea de modelos de encadenamiento de salidas simples. El entrenamiento del RC consiste en seleccionar una cadena aleatoria (permutación) del conjunto de variables de salida y luego construir un modelo de regresión separado para cada salida. Asumiendo que la cadena por defecto $C = Y_1, Y_2, \ldots, Y_m$ (donde C es representado como un conjunto ordenado) es seleccionada, el primer modelo se refiere a la predicción de Y_1, tiene la forma $h_1 : X \rightarrow R$ y es igual al modelo construido por el método *Single Transformation* (ST) para esta salida. La diferencia en RC es que los modelos subsiguientes $h_j, j > 1$ son entrenados en el conjunto de datos transformados $D_j^* = (x_j^{*1}, y_j^1), \ldots, (x_j^{*n}, y_j^n)$ donde $x_j^{*i} = [x_1^i, \ldots, x_1^i, y_1^i, \ldots, y_{j-1}^i]$ son vectores de entradas transformadas que consisten en los vectores de entrada originales de los ejemplos de entrenamiento aumentados por los valores reales de todos los objetivos anteriores en la cadena. De esta manera, los modelos construidos para salidas $Y_{j,j>1}$, tiene la forma $h_j : X \times R^{j-1} \rightarrow R$. Dado como un encadenamiento de los modelos, el vector de salida $\hat{y}^q$ de una instancia desconocida x^q es obtenido por la aplicación secuencialmente de los modelos h_j así $\hat{y}^q = [h_1(x^q), h_2(x_2^{*q}), \ldots, h_m(x_m^{*q})]$ donde $x_{j,j>1}^{*q} = [x_1^q, \ldots, x_d^q, \hat{y}_1^q, \ldots, \hat{y}_{y-1}^q]$. Los valores verdaderos $y_1^q, \ldots, y_{j-1}^q$ de las variables de salida no están

disponible en tiempo de predicción, el método se basa en estimaciones de estos valores obtenidos mediante la aplicación de los modelos $h_1, \ldots, h_{j-1}$.

Algoritmo 3: Entrenamiento RCC

Data: Conjunto de entrenamiento D,número de *cross-validation folds* interna f

Result: Modelos encadenados $h_j, j = 1..m$

Generar D_1^*

$D_1^* = (x^1, y_1^1), ..., (x^n, y_1^n)$ //Ejecutar la transformación ST de D para crear D_1^*

for $j = 1, ..., m$ **do**

  $h_j : D_j^* \rightarrow R$ // Construir un modelo encadenado para Y_j usando todo D_j^*

  if $j < m$ **then**

    Generar D_{j+1}^*

    $D_{j+1}^* \leftarrow \emptyset$ // Inicializar D_{j+1}^*

    Dividir D_j^* *en* f *partes* $D_j^{*i}, i = 1..f$

    for $i = 1, ..., f$ **do**

      $h_j^i : D_j^*\ D_j^{*i} \rightarrow R$ // Construir uno modelo encadenado Y_j usando $D_j^*\ D_j^{*i}$

      for $x_j^{*i} \in D_j^{*i}$ **do**

        $x_{j+1}^{*i} = x_j^{*i}$ // Inicializar x_{j+1}^{*i}

        $\hat{y}_j^i = h_j^i(x_j^{*i})$

        $x_{j+1}^{*i} = [x_j^{*i}, \hat{y}_j^i]$ // Añadir x_{j+1}^{*i} con la estimación de $\hat{y}_j^i$

        $D_{j+1}^* = D_{j+1}^* \cup (x_{j+1}^{*i}, y_{j+1}^i)$

Algoritmo 4: Predicción RC

Data: Instancia desconocida x^q, modelos encadenados $h_j, j = 1..m$

Result: Vector de salida $\hat{y}^q$

$\hat{y}^q = 0$ // Inicializar el vector de salida

$x_1^{*q} = x^q$

for $j = 1, ..., m$ **do**

  $\hat{y}_j^q = h_j(x_j^{*q})$

  $x_{j+1}^{*q} = [x_j^{*q}, \hat{y}_j^q]$ // Concatenar x_j^{*q} y $\hat{y}_j^q$

2.2.3. *Curds and Whey (CW)*

En presencia de un conjunto de datos limitados de entrenamiento para la predicción de múltiples variables de salidas, el empleo de las coordenadas canónicas resulta efectivo debido a que contrae el tamaño del espacio multivariado representado.

Breiman y Friedman en su trabajo [32] proponen la contracción simultánea tanto en el espacio de entrada como en el de salida.

En general para q variables de salida $y = (y_1, \ldots, y_q)$ con valores de regresión de mínimos cuadrados separados $\hat{y} = (\hat{y}_1, \ldots, \hat{y}_q)$ existe la posibilidad de que si las variables de salida están correlacionadas, se puede obtener un predictor más exacto $\tilde{y}_i$ para cada y_i usando una combinación lineal:

$$\tilde{y}_i = \bar{y}_i + \sum_{k=1}^{q} b_{ik}(\hat{y}_k - \bar{y}_k), i = 1, \ldots, q \tag{2.1}$$

donde:

$$\hat{y}_i = \bar{y}_i + \sum_{j=1}^{p} \hat{a}_{ij}(x_j - \bar{x}_j) + \lambda \sum_{j=1}^{p} \theta_j^2, \tag{2.2}$$

donde:

$$\lambda \sum_{j=1}^{p} \theta_j^2$$

es el factor de regularización de la regresión y

$$\{\hat{a}_{ij}\}_{j=1}^{p} = \arg\min_{\{a_j\}_1^p} \left(\sum_{n=1}^{N} \left[y_{ni} - \bar{y}_i - \sum_{j=1}^{p} a_j (x_{nj} - \bar{x}_j) \right]^2 + \lambda \sum_{j=1}^{p} \theta_j^2 \right), \tag{2.3}$$

Se asume además que las variables predictoras y las variables de salida están centradas por la media correspondiente del conjunto de entrenamiento $\{y_i \leftarrow y_i - \bar{y}_i\}_1^q$, $\{x_j \leftarrow x_j - \bar{x}_j\}_1^p$. Así como todas las salidas estimadas son centradas por la media correspondiente de la muestra $\{\hat{y}_i \leftarrow \hat{y}_i - \bar{y}_i\}_1^q$, $\{\tilde{y}_i \leftarrow \tilde{y}_i - \bar{y}_i\}_1^q$.

El método *Curds and Whey* es el encargado de encontrar el conjunto óptimo de valores para los coeficientes $\{b_{ik}\}$. Partiendo de una notación matricial y vectorial para las respectivas medidas(centradas):

$$\tilde{y} = \{\tilde{y}\}_1^q, \hat{y} = \{\hat{y}\}_1^q, \mathbf{B} = [b_{ik}] \in R^{q \times q} \tag{2.4}$$

la ecuación (2.1) se expresa como:

$$\tilde{y} = \mathbf{B}\hat{y}, \tag{2.5}$$

y la estimación de $\mathbf{B}$ toma la forma $\mathbf{B} = \mathbf{T}^{-1}\mathbf{D}\mathbf{T}$ donde $\mathbf{T}$ es la matriz de coordenadas canónicas y $\mathbf{D} = diag(d_1, \ldots, d_q)$ es una matriz diagonal.

Para una variable de salida y, la variable de salida estimada $\tilde{y}$ se expresa como

$$\tilde{y} = b\hat{y} = \sum_{j=1}^{p} (b\hat{a}_j) x_j, \tag{2.6}$$

donde $\hat{y}$ se calcula mediante (2.2).

Luego de un conjunto de procedimientos algebráicos descritos en [32], para estimar la matriz de contracción óptima $\mathbf{B}$ se hace una descomposición en auto valores y autovectores de la matriz $\mathbf{Q}$:

$$\mathbf{Q} = (YY^T)^{-1} Y^T X (XX^T + \lambda I_p)^{-1} X^T Y, \tag{2.7}$$

donde Y es la $n \times q$ matriz de variables de salida, X es la $n \times p$ matriz de variables predictoras, λ es el coeficiente de regularización y I_p es la $n \times p$ matriz de identidad. Posteriormente

$$\mathbf{B} = V^{-1} D^* V, \tag{2.8}$$

donde $D^* = diag(d_1, \ldots, d_q)$ es la matriz de contracción óptima (coeficientes de correlación canónica) de autovalores de la matriz Q y V es la matriz de autovectores (coordenadas canónicas) de la matriz Q . En el caso más simple dado: $r = p/N$ con N igual al número total de observaciones y p el número de variables predictoras, define el d_k^* como

$$d_k^* = \frac{c_k^2}{c_k^2 + r(1 - c_k^2)}, k = 1, 2, \ldots, q \tag{2.9}$$

siendo c_k^2 el ***i-ésimo*** coeficiente de correlación canónica. Este enfoque ofrece mejores predicciones en comparación con mínimos cuadrados ordinarios, pero no proporciona suficiente contracción para ser óptima.

Un segundo enfoque, basado en la validación cruzada generalizada establece el d_k^* de la forma siguiente:

$$d_k^* = \frac{(1 - r)(c_k^2 - r)}{(1 - r)^2 \times c_k^2 + r^2 \times (1 - c_k^2)}, k = 1, 2, \ldots, q \tag{2.10}$$

Sujeto a $d_k^* \leftarrow \max(d_k^*, 0)$, donde D^* se obtiene a partir de (2.10) usando los coeficientes de correlación canónica correspondiente c_k^2 y los grados de libertad efectivos

$$r = trace(X^T(XX^T + \lambda I_p)^{-1}X). \tag{2.11}$$

Algoritmo 5: Entrenamiento *Curds and Whey* (CW)

Data: Factor de regularización de la regresión λ, matriz de variables predictoras X del conjunto de entrenamiento, matriz de variables de salida Y del conjunto de entrenamiento, matriz identidad I_p.

Result: Matriz de contracción B

1 *Centrar la matriz X y la matriz Y por la media correspondiente del conjunto de entrenamiento* $\{y_i \leftarrow y_i - \bar{y}_i\}_1^q$, $\{x_j \leftarrow x_j - \bar{x}_j\}_1^p$.

2 *Determinar la matriz* $Q = (YY^T)^{-1}Y^TX(XX^T + \lambda I_p)^{-1}X^TY$.

3 *Descomponer en autovectores V y autovalores* D^* *la matriz Q.*

4 *Calcular* D^* *, mediante* $d_k^* = \frac{c_k^2}{c_k^2 + r(1-c_k^2)}, i = 1, 2, \ldots, q$*, usando los coeficientes de correlación canónica correspondiente* c_k^2 *para* $k = 1, 2, \ldots, q$ *y* $r = p/N$.

5 *Calcular la matriz de contracción óptima* $B = V^{-1}D^*V$.

Algoritmo 6: Entrenamiento *Curds and Whey* con validación cruzada (CWVC)

Data: Factor de regularización de la regresión λ, matriz de variables predictoras X del conjunto de entrenamiento, matriz de variables de salida Y del conjunto de entrenamiento, matriz identidad I_p.

Result: Matriz de contracción óptima B

1 *Centrar la matriz X y la matriz Y por la media correspondiente del conjunto de entrenamiento* $\{y_i \leftarrow y_i - \bar{y}_i\}_1^q$, $\{x_j \leftarrow x_j - \bar{x}_j\}_1^p$.

2 *Determinar la matriz* $Q = (YY^T)^{-1}Y^TX(XX^T + \lambda I_p)^{-1}X^TY$.

3 *Descomponer en autovectores V y autovalores* D^* *la matriz Q.*

4 *Calcular* D^* *, mediante* $d_k^* = \frac{(1-r)(c_k^2 - r)}{(1-r)^2 \times c_k^2 + r^2 \times (1-c_k^2)}$*, usando los coeficientes de correlación canónica correspondiente* c_k^2 *para* $k = 1, 2, \ldots, q$ *y* $r = trace(X^T(XX^T + \lambda I_c)^{-1}X)$.

5 *Calcular la matriz de contracción óptima* $B = V^{-1}D^*V$.

Algoritmo 7: Predicción *Curds and Whey (CW)*

Data: Instancia desconocida $x^{1\times p}$, matriz de coeficientes $\hat{Y}$, matriz de contracción óptima B, vector de medias $\bar{y}^{1\times q}$ de las variables de salida del conjunto de entrenamiento.

Result: $\tilde{y}$ predicha.

1 *Centrar* $x^{1\times p}$ *por la media correspondiente del conjunto de entrenamiento* $\{x_j \leftarrow x_j - \bar{x}_j\}_1^p$.

2 *Calcular* $\tilde{y} = x^{1\times p}\hat{Y}B + \bar{y}^{1\times q}$

2.3. Integración del *Curds and Whey* en Mulan

Mulan es una librería de código abierto escrita en Java para el aprendizaje automático, incluye una variedad de algoritmos que solucionan tareas de aprendizaje multiqueta como la clasificación, el ranking o la combinación de estas dos. Además permite la selección de características y la evaluación de un grupo de métricas a través de la validación cruzada y evaluación *hold out*.

Para la creación de un clasificador se debe implementar la interfaz *MultiLabelLearner*, la cual define operaciones básicas como la construcción de un modelo de aprendizaje a partir de *MultiLabelInstances*, y ejecutar la predicción a partir de una base de datos de pruebas. Cada clasificador debe implementar dos métodos básicos: *buildInternal* y *makePredictionInternal*.

El método *buildInternal* se emplea para la creación un modelo de aprendizaje que utiliza una base de datos de entrenamiento (*MultiLabelInstances*) y es llamado desde el método *build(MultiLabelInstances)* contenido en la clase *MultiLabelLearnerBase*.

El método *makePredictionInternal* se utiliza para, dada una instancia de una base de datos de prueba, predecir un resultado empleando el modelo que fue implementado en el método *buildInternal*. La llamada al *makePredictionInternal* se produce desde el método *makePrediction(Instance)* contenido en la clase *MultiLabelLearnerBase* y retorna el resultado de la predicción. La predicción retornada es un objeto del tipo *MultiLabelOutput*, definido en la clase *MultiLabelOutput*, a partir de la cual se puede determinar la capacidad del clasificador, mediante una métrica determinada, al ejecutar la predicción.

Para la implementación de la solución propuesta se incluye dentro del paquete *mulan.regressor.transformation*

la clase *CW* que extiende de la clase abstracta *TransformationBasedMultiTargetRegressor* quien a su vez extiende de la clase *MultiLabelLearnerBase*, la cual implementa la interfaz *MultiLabelLearner*.

Además se incluyen las clases *Utiles*, que contiene los métodos necesario para el trabajo con las bases de datos y las matrices, y *Curds and Whey*, que contiene la implementación de las variantes del algoritmo como se puede apreciar en la figura 2.1.

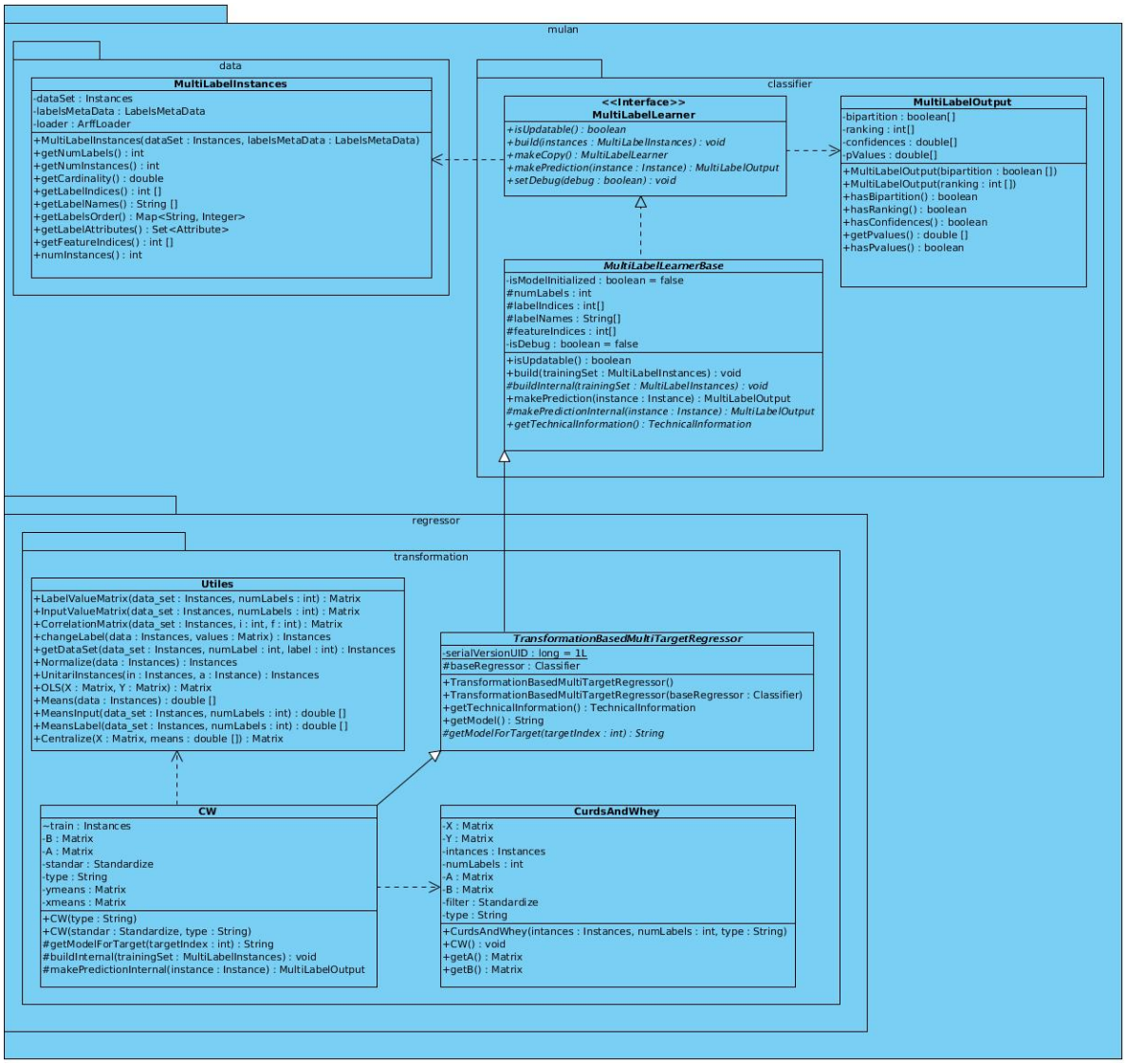

Figura 2.1: Diagrama de clases.

2.4. Patrones de Diseño

Para la integración del algoritmo *Curds and Whey* y sus variantes a la arquitectura de la herramienta Mulan, se utilizaron diferentes patrones de diseño para describir y organizar las clases y objetos. Según [64] un patrón de diseño es una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular. Estos patrones identifican: Clases, Instancias, Roles, Colaboraciones y la distribución de responsabilidades, además es una técnica muy eficaz para flexibilizar el código.

2.4.1. Patrones para Asignar Responsabilidades (GRASP)

Los patrones GRASP *General Responsibility Assignment Software Patterns* describen los principios fundamentales de la asignación de responsabilidades a objetos en una aplicación. Se considera que más que patrones, son una serie de buenas prácticas recomendadas en el diseño de software [65].

Los patrones GRASP utilizados para la implementación del algoritmo *Curds and Whey* y su variante para validación cruzada son los siguientes:

Creador

Una clase B crea una instancia de otra clase A si: B agrega los objetos A, B contiene los objetos A, B registra las instancias de los objetos A, B utiliza específicamente los objetos A o B tiene los datos de inicialización que serán transmitidos a A cuando este objeto sea creado.

El uso de este patrón se ve evidenciado en la clase CW cuando al crear una instancia de la clase *CurdsAndWhey*. Su utilización conduce a un bajo acoplamiento, lo que implica menos dependencias respecto al mantenimiento y mejores oportunidades de reutilización. A continuación se muestra un ejemplo de este patrón:

```
CurdsAndWhey curds = new CurdsAndWhey(train, numLabels,type);
```

Figura 2.2: Ejemplo de uso de patrón Creador.

Bajo acoplamiento

Se aplica en las clases CW, Utiles y *CurdsAndWhey* con el objetivo de que estén lo menos relacionadas entre sí. Y de tal manera de que al producirse algún cambio en alguna de ellas no repercuta grandemente en las otras, y así potenciar la reutilización.

Alta cohesión

Se utiliza para las clases CW, Utiles y *CurdsAndWhey*, de forma tal que cada una tenga asignada una responsabilidad específica sin sobrepasar los límites manejables. Con la utilización de este se facilita el análisis del diseño y simplifica el mantenimiento.

Experto

Se selecciona para ser aplicado en las clases CW, Utiles y *CurdsAndWhey* de manera que contengan la información necesaria para ejecutar una acción específica, siendo cada clase la más indicada para realizar una tarea. Con su utilización se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide, propiciando un bajo acoplamiento. Además su uso proporciona clases más sencillas y cohesivas, fáciles de comprender y mantener.

Un ejemplo donde se evidencia Bajo acoplamiento, Alta cohesión y Experto es las clases CW y *CurdsAndWhey*. En la imagen se puede observar como entre esta solo existe una relación de uso y que cada clase tiene sus funcionalidades específicas par realizar sus tareas por lo que además se evidencia el patrón Experto debido a que ambas clases son expertas en lo que hacen.

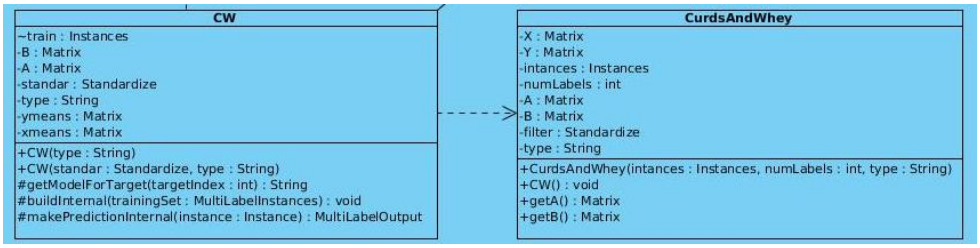

Figura 2.3: Ejemplo de uso de los patrones Creador, Bajo acoplamiento, Alta cohesión y Experto.

2.5. Conclusiones del capítulo

En este capítulo se definieron los algoritmos ERCC y MTSC, que explotan la interdependencia de las variables de salida, poropuestos en [8] debido a que obtienen los mejores resultados predictivos entre los algoritmos del estado del arte . Además se describió la fundamentación teórica e implementación del algoritmo de estadística multivariada que explota la interdependencia del espacio de entrada y de salida mediante la correlación canónica *Curds and Whey* y su variante de validación cruzada, para problemas de predicción con salidas compuestas. Para la implementación en la herramienta Mulan del algoritmo *Curds and Whey* se emplearon los patrones de diseño Creador, Bajo acoplamiento, Alta cohesión y Experto.

Capítulo 3

Experimento y Resultados.

En este capítulo se explica el experimento realizado. Primeramente se describe la medida de evaluación y el proceso seguido para la comparación estadística de los algoritmos. Además se hace una descripción de las base de datos utilizadas para los experimentos y se muestra un ejemplo de un caso de estudio empleando las variantes del algoritmo *Curds and Whey* con la base de datos *The river flow 2*.

3.1. Evaluación

Como medida de evaluación se empleó el promedio del error cuadrático medio (aRRMSE) entre lo predicho por el algoritmo y lo medido en cada base de dato para las variables de salidas. La misma se determina de la siguiente manera:

$$aRRMSE(h; D_{test}) = \frac{1}{q}\sum_{j=1}^{q}\sqrt{\frac{\sum_{(x,y)\in D_{test}} \left(h\left(x_j\right) - y_j\right)^2}{\sum_{(x,y)\in D_{test}} \left(\bar{y}_j - y_j\right)^2}}$$

3.2. Base de Datos

En la tabla (3.1) se describen las 12 bases de datos utilizadas en el experimento, donde la primera columna indica el nombre de la base de datos, la segunda se refiere al número de observaciones de la base de datos. En

esta columna además se indica si los datos están particionados en datos de prueba *test* y datos de entrenamiento *train* o si la base de datos contiene todos los datos. En el experimento es necesario aplicar una validación cruzada para seleccionar los datos de entrenamiento y los de prueba, se conoce como *K-Fold Cross Validation* (se señala como cv). La tercera y cuarta columna indica la cantidad de atributo en el espacio de entrada y la cantidad en el espacio de salida. La última columna da una idea del dominio de aplicación en el cual fueron colectados los datos.

Datos	Instancias	# Atributos	# Targets	Dominio
EDM	154/cv	16	2	Generación de electricidad
Solar Flare1	323/cv	10	3	Veces que se produce destellos repentinos
Solar Flare2	1066/cv	10	3	en la superficie del sol durante 24 horas
OES97	343/cv	263	16	Estima el numero de empleados a tiempo
OES10	403/cv	298	16	completo para una ciudad
Atp1d	201/136	411	6	Precio de los boletos de avión para una
Atp7d	188/108	411	6	Aerolínea para intervalos de tiempo
RF1	4108/5017	64	8	Predice el comportamiento de las redes
RF2	4108/5017	576	8	fluviales de ríos en 48 horas
SCM1d	8145/1658	280	16	Predice los precios de los productos futuros
SCM20d	7463/1503	61	16	en una competencia
Water Quality	1060/cv	16	14	Abundancia de 14 plantas y especies en ríos de Eslovenia

Tabla 3.1: Datos generales de las bases de datos empleadas para evaluar los algoritmos.

EDM (*Electrical Discharge Machining*) es una base de datos que representa un problema de regresión con dos variables de salida y 16 variables de entrada continuas. Cada variable de salida puede tomar 3 valores distintos (-1,0,1). La tarea es acortar el tiempo de mecanizado mediante la reproducción del comportamiento de un operador humano que controla los valores de dos variables.

SF1, SF2 (*Solar Flare*) [66] es una base de datos que contiene 3 variables de salida correspondientes a 3 tipos de llamas solares (común, moderada y severa) que son observadas durante 24 horas. Hay dos versiones de esta base de datos, SF1 contiene los datos del año 1969 y SF2 del año 1978.

WQ (*Water Quality*) [6] es una base de datos que contiene 14 variables de salida que representan las especies de plantas y animales en los ríos de Eslovenia y 16 variables de entrada que se refieren a los parámetros físicos y químicos que miden la calidad del agua.

OES97, OES10 (*Occupational Employment Survey*) son bases de datos obtenida en los años 1997 (OES97) y 2010(OES10) del *Occupatinal Employement Survey* compilada por *US Bareu of Labor Statistics*. Cada instancia (ciudad) provee un estimado del tiempo completo equivalente a la cantidad de empleados entre un número determinado de empleos para un área metropolitana específica. Existen 334 y 403 ciudades (instancias) en la base de datos de 1997 y mayo de 2010 respectivamente. Las variables de entrada en estas bases de datos son una secuencia aleatoria de un conjunto de empleos (ejemplo: doctor, dentista, mecánico) observados en al menos el 50 % de las ciudades(algunas de las categorías no tiene valor para determinadas ciudades). Las variables de salida en los dos años son aleatorias, seleccionadas de todo el conjunto de categorías alrededor del 50 % del umbral. Los valores perdidos tanto en las variables de entrada como en las de salida son reemplazados por la media de la muestra.

SCM1d, SCM20d (*The Supply Chain Management*) los conjuntos de datos de gestión de la cadena de suministro se derivan de la Competencia Agente Operador en la cadena de suministro (SCM TAC) del torneo a partir de 2010. Los métodos precisos para pre-procesamiento de datos y la normalización se describen en detalle en [67]. Algunos valores de referencia para la precisión de la predicción en este dominio están disponibles en la Predicción Desafío TAC SMC [68]. Estos datos corresponden sólo al "Producto Futuro"tipo de predicción. Cada fila corresponde a un día de observación en el torneo (hay 220 días en cada juego y 18 partidos del torneo en un torneo). Las variables de entrada en este ámbito son los valores de un día de torneo específico. Además, se incluyen 4 observaciones temporizadas para cada producto observado y el componente (1,2,4 y 8 días de retardado) para facilitar cierta anticipación de las tendencias en el futuro. Los conjuntos de datos contienen 16 objetivos, cada objetivo se corresponde con el valor medio del día siguiente (SCM1d) o el valor por 20 días en el futuro (SCM20d). Los días sin valores objetivos se excluyen de los conjuntos de datos

(es decir, los días con etiquetas que están más allá del final del juego están excluidos).

RF1, RF2 (*The river flow*) los conjuntos de datos de flujo de los ríos se refieren a la predicción de la red fluvial fluye durante 48 horas en el futuro en lugares específicos. El conjunto de datos contiene los datos de las observaciones de flujo por hora de los 8 sitios en la red del río Mississippi en los Estados Unidos y se obtuvieron del Servicio Meteorológico Nacional de Estados Unidos. Cada fila incluye la observación más reciente para cada uno de los 8 sitios, así como observaciones en tiempo lag de 6, 12, 18, 24, 36, 48 y 60 horas en el pasado. En RF1, cada sitio contribuye 8 variables de atributos para facilitar la predicción. Hay un total de 64 variables más 8 variables objetivo. El conjunto de datos RF2 extiende los datos RF1 mediante la adición de información de pronóstico de precipitación para cada uno de los 8 sitios (lluvia esperada reportado como valores discretos: 0,0, 0,01, 0,25, 1,0 pulgadas). Para cada sitio de observación y de calibre, el pronóstico de precipitación para las ventanas de 6 horas hasta 48 horas en el futuro se añade (6, 12, 18, 24, 30, 36, 42, y 48 horas). Los dos conjuntos de datos tanto contienen más de 1 año de observaciones horarias (>9000 horas) recogidos a partir de septiembre de 2011 hasta septiembre de 2012. El período de formación es el primer 40 % de las observaciones, y el período de prueba es el resto. El dominio es un candidato natural para la regresión multi-objetivo porque hay relaciones físicas claras entre las lecturas en la red fluvial contiguo.

ATP1d, ATP7d (*The Airline Ticket Price*) los conjunto de datos se refiere a la predicción de los precios de los billetes de avión. Las filas son una secuencia de observaciones en tiempo-ordenado lo largo de varios días. Cada muestra en esta base de datos representa un conjunto de observaciones de una fecha de observación y salida par específico. Las variables de entrada para cada muestra son valores que pueden ser útiles para la predicción de los precios de los billetes de avión para una fecha de salida específica. Las variables objetivo en estos conjuntos de datos están al día siguiente (ATP1d) precio o el precio mínimo observado en los próximos 7 días (ATP7d) para las preferencias de vuelo 6 de destino (cualquier línea aérea con cualquier número de paradas, cualquier línea aérea sin parar solamente, Delta Airlines, Continental Airlines, Airtrain Airlines y United Airlines). Las variables de entrada incluyen los siguientes tipos: el número de días entre la fecha de la observación y la fecha de salida (1 función), las variables booleanas para el día-de-la-semana de la fecha de observación (7 funciones), la enumeración completa de la siguiendo 4 valores: 1) el precio mínimo, el precio, y el número de citas de 2 media) todas las compañías aéreas y de cada aerolínea citando a más del 50 % de

los días de observación 3) para no dejar, de una sola parada, y de dos paradas vuelos, 4) para el día actual, el día anterior, y dos días anteriores. El resultado es un conjunto de características de 411 variables. Para obtener detalles específicos sobre cómo se construyen estos conjuntos de datos por favor consulte [69]. La naturaleza de estos conjuntos de datos es heterogénea con una mezcla de varios tipos de variables que incluyen variables booleanas, los precios, y los recuentos.

3.3. Resultados

En la tabla 3.2 se muestra el aRRMSE obtenido al evaluar los algoritmos: *Curds and Whey*(CW), *Curds and Whey* en su variante de validación cruzada(CWCV), *Multi-objetive Random Forest* (MORF), *Single Target*(ST), *Multi-Target Stacking* (MTS), *Multi-Target Stacking Corrected* (MTSC), *Ensemble of Regressor Chains* (ERC), *Ensemble of Regressor Chains Corrected* (ERCC) y *k-Nearest Neighbors for Structured Prediction*(KNN-SP) en las 12 bases de datos anteriormente descritas. Los aRRMSE obtenidos para cada variable de salidas se muestran de forma detallada en el anexo A.

Los algoritmos MTSC y ERCC, son ejecutados con $f = 10$ *folds* de validación cruzada interna y 10 modelos a ensamblar para el ERC y el ERCC. Además ese emplean 100 arboles para la configuración del MORF. Una explicación mas detalla de la configuración empleada para estos algoritmos se encuentra en [8].

El CW y CWCV emplean para determinar los $\hat{y}_i$ un valor de $\lambda = 100$ así como para la matriz de contracción óptima **B** un valor de $\lambda = 0{,}001$ ambos determinados a través del método de prueba y error.

De manera general para las bases de datos que se evalúan mediante el método de validación cruzada se usa una configuración de $f = 10$ *folds*.

Dataset	CW	CWCV	MORF	ST	MTS	MTSC	ERC	ERCC	KNNSP
WQ	95.87	95.18	**89.94**	90.83	91.10	90.95	90.97	90.59	93.50
EDM	84.82	84.99	**73.38**	74.21	74.30	73.96	74.35	74.07	85.70
SF1	**100.04**	104.53	128.25	113.54	112.70	106.80	105.01	108.87	107.50
SF2	116.85	118.11	142.48	114.94	**94.48**	105.53	105.32	108.79	136.50
OES97	46.18	**45.99**	54.90	52.48	52.59	52.43	52.54	52.39	60.10
OES10	36.36	**35.70**	45.18	42.00	42.01	42.05	42.02	41.99	51.80
SCM1d	**42.99**	43.00	56.63	47.75	47.41	47.01	47.09	46.63	60.50
SCM20d	**75.26**	75.73	77.75	77.68	78.62	78.54	77.55	75.97	91.70
ATP1d	42.01	41.17	42.22	37.35	37.16	37.17	**37.10**	37.24	62.40
ATP7d	56.76	56.25	55.08	52.48	51.43	**50.74**	53.43	51.24	89.50
RF1	91.48	**58.24**	85.13	69.63	82.37	69.82	79.47	69.89	102.40
RF2	53.93	**50.17**	91.89	69.64	81.75	69.86	79.61	69.82	83.60

Tabla 3.2: Resultados del aRRMSE obtenidos al evaluar las 12 bases de datos para los 9 algoritmos.

En el análisis de la tabla 3.2 se observa que tanto el algoritmo CW como su variante para validación cruzada CWCV obtienen los menores valores de aRRMSE en la predicción para las bases de datos: SCM1d, SCM20d, RF1, RF2, OES97, OES10 y SF1.

Es importante resaltar que en las 4 bases de datos de mayor dimensionalidad: SCM1d, SCM20d, RF1 y RF2 el rendimiento predictivo del CW y CWCV sobresale del resto.

Una vez obtenida la tabla 3.2 se procede a aplicar el test de Friedman para conocer si existen diferencias entre los algoritmos evaluados. Los resultados del test se exponen en la tabla 3.3.

Algoritmos	Ranking
ERCC	**3.41**
CWCV	3.83
MTSC	3.92
ERC	4.50
CW	4.50
ST	4.58
MTS	5.25
MORF	6.67
KNNSP	8.33

Tabla 3.3: Estadístico de Friedman (distribuidos según chi-cuadrado con 8 grados de libertad: 31.46. P-valor calculado por el test de Friedman: 1.1601337337818762E-4.

Al aplicarse el test de Friedman para los 9 algoritmos, los valores obtenidos difieren por tanto se rechaza la hipótesis nula y se procede a aplicar un test *post-hoc*. Se escoje el test de Nemenyi debido que compara todos los algoritmos entre sí. Los resultados son expuestos en la figura 3.1 para $\alpha = 0{,}05$, $k = 9$ y $N = 12$ empleando la ecuación (1.12).

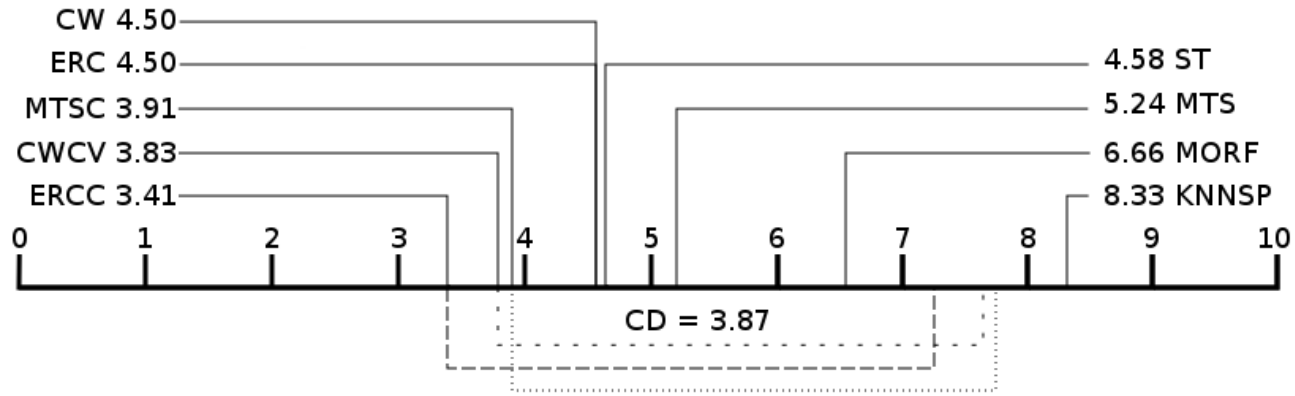

Figura 3.1: Test de Nemenyi para $\alpha = 0{,}05$, $k = 9$ y $N = 12$

En el análisis del test de Nemenyi se aprecia que para la distancia crítica obtenida, $CD = 3{,}87$, el algoritmo KNNSP presenta diferencias significativa con respecto a ERCC, CWCV y ERCC. Además entre los restantes algoritmos no se aprecian estadísticamente diferencias significativas en cuanto a la precisión de la predicción.

3.4. Implementación de Caso de Estudio para la variante de validación cruzada del *Curds and Whey*.

Para el desarrollo del caso de estudio se emplea como base de datos *The river flow 2*, una base de datos de ata dimensionalidad, la cual tiene 4108 observaciones para entrenamiento, 5017 observaciones para prueba, 576 atributos y 8 salidas.

Primeramente se centralizan todo el conjunto de datos, se centra la matriz $X^{n\times p}$, de variables predictoras, y la matriz $Y^{n\times q}$, de variables de salida, por la media correspondiente del conjunto de entrenamiento $\{y_i \leftarrow y_i - \bar{y}_i\}_1^q$, $\{x_j \leftarrow x_j - \bar{x}_j\}_1^p$.

```
Utiles utiles = new Utiles();
double[] xmeans = utiles.MeansInput(intances, numLabels);
X = utiles.Centralize(utiles.InputValueMatrix(intances, numLabels),
xmeans);
double[] ymeans = utiles.MeansLabel(intances, numLabels));
Y = utiles.Centralize(utiles.LabelValueMatrix(intances, numLabels),
ymeans);
```

Se calcula el *OLS* con regularización, *ridge regression* con un valor de $\lambda = 100$, de las variables de salida Y, mediante las variables predictoras X.

$$X = \begin{bmatrix} x_{11}.....x_{1n} \\ x_{21}.....x_{2n} \\ . \\ . \\ x_{n1}.....x_{nn} \end{bmatrix} Y = \begin{bmatrix} y_{11}.....y_{1n} \\ y_{21}.....y_{2n} \\ . \\ . \\ y_{n1}.....y_{nn} \end{bmatrix}$$

```
A= utiles.OLS(X, Y);
```

Luego, se determina la matriz de correlación canónica *Q*, mediante la ecuación (2.7), y se descompone en las matrices de autovalores, *D*, y autovectores, *V*.

```
Matrix XT = X.transpose().times(X);
Matrix I = new Matrix(XT.getRowDimension(), XT.getColumnDimension());
for (int i = 0; i < I.getRowDimension(); i++) {
 for (int j = 0; j < I.getColumnDimension(); j++) {
         if (i == j) {
                 I.set(i, j, 1);
 }}}
I = I.times(0.001);
Matrix OLS_Y = XT.plus(I).inverse().times(X.transpose()).times(Y);
Matrix OLS_X = Y.transpose().times(Y).inverse().times(Y.transpose())
.times(X);
Matrix Q = OLS_X.times(OLS_Y);
EigenvalueDecomposition eig = Q.eig();
Matrix D = eig.getD();
Matrix V = eig.getV();
```

Luego, se determina el valor de r mediante la ecuación (2.11) y se procede al cálculo de los coeficientes

de la matriz D empleando la ecuación (2.10).

```
r = utiles.OLS(X, X).trace();
for (int i = 0; i < numLabels; i++) {
 D.set(i, i, (1-r)*(D.get(i, i)-r) /
  (Math.pow((1-r),2)*D.get(i, i)+ Math.pow(r, 2)*(1 - D.get(i, i))));
 if(D.get(i, i)<0){
        D.set(i, i, 0);
 }}
```

Se obtiene la matriz B, que son los coeficientes de contracción óptima para el modelo.

```
B = V.inverse().times(D.times(V));
```

Para ejecutar la predicción se obtiene el vector de variables predictoras de la instancia a predecir, se centraliza por la media correspondiente del conjunto de entrenamiento y finalmente se obtiene un vector con los resultados de la predicción.

```
Utiles utiles = new Utiles();
Instances test_instance = utiles.UnitariInstances(train, instance);
Matrix X = utiles.Centralize(utiles.InputValueMatrix
(test_instance, numLabels), xmeans.getRowPackedCopy());
predictions = X.times(A).times(B).plus(ymeans).getRowPackedCopy();
```

Se calcula el *aRRMSE* y se muestran los resultados.

Dataset	CW	CWCV	MORF	ST	MTS	MTSC	ERC	ERCC	KNNSP
aRRME	53.93	**50.11**	91.89	69.64	81.75	69.86	79.61	69.82	83.60
Targets									
CHSI2	**20.59**	21.22	59.04	49.30	53.06	47.56	51.06	48.63	64.29
NASI2	135.26	118.10	**100.55**	103.90	100.76	102.49	103.01	103.25	99.19
EADM7	30.60	**30.39**	56.91	52.42	50.32	50.35	50.08	51.88	71.62
SCLM7	49.46	**48.57**	95.89	67.41	62.50	66.12	62.81	66.38	77.78
CLKM7	29.70	**27.30**	83.62	74.09	75.01	73.57	74.00	74.02	85.75
VALI2	50.12	**38.54**	97.87	59.50	153.91	61.21	142.76	60.15	103.56
NAPM7	**52.17**	55.36	148.90	58.01	62.41	57.09	59.44	57.81	59.16
DLDI4	63.51	**61.42**	92.33	92.51	96.00	100.47	93.72	96.43	107.33

Tabla 3.4: Resultados de la ejecución del CWCV para $f = 10$ *folds* de validación cruzada sobre la base de datos RF2. Los mejores resultados son los que se resaltan.

3.5. Conclusiones del capítulo

Los algoritmos que tratan la interdependencia entre las variables predictoras y las variables de salida superan en el aRRSME a los que no la tratan. Además el empleo de la estadística multivariada para el tratamiento de la interdependencia supera en cuanto al aRRSME a los algoritmos del estado del arte en dominios de aplicación específico.

Aplicando la prueba de Friedman se obtuvo CWCV es superior al CW y se ubica en el segundo lugar del ranking entre los nueve algoritmos evaluados en cuanto al error predictivo. Así como el análisis *post-hoc* de Nemenyi demostró que no existen diferencias significativas entre los algoritmos propuestos de estadística multivariada CW y CWCV, y el mejor del estado del arte ERCC. También que los algoritmos CW y CWCV superan al resto en aRRSME en las bases de datos de alta dimensionalidad dígase RF1, RF2, SCM1d y

SCM20d además de OES97, OES10 y SF1.

Conclusiones

Al término de la presente investigación se arriba a las siguientes conclusiones:

- El estudio desarrollado sobre los problemas de predicción con salidas compuestas demostró que los algoritmos que tratan la interdependencia muestran mejoras con respecto a los que no la tratan en cuanto al error predictivo.
- El empleo de lo métodos de estadística multivariada para tratar la interdependencia mejora los resultados en cuanto al error predictivo para dominios de aplicación específicos como son: RF1, RF2, SCM1d y SCM20d, además de OES97, OES10 y SF1 con respecto a los algoritmos del estado del arte que tratan la interdependencia en problemas de predicción con salidas compuestas.
- Aplicando la prueba de Friedman se obtuvo CWCV es superior al CW y se ubica en el segundo lugar del ranking entre los nueve algoritmos evaluados en cuanto al error predictivo. Así como el análisis *post-hoc* de Nemenyi demostró que no existen diferencias significativas entre los algoritmos propuestos de estadística multivariada CW y CWCV, y el mejor del estado del arte ERCC. Por el contrario si existe diferencias significativas entre el CWCV y el KNN-SP.

Recomendaciones

Al concluir la presente investigación se proponen, a manera de recomendaciones, una serie de tareas para ampliar y dar continuidad al trabajo realizado:

- Estudiar el desempeño de algoritmos de estadística multivariada como el FES y el MRCE para problemas de predicción con salidas compuestas.
- Experimentar la combinación del algoritmo KNN con los diferentes algoritmos de estadística multivariada.

Referencias bibliográficas

[1] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[2] W. Ilg, G. Bakir, J. Mezger, M. Giese, S. Nowozin, G. BakIr, S. Nowozin, G. BakIr, K. Tsuda, S. Nowozin, *et al.*, "Predicting structured data," *Journal of Machine Learning Research*, vol. 1, pp. 996–1003, 2012.

[3] T. Li, C. Zhang, and S. Zhu, "Empirical studies on multi-label classification.," in *IcTAI*, vol. 6, pp. 86–92, 2006.

[4] D. Kuznar, M. Mozina, and I. Bratko, "Curve prediction with kernel regression," in *Proceedings of the 1st Workshop on Learning from Multi-Label Data*, pp. 61–68, 2009.

[5] D. Kocev, S. Džeroski, M. D. White, G. R. Newell, and P. Griffioen, "Using single-and multi-target regression trees and ensembles to model a compound index of vegetation condition," *Ecological Modelling*, vol. 220, no. 8, pp. 1159–1168, 2009.

[6] S. Džeroski, D. Demšar, and J. Grbović, "Predicting chemical parameters of river water quality from bioindicator data," *Applied Intelligence*, vol. 13, no. 1, pp. 7–17, 2000.

[7] S. Džeroski, A. Kobler, V. Gjorgjioski, and P. Panov, "Using decision trees to predict forest stand height and canopy cover from lansat and lidar data," *Managing environmental knowledge: EnviroInfo*, pp. 125–133, 2006.

[8] E. Spyromitros-Xioufis, G. Tsoumakas, W. Groves, and I. Vlahavas, "Multi-label classification methods for multi-target regression," *arXiv preprint arXiv:1211.6581*, 2012.

[9] H. Hotelling, "Relations between two sets of variates," *Biometrika*, pp. 321–377, 1936.

[10] R. Statistical Package, "R: A language and environment for statistical computing," *Vienna, Austria: R Foundation for Statistical Computing*, 2009.

[11] M. U. Guide, "The mathworks," *Inc., Natick, MA*, vol. 5, p. 333, 1998.

[12] I. H. Witten and E. Frank, "Weka," *Machine Learning Algorithms in Java*, pp. 265–320, 2000.

[13] J. Struyf, B. Zenko, H. Blockeel, C. Vens, and S. Dzeroski, "Clus: User's manual," 2010.

[14] J. Read, "Tutorial. meka 1.7. 3," 2014.

[15] D. Peña, *Análisis de datos multivariantes*, vol. 24. McGraw-Hill Madrid, 2002.

[16] J. Soley and E. Alfaro, "Aplicación de análisis multivariado al campo de anomalías de precipitación en centroamérica," *Tópicos Meteorológicos y Oceanográficos*, vol. 6, no. 2, pp. 71–93, 1999.

[17] I. Antúnez-Potashkina, R. Coro-Antich, and S. Rodriguez Ceballos, "Morfometría computerizada en aspirados celulares del carcinoma ductal infiltrante de la mama. modelo de estadística multivariada," *Rev Esp Patol*, vol. 30, no. 3, pp. 193–199, 1997.

[18] E. d. Uña Álvarez, "Estudio multivariado del micromodelado granítico: Interpretación comparada de la génesis y evolución de la gnammas en macizos antiguos," 1998.

[19] J. C. Rodríguez Campo, "Estudio comparativo de técnicas estadística multivariada versus las redes neuronales artificiales en el análisis de datos de calidad de agua," 2008.

[20] A. C. Rencher, *Methods of multivariate analysis*, vol. 492. John Wiley & Sons, 2003.

[21] M. Reiter, *Enhanced multiple output regression based on canonical correlation analysis with applications in computer vision*. na, 2010.

[22] T. Maldonado and E. Alfaro, "Predicción estacional para aso de eventos extremos y días con precipitación sobre las vertientes pacífico y caribe de américa central, utilizando análisis de correlación canónica," *InterSedes*, vol. 12, no. 24, 2012.

[23] J. A. Urrutia, E. A. C. Trejos, and R. Palomino, "Una aplicación de las correlaciones canónicas en climatología," *Scientia et Technica*, vol. 2, no. 48, pp. 263–268, 2011.

[24] Y. N. Díaz Valdés, M. Moreno Miravalles, J. Bacallao Gallestey, and J. E. Fernández-Britto Rodríguez, "Aterosclerosis coronaria y daño miocárdico. estudio de autopsias utilizando el sistema aterométrico," *Revista Habanera de Ciencias Médicas*, vol. 12, no. 1, pp. 57–65, 2013.

[25] B. Carpentieri, I. S. Duff, and L. Giraud, *Some sparse pattern selection strategies for robust Frobenius norm minimization preconditioners in electromagnetism*. Citeseer, 2000.

[26] J. D. Hamilton, *Time series analysis*, vol. 2. Princeton university press Princeton, 1994.

[27] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[28] A. J. Izenman, "Reduced-rank regression for the multivariate linear model," *Journal of multivariate analysis*, vol. 5, no. 2, pp. 248–264, 1975.

[29] M. Yuan, A. Ekici, Z. Lu, and R. Monteiro, "Dimension reduction and coefficient estimation in multivariate linear regression," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 3, pp. 329–346, 2007.

[30] A. J. Rothman, E. Levina, and J. Zhu, "Sparse multivariate regression with covariance estimation," *Journal of Computational and Graphical Statistics*, vol. 19, no. 4, pp. 947–962, 2010.

[31] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "A sparse-group lasso," *Journal of Computational and Graphical Statistics*, vol. 22, no. 2, pp. 231–245, 2013.

[32] L. Breiman and J. H. Friedman, "Predicting multivariate responses in multiple linear regression," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 59, no. 1, pp. 3–54, 1997.

[33] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[34] D. Kocev, C. Vens, J. Struyf, and S. Džeroski, *Ensembles of multi-objective decision trees*. Springer, 2007.

[35] B. Ženko and S. Džeroski, *Learning classification rules for multiple target attributes*. Springer, 2008.

[36] M. Pugelj and S. Džeroski, "Predicting structured outputs k-nearest neighbours method," in *Discovery Science*, pp. 262–276, Springer, 2011.

[37] T. Aho, B. Ženko, S. Džeroski, and T. Elomaa, "Multi-target regression with rule ensembles," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 2367–2407, 2012.

[38] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[39] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[40] H. Blockeel, L. De Raedt, and J. Ramon, "Top-down induction of clustering trees," *arXiv preprint cs/0011032*, 2000.

[41] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 1, pp. 66–75, 1994.

[42] J. Kittler, M. Hatef, R. P. Duin, and J. Matas, "On combining classifiers," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 3, pp. 226–239, 1998.

[43] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants," *Machine learning*, vol. 36, no. 1-2, pp. 105–139, 1999.

[44] J. Schneider, "Cross validation," *A Locally Weighted Learning Tutorial Using Vizier*, vol. 1, 1997.

[45] F. Joanneum, "Cross-validation explained," 2005.

[46] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[47] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937.

[48] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *The Annals of Mathematical Statistics*, vol. 11, no. 1, pp. 86–92, 1940.

[49] J. Zar, "Biostatistical analysis. the university of chicago press upper saddle river," *New Jersey*, 1999.

[50] D. J. Sheskin, *Handbook of parametric and nonparametric statistical procedures*. crc Press, 2003.

[51] R. L. Iman and J. M. Davenport, "Approximations of the critical region of the fbietkan statistic," *Communications in Statistics-Theory and Methods*, vol. 9, no. 6, pp. 571–595, 1980.

[52] R. A. Fisher, "Statistical methods and scientific inference.," 1956.

[53] P. Nemenyi, "Distribution-free multiple comparisons," in *Biometrics*, vol. 18, p. 263, INTERNATIONAL BIOMETRIC SOC 1441 I ST, NW, SUITE 700, WASHINGTON, DC 20005-2210, 1962.

[54] J. W. Tukey, "Comparing individual means in the analysis of variance," *Biometrics*, pp. 99–114, 1949.

[55] O. J. Dunn, "Multiple comparisons among means," *Journal of the American Statistical Association*, vol. 56, no. 293, pp. 52–64, 1961.

[56] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian journal of statistics*, pp. 65–70, 1979.

[57] Y. Hochberg, "A sharper bonferroni procedure for multiple tests of significance," *Biometrika*, vol. 75, no. 4, pp. 800–802, 1988.

[58] G. Hommel, "A stagewise rejective multiple test procedure based on a modified bonferroni test," *Biometrika*, vol. 75, no. 2, pp. 383–386, 1988.

[59] C. Adamson, "What is java," 2006.

[60] B. Durrant, E. FRANK, L. HUNT, G. HOLMES, M. MAYO, B. PFAHRINGER, T. SMITH, and I. WITTEN, "Weka 3: Data mining software in java," 2013.

[61] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas, "Mulan: A java library for multi-label learning," *The Journal of Machine Learning Research*, vol. 12, pp. 2411–2414, 2011.

[62] O. Corporation, "¿qué es netbeans?," 2015.

[63] V. Paradigm, "Visual paradigm for uml," *Visual Paradigm for UML-UML tool for software application development*, 2013.

[64] R. Pressman, "Ingeniería del software. un enfoque práctico. sexta edición. editoria l mcgraw-hill," 2005.

[65] C. Larman, *UML y Patrones*. Pearson, 1999.

[66] A. Asuncion and D. Newman, "Uci machine learning repository," 2007.

[67] W. Groves and M. Gini, "Improving prediction in tac scm by integrating multivariate and temporal aspects via pls regression," in *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets*, pp. 28–43, Springer, 2013.

[68] D. Pardoe and P. Stone, "The 2007 tac scm prediction challenge," in *Agent-Mediated Electronic Commerce and Trading Agent Design and Analysis*, pp. 175–189, Springer, 2010.

[69] W. Groves and M. Gini, "A regression model for predicting optimal purchase timing for airline tickets," tech. rep., Technical Report 11-025, University of Minnesota, Minneapolis, MN, 2011.

Anexo A

Anexo

La siguiente tabla muestra los resultados para cada algoritmo de cada salida de las base datos.

Dataset	CWCC	CWCV	MORF	ST	MTS	MTSC	ERC	ERCC	KNNSP
rf1									
CHSI2	24.73	24.69	54.98	49.32	53.25	47.64	50.98	48.65	59.31
NASI2	357.73	157.27	101.19	103.84	100.82	102.33	102.83	103.15	98.97
EADM7	34.99	34.98	55.52	52.36	49.79	50.50	50.04	51.85	60.91
SCLM7	59.19	53.59	75.09	66.82	62.56	64.95	62.93	65.86	120.95
CLKM7	35.64	29.71	87.23	73.81	74.72	72.89	73.71	73.53	81.43
VALI2	102.35	53.28	93.17	60.34	158.93	62.15	142.05	61.62	103.26
NAPM7	51.72	51.38	94.55	92.55	96.02	100.65	93.67	96.60	195.91
DLDI4	65.55	61.04	94.55	92.55	96.02	100.65	93.67	96.60	98.59
rf2									
CHSI2	20.59	21.22	59.04	49.30	53.06	47.56	51.06	48.63	64.29
NASI2	135.26	118.10	100.55	103.90	100.76	102.49	103.01	103.25	99.18
EADM7	30.61	30.39	56.91	52.42	50.32	50.35	50.08	51.88	71.62
SCLM7	49.46	48.58	95.89	67.41	62.50	66.12	62.81	66.38	77.78
CLKM7	29.70	27.30	83.62	74.09	75.01	73.57	74.00	74.02	85.73
VALI2	50.12	38.55	97.87	59.50	153.91	61.21	142.76	60.15	103.56

NAPM7	52.17	55.37	148.90	58.01	62.41	57.09	59.44	57.81	59.16
DLDI4	63.52	61.43	92.33	92.51	96.00	100.47	93.72	96.43	107.33
scm1d									
LBL	35.89	35.89	51.18	37.19	38.01	36.58	37.28	36.87	58.62
MTLp2	35.53	35.53	50.47	38.18	37.06	37.32	36.61	37.19	60.59
MTLp3	41.29	41.29	53.12	46.85	44.02	43.59	44.01	44.10	60.42
MTLp4	41.98	41.98	56.22	46.98	50.71	48.48	47.34	46.89	62.65
MTLp5	51.31	51.32	72.79	59.56	63.64	58.87	64.17	58.20	77.65
MTLp6	54.99	54.99	76.07	66.05	65.82	64.95	65.63	64.77	79.00
MTLp7	55.27	55.27	73.23	64.51	64.42	63.59	63.82	61.05	77.96
MTLp8	56.01	56.01	77.48	70.56	69.13	67.74	70.40	66.17	81.28
MTLp9	40.02	40.02	51.11	41.31	42.07	40.82	40.35	40.83	51.10
MTLp10	39.58	39.58	51.57	45.13	40.33	44.24	41.83	44.56	53.10
MTLp11	43.38	43.38	52.21	47.31	43.74	46.34	44.40	46.56	50.89
MTLp12	42.76	42.76	56.19	43.06	43.80	43.21	42.93	42.96	54.58
MTLp13	36.04	36.04	49.34	39.18	39.30	39.09	38.67	38.86	52.75
MTLp14	36.20	36.20	45.45	37.56	38.27	37.37	37.23	37.17	48.83
MTLp15	37.85	37.85	45.39	39.62	39.05	39.95	38.99	39.54	50.37
MTLp16	39.75	39.74	44.17	40.97	39.28	40.08	39.72	40.30	49.04
scm20d									
LBL	61.44	61.74	65.64	62.86	64.90	62.28	64.47	61.76	71.59
MTLp2A	65.20	63.47	65.68	64.91	63.84	64.58	64.96	62.20	74.69
MTLp3A	62.46	61.41	64.21	71.19	65.77	67.85	62.58	65.17	79.68
MTLp4A	70.85	65.52	69.04	71.74	74.78	72.93	67.33	69.11	85.33
MTLp5A	95.00	99.14	91.85	92.67	103.83	93.72	98.98	92.68	106.40

MTLp6A	93.58	101.13	94.61	97.17	99.64	99.38	99.02	95.80	112.36
MTLp7A	94.07	102.11	89.68	100.13	97.33	96.79	96.72	94.43	109.00
MTLp8A	97.20	102.00	94.01	99.00	103.54	101.70	100.99	99.27	118.78
MTLp9A	76.30	70.08	71.51	71.12	69.90	73.44	71.43	69.62	79.26
MTLp10A	70.87	71.65	76.43	74.16	74.04	77.84	73.94	74.76	86.37
MTLp11A	73.25	72.35	77.15	73.82	72.48	76.30	74.24	72.43	92.32
MTLp12A	72.45	70.91	81.80	74.52	77.69	78.00	77.25	73.03	98.24
MTLp13A	69.04	69.84	77.53	72.92	71.98	72.00	72.76	72.22	88.16
MTLp14A	67.05	67.00	75.35	72.49	71.41	73.78	71.48	71.93	87.80
MTLp15A	69.33	66.35	74.51	70.22	72.80	72.19	71.28	69.19	86.71
MTLp16A	66.21	67.01	75.02	73.92	73.98	73.81	73.35	71.91	89.76
atp1d									
ALLminpA	47.48	47.55	47.36	48.23	48.51	48.19	48.54	48.18	61.98
ALLminp0	41.16	41.28	43.65	42.97	42.00	42.20	42.64	42.79	65.87
aDLminpA	42.51	41.99	42.44	41.57	40.46	40.96	39.76	41.20	61.91
aCOminpA	36.39	34.16	35.63	24.19	23.32	24.16	24.17	23.94	63.14
aFLminpA	48.18	47.13	48.69	47.12	47.97	47.30	46.97	47.31	56.05
aUAminpA	36.39	34.95	35.57	20.03	20.71	20.19	20.52	20.05	65.28
atp7d									
ALLminpA	70.84	70.66	63.56	64.07	61.89	62.93	69.57	62.60	95.29
ALLminp0	55.35	755.11	60.16	66.99	67.22	65.30	67.50	65.99	84.38
aDLminpA	55.44	55.32	52.39	54.64	53.32	53.70	55.29	54.21	90.87
aCOminpA	44.69	43.42	43.71	31.62	26.90	27.81	29.36	29.05	87.48
aFLminpA	68.65	68.67	67.44	68.92	72.84	70.04	70.74	69.38	92.15
aUAminpA	45.63	44.35	43.20	28.61	26.43	24.65	28.10	26.23	97.08

oes10									
513021	38.58	36.60	45.34	43.81	43.79	43.72	43.64	43.79	53.18
292071	33.67	35.31	39.32	38.50	38.27	38.16	38.70	38.28	46.33
392021	41.75	41.93	39.72	41.22	41.25	41.36	41.26	41.24	51.59
151131	35.42	35.22	45.70	43.05	43.14	43.07	43.10	43.03	55.72
151141	38.39	36.42	49.61	43.64	43.88	44.05	43.80	43.80	59.22
291069	52.40	50.93	65.01	61.64	61.70	61.63	61.72	61.64	62.04
119032	28.68	27.43	40.17	36.97	36.89	37.09	36.97	37.01	53.13
432011	32.06	31.68	39.69	39.15	39.22	39.22	39.19	39.17	49.81
419022	53.99	52.80	65.44	64.41	64.44	64.44	64.43	64.40	65.88
292037	30.69	30.60	38.24	33.49	33.70	33.69	33.53	33.55	47.48
519061	28.47	28.02	55.66	39.48	39.67	39.66	39.52	39.55	53.25
291051	19.65	18.15	26.65	26.46	26.72	26.58	26.51	26.40	37.47
172141	56.12	56.54	59.30	49.44	49.63	49.56	49.49	49.45	58.54
431011	17.47	14.55	22.72	26.86	26.72	27.09	26.84	26.90	41.31
291127	37.22	38.10	44.16	46.15	45.28	45.60	46.08	45.78	51.44
412021	37.30	36.95	46.15	37.66	37.85	37.88	37.63	37.79	42.61
oes97									
58028	19.73	18.88	26.46	33.62	33.48	33.69	33.67	33.64	42.38
15014	28.98	27.64	46.50	35.83	36.54	35.16	35.88	35.24	53.73
32511	74.46	73.90	75.16	71.67	71.50	71.37	71.58	71.42	76.85
15017	30.13	30.69	40.25	36.49	36.96	36.79	36.47	36.61	43.34
98502	59.84	59.50	71.49	66.49	66.61	66.54	67.05	66.50	78.21
92965	57.08	58.00	69.84	64.76	64.84	64.73	64.92	64.72	72.54
32314	70.20	72.04	70.43	61.41	61.54	61.72	61.29	61.52	65.86

13008	21.97	21.34	27.17	33.94	33.97	33.74	33.92	33.79	44.80
21114	22.33	20.72	29.07	30.69	30.72	30.82	30.70	30.73	43.57
85110	44.22	44.06	63.60	56.69	57.45	56.83	56.68	56.63	61.99
27311	52.90	52.12	56.59	60.08	59.47	59.51	59.90	59.80	64.22
98902	41.97	42.80	54.28	53.59	53.72	53.63	53.63	53.52	55.43
65032	47.10	46.84	53.46	55.20	55.43	55.01	55.18	55.14	59.38
92998	59.90	58.45	70.27	64.36	64.16	64.42	64.69	64.25	76.72
27108	54.38	54.61	63.64	58.02	57.63	57.66	58.07	57.70	67.40
53905	53.72	54.31	60.21	56.89	57.41	57.17	57.00	57.00	69.48
flare1									
c-class	98.06	97.85	103.46	101.68	103.72	100.08	100.03	100.67	99.65
m-class	98.75	98.48	121.16	109.63	113.68	100.50	99.18	103.35	95.87
x-class	103.34	117.27	160.12	129.29	120.70	119.81	115.84	122.61	126.93
flare2									
c-class	93.33	93.20	99.59	98.01	97.45	96.36	97.33	96.43	93.93
m-class	105.23	104.08	115.95	107.54	99.35	99.20	102.18	102.98	115.77
x-class	151.99	157.07	211.89	139.28	86.64	121.04	116.46	126.97	199.94
water-quality									
25400	95.37	94.80	92.38	92.45	93.05	92.81	92.95	92.72	94.08
29600	99.01	98.60	97.63	98.65	98.62	98.38	98.54	98.26	99.39
30400	97.00	95.92	94.20	94.52	94.49	95.11	94.37	94.61	95.21
33400	95.14	94.64	89.32	91.20	90.37	90.23	91.21	90.18	93.42
17300	97.51	96.54	89.53	90.21	92.10	91.39	90.63	90.79	93.10
19400	91.14	90.24	82.79	83.42	82.93	82.94	83.55	82.85	87.63
34500	97.45	96.93	95.95	96.95	96.14	96.34	96.50	95.71	96.97

38100	95.31	95.45	90.70	91.20	90.77	90.80	91.24	91.05	92.87
49700	92.72	91.56	79.31	79.48	81.49	80.82	79.90	79.57	87.16
50390	97.00	95.93	89.15	89.16	90.11	89.91	89.21	89.19	94.76
55800	96.49	95.78	90.32	92.36	93.11	92.88	92.59	92.40	94.23
57500	96.81	95.98	89.63	91.76	91.73	91.86	91.75	91.42	92.57
59300	99.16	99.11	93.10	94.67	95.66	95.09	95.42	94.69	97.90
37880	92.08	91.14	85.08	85.57	84.82	84.69	85.68	84.87	89.78
edm									
DFlow	92.40	92.80	77.54	81.53	81.68	81.23	81.79	81.37	94.84
DGap	77.25	77.19	69.22	66.89	66.92	**66.70**	66.90	66.77	76.65