# Estimación parámetros de Sorter
# Trabajo fin de máster

Guillermo Portela Vázquez

**Resumen**

Se presenta a continuación una versión reducida del trabajo completo entregado a la coordinación del máster pero su publicación no está autorizada por motivos de confidencialidad de la empresa. Se expondrán secciones resumiendo los apartados que aparecen en el trabajo original sin revelar información sensible de la empresa.

## 1. Explicación de la problemática

En las instalaciones de la empresa INDITEX se realiza un cierto proceso involucrando una máquina llamada Sorter. A la hora de realizar el proceso se necesita organizar y disponer adecuadamente a un número de trabajadores. Para confeccionar este reparto de labores entre trabajadores existe un programa de optimización, del cual la empresa no es dueña intelectual pero sí tiene acceso a su ejecución. Este programa produce tablas candidatas de reparto de labores entre trabajadores. La utilización de las tablas propuestas por el programa no es obligatoria y realmente suele ser usada como guía para el encargado de organización que tiene la palabra final al respecto. Dicho programa de optimización requiere varios parámetros de entrada, algunos de ellos son observables teniendo en cuenta características del proceso y las instalaciones donde se va a realizar. Pero dos de ellos, el número de trabajadores a emplear y un parámetro que llamaremos *vid*, son valores que aquel que ejecute el programa tiene que decidir intuitivamente. El objetivo de las prácticas consistió en construir un criterio que decida, a partir del resto de valores sí observables antes de la ejecución, el valor del parámetro *vid* que provoque en el programa de optimización la generación de la mejor tabla de reparto de trabajos posible. Para analizar si una tabla de reparto de trabajo es mejor o peor que otra se atiende a 4 métricas de calidad que son observables una vez obtenida la tabla de reparto de trabajos en cuestión y son valores numéricos. La necesidad de crear una herramienta así surge del precio computacional que supone ejecutar el programa de optimización, tardando hasta 10 minutos en acabar desde que se inicia el programa. Además, realizar el proceso que estas tablas "organizan" también es realmente costoso. Se observó que el programa es altamente sensible al parámetro *vid* pudiendo producir cargas de trabajo muy desiguales entre dos trabajadores, algo que la empresa quiere evitar.

## 2. Explicación de la base de datos

En este apartado se cuenta la forma de obtener la información relativa al proceso a partir de la base de datos de la empresa. Comentando vicisitudes sobre el software empleado, `databricks`, y herramientas de trabajo disponibles. Se explican las conexiones entre tablas de una pequeña parte de la base de datos y se dirige a los anexos del trabajo adecuados donde se detallan las consultas de `sql` necesarias para recabar la información de una manera útil para el resto de tareas del trabajo. Debido a un cambio en la forma de almacenar ciertos datos, el estudio de relaciones entre tablas en la base de datos tuvo que hacerse dos veces por separado. También se comentan limitaciones por falta de documentación acerca de ciertos aspectos del proceso a estudiar.

# 3. Cómo puntuar una tabla de reparto

En este capítulo se proponen dos formas de evaluar las tablas de reparto comprimiendo la información de las 4 métricas de calidad en una sola. Es decir, se construyen 2 funciones $f_1$ y $f_2$ que parten del mismo subconjunto de $\mathbb{R}^4$ y tienen como conjunto de llegada $\mathbb{R}$. De forma que se preserve cierta estructura de orden en las puntuaciones.

#### 3.0.1. Método inspirado en PCA

El primero de los métodos propone encontrar una función lineal de la forma

$$f_1(\overrightarrow{x}) = (\overrightarrow{x}) = \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \alpha_4 x_4 = \overrightarrow{\alpha} \cdot \overrightarrow{x} \tag{1}$$

donde $\overrightarrow{\alpha}$ es un vector de pesos y $\overrightarrow{x}$ es el vector conteniendo las 4 métricas de la tabla. Este método toma inspiración del análisis de componentes principales propio del análisis multivariante para obtener los valores del vector $\overrightarrow{\alpha}$ a partir del histórico de las métricas de tablas anteriores. Por características propias de las métricas de calidad, se tienen que restringir los posibles valores de las coordenadas de forma que obligue a $\overrightarrow{\alpha}$ a pertenecer a uno concreto de los hexadecantes de la esfera $\mathbb{S}^3$.

## 3.1. Mediante función de distribución empírica

Este método propone asignar una puntuación a través de la homogeneización por cuantiles respecto al histórico de los valores observados en resto de tablas disponibles. Es decir, para una tabla y una métrica concreta se asigna la proporción de las otras tablas registradas que obtuvieron una puntuación peor que la tabla escogida concreta en esa métrica. Este proceso se realiza para las 4 métricas y luego se suman las nuevas 4 puntuaciones siguiendo unas restricciones dictaminadas por el concepto que representa cada una de esas métricas.

# 4. Modelos considerados

Se propuso aplicar modelos de machine learning para intentar predecir el valor de cada una de las 4 métricas de calidad en función de los diferentes valores de *vid*. En este capítulo se hace una introducción teórica a cada uno de los modelos empleados:

1. Regresión lineal Múltiple. Método clásico empleado de forma introductoria
2. Modelos aditivos.
3. Bosques aleatorios.
4. XGBoost.

# 5. implementación

En este capítulo se discuten detalles sobre la implementación en `Python` de cada uno de los modelos listados en la sección anterior. También se discute la calidad de las predicciones producidas los distintos modelos. Para esta labor se ha seguido una dinámica de trabajo clásica del aprendizaje estadístico. Separando los datos en entrenamiento y test, sobre los primeros se construyen los modelos y se evalúa su efectividad en el conjunto de test. Se ve que una de las 4 métricas es muy fácil de predecir con casi cualquier modelo y que otra de las 4 es difícil también para todos los modelos. Finalmente se determina que para dos de ellos es mejor utilizar bosques aleatorios y para los otros dos es mejor utilizar XGBoost.
En este capítulo se construye también una herramienta donde se introducen los valores observables antes de obtener la tabla de reparto y a través de los modelos con mejor comportamiento se crean gráficas de predicción para cada una de los 4 métricas de calidad, junto a las dos versiones comprimidas discutidas en el capítulo 3, en función de qué parámetro de entrada *vid* se escoja. Resaltando en esas gráficas el valor que se considera óptimo a introducir como parámetro de entrada al programa. Se genera también una versión opcional donde se obtienen gráficas 3D incluyendo también predicciones del comportamiento en función del segundo parámetro no fijado, el número de trabajadores, resaltando en estas gráficas de igual manera los valores que se postulan como óptimos. Por otra parte en esta herramienta se deja modificar por el usuario los valores de $\overrightarrow{\alpha}$ por si quiere utilizar una ponderación diferente de las métricas de calidad.

# 6. Conclusiones y trabajo futuro

En este apartado se discuten las limitaciones del programa y posibles mejoras en un futuro. Estas se tratan, esencialmente, de mejoras en la interfaz de usuario y conexiones entre otros programas que, principalmente, son tarea de un trabajador con un perfil informático.