UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERIA INDUSTRIAL

OPTIMIZACIÓN DE RUTEO MULTIMARCAS DE PRODUCTOS PARA REPONEDORES DE SUPERMERCADOS

TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN GESTIÓN DE OPERACIONES

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL INDUSTRIAL

GERALD ESTEBAN MATUS CONTRERAS

PROFESOR GUÍA:
ANDRES MUSALEM SAID

MIEMBROS DE LA COMISIÓN:
FERNANDO ORDOÑEZ PIZARRO
CHARLES THRAVES CORTÉS-MONROY

SANTIAGO DE CHILE
2023

Resumen

Según América Económica se mueven aproximadamente 10,000 millones de dólares al año en ventas en supermercados en Chile. Cada uno de estos supermercados tiene por objetivo hacer la mayor cantidad de ventas posibles, para ello es necesario que cada cliente tenga disponible en el local sus productos de interés. Estos productos pueden ser bebestibles, alimentos perecibles, alimentos no perecibles o cualquier otro producto que sea de utilidad para el cliente final, tales como elementos de aseo o loza. Productos que se pueden ver disponibles dentro de la gran gama de artículos que ofrecen los supermercados en la actualidad.

Es aquí donde una actividad primordial actúa, la reposición. La cual consiste en que cada trabajador, conocido como reponedor, mueve productos desde la bodega hasta las góndolas del supermercado para su posterior comercialización. El cliente que busca su producto de interés en la góndola, lo hace casi en el mismo horario que el reponedor coloca el artículo en el estante. Es por ello que surge el interés de analizar este proceso de reposición, con el fin de encontrar un modelo de reposición óptimo para la empresa.

Para resolver el problema del orden de reposición, se hará uso de herramientas de Machine Learning combinadas con optimización. La meta es maximizar las ventas de los productos que se repondrán, analizando el comportamiento actual que tienen cinco marcas dentro del local piloto.

Usando diferentes modelos de predicción, se buscó anticipar las ventas que ocurrirán por hora durante el día. Este dato obtenido se usa en el modelo de optimización, para que vaya cambiando su función objetivo a través del tiempo. El interés es simular que, si se repone antes de que un cliente llegue con intenciones de comprar, el supermercado obtendrá ganancia por haber vendido ese producto.

De esta manera, se realizaron distintas pruebas, con el objetivo de testear diferentes versiones del modelo de optimización. Las iteraciones tenían como fin, lograr un modelo lo más cercano a la realidad. Así poder aplicar este resultado al supermercado piloto, que fue el Santa Isabel de Avenida Grecia en la Región Metropolitana.

Respecto al último modelo presentado, se logró que funcionara con una predicción de demanda calculada la noche anterior al día de reposición y el modelo de optimización durante la mañana del día a reponer. Con un gap de optimalidad menor a 5 % cuando el número a reponer es del orden de 60 productos y con optimalidad cuando estaba en el orden de 30 producto. Es decir, para el último caso, se demora menos de 1 en definir la ruta óptima.

Para mis padres, que siempre confiaron en mi.

Agradecimientos

Este documento marca el inicio y un fin de dos grandes etapas de mi vida, primeramente, marca el final de mi época universitaria hasta ahora, durante bastantes años (no vale la pena decir cuántos jaja) empecé en un pequeño colegio en Máfil un pequeño pueblo de Valdivia, hasta que por cosas de la vida termine mi enseñanza media en Santiago, buscando siempre mi camino y mis objetivos, hasta que llego en ese punto de mi vida que veía bastante lejano hace no mucho tiempo.

Se que no es fácil criar a una persona, por lo que mis primeros agradecimientos son a mis padres Gerald Matus Rojas y María Contreras Zamora, que, aunque ni yo mismo confiara en mis capacidades, nunca dudaron de que podría lograr todo lo que me propusiera en los diferentes ámbitos de la vida, pero por sobre todo en los estudios, definitivamente no estaría terminando esta etapa si no fuera por ellos.

Así mismo tengo que agradecer a mis abuelos, varios de ellos ya no se encuentran conmigo, pero tengo claro que si me apoyaban de esa manera en vida mucho más lo hacen desde donde estén, acompañándome en cada momento, apoyándome y ayudándome en cada etapa de mi vida, ya sea malcriándome y simplemente enseñándome a ser mejor persona.

A mis amigos y compañeros, definitivamente cuando llegué a esta universidad tuve que aprender (a la mala) de que cada persona me aporta su granito de arena para ser mejor estudiante, mejor persona y definitivamente mejor amigo.

Por último, a mi profesor guía Andres Musalem, que confió en mi a pesar de no ser un estudiante tan destacado como muchas personas de mi generación, pero que incluso en periodo de vacaciones me apoyo para que toda esta investigación saliera lo mejor posible apoyándome en muchas reuniones, a pesar de que en la mayoría del tiempo el estuviera en otro país.

Es en este documento, donde finaliza mi periodo universitario y empieza mi etapa (espero) como profesional, ser un inJeniero Civil Industrial con Magister en Gestión de Operaciones, en donde siempre quiero seguir aprendiendo y descubriendo todo lo que puedo hacer por este mundo.

Tabla de Contenido

1. Introducción **1**

1.1. Objetivos 2

1.1.1. Objetivo Principal 2

1.1.2. Objetivos Específicos 2

1.2. Estructura de la tesis 2

2. Estado del Arte **3**

3. Descripción Empírica y Datos Disponibles **6**

4. Metodología **10**

4.1. Predicción de Demanda 10

4.2. Modelo de Optimización 13

4.2.1. Primer Modelo de Optimización: 14

4.2.2. Segundo Modelo de Optimización: 20

5. Análisis y puesta en marcha **27**

5.1. Estimación de Parámetros 27

5.2. Análisis previo y pruebas previas 28

5.2.1. Primera Prueba Real 28

5.2.2. Evaluación con parámetros reales, 70 productos 29

5.2.3. Evaluación con parámetros reales, 61 productos 30

5.2.4. Evaluación con parámetros reales, 30 productos 31

5.3. Puesta en Marcha 33
5.3.1. Antecedentes 33
5.3.2. Prueba en terreno 33

6. Conclusión **35**

Bibliografía **38**

Índice de Tablas

3.1. Resumen Stock Movilizado 8

3.2. Ejemplo de datos reales stock 9

3.3. Ejemplo de datos reales stock, agregando información de Reposición y de Ventas 9

4.1. Variables dummy para los días 10

4.2. Variables dummy para los bloques de horario 11

5.1. Resumen prueba con 70 productos a reponer 30

5.2. Resumen prueba con 61 productos a reponer 31

5.3. Resumen 30 productos (día real) 32

Índice de Ilustraciones

2.1. Representación gráfica de los conjuntos con Nested Cross Validation. 4

3.1. Algunos de los productos presentes en Santa Isabel Grecia. 7

3.2. Total de Ventas en la tienda. 8

4.1. Resultado de la predicción. 13

4.2. Representación grafos primera prueba simplificado, ya que todos los nodos estan conectados entre sí. 16

4.3. Resultados de la primera prueba, ruta sugerida siguiendo de manera decreciente el aporte a la F.O. 16

4.4. Representación grafos segunda prueba simplificado, ya que todos los nodos estan conectados entre sí. 17

4.5. Resultados de la segunda prueba. 18

4.6. Representación grafos primera prueba simplificado, ya que todos los nodos estan conectados entre sí. 18

4.7. Resultados de la tercera prueba. 19

4.8. Ejemplo output modelo. 19

4.9. Representación grafos primera prueba con capacidad del carro, simplificado ya que todos los nodos estan conectados entre sí. 23

4.10. Resultados de la primera prueba con capacidad del carro. 24

4.11. Representación grafos segunda prueba con capacidad del carro, simplificado ya que todos los nodos estan conectados entre sí. 24

4.12. Resultados de la segunda prueba con capacidad del carro. 25

4.13. Representación grafos tercera prueba con capacidad del carro, simplificado ya que todos los nodos estan conectados entre sí. 25

4.14. Resultados de la tercera prueba con capacidad del carro. 26

Capítulo 1

Introducción

Según América Económica, se mueven aproximadamente 10,000 millones de dólares al año en ventas en supermercados en Chile. Este gran volumen depende de una actividad clave de cada supermercado, la reposición, que consiste en asegurarse de posicionar los productos presentes en la bodega a las distintas góndolas para ser vendidos.

Actualmente, la actividad de reposición se realiza por las marcas de los proveedores, es decir, un proveedor del supermercado envía a personas a este lugar para que, desde la bodega del mismo local, coloque los productos en las góndolas. Así queda al alcance del cliente final para su posterior compra. Luego de esta tarea, el reponedor se traslada a otro supermercado a realizar la misma tarea, incurriendo en costos de tiempo y movilización.

Es aquí donde Zippedi, empresa chilena que ofrece monitorear el stock en góndola, a través de fotos con robots, que durante la noche recorren los pasillos de los supermercados.

Complementando el servicio a sus clientes, Zippedi decidió extender su labor a facilitar servicios de reposición a diferentes marcas, aportando a la eficiencia de la empresa proveedora y al bienestar para el trabajador encargado de la tarea; pero, para decidir el orden nacen las siguientes preguntas: ¿cuál producto se debería reponer primero?, ¿se debería velar por la ruta mínima y/o por la ventas?.

Actualmente el orden en el cual se hace la reposición no presenta un orden por escrito, sino que más bien el reponedor tiene la libertad de realizarlo de manera libre en la prueba piloto de Zippedi. Son un total de 461 productos de 5 marcas, que se reponen de la siguiente forma: entre 20 y 60 productos diarios con aproximadamente 4-5 idas a la bodega para repartir en los pasillos.

Es así como en este documento, el problema se tratará como un problema de ruteo de vehículos, en donde la ubicación de los productos vendría siendo los nodos, el reponedor sería el vehículo y la bodega sería el centro de abastecimiento.

La hipótesis que usaremos en esta tesis es que existen ciertos productos que se venden más en determinados periodos de tiempo, y, dependiendo de cuales sean, es que se deberían priorizar. Es así como el objetivo de esta tesis se enfoca en darle un orden para que este reponedor pueda realizar su tarea con distintas marcas combinando técnicas de Machine

Learning con optimización, maximizando las ventas esperadas por el modelo de predicción.

1.1. Objetivos

1.1.1. Objetivo Principal

El objetivo principal del trabajo es generar una metodología, para definir un orden de reposición en los supermercados en función de las ventas pronosticadas y la ubicación de los productos, los tiempos necesarios para realizar cada etapa de la reposición, las idas a la bodega para rellenar el carro, entre otras variables.

1.1.2. Objetivos Específicos

- Desarrollar un modelo de predicción por hora para cada producto que la empresa está encargada de reponer.
- Formular un modelo de optimización que permita maximizar las ventas de la empresa con la predicción de lo que se espera vender.
- Realizar una prueba piloto en algun supermercado.

1.2. Estructura de la tesis

Ya teniendo los objetivos explícitos, la estructura de esta tesis está dada, primero, por una investigación previa, en donde, se buscará información para realizar una predicción por hora y una optimización tanto de información en la web como de papers de optimización.

Luego de esto se explicará los datos disponibles y la manera en la cual se realizará la predicción de la demanda para todos los productos que tienen a cargo la empresa.

Posterior a esto, se explicará los distintos de modelo de optimización implementados buscando acercarse lo más posible a una situación realista de reposición.

Finalmente, se explicará la puesta marcha en un supermercado y las distintas mejoras propuestas una vez realizada las primeras pruebas.

Capítulo 2

Estado del Arte

Primero que nada, para realizar la predicción de la demanda por hora, se buscó información de la web específicamente, ya que en este medio aparecen distintas opciones para esta tarea, desde la implementación en código con Python hasta distintos algoritmos para realizarla.

Al estudiar algoritmos de predicción, se eligieron entre varios: regresión Lasso con Cross Validation, regresión Random Forest, Stepwise Forward y Facebook Prophet.

Regresión Random Forest: es un algoritmo de aprendizaje, que se basa en múltiples predicciones para luego promediarlas y tener el resultado de la predicción, [1].

Step Wise Forward: es un algoritmo que va agregando variables que aportan significancia al modelo y eliminan las que empeoran el modelo, todo esto a través del p-valor. Para el caso de este documento se consta de una significancia del 10 %, [8].

Fb Prophet: Es un modelo de regresión aditiva univariada con una tendencia de curva de crecimiento lineal o logística por partes. Incluye un componente estacional anual modelado usando series de Fourier y un componente estacional semanal usando variables ficticias. El Modelo funciona bien cuando existe un tiempo extendido de observaciones históricas detalladas y que tengan estacionalidades marcadas, [9].

Regresión Lasso con CV: Es el método Lasso para elección de variables significantes, pero el parámetro penalizador se escoge a través de un algoritmo de Validación Cruzada, [4].

Como se tenían todas estas opciones, se necesitaba una manera de elegir cual modelo funcionaba mejor para cierto producto, de esta manera se estudió la técnica llamada, "Nested Cross Validation"[2], el cual consta de dividir los datos temporales en 3 conjuntos (a diferencia de los 2 más naturales) los cuales son Entrenamiento, validación Cruzada y testeo, representada en la Figura 2.1, las cuales son ordenadas de manera temporal para los datos que se tienen disponibles, luego al promediar el error cuadrático medio de los conjuntos de validación cruzada, de esta manera se puede obtener una estimación casi imparcial del error según VARMA Y SIMON 2006 [12].

Luego de esto, viene el modelo de optimización el cual se decidió implementar con python-

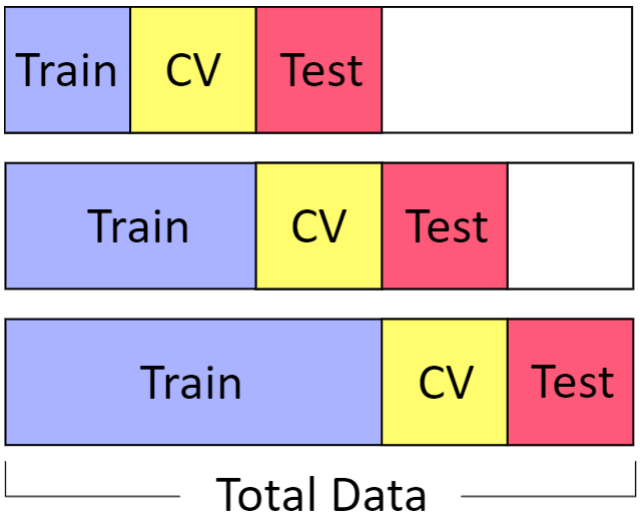

Figura 2.1: Representación gráfica de los conjuntos con Nested Cross Validation.

gurobi [10] , ya que ha sido el más utilizado hasta ahora. Para esto se buscaron varias investigaciones para proponer el correcto, en particular, se busca un modelo donde se visiten diferentes nodos (para nuestro caso productos), y donde no necesariamente se visiten todos, ya que, un reponedor puede no alcanzar a realizar todas sus tareas en un día. Además de esto, se debe encontrar un modelo que cambie a través del tiempo su función objetivo. De esta forma se añade a los modelos investigados ventanas de tiempo y variación de periodos de tiempo, es así como, por ejemplo, se le añade diferentes pesos si el producto es visitado en la mañana a que si se visita en la noche.

Traveling Salesman Problems with Profits: Este es un paper en donde se presenta el caso donde visitan distintos nodos y que no necesariamente se pasan por todos. En él, la función objetivo del problema muestra una minimización entre el costo de ir a los lugares y el beneficio de haber visitados esos nodos. Método que resulta aplicable al modelo que se busca obtener en este documento, maximizando la ganancias de haber repuesto los productos, quizás no realizar la resta del costo, ya que, se necesitan otras informaciones más específicas, como el sueldo del reponedor para ponderarlo con el tiempo, además de que existen otros beneficios intangibles que no se pretenden simular en el modelo, [5].

Heuristics for the multi-period orienteering problem with multiple time windows: Este en particular tiene bastantes cosas interesantes a tener en cuenta para este documento. Primero que nada, nos presenta dos tipos de conjuntos, un conjunto de nodos que tiene que visitar sí o sí y, además de esto, un conjunto de nodos opcionales que tiene que maximizar. Si bien, se cree que en esta tesis no deberían existir conjuntos opcionales para reponer, muestra una función objetivo que solo busca maximizar ganancias.

Otro punto por destacar de este paper es que presenta ventanas de tiempo, en donde cada nodo tiene que ser visitado en cierto periodo. Estos nodos tienen un tiempo de atención y de viaje, así como restricciones en que, si tiene que salir de cierto nodo debe haber salido del anterior, haber viajado desde el nodo de salida a nodo de llegada, y posteriormente tiene que haber terminado de atenderlo. Esta restricción parece estar acorde a lo que se necesita, ya que, el mercaderista termina de reponer el producto cuando sale del anterior, va hacia el

que tiene que reponer y llegando ya al pasillo, tiene que dejar la mercadería en la góndola. Posiblemente también se pueda adaptar para que genere los bloques de horario que se necesita para la Función objetivo, es decir, que cierto producto empiece y termine en cierto bloque, esto puede generar un tiempo en que el reponedor no haga nada, pero considerando los tiempos que una persona se mueve en el supermercado y repone un producto en la góndola, se puede asumir que este tiempo muerto debería ser de unos pocos minutos, por lo que no intervendrá de gran manera a la eficiencia del tiempo de la persona que siga la planificación, [6].

A Periodic Traveling Politician Problem with Time-Dependent Rewards en este paper, nos muestra un ruteo de un político que viaja a distintas ciudades. Lo interesante de este paper es que muestra una función objetivo dependiente del tiempo, en donde mientras más tiempo se quede en un lugar, la recompensa de quedarse es menor, es decir, es decreciente con el paso del tiempo. Si bien es parecido de cierta manera al objetivo que se tiene del modelo de optimización del presente documento, no se aprecia una característica que se pueda aplicar en este caso, ya que, en el modelo del documento se asumirá de que si parte la reposición del producto, el mercaderista se quedará en el lugar hasta que finalice la reposición de ese producto, [3].

Capítulo 3

Descripción Empírica y Datos Disponibles

Cabe mencionar que todos los datos fueron proporcionados por Zippedi y, además, se trabajó con ellos en un supermercado de Cencosud para realizar la prueba de reposición. Para este objetivo se contaba con la actualización de Stock total (ubicado tanto en la góndola como en bodega) por hora de cada producto, así fue posible estimar la venta como la diferencia de stock por hora. El piloto usado para aplicar el modelo fue el supermercado Santa Isabel de Av. Grecia en la Región Metropolitana, mientras que los clientes analizados de esta empresa fueron son S.C. Johnson, Essity, Chocolates Valor y Loreal.

Además de esto, en los supermercados que se encuentra el robot se pueden obtener datos como la posición de cada producto, pasillo, altura, precio, entre otras cosas. Este robot, genera automáticamente una alerta de reposición, que avisa si cierto producto tiene poco stock en góndola y si en la bodega existe el producto o no (con el dato de stock interno del supermercado).

En el caso de que no exista robot (para Cencosud estos son los supermercados Santa Isabel), las alertas se deben realizar de manera manual, donde el reponedor realiza una inspección visual, observando si existen pocas unidades del producto en la primera parte del día, en este caso marca aquellos productos como una Alerta de Reposición

Para conocer la situación actual, se realizaron visitas a 4 locales de supermercados en Santiago. En todos estos supermercados trabaja un Reponedor con jornada de lunes a sábado de 8 a.m. a 3 p.m. Este individuo se dedica a reponer, ordenar sala y bodega, además de pedir los productos que presentan quiebre en bodega.

De momento, en los supermercados Jumbo el robot recorre, durante la noche, la sala y deja alertas de reposición que debe realizar el reponedor; pero, en caso de Santa Isabel, esta tarea tiene que hacerla el trabajador a primera hora del día. Comienza escaneando cada código de barra que le falta stock en la góndola. En ambas cadenas de supermercados, cuando termina de reponer un producto, escanea el código dejando como tarea realizada, quedando con una marca de fecha/hora en Zippedi.

Figura 3.1: Algunos de los productos presentes en Santa Isabel Grecia.

Por otro lado, también interesan las ventas que existieron en cierto periodo de tiempo, por lo que una opción era conseguir estos datos directamente de los productos que pasaron por la caja, pero debido a que era un dato sensible de los supermercados, en reemplazo se hace uso del stock por hora de cada producto en la tienda como se muestra en la tabla 3.2.

En este punto del modelado, al mostrar el stock se puede hacer una suposición de que la diferencia del stock entre una hora y la otra es la venta estimada que hubo en ese periodo de tiempo. Esto presenta una dificultad cuando el stock aumenta, ya que eso puede significar que llegó un camión con una cantidad de productos que dejó en bodega, pero no se podrá saber cuánta venta real hubo en ese instante. También se asume que si repone menos cantidad de la que se vendió, se subestimará esta cantidad. Estos últimos dos casos son importantes tenerlos en cuenta a la hora de realizar los análisis (ver en tabla 3.3).

La cantidad de ventas en la tienda con los productos que tienen a cargo es más o menos constante, pero existen algunos outliers en donde las ventas se disparan posiblemente por un ajuste de stock de la misma tienda que se debe a robos, datos perdidos, etc.

Además de esto, los datos se trabajarán por hora, que es la lógica de los datos brindados, pero estos pueden venir con datos perdidos y el lapsus entre una observación y otra puede ser mayor, como se puede apreciar en la tabla 3.3, pueden haber 'Saltos' de bastante más tiempo. Este tipo de datos faltantes se tratarán por imputación de datos, de manera que para solucionar estos casos se va a rellenar con las horas faltantes y las ventas se repartirán en esas horas, de esta manera podemos tener una tabla con las ventas estimadas por hora.

En la tabla resumen 3.3 se puede ver que existen 461 productos distintos, con un promedio de 3.7 horas, y que el percentil 75 se encuentra en un 3.62. Esto quiere decir que casi todos los ítems tienen datos perdidos. Para resolver este dilema se hace uso nuevamente de la imputación de datos. Un punto que cabe destacar de esta tabla con relación a los datos perdidos es que existen unos pocos productos que tienen más datos perdidos al ver la fila del máximo, por lo que posiblemente la reposición de este producto no pueda ser predicho correctamente al realizar esta misma por hora.

Complementado a lo que se dijo anteriormente, se procede a realizar métodos de predicción. Cabe destacar que los métodos que se caracterizarán en la siguiente sección asumen

Tabla 3.1: Resumen Stock Movilizado

Item	store
Count	461
mean	3.7
std	0.9
min	1.8
25 %	3.62
50 %	3.62
75 %	3.63
Max	9.7

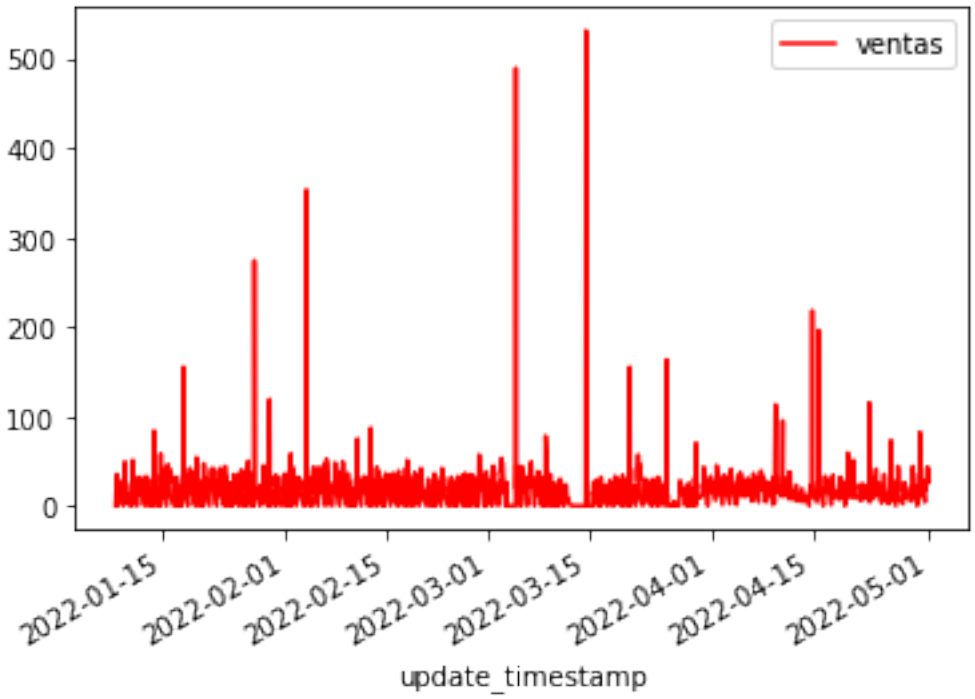

Figura 3.2: Total de Ventas en la tienda.

que se tiene todos los ítems con las respectivas ventas por hora sin datos perdidos. Es decir, se realiza la imputación de datos, de tal manera que aquellas horas en que no tenían datos, ahora tendrán datos, completando toda la jornada con ventas.

Tabla 3.2: Ejemplo de datos reales stock

Item	**store**	**update timestamp**	**stock**
1335736016	J501	2021-12-27 13:41:14.501831 UTC	47
1335736016	J501	2021-12-27 15:47:31.573646 UTC	53
1335736016	J501	2021-12-27 17:00:35.360918 UTC	52
1335736016	J501	2021-12-28 02:27:12.175715 UTC	50

Tabla 3.3: Ejemplo de datos reales stock, agregando información de Reposición y de Ventas

Item	**store**	**update timestamp**	**Stock**	**Reposición**	**Ventas**
1335736016	J501	2021-12-27 13:41:14.501831 UTC	47	0	1
1335736016	J501	2021-12-27 15:47:31.573646 UTC	53	1	0
1335736016	J501	2021-12-27 17:00:35.360918 UTC	52	0	1
1335736016	J501	2021-12-28 02:27:12.175715 UTC	50	0	2

Capítulo 4

Metodología

4.1. Predicción de Demanda

El siguiente análisis de prueba se realizó en una tienda específica, en particular se escogió el local de la cadena Santa Isabel de Avenida Grecia ya que presentaba un número mayor de productos que las demás tiendas analizadas con 461 productos en todo el supermercado los cuales están repartidos en 5 pasillos.

Para las horas en donde no existían datos de stock en la hora anterior, se buscaron variables explicativas de tipo dummy para realizar un modelo de predicción de la demanda de cada producto, las cuales se aprecian en la Tabla 4.1 y Tabla 4.2. Para cualquier otra situación que no sea el que se exige para que tome valor '1', la variable dummy tomará valor '0'.

Tabla 4.1: Variables dummy para los días

Grupo 1	**Valor "1"si se realiza la venta:**
lunes	ese día
martes	ese día
miércoles	ese día
jueves	ese día
viernes	ese día
sabado	ese día
domingo	ese día
DíaSemana	de lunes a viernes
finde	sábado a domingo
mar-jueves	martes a jueves
Feriado	feriados

Tabla 4.2: Variables dummy para los bloques de horario

Grupo 2	**Valor "1"si se realiza la venta:**
9hrs	8 a 9 hrs. inclusive
10hrs	9 a 10 hrs. inclusive
11hrs	10 a 11 hrs. inclusive
12hrs	11 a 12 hrs. inclusive
13hrs	12 a 13 hrs. inclusive
14hrs	13 a 14 hrs. inclusive
15hrs	14 a 15 hrs. inclusive
16hrs	15 a 16 hrs. inclusive
17hrs	16 a 17hrs. inclusive
18hrs	17 a 18 hrs. inclusive
19hrs	18 a 19 hrs. inclusive
20hrs	19 a 20 hrs. inclusive
21hrs	20 a 21 hrs. inclusive
22hrs	21 a 22 hrs. inclusive
9a10	8 a 10 hrs. inclusive
11a12	10 a 12 hrs. inclusive
13a14	12 a 14 hrs. inclusive
15a16	14 a 16 hrs. inclusive
17a18	16 a 18 hrs. inclusive
19a20	18 a 20 hrs. inclusive
21a22	20 a 22 hrs. inclusive
9a11	8 a 11 hrs. inclusive
12a14	11 a 14 hrs. inclusive
15a17	14 a 17 hrs. inclusive
18a21	17 a 22 hrs. inclusive
mañana	8 a 12 hrs. inclusive
tarde	12 a 18 hrs. inclusive
noche	18 a 22 hrs. inclusive

Dado el supuesto de que estas variables definen el comportamiento de las ventas del supermercado, conviene realizar una regresión múltiple que permita evaluar su impacto en la variable dependiente. Así se obtiene el siguiente modelo con las variables presentes en las tablas 4.1, 4.2:

$$Ventas = \beta_0 + \beta_1 DiaSemana + \beta_2 finde + \beta_3 lunes + ... + \beta_{10} Domingo + \sum_{horas} \beta_{horas} horas + \sum_{bloques} \beta_{bloques} BloquesHorarios + \varepsilon$$

En los resultados de la regresión, el R cuadrado ajustado era cercano a 0.3 usando todas las variables mencionadas, pero se optó por intentar mejorar este valor por lo que se decidió usar una nueva variable a que se llamará 'Ventas Anteriores' como variable explicativa. Este nuevo atributo consiste en usar el día anterior, es decir, ventas de la misma hora, pero del día anterior como variable independiente y también la misma variable, pero del mismo día de la semana anterior a esta.

Además, se agregaron variables llamadas 'Ventas Bloques' que constan de lo que se vendió en la mañana, tarde y noche del día anterior y de la semana pasada como variable explicativa.

$$\text{Ventas}_t = ModeloAnterior + \beta_1 Ventas_{t-1} + \beta_2 VentasBloqueMañana_{t-1} + \beta 3 VentasBloqueTarde_{t-1} + \beta_4 VentasBloqueNoche_{t-1} + \beta_5 Ventas_{t-7} + \beta_6 VentasBloqueMañana_{t-7} + \beta_7 VentasBloqueTarde_{t-7} + \beta_8 VentasBloqueNoche_{t-7} + \varepsilon$$

En donde el ModeloAnterior es el especificado antes de agregar las ventas del periodo anterior, y el t se considera hoy, por lo que t-1 es el día anterior y t-7 es el mismo día, pero de la semana pasada. Cabe destacar que no se considera la variable $Precio_t$ ya que la variación de esta variable solo existirá cuando el supermercado conste de robot, puesto que el supermercado se abstiene de otorgar estos datos.

Luego al evaluar el modelo con la regresión lineal especificada, se observó que los productos escogidos mejoraron su RCuadadroAjustado cercano a un 0.6 por lo que se dejó estas últimas variables y se procedió a estudiar modelos de predicción más complejos que la regresión lineal, presentes en el capítulo 2.

Usando alguno de estos modelos, puede pasar que, dependiendo de cada ítem/producto, algunos de estos modelos de predicción funcione mejor que otro, por lo que realizando Nested Cross Validation y comparando el promedio de sus errores cuadráticos medios se elegirá cuáles de los modelos elegir, pero antes de esto, se decidió descartar algunos de los modelos de predicción con el fin de equilibrar mejora en la predicción vs tiempo que demora en entregar la predicción del día, de esta manera se dejó Regresion Random Forest y Regresion Lasso con CV. En otras palabras, se realiza todo lo anteriormente mencionado y el que tenga menor error se elegirá para que prediga cierto producto.

Finalmente, para finalizar la predicción se realizó un ciclo con los 461 productos para predecir las ventas del día, el output de esto se representa en la figura 4.1.

En cada columna representa la predicción de un producto con su mejor estimación casi imparcial del error, caracterizando columna por columna:

Producto: Es el código de los productos presente en la predicción, en particular cada código ('ean') representa una diferente combinación de producto/tamaño, por ejemplo, el

producto	ventana	prediccion	acumulado	precio	llave	obj
157979	1	0.006896	0.2341234	3090	157979-1	723.44144
157979	2	0.006896	0.2272274	3090	157979-2	702.13268
157979	3	0.006896	0.2203314	3090	157979-3	680.82392
157979	4	0.0197352	0.2134353	3090	157979-4	659.51516
158609	1	0	0.4173819	3590	158609-1	1498.4011
158609	2	0	0.4173819	3590	158609-2	1498.4011
158609	3	0.012778	0.4173819	3590	158609-3	1498.4011
158609	4	0.0415771	0.4046039	3590	158609-4	1452.528

Figura 4.1: Resultado de la predicción.

producto 120734 representa 'Insecticida Raid Todo Insecto 360cc'

Ventana: Hora en la cual se realiza la predicción, son en total 14 ventanas en donde ventana 1 representa de 8-9 hrs, ventana 2 de 9-10 hrs, etc., hasta llegar a las 22 horas.

Predicción: Cantidad de productos vendida según el modelo.

Acumulado: Cantidad vendida desde esa hora hasta el final del día.

Precio: Precio del producto.

Obj: Valor que llegará al modelo de optimización como función objetivo, es el producto entre el acumulado y precio.

Cabe destacar que, en esto último, podría usarse perfectamente el margen del supermercado para ese producto, ya que lo que se busca es maximizar la ganancia que representa reponer ese producto antes que se venda, pero al ser una información sensible del supermercado se usa el precio para efectos de esta tesis. Cabe recordar que la idea de este modelo es que si se repone antes de que se venda se tendrá toda la ganancia de esta venta, pero si llega a reponer el producto después de que se hubiera vendido se pierde esta venta.

Dado que, como en los Santa Isabel no existe este robot, se tiene unos pocos minutos para realizar la predicción y posteriormente realizar la optimización. De esta manera se opta por realizar la predicción el día anterior para poder tener esa información el día siguiente en la mañana. Para la optimización hay que tener en cuenta que se tiene un tiempo acotado para tener una óptima solución.

Para cada producto la predicción se demora en promedio 16 segundos por cada artículo. Esto implica que para tener la predicción completa es necesario que paseen aproximadamente 2 horas, en el caso del proyecto piloto. Por lo tanto, por temas prácticos, podría iniciar el proceso de predicción cerca de las 11 p.m. y tener el programa de reposición de todos los productos completado a las 1 a. m.

4.2. Modelo de Optimización

Llegado a este punto, hay que recordar que el objetivo de esta predicción es usarla como input para el modelo de optimización para luego usarla de Función Objetivo, tratando de simular que dependiendo del tiempo en que llegue el mercaderista a reponer influirá en su

respectiva recompensa, de tal manera que si llega muy tarde al producto se perderán las ventas de lo que va del día, pensando en esto se presenta el modelo de optimización propuesto.

4.2.1. Primer Modelo de Optimización:

Conjuntos:

- Bloques: Bloques Horarios en el cual definimos, en particular, las primeras 7 variables de la Tabla 4.2. B.
- Productos: N.
- Productos con bodega: Conjunto de productos y la bodega No.

Parámetros:

- R_{ib}: Recompensa por reponer Producto i en el bloque b (output de la predicción).
- A_i: Tiempo de Reposición del producto i.
- $TIME_{ij}$: Tiempo de ir del producto i al producto j.
- END: Tiempo maximo de permanencia (fin de la jornada).
- $[S_b, E_b]$: Ventana de tiempo del bloque b.

Variables de decisión:

$$\mathrm{X}_{ij} = \begin{cases} 1 & \text{Si se va del producto } i \text{ al producto } j. \\ 0 & \text{Si no.} \end{cases}$$

$$\mathrm{Y}_{ib} = \begin{cases} 1 & \text{Si se repone el producto } i \text{ en el bloque } b. \\ 0 & \text{Si no.} \end{cases}$$

α_i = Tiempo en que sale del producto i.

Función objetivo:

$\text{máx} \sum_i \sum_b Y_{ib} * R_{ib}$

Restricciones:

1) Para salir de un producto tiene que haber entrado:
   $\sum_{j \neq i, j \in No} X_{ji} = \sum_{j \neq i, j \in No} X_{ij}, \forall i \in No$
2) No es necesario reponer todos los productos, puede que no lo alcance a reponer:
   $\sum_b Y_{ib} \leq 1, \ \forall i \in No$
3) Para ser repuesto tiene que haber llegado el reponedor:
   $\sum_b Y_{ib} = \sum_{j \neq i, j \in No} X_{ij}; \forall i \in No$
4) No puede irse de j antes de haber terminado de reponer:
   $\alpha_i + TIME_{ij} + A_j \leq \alpha_j + M(1 - X_{ij}), \forall i \in No, \forall j \in N$
5) Tiene que haber terminado de reponer dentro de la ventana:
   $(\text{S}_b) \cdot Y_{ib} \leq \alpha_i \leq E_b + M \cdot (1 - Y_{ib}), \forall i \in N, \forall b \in B$
6) Tener que terminar todo dentro de la jornada:
   $\alpha_i \leq END, \forall i \in No$
7) Naturaleza de las variables:
   (a) $X_{ij} \in \{0, 1\}$
   (b) $Y_{ib} \in \{0, 1\}$
   (c) $\alpha_i \geq 0$

Luego de formulado el modelo de optimización se procedió a realizar distintas pruebas para observar su comportamiento, cada prueba con cinco productos y cambiando los parámetros para ver cómo se comporta el modelo en las diferentes situaciones.

Pruebas

Primera Situación: Cinco productos a reponer con tiempos de traslado iguales para cada producto y tiempos de reposiciones iguales, pero con un diferente aporte a la función objetivo con la diferencia entre período a período de misma magnitud en cada producto, es decir, se vende la misma cantidad en cada periodo/hora, además restringido a un tiempo en donde no pueda pasar por todos los productos, así que tiene que dejar alguno fuera de la ruta. De esta manera el modelo sugiere que en esta situación el reponedor recorra desde el producto que tiene más aporte a la función objetivo y vaya recorriendo de esta manera de forma descendiente bajo este criterio, dejando al cual menos aporta fuera de la ruta.

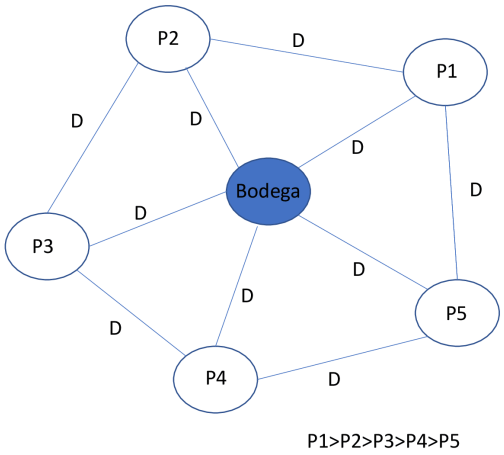

Figura 4.2: Representación grafos primera prueba simplificado, ya que todos los nodos estan conectados entre sí.

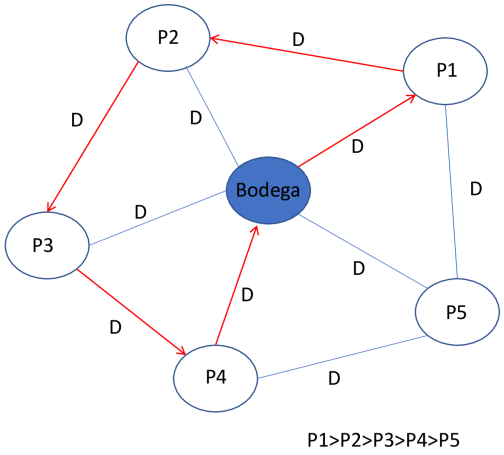

Figura 4.3: Resultados de la primera prueba, ruta sugerida siguiendo de manera decreciente el aporte a la F.O.

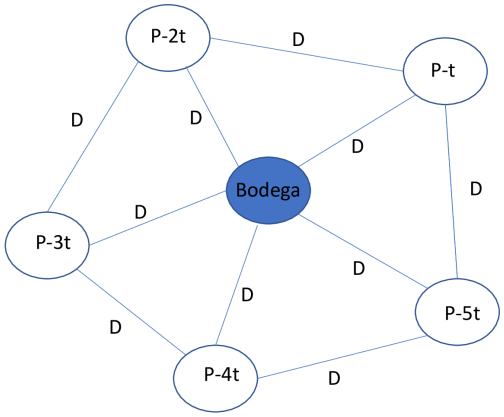

Figura 4.4: Representación grafos segunda prueba simplificado, ya que todos los nodos estan conectados entre sí.

Segunda Situación: Cinco productos a reponer con tiempos de traslado iguales, tiempo de reposición iguales y con recompensas iguales, pero con una diferencia del aporte a la F.O. menor, es decir, viendo una situación real, existen algunos productos que se venden en mayor cantidad que otros. Bajo estas condiciones, el modelo propone ir primero al producto que se vende más en el primer período y seguir este mismo orden, dejando el que tiene menos ventas al principio del día para el final.

Tercera Situación: Seis productos a reponer, uno se encuentra cerca de la bodega y otro lejos de la misma. Los otros cuatro se encuentran a la misma distancia entre ellos, tiempos de reposición iguales, mismo aporte y decaimiento de la función objetivo. El producto que está lejos de todo tiene una recompensa más grande que los demás y un decaimiento igual de grande. Bajo estas circunstancias la solución sugiere ir primero al producto que se encuentra más cerca y posteriormente se dirige al producto más lejano, teniendo la misma recompensa de este último producto si hubiera ido de la bodega directamente y, posterior a esto, al modelo le es indiferente a cuál de los 4 restantes ir a reponer.

Con estas pruebas, se puede dar cuenta que los resultados están alineados con el sentido común para alguien que observa el grafo, pero se intentará extenderlo a la realidad dándole alguna noción de capacidad que tiene el mercaderista de reponer los productos en góndola.

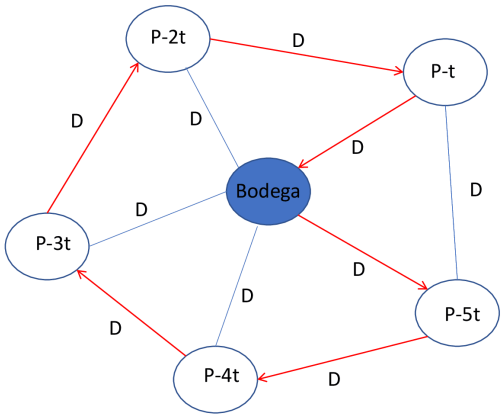

Figura 4.5: Resultados de la segunda prueba.

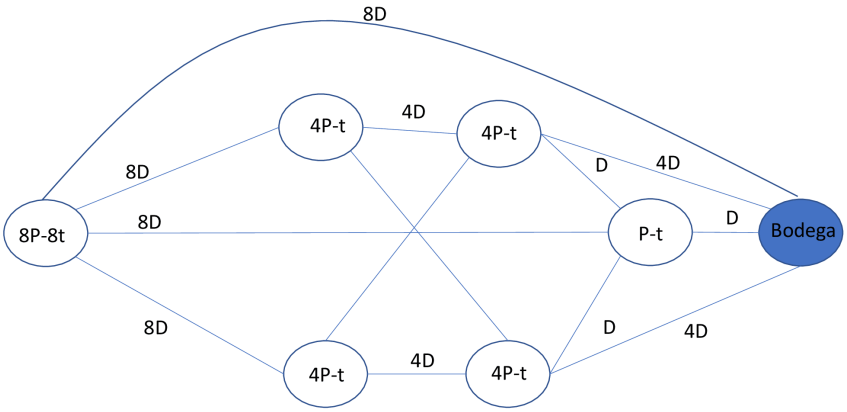

Figura 4.6: Representación grafos primera prueba simplificado, ya que todos los nodos estan conectados entre sí.

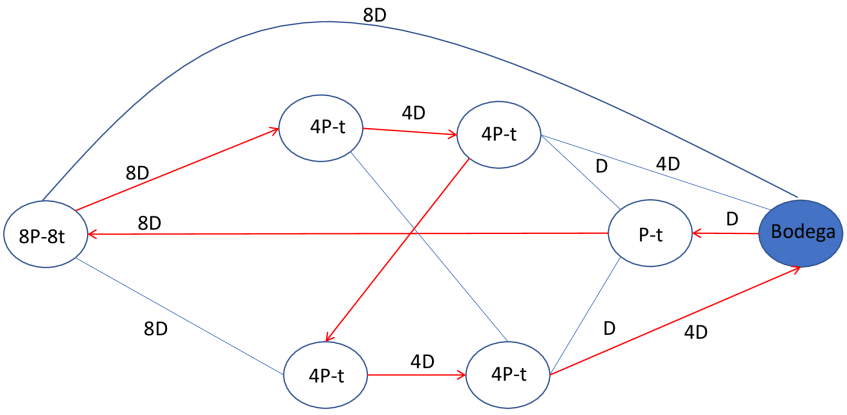

Figura 4.7: Resultados de la tercera prueba.

```
x[bodega,producto0]=1.0
x[producto0,producto1]=1.0
x[producto1,producto2]=1.0
x[producto2,producto3]=1.0
x[producto3,producto4]=1.0
x[producto4,bodega]=1.0
y[bodega,2]=1.0
y[producto0,1]=1.0
y[producto1,2]=1.0
y[producto2,3]=1.0
y[producto3,4]=1.0
y[producto4,4]=1.0
alpha[producto0]=8.0
alpha[producto1]=15.999999999999929
alpha[producto2]=24.00000000000012
alpha[producto3]=32.0
alpha[producto4]=40.0
```

Figura 4.8: Ejemplo output modelo.

4.2.2. Segundo Modelo de Optimización:

Una vez hecho el primer modelo, se agregará una noción de capacidad ya que el reponedor no puede hacer su trabajo en una vuelta para dejar todos los productos en las góndolas. Al haber varios productos la persona tiene que ir varias veces a la bodega para abastecerse y seguir dejando los productos en su lugar, es por esto que, en esta iteración del modelo se verá una forma de estimar el tamaño del carro y el tamaño de los productos en volumen, agregando la restricción de que la combinación de los productos que llevará en el carro no puede superar la capacidad de este.

En términos de programación, para que el modelo reconozca que va pasando tiempo en cada ida a la bodega y no ocurren de manera simultánea, se crean varías bodegas que cumplen la misma función. En otras palabras, en el output del modelo dirá que pasa a bodega 1, bodega 2, etc., pero, en realidad, la bodega es la misma. De esta misma manera se agregan conjuntos donde se explicita que no considera la primera bodega, que es de donde inicia la ruta.

Conjuntos:

- Bloques: Bloques Horarios en el cual definimos. B.
- Productos: N.
- Productos con bodega: Conjunto de productos y la primera bodega N_0.
- Productos sin primera bodega: $Nbod$
- Bodegas: Bod_0.
- Bodegas sin la primera: Bod.

Parámetros:

- R_{ib}: Recompensa por reponer producto i en el bloque b.
- A_i: Tiempo de Reposición del producto i.
- $TIME_{ij}$: Tiempo de ir de i a j.
- END: Tiempo maximo de permanencia.
- $[S_b, E_b]$: Ventana de tiempo del bloque b.
- cap: Capacidad del carro.
- q_i: Cantidad a reponer del producto i.

Variables de Decisión:

$$\mathrm{X}_{ij} = \begin{cases} 1 & \text{Si usa el camino del producto } i \text{ a } j. \\ 0 & \text{Si no.} \end{cases}$$

$$\mathrm{Y}_{ib} = \begin{cases} 1 & \text{si se repone el producto } i \text{ en el bloque } b. \\ 0 & \text{Si no.} \end{cases}$$

α_i = Tiempo en que sale del producto i.

u_i = Volumen para reponer el producto i más el volumen de todos los productos que fueron visitados anteriormente, desde la ultima vez que se fue a la bodega.

Función objetivo:

máx $\sum_i \sum_b Y_{ib} * R_{ib}$

Restricciones:

1) Para salir de un producto tiene que haber entrado:
   $\sum_{j \neq i, j \in No} X_{ji} = \sum_{j \neq i, j \in No} X_{ij}, \forall i \in No$
2) De un mismo producto solo se puede salir 1 vez:
   $\sum_{j \neq i, j \in No} X_{ij} \leq 1, \forall i \in No$
3) No es necesario reponer por todos los productos:
   $\sum_b Y_{ib} \leq 1, \quad \forall i \in Nbod$
4) Para ser repuesto tiene que haber llegado el reponedor:
   $\sum_b Y_{ib} = \sum_{j \neq i, j \in No} X_{ij}; \ \forall i \in No$
5) No puede irse de j antes de haber terminado de reponer:
   $\alpha_i + TIME_{ij} + A_j \leq \alpha_j + M(1 - X_{ij}), \ \forall i \in No, \ \forall j \in Nbod$
6) Tiene que haber terminado de reponer dentro de la ventana:
   $(\mathrm{S}_b) \cdot Y_{ib} \leq \alpha_i \leq E_b + M \cdot (1 - Y_{ib}), \ \forall i \in Nbod, \ \forall b \in B$
7) Terminar dentro de la jornada:
   $0 \leq \alpha_i \leq END, \ \forall i \in No$
8) No se puede superar la capacidad del carro :
   $\mathrm{u}_i \leq CAP, \ \forall i \in N$
9) No puede llevar menos de lo que necesita reponer:
   $\mathrm{q}_i \leq u_i, \ \forall i \in N$

10) Unidades del carro deben ser suficientes para reponer el producto i y todo los productos visitados anteriormente desde la última vez que se visitó la bodega:
$\mathrm{u}_j \leq X_{ij} \cdot M,\ \forall i \in N\ , \forall j \in N, M \gg 0$
$\mathrm{u}_j - u_i \leq q_i,\ \forall i \in N\ , \forall j \in N$
$\mathrm{u}_j - u_i \geq q_i - (1 - X_{ij}) \cdot M,\ \forall i \in N\ , \forall j \in N, M \gg 0$

11) No se puede realizar viajes entre bodegas:
$\mathrm{X}_{ij} = 0\ ,\ \forall i \in Bod,\ \forall j \in Bod$

12) Naturaleza de las variables:

(a) $X_{ij} \in \{0, 1\}$

(b) $Y_{ib} \in \{0, 1\}$

(c) $\alpha_i \geq 0$

(d) $u_i \geq 0$

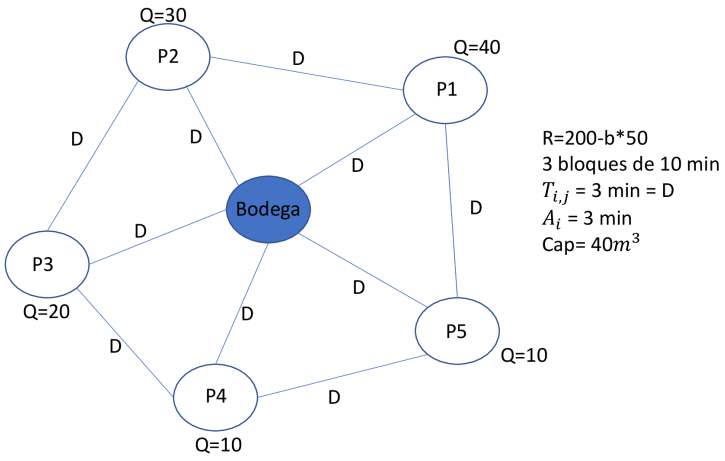

Figura 4.9: Representación grafos primera prueba con capacidad del carro, simplificado ya que todos los nodos estan conectados entre sí.

Primera Prueba: Cinco productos con tiempos de traslados entre si iguales ($T_{ij} = 3$), 3 bloques de 10 minutos, tiempos de reposición iguales ($A_i = 3$), y aporte a la funcion objetivo igual en cada producto ($R = 200 - 50b$), capacidad del carro 40, y volumen de los productos: $q_1 = 40$, $q_2 = 30$, $q_3 = 20$, $q_4 = 10$, $q_5 = 5$. Bajo estas condiciones el modelo realiza el recorrido a los productos que usan menos espacio en el carro ($q_4 \rightarrow$q$_3 \rightarrow$q$_5$) para posteriormente ir a cualquiera de los 2 restantes, ya que tienen todos los parametros iguales y le queda tiempo solo para ir a uno de los 2.

Segunda Prueba: Para esta prueba, se realizó exactamente la misma situación que la primera prueba, pero dejando un producto con un poco más de aporte a la función objetivo ($R_1 = 201 - 50b$), por lo que al correr el modelo de optimización, hizo exactamente lo mismo que la primera prueba, pero esta vez el último producto que visita es el que tiene más aporte a la F.O.

Tercera Prueba: es igual a la segunda prueba, pero se agregó un bloque más, de 10 minutos, por lo que agrega el producto que antes no reponía.

Una vez realizado un modelo más cercano a la realidad y observando el comportamiento de este en productos pequeños el siguiente paso será poner en marcha en un lugar real. Para esto se tiene que realizar la estimación de todos los parámetros y observando datos importantes a la hora de realizar una optimización como, por ejemplo, el tiempo que se demora.

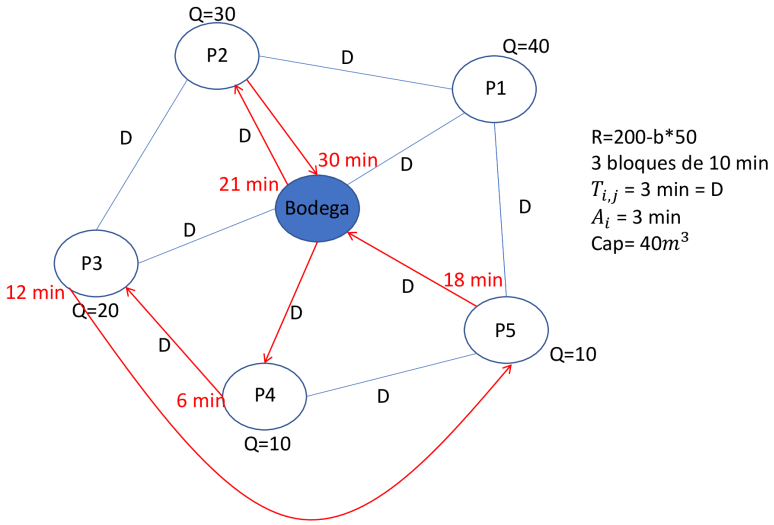

Figura 4.10: Resultados de la primera prueba con capacidad del carro.

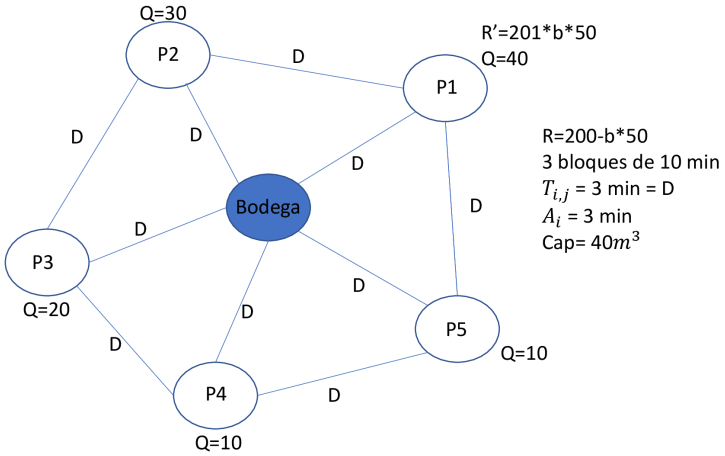

Figura 4.11: Representación grafos segunda prueba con capacidad del carro, simplificado ya que todos los nodos estan conectados entre sí.

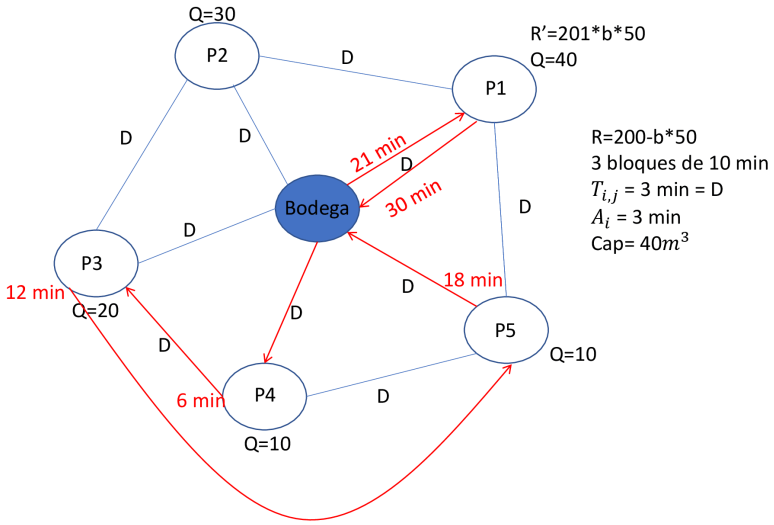

Figura 4.12: Resultados de la segunda prueba con capacidad del carro.

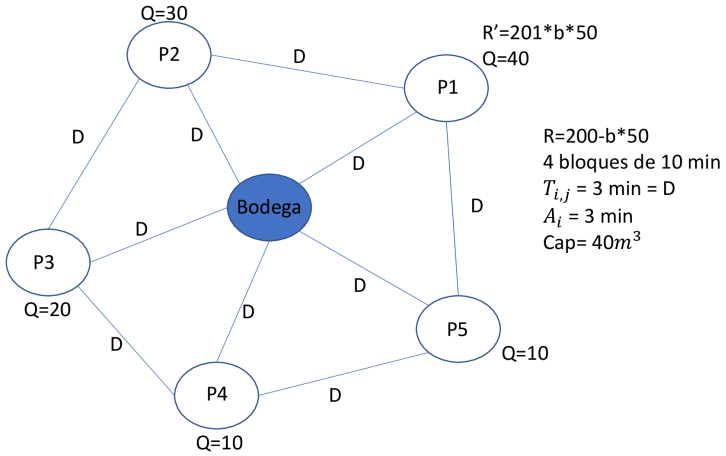

Figura 4.13: Representación grafos tercera prueba con capacidad del carro, simplificado ya que todos los nodos estan conectados entre sí.

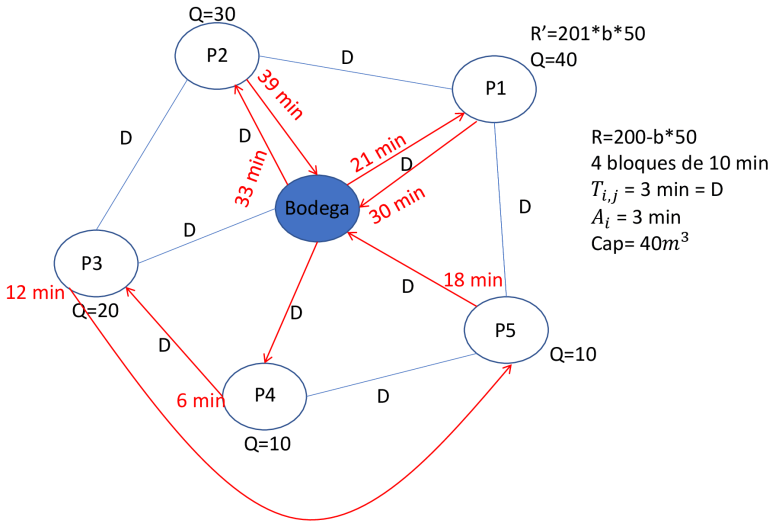

Figura 4.14: Resultados de la tercera prueba con capacidad del carro.

Capítulo 5

Análisis y puesta en marcha

5.1. Estimación de Parámetros

Luego de haber formulado los modelos y hacer pruebas de menor escala, se procede a usar el modelo con datos reales en el supermercado Santa Isabel de Grecia, ya que de los supermercados que Zippedi se encargaba de reponer era el que presentaba más productos y observar al reponedor realizando su trabajo.

De esta manera se procede a estimar los parámetros necesarios. En primer lugar, los **tiempos de viaje** se estimaron midiendo el tiempo que se demora el reponedor en visitar cada pasillo del supermercado. Con el fin de corroborar la veracidad del tiempo medido, se realiza, además, el mismo procedimiento con otras personas. Se observó que en el trayecto no existe una diferencia notable entre la caminata de una persona con el carro que sin él, por lo que se puede hacer una estimación con una persona caminando a una velocidad normal, realizando el recorrido varias veces. De esta manera se tiene una estimación de lo que se demora en ir de un pasillo a cualquier otro. Cabe mencionar que, por simplicidad, se usa un valor constante del tiempo de traslado en ir de un pasillo al otro, ya que dentro del pasillo la demora de ir de un producto a otro es muy pequeña y no constituye objeto de interés para el modelo final.

Luego para los **Tiempos de Reposición**, se tomaron de la misma manera que para los tiempos de viaje. Es decir, midiendo a través de un cronómetro el tiempo que se demora en reponer los productos ya estando en el lugar, este tiempo no tenía mucha diferencia entre los distintos productos, excepto en ciertos productos que el supermercado da la instrucción de ponerle alarma para evitar robo, como, por ejemplo, aquellos productos de perfumería, llegando a demorarse hasta 4 veces más que el resto. Cabe destacar que, además de este tiempo, se decidió dar un tiempo extra, puesto que, además de reponer, el mercaderista también tiene que ordenar el pasillo en los que está encargado, y esta tarea la realiza mientras va reponiendo los productos del pasillo.

En el caso de la **Capacidad**, se realizó midiendo el carro versus las medidas de las cajas de los productos, por lo que uno de los supuestos, para esta forma de estimación, es que cada vez que el reponedor hace su labor, este lleva la caja completa del producto que tiene

que dejar en góndola, se asumió que de esta manera se está subestimando la cantidad de productos que caben en el carro ya que, cuando el supuesto es incorrecto, la persona deja los productos sin empaque en el carro por lo que debería usar menos espacio.

En términos de la bodega, se decidió tener un tiempo fijo en donde se abastece de los productos a la vuelta, esto debido a que dentro de la bodega existía, muchas veces, un desorden para buscar el producto por lo que este podía variar entre los 25-40 minutos en llenar el carro para ir a las góndolas, por lo que, para este modelo, ir a la bodega le toma 40 minutos indistintamente de la cantidad de productos que este lleve.

Luego de tener todos estos parámetros más los datos respectivos de ventas de la predicción de demanda y una lista de productos que tiene que dejar en góndola durante el día, se está preparado para realizar pruebas reales del modelo.

5.2. Análisis previo y pruebas previas

Para tener un buen resultado a la hora de optimizar, lo primero que se necesita ver es la cantidad de tareas de reposición que la persona hace durante el día, en particular, según datos de Zippedi, era una cantidad que iba cambiando bastante a través de los días, habiendo fechas en la que habían 18 tareas y otros donde habían 60 productos que se reponían, por lo que se tenían que evaluar el tiempo en el que encontraba la solución óptima o con un gap aceptable, ya que mientras más productos se tengan que reponer el modelo será mucho más complejo de resolver.

5.2.1. Primera Prueba Real

De esta manera se realizó una prueba con 70 productos al azar, pero al manejar el resultado alcanzaba un gap de 107 % a los 12 minutos del correr, es decir, la solución no era muy buena con ese tamaño. Además, a veces en la ruta se cambiaba de pasillo dentro de la misma ventana, es decir, visitaba un pasillo, se cambiaba de pasillo y después volvía al primer pasillo en la misma hora, lo cual al parecer no tenía mucho sentido aparente ya que tenía exactamente el mismo aporte a la Función Objetivo, por lo que, se tuvo que realizar un ajuste al modelo para que penalice de cierta manera estas situaciones.

De esta manera, se decidió hacer un cambio a la Función Objetivo creando un nuevo parámetro, definido como:

$$\mathrm{D}_{ij} = \begin{cases} 1 & \text{si } i \text{ y } j \text{ estan en diferente pasillo} \\ 0 & \text{Si no} \end{cases}$$

Así se puede recoger las veces en que ir de cierto producto a otro cambia de pasillo y poder penalizar la F.O de la siguiente manera:

$$\text{máx} \sum_i \sum_b Y_{ib} * R_{ib} - \lambda \sum_i \sum_j X_{ij} * D_{ij}$$

En donde también cuenta con un ponderador λ el cual puede tomar distintos valores de-

pendiendo de cuanto se le quiera penalizar, desde aquí en adelante se harán pruebas distintas con datos reales, y se fijará en cosas importantes tanto del modelo como logísticamente, como lo son: gap, tiempo que alcanza dicha solución, cantidad de tareas/productos a reponer, cambios del pasillo en el proceso, etc.

Cabe mencionar que para comparar la F.O con los distintos λ, se toma la ruta resultante de ese parámetro y se evalúa con la F.O original (sin penalización), además de esto, a cada una de las situaciones se agregó un límite de tiempo igual, pero Gurobi cuando no encontraba una buena solución en el tiempo límite, este hacia otra iteración con el doble de tiempo.

5.2.2. Evaluación con parámetros reales, 70 productos

Para esta prueba se eligieron 70 productos al azar de la lista, el reponedor tiene que ir a un total de 6 pasillos para completar todas esas tareas, en los siguientes puntos se evaluarán escenarios de cada λ, por motivos de simplicidad se decidió dejar al modelo un tiempo de 6 minutos para encontrar la mejor solución, pero por temas del programa a veces hacía un iteración completa llegando al doble de tiempo en ciertos casos.

λ=0

Para este escenario, el modelo alcanzó un gap de 107 % y encontró dicha solución en 720 segundos (12 minutos) llegando a reponer solo 14 productos y a una Función Objetivo de 56,475 unidades monetarias, cabe mencionar que considerando ese gap y la cantidad de productos repuestos no es una solución muy recomendada para que el reponedor la lleve a cabo, por lo que este fue uno de los casos en que se decidió cambiar la función objetivo con los distintos parámetros penalizando la misma.

λ= 0.001

Este caso fue el que tuvo mejor desempeño, llegando a un gap de 4.07 % y teniendo una función objetivo de 112,537, pero demorando 12 minutos en realizar el ruteo.

λ =0.1

Para este caso, el modelo logra reponer 69 productos con un gap de 7.6 % y llegando a una Función Objetivo de 108,663.

λ=1

Para esta iteración, se repusieron 32 productos con las mismas condiciones, llegando a un gap de un 39 % con la función objetivo de 84,276, si bien no es una gran solución, es bastante

Tabla 5.1: Resumen prueba con 70 productos a reponer

Penal	Tiempo(s)	% gap	F.O	Reposiciones	Cambios de pasillo	Ratio
0	720	107 %	56,475	13	11	0.84
0.001	720	4.1 %	112,537	69	12	0.17
0.1	360	7.6 %	108,663	69	12	0.17
1	360	38 %	84,176	32	9	0.28
10	360	10.5 %	106,100	68	11	0.16

mejor que la solución anterior sin ninguna penalización y, además, encontró la solución en la mitad del tiempo.

λ= 10

En esta ocasión los resultados son bastantes más positivos que las soluciones anteriores, para este caso el reponedor podría realizar 68 tareas, alcanzando un gap de 10.5 % y llegando a una función objetivo de 106,100, bastante superior a casos anteriores.

5.2.3. Evaluación con parámetros reales, 61 productos

Para esta prueba se eligieron 61 productos al azar de la lista, tratando de observar cómo se comporta λ cuando cambia su valor, además ver si este mismo hace que la búsqueda de una solución sea eficiente y efectivo.

λ=0

Para este escenario, el modelo alcanzó un gap de 6 % llegando a esa solución en 2 minutos y alcanzando una función objetivo de 170,311 dejando afuera un producto y realizando 40 cambios de pasillo durante todo el proceso, así el ratio entre $CambiosPasillo/producto$ es de 0.66, es decir, por cada producto repuesto, se cambió de pasillo 0.66 veces.

λ=0.001

Para este caso, el modelo logra reponer todos productos con un gap de 7.3 % y llegando función objetivo de 168,030 con 17 cambios de pasillo, llegando a un ratio de 0.28.

λ=0.1

Para este caso, el modelo logra reponer todos productos con un gap de 4.34 % y llegando función objetivo ganadora de 172,797 con 17 cambios de pasillo, llegando a un ratio de 0.28.

Tabla 5.2: Resumen prueba con 61 productos a reponer

Penal	Tiempo(s)	% gap	F.O	Reposiciones	Cambios de pasillo	Ratio
0	120	6 %	170,311	60	40	0.67
0.001	60	7.3 %	168,030	61	17	0.28
0.1	60	4.34 %	172,797	61	17	0.28
1	60	9 %	165,715	61	13	0.21
10	60	13.5 %	158,953	61	19	0.31

λ=1

Para esta iteración, se repuso todos los productos, alcanzando la solución en un minuto con una Función Objetivo más baja que la original, 165,715, y con un gap de 9 % y realizando solo 13 cambios de pasillo con un ratio de un 0.21

λ=10

En esta situación, el modelo llego a una F.O. de 158,953 en un minuto reponiendo todos los productos con un gap de un 13.5 % por lo que le costó más llegar a una buena solución que los casos anteriores, además teniendo 19 cambios de pasillo, considerando que debería ser el que tenga menos cambios por la alta penalización, el ratio $\frac{CambiosPasillo}{producto}$ de un 0.31.

5.2.4. Evaluación con parámetros reales, 30 productos

Para esta prueba se eligieron 30 productos de la lista de un día real según la base de Zippedi, en particular se trata de los productos marcados el 25 de abril de este año, considerando que son bastantes menos productos que las pruebas anteriores, se dejará igual que la prueba anterior, un máximo de un minuto para encontrar la mejor solución esperando que esta vez encuentre el óptimo.

λ=0

Para este escenario, el modelo alcanzó un gap de 0.04 % llegando a esa solución en 2 minutos y alcanzando una función objetivo de 101,826 reponiendo los 30 productos con 12 cambios de pasillo durante el proceso, así el ratio entre $CambiosPasillo/producto$ es de 0.4.

λ= 0.001

Para este caso, el modelo logra reponer todos productos con un gap de 0 % (óptimo) y llegando función objetivo igual que lamba 1 y 10, pero en solamente 0.04 segundos (básicamente instantánea) con 4 cambios de pasillo, llegando a un ratio de 0.13.

Tabla 5.3: Resumen 30 productos (día real)

Penal	**Tiempo(s)**	**% gap**	**F.O**	**Reposiciones**	**Cambios de pasillo**	**Ratio**
0	120	0.04 %	101,826	30	12	0.4
0.001	0.04	0 %	101,870	30	4	0.13
0.1	11	0.007 %	101,863	30	3	0.1
1	15	0.003 %	101,870	30	3	0.1
10	60	0.03 %	101,870	30	2	0.06

λ=0.1

Para este caso, el modelo logra reponer todos productos con un gap de 0.007 % y llegando función objetivo un poco menor 101,863 con 3 cambios de pasillo, llegando a un ratio de 0.1.

λ=1

Para esta iteración, se repuso todos los productos, alcanzando la solución en 15 segundos con una funcion objetivo mejor que la original de 101,870 y con un gap de 0.003 % y realizando solo 3 cambios de pasillo, y el ratio llego a un 0.1.

λ=10

En esta situación, el modelo llego a una F.O de 101,870, misma de la situación anterior, llegando a esa solución en un minuto reponiendo todos los productos con un gap de un 0.03 %, además teniendo 2 cambios de pasillo, llegando un valor con sentido por la penalización, y llegando a un ratio $\frac{CambiosPasillo}{producto}$ de un 0.06.

Luego de revisadas todas las pruebas con parámetros reales, se puede observar que, en general, agregar una penalización a la Función Objetivo ayudó bastante a que las soluciones convergieran más rápido a un buen ruteo. Esto debido posiblemente a que si se mira una sola ventana, sin la penalización, le es indiferente ir del producto i al producto j si son de diferente pasillo; en cambio, con la penalización hace que la recompensa de ir de i a j es menor en la misma ventana. Elección que implica que elegir la ruta en la misma hora sea más fácil, llegando a demorarse menos tiempo y además con un F.O mayor al caso sin penalización, y, en general, se recomienda usar penalizaciones pequeñas por los buenos resultados que en general se mostraron en las pruebas presentadas anteriormente.

5.3. Puesta en Marcha

5.3.1. Antecedentes

Recapitulando, para realizar la prueba en marcha se necesita realizar la predicción de demanda de cada uno de los productos el día anterior ya que se necesita las ventas que hubo el día anterior al cierre de la tienda, de esta manera tenemos la mejor predicción entre Regresion Random Forest y Lasso con CV, todo este periodo de predicción dura aproximadamente 2-3 horas donde solo se necesita agregar los datos de ventas que tiene en su poder Zippedi.

Posterior a esto, durante la mañana del día siguiente, el reponedor marca a través de la aplicación de Zippedi interna llamada 'Bruno' aquellos productos que hace falta reponer, y de esta lista poner en marcha el Modelo de Optimización con una penalización pequeña (como por ejemplo 0.1 o 0.01) tanto como para que le cueste menos encontrar una solución como además que no realice cambios de pasillo innecesarios.

5.3.2. Prueba en terreno

Realizado los pasos anteriores en el Santa Isabel de Grecia, se procedió a correr el Modelo dando una ruta propuesta para el modelo con un total de 26 productos y dando un resultado en un minuto con un gap menor al 1 %, al realizar el reponedor cada paso de ella se observó que los tiempos estimados no estaban tan lejanos a la realidad, además de estos aparecían otras restricciones necesarias específicamente para el supermercado. Como, por ejemplo, que cada vez que la persona tenía que buscar aquellos productos de perfumería debía realizarlos de corrido, es decir, todos los productos en la misma vuelta de la reposición ya que existía una bodega con cerradura para sacar todos los productos de esa categoría que se tenían que pedir a uno de los jefes del mismo supermercado. Esta situación se puede arreglar agregando una penalización distinta a esos productos de la siguiente manera:

$$P_{ij} = \begin{cases} 1 & \text{si al producto } i \text{ es de perfumería y el de } j \text{ no.} \\ 0 & \text{Si no.} \end{cases}$$

$$\text{máx} \sum_i \sum_b Y_{ib} * R_{ib} - \lambda \sum_i \sum_j X_{ij} * D_{ij} - \mu \sum_i \sum_j X_{ij} * P_{ij}$$

Con μ un número mayor a 1 de tal manera de penalizarlo si realiza un camino que no es entre un producto de perfumería-perfumería, pero no tan alto de tal manera que prefiera seguir la ruta en ciertos casos (para que reponga lo más productos que pueda) de esta manera, el modelo tiene resultados parecidos a los anteriormente mencionados y conserva la eficiencia para la solución.

Pero de la misma manera, se presentó otro quiebre al realizar esta prueba. La mayoría de los 27 productos no eran quiebres en góndola, sino que, en su mayoría los artículos tenían stock pero que no tenía la cantidad máxima en los pasillos. Es decir, para el modelo de optimización anteriormente presentado durante este documento, la F.O básicamente busca Maximizar Ganancias ($\text{máx} \sum_i \sum_b Y_{ib} * R_{ib} - Penalizaciones$). En este caso se asume que si el producto no se repone se perdería esta venta, esto porque de los 27 productos no existían

más de 5 ó 6 productos inexistentes en los pasillos, el modelo pierde la utilidad, siendo, por ejemplo, un modelo de ruta mínima mucho más útil para este caso.

Capítulo 6

Conclusión

La reposición es una tarea que realizan todos los supermercados a lo largo del mundo y al parecer, nunca se ha realizado un avance más allá de la experiencia de cada una de las personas que realizan esta tarea. Considerando todas las herramientas de optimización y machine learning que se están desarrollando en el mundo, se cree que se pueden implementar más de algunas herramientas y la presentada en este documento puede ser una de ellas.

La mayoría de los proveedores actualmente envían a sus trabajadores para realizar esta tarea, al tener que realizar la reposición en varios supermercados es improbable estar a primera hora del día reponiendo los productos más vendidos durante la primera parte del día, más aún cuando los supermercados son lejanos unos de otros. Para ello se propone que una persona, por lo menos, de la misma marca de supermercado realice esta tarea para muchas marcas de productos ayuda a ser eficiente todo lo que es viaje entre los locales y, además adelantarse a las ventas, por lo que los ingresos tanto de la marca como a los mismos supermercados deberían subir, al realizar este modelo y mejorándolo en el futuro.

Si bien con los 461 productos encargados de reponer por Zippedi, resultó ser un problema muy pequeño para el modelo por los pocos quiebres en góndola, este mismo modelo puede ser utilizado con otra entidad, por ejemplo, el mismo supermercado ofreciendo el servicio de reposición para aquellos productos que existen en sus bodegas, pero se encuentran con quiebre en góndola. De esta manera, pueden dejar a otras personas o a los mismos proveedores el orden y otras tareas que realizan los mercaderistas.

Para futuras investigaciones: como el lector pudo apreciar en esta tesis, se presentan 2 focos, la primera parte de predicción y la segunda de optimización. Para el caso de una posterior investigación referente a este caso, la predicción podría aplicar otro método de imputación de datos y estudiarlos de otra manera. Es importante mencionar que existen muchos más métodos de predicción que se pueden utilizar en ciertos productos, como por ejemplo medias móviles, Arima, etc. Posiblemente haya algún modelo el cual pueda disminuir los errores a la hora de realizar la predicción de demanda. Otro método distinto al explicado en el documento, no se aplicó porque, por temas de intereses con Zippedi, la predicción que se realizó cumplía con el objetivo que ellos buscaban del modelo y en caso de requerir más variables a tomar en cuenta se dejará para una tarea del futuro.

Por otro lado, para la optimización aún se puede realizar algunas modificaciones, como por ejemplo penalizar proporcional a la distancia entre pasillos, esto podría hacer que se prefiera reponer productos más cercanos cuando la penalización por distancia es mayor, pero habría que tener cuidado en el caso de que el aporte de reponer cierto producto i sea menor que la penalización, en estos casos, el modelo debería preferir no reponerlo.

Para el caso de la optimización, se cree necesario realizar una prueba con un problema con una mayor magnitud de productos a reponer para lograr encontrar algunos elementos necesarios que puedan mejorar el modelo y/o ajustar a dicha situación. Este último paso, que se tenía pensado para este documento, no se pudo realizar por la finalización de la prueba piloto de reposición de Zippedi y la imposibilidad de encontrar posteriormente un supermercado que quiera probar el modelo en los próximos meses.

Lo último que quedó pendiente fue realizar una evaluación económica para implantar el modelo. Si bien funciona, por temas de la finalización de la prueba piloto, no se logró realizar una prueba que cambiara significativamente el beneficio para el supermercado (al haber 5-6 quiebres en todo el supermercado). También se necesitaría datos sensibles del supermercado, como lo es el margen que tienen para vender todos los productos y modelarlos computacionalmente, comparando, por ejemplo, el modelo de ruta mínima vs el que propone esta tesis.

Bibliografía

[1] *Random Forest Regression in Python* , https://www.geeksforgeeks.org/random-forest-regression-in-python/, *[consulta: 05 Diciembre 2021].*

[2] *Time Series Nested Cross-Validation with scikit-learn* , https://www.angioi.com/time-nested-cv-with-sklearn, *[consulta: 08 Diciembre 2021].*

[3] Deniz Aksen and Masoud Shahmanzari. *A Periodic Traveling Politician Problem with Time-Dependent Rewards.* Springer, Berlin, 2016.

[4] Jason Brownlee. *How to Develop LASSO Regression Models in Python,* https://machinelearningmastery.com/lasso-regression-with-python/, *[consulta: 07 Diciembre 2021].*

[5] Dominique Feillet, Pierre Dejax, and Michel Gendreau. *Traveling Salesman Problems with Profits.* Transportation Science, 2005.

[6] Damianos Gavalas, Charalampos Konstantopoulos, Konstantinos Mastakas, Grammati Pantziou, and Nikolaos Vathis. *Heuristics for the Time Dependent Team Orienteering Problem: Application to Tourist Route Planning.* Elsevier, 2015.

[7] Michel Gendreau, Gian paolo Ghiani, and Emanuela Guerriero. *Time-dependent routing problems: A review.* Elsevier, 2015.

[8] Fakhredin Khorasani. *Forward Selection to Find Predictive Variables with Python Code,* https://fakhredin.medium.com/forward-selection-to-find-predictive-variables-with-python-code-3c33f0db2393/, *[consulta: 11 Diciembre 2021].*

[9] Mitchell Krieger. *Time Series Analysis with Facebook Prophet: How it works and How to use it* , https://towardsdatascience.com/time-series-analysis-with-facebook-prophet-how-it-works-and-how-to-use-it-f15ec *[consulta: 10 Diciembre 2021].*

[10] GUROBI OPTIMIZATION. *Gurobi Optimizer Reference Manual* , https://www.gurobi.com/documentation/9.1/refman/index.html, *[consulta: 05 Marzo 2022].*

[11] Fabien Tricoire, Martin Romauch, Karl F. Doerner, and Richard F. Hartla. *Heuristics for the multi-period orienteering problem with multiple time windows.* Elsevier, Austria, 2009.

[12] Sudhir Varma and Richard Simon. *Bias in error estimation when using cross-validation for model selection.* BMC Bioinformatics 7, 2006.

[13] Brian Wheatman, Sertac Karaman, and Daniela Rus. *Tourist Path Optimization Problem.*