Optimización de Ingresos en sistema de bicis compartidas

Optimización con método Simplex en Citi Bike

-Yalidt, José, Yedam y Fernanda

Mayo 2021

Introducción

CitiBike es el sistema de bicicletas compartidas de la ciudad de Nueva York.

Comenzó en 2013 y actualmente es el más grande de Estados Unidos.

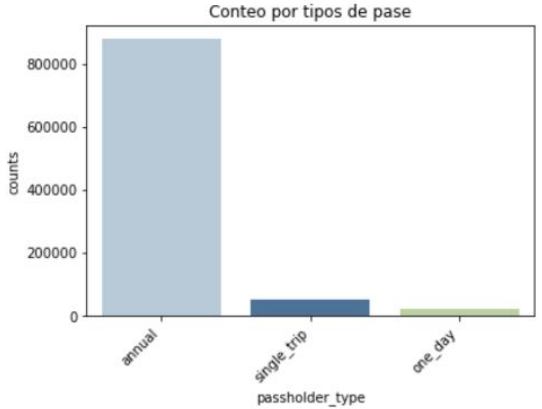
Optimizar los ingresos de la compañía dependiendo del tipo de pases y precios

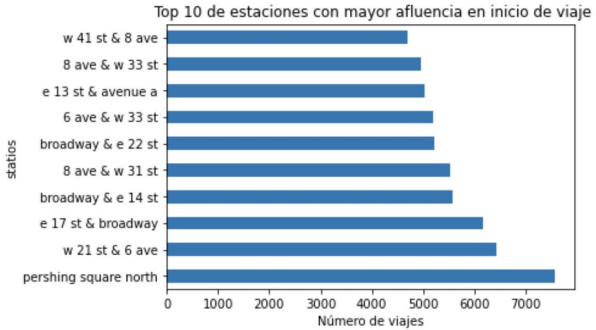
- Disponible las 24 horas, los 365 días del año.
- Estaciones en Manhattan, Brooklyn, Queens y Jersey City .

Objetivos

- Definir esquema óptimo de precios para optimizar ganancias de Citi Bike.
- Implementación del método simplex.
- Perfilamiento del algoritmo.
- Implementación y creación de pipeline vía cómputo en la nube.

Análisis exploratorio





- El pase más común es el anual.
- Las estaciones con mayor afluencia de viajes son Pershing Square North seguido de W 21 St & 5 Ave.
- La base cuenta con **9,518,296 registros y 15 columnas.**
- Se eliminaron unos datos atípicos de usuarios cuyo tiempo de uso de la bicicleta era excesivo.
- La mayoría de los viajes son tipo *one_way.*

Características del problema

Se consideraron las siguientes variables para la modelación del problema:

- *trip_id* - número de viaje
- *duration* - duración del viaje en minutos
- *startime* - fecha de inicio de viaje
- *stoptime* - fecha de término de viaje
- *Type_pass* - tipo de pase: anual, diario o un sólo viaje
- *trip_category* - tipo de viaje: viaje ida y vuelta o un sólo viaje

Características del problema

Type	Costo del pase	Tarifa adicional (15 min)
annually	$180 dlls/año	$1.8 dlls
daily	$15 dlls/día	$2.7 dlls
single	$3.5 dlls/viaje	$2.7 dlls

Cuadro 2. Precios vigentes en 2019

La función de ingresos a maximizar:

$$income = 3{,}5 * z_1 + 2{,}7 * z_2 + 1{,}8 * z_3 + z_4$$

donde:

- z_1 =passes_sold_single
- z_2 =total_15min_blocks_post_free of daily passes
- z_3 =total_15min_blocks_post_free for anual passes
- z_4 =passes_sold * new_pass_prices (fix income)

sujeto a :

$$day\_yes + day\_no = 1$$

$$annual\_yes + annual\_no = 1$$

Método de solución

Programación lineal de enteros

Variables objetivos toman valores binarios de 0 y 1

Decisiones óptimas para **llevar a cabo o no** una actividad

Método simplex

Puede resolver problemas de programación entera, si se cumple que la matriz de restricción A es **totalmente unimodular**

Es decir, cada submatriz cuadrada tiene determinante **0, +1 o -1**

Algoritmo

Para cada paso dentro del método simplex

Dados $\mathcal{B}, \mathcal{N}, x_B = B^{-1}b \geq 0, x_N = 0$

Resolver $B^T v = c_B$ para v

Calcular $\lambda_N = c_N - N^T v$

Si $\lambda \geq 0$ se encontró un punto óptimo, si no:

Seleccionar $nb \in \mathcal{N}$ con $\lambda_{nb} < 0$ como el índice que entra

Resolver $Bd = A_{nb}$ para d.

Si $d \leq 0$ detenerse, el problema es no acotado.

Calcular $x_{nb}^+ = \min\{\frac{x_{B_i}}{d_i} : d_i > 0\}$ y sea ba el índice que minimiza.

Actualizar $x_B^+ = x_B - dx_{nb}^+, x_N^+ = (0, \dots, 0, x_{nb}^+, 0, \dots, 0)^T$

Cambiar $\mathcal{B}$ al añadir nb y remover la variable básica correspondiente a la columna ba de B.

Infraestructura

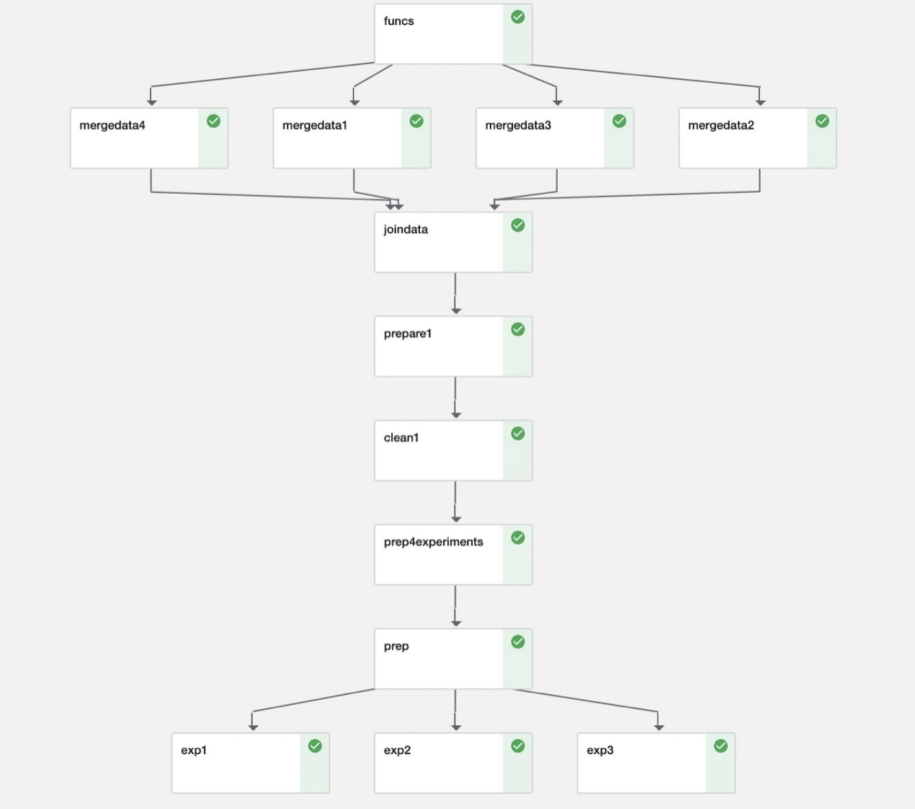
Programación

Desarrollo del código

Ejecución final

Flujo de trabajo

Ejecución del algoritmo



Se realizaron tres experimentos en paralelo:

- 766 seg sin compilación en C
- 764 seg con compilación en C

Infraestructura en *AWS*, utilizando *Kale* y *Minikube* para el pipeline.

AMI opt2-aws-educate-17-03-2021 con m5.2xlarge, la cual tiene 8 CPU's

citi bike

Resultados

Para el periodo examinado, tenemos los viajes registrados de la siguiente manera

Type	Total trips	Total free blocks	Total 15min blocks post free	Total minutes
annually	8,230,870	9,354,633	1,765,973	108,470,986
daily	872,372	927,538	76,574	30,526,375

Cuadro 1. Viajes registrados de Junio a Diciembre 2019

Resultados

Type	Costo del pase	Tarifa adicional (15 min)
annually	$180 dlls/año	$1.8 dlls
daily	$15 dlls/día	$2.7 dlls
single	$3.5 dlls/viaje	$2.7 dlls

Cuadro 2. Precios vigentes en 2019

Ingresos: 40,709,327

Pase diario: Si
Pase anual: Si

Type	Costo del pase	Tarifa adicional (15 min)
annually	$169 dlls/año	$2.5 dlls
daily	$14.95 dlls/día	$4 dlls
single	$3 dlls/viaje	$2.7 dlls

Cuadro 3. Precios vigentes en 2018

Ingresos: 35,716,840

Pase diario: Si
Pase anual: Si

Type	Costo del pase	Tarifa adicional (15 min)
annually	$180 dlls/año	$1.8 dlls
daily	$5 dlls/día	$2.7 dlls
single	$3.5 dlls/viaje	$2.7 dlls

Cuadro 4. Precios propuestos en 2020

Ingresos: 39,879,308

Pase diario: No
Pase anual: Si

Conclusiones

- De acuerdo a los experimentos evaluados, se maximiza las ganancias ofreciendo ambos pases.
- Se sugiere evaluar la incorporación de un pase mensual dado que los pases actuales se mantendrían dada la evaluación.
- *Kale* y *Minikube* permitieron ejecutar pipelines con seis meses de historia con alrededor de 10 millones de líneas.

.'

- Se mejoró la eficiencia del algoritmo en tiempo de ejecución y se obtuvo el mismo resultado que con paquetes de *Python*.

Referencias

Citi Bike, oficial page: https://www.citibikenyc.com/pricing

Citi Bike, data: https://www.citibikenyc.com/system-data

Algoritmos genéticos: https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3: :text=A

Notas del curso de Temas Selectos de Modelado, ITAM, Moreno Palacios Erick, 2021 https://itam-ds.github.io/analisis-numerico-computo-cientifico/README.html

A Metro Bike Share Data Part 1 — Linear Optimization with PuLP, Qiao Finn https://towardsdatascience.com/la-metro-bike-share-data-part-1-linear-optimization-with-pulp-bc8ed4c85cd2