UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA CIVIL INDUSTRIAL

MODELO DE PREDICCIÓN Y ESTIMACIÓN DE TIEMPOS DE TRASLADO ENTRE DOS PUNTOS UTILIZANDO DATOS DE GPS

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN
GESTIÓN DE OPERACIONES

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERA CIVIL INDUSTRIAL

JAVIERA FERNANDA MORALES BENÍTEZ

PROFESOR GUÍA:
CRISTIÁN CORTÉS CARRILLO

PROFESOR CO-GUÍA:
ÁLVARO ECHEVERRÍA SOLIS

MIEMBROS DE LA COMISIÓN:
FERNANDO ORDOÑEZ PIZARRO
PABLO ANDRÉS REY

Este trabajo ha sido parcialmente financiado por SimpliRoute

SANTIAGO DE CHILE
JULIO 2020

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERA CIVIL INDUSTRIAL Y
MAGÍSTER EN GESTIÓN DE OPERACIONES
POR: JAVIERA FERNANDA MORALES BENÍTEZ
FECHA: JULIO 2020
PROF. GUÍA: CRISTIÁN CORTÉS CARRILLO

MODELO DE PREDICCIÓN Y ESTIMACIÓN DE TIEMPOS DE TRASLADO ENTRE DOS PUNTOS UTILIZANDO DATOS DE GPS

En el mundo de la logística de transporte es importante "la última milla", por lo que existen empresas dedicadas a entregar un servicio de plataforma a clientes encargados de realizar logística de última milla. SimpliRoute es una empresa donde su principal función es el desarrollo de un software que permite gestionar rutas múltiples de despachos de forma sencilla pero inteligente. Entregando rutas óptimas mediante la heurística de VRP (Vehicle Routing Problem).

En la actualidad SimpliRoute busca mejorar el servicio a sus clientes por medio de la mejora de la predicción de los tiempos de traslado entre dos visitas o puntos de entrega. Esto actualmente se calcula realizando un ajuste mediante regresión lineal a los resultados de la matriz de distancia (tiempos) de OSRM (Open Source Routing Machine) para aproximar estos valores a los entregados por la API de Matriz de Distancia de Google Maps, lo que deja espacios de mejora.

Se plantea un modelo de predicción de tiempos de traslado mediante el modelo de machine learning de Random Forest programado en el lenguaje de programación Python, esto utilizando GPS de vehículos registrados en la plataforma de SimpliRoute y de Transantiago para complementar zonas faltantes en el mapa. En primer lugar se calcula la velocidad promedio de movimiento de los vehículos, para luego obtener un algoritmo de cálculo de tiempos históricos de traslado. Con los valores históricos obtenidos se realiza un modelo de entrenamiento de Random Forest que realiza una predicción de los tiempos en base a datos históricos. Dicha predicción se realiza con un 96.88 % de precisión, calculado utilizando la medida de error porcentual MAPE.

Para asegurar que la predicción obtenida sea certera se realiza una comparación con los valores obtenidos de una llamada a la API de Google Maps, obteniendo como resultado que la predicción calculada por el modelo de Random Forest tiene una diferencia de ± 5 minutos con los obtenidos por Google Maps. Con lo que se concluye que el modelo desarrollado en esta tesis es satisfactorio para la obtención de tiempos de traslado futuros para los vehículos de SimpliRoute.

Este trabajo está dedicado a mi familia, a Seba, amigos y a SimpliRoute

Agradecimientos

Gracias papá y mamá por siempre motivarme a estudiar, a ser ordenada con todo en mi vida y nunca limitar mis deseos de continuar aprendiendo. Gracias por siempre estar ahí y ser un apoyo constante en mi vida. Gracias Yumni y Fran por acompañarme siempre, Yumni por darme comida en mis noches de estudio en los primeros años de la U, por estar siempre atenta a cualquier necesidad que tuviera, Fran por hacerme reír y por escucharme cuando tenía ganas de conversar sobre cualquier cosa. Mimi, Sofi, gracias por ser parte de mi vida, las conozco hace tan poco, pero parece toda una vida, las adoro y muchas gracias por sus locuras y largas conversaciones.

Seba, muchísimas gracias por ser tú, por amarme a pesar de todo, por acompañarme en todo este camino, por aguantar todos mis cambios de humor y por estar siempre conmigo a mi lado, gracias por ser un gran apoyo en mi vida, eres el mejor.

Cata, Vanti, Gaspar quiero agradecerles por acompañarme en estos 6 largos años en la U, hicieron de estos años los mejores de mi vida. Por todos los almuerzos juntos, las largas tardes de estudio en plan común, por escuchar quejas de los ramos y de la vida. Los quiero mucho! Fran Lagos, por siempre estar dispuesta a escucharme, por la gran amistad que se formó entre nosotras a pesar del poco tiempo de conocernos. Esteban C, Esteban N, Simón, Pauli por su amistad y compañía en tareas y largas tardes/noches de estudio en la U. A todos los chicos del MGO, Seba, Pipe, Romi, Feña, Gabi, Ingrid, hicieron del magíster una experiencia inolvidable, gracias por las tardes de estudio y juegos en la Salita, por los almuerzos entretenidos y por las millones de risas. Manu, por tu tremenda amistad, por escuchar, por ayudar, por las risas y por ser el mejor amigo que alguien podría desear, gracias por siempre estar presente, por tus grandes consejos y por tu honestidad.

Álvaro, muchas gracias por confiar en mí y por darme una oportunidad para desarrollar mi potencial. Víctor, gracias por ser el gran líder que eres, sin tu apoyo no habría podido terminar nunca esta tesis. Lucas, Gonzalo, Ariel, Gabriel y José, por toda su ayuda en el desarrollo que necesité para terminar esto. Leidy, Gloria, Fabi, Azu, por ser muy buenas amigas y por su compañía en Simpli, las quiero mucho! Y finalmente, gracias a todo SimpliRoute por ser el mejor equipo de la vida y hacer de este el mejor lugar para trabajar.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Introducción

En logística de transporte es importante considerar "la última milla", que corresponde al proceso final de entrega de un pedido al comprador. El concepto de última milla es importante para el cliente o consumidor final, debido a que se relaciona directamente con la satisfacción de cliente, lo que permite aumentar las ventas de la empresa. Existen empresas que se dedican a entregar un servicio online vía software o plataformas para que las empresas encargadas de realizar logística de última milla puedan entregar un buen servicio a sus clientes. Es aquí donde SimpliRoute juega un rol fundamental.

SimpliRoute es una empresa donde su principal función es el desarrollo de un software que permite gestionar rutas múltiples de despachos de forma sencilla pero inteligente. Permite crear rutas óptimas, hacer seguimiento en tiempo real y acceder a las estadísticas de las visitas programadas. SimpliRoute cuenta con un equipo completo de personas trabajando desde Chile y con partners a lo largo del mundo, teniendo clientes en más de 13 países.

Dado el constante crecimiento de la empresa y a la alta demanda presente, siempre se busca mejorar los servicios entregados a los clientes, dentro de los que se encuentran los tiempos estimados por la plataforma de SimpliRoute. Actualmente se utiliza en la empresa una regresión que ajusta los datos de OSRM[1] a los datos de la API de Google Maps[2]. Económicamente es infactible para la empresa utilizar solo la API de Google Maps, por lo que han debido desarrollar un método para mejorar la estimación de los tiempos de traslado de los vehículos, pero la solución encontrada no es escalable en el tiempo, debido a que se encuentra desarrollada para la Región Metropolitana y es un proceso mecánico y no automático.

En esta tesis se aborda el problema de estimación de tiempos de viaje, a partir de un modelo de estimación y predicción de los tiempos de traslado entre distintos puntos desde información de mediciones de GPS. Dicha estimación se comparará con la entregada por la API de matriz de distancia de Google Maps.

[1]OSRM: Open Source Routing Machine. Motor de enrutamiento para algoritmo de caminos cortos en redes de caminos.

[2]Application Programming Interface. Entrega matrices de distancia entre puntos de GPS, desarrollado por Google.

Capítulo 1

Objetivos y Alcances

1.1. Objetivos

1.1.1. Objetivo General

Esta tesis es parte de la línea de investigación de SimpliRoute en busca de entregar un mejor servicio a los clientes, para lo cual busca mejorar los tiempos de traslado entre visitas entregado por la plataforma a los algoritmos de ruteo de vehículos. El objetivo de esta tesis es formular e implementar un modelo que mejore la estimación de los tiempos de traslado de la plataforma de ruteo, con el objetivo de lograr una estimación comparable a la entregada por Google Maps.

1.1.2. Objetivos Específicos

1. Estudiar la literatura sobre métodos de estimación de tiempos de traslado.

2. Procesar los datos, es decir, limpiar y eliminar outliers, para obtener una base de datos robusta para trabajar con el modelo.

3. Desarrollar un modelo de estimación de tiempos de traslado mediante el uso de señales de GPS, utilizando datos de traslado entre puntos GPS de la API de Google Maps, de buses de Transantiago y de vehículos inscritos en la plataforma de SimpliRoute.

4. Desarrollar un modelo de predicción de los tiempos de traslado, mediante una metodología de Random Forest.

5. Comparar los resultados obtenidos con los entregados por la API de Google Maps y reportar estadísticas de rendimiento del método desarrollado.

1.2. Alcances del Estudio

Es difícil desarrollar modelos que consideren todas las variables existentes en la realidad, por lo que a continuación se describe el alcance y las limitaciones consideradas en el presente trabajo de tesis.

1.2.1. Zona geográfica de estudio e implementación

El estudio y la implementación de la tesis se desarrolla en la Región Metropolitana, debido a que se cuenta con una mayor cantidad de datos que se utilizan para calibrar el modelo, donde estos datos corresponden a matrices de distancia obtenidos de la API de Google Maps, datos de los buses de Transantiago y finalmente, los datos entregados por SimpliRoute de los GPS de los vehículos inscritos en la plataforma.

Se desarrolla un modelo escalable, que pueda ser fácilmente implementado en los distintos países y regiones donde opera la empresa, extender el modelo a otros países es un proyecto futuro que no se considera dentro del alcance de esta tesis.

1.2.2. Variables a considerar

Debido a la gran cantidad de variables que se pueden considerar dentro de un problema de ruteo, en esta tesis se trabaja con las variables de dirección de traslado y densidad de tráfico. No se consideran las condiciones climáticas del día del ruteo y tampoco las eventualidades que ocurran en un día, como marchas o accidentes de tránsito, esto debido a que se realiza una predicción de tráfico en condiciones normales, considerando el tráfico a distintas horas del día y distintos días de la semana.

Cabe destacar que no se consideran las diferentes horas del día que pueden afectar al tráfico, debido a que el modelo desarrollado en esta tesis corresponde a un trabajo preliminar para tener una primera aproximación de los tiempos de traslado, en trabajos futuros se considerará esta variable.

Por otro lado, no se consideran los tipos de calles en el modelo debido a que es compleja la obtención de estos datos y además, compatibilizarlos con los datos con los que se cuenta.

1.3. Estructura de trabajo

Esta tesis está estructurada de la siguiente forma. En primer lugar, en el Capítulo 2, se llevará a cabo una revisión de la literatura sobre los modelos existentes de estimación de tiempos de traslado, utilizando datos de señales de GPS, lo que permite obtener una idea sobre el modelo que se desarrollará en esta tesis.

En el Capítulo 3 se detalla el problema a resolver. Se realiza una descripción de la solicitud de los clientes a la empresa, una descripción del problema que actualmente tiene la empresa, justificando cualitativa y cuantitativamente la decisión de realizar el trabajo desarrollado.

En el Capítulo 4 se presenta el modelo desarrollado junto con los cálculos matemáticos considerados para la construcción del modelo, realizando una descripción general de la creación del modelo para luego detallar cada paso.

En el Capítulo 5, se muestra una descripción de los datos utilizados para evaluar el modelo, los resultados obtenidos y un análisis posterior realizado en base a la calidad de la estimación desarrollada y sus beneficios para la empresa.

Finalmente, se presentan las conclusiones obtenidas a partir del trabajo desarrollado, su escalabilidad y planes futuros para la utilización del modelo por parte de la empresa.

Capítulo 2

Revisión Bibliográfica

En esta tesis se aborda el problema de estimación de tiempos de traslado entre puntos GPS. Se encontraron referencias importantes en este tema y con distintos métodos de resolución del cálculo de estimaciones. En este capítulo se revisa la literatura estudiada para el desarrollo del trabajo de tesis.

Para la estimación de los tiempos de traslado se considera la literatura correspondiente a *trip-based models* y *links*. Además, se considera literatura respecto a Random Forest para la predicción de tiempos, el cálculo de velocidades utilizando datos de GPS y un paper relacionado con *map-matching*[1]. A continuación se presenta la literatura respectiva a los modelos mencionados.

2.1. Modelos basados en viajes y de arcos

En los modelos basados en viajes y de arcos se consideran las distancias y velocidades desde un punto GPS hasta otro, considerando varias mediciones de puntos GPS intermedios. Considerando el promedio de las medidas mencionadas, se calcula el tiempo de traslado desde un punto GPS a otro.

Li et al. (2017) desarrollan un algoritmo que aprende sobre la congestión local considerando patrones de un grupo reducido de caminos con datos históricos. Generan trayectorias que se dividen en viajes. Dada una consulta de predicción del tiempo de viaje, los autores identifican los patrones de congestión actuales alrededor de la ruta de la consulta a partir de trayectorias recientes, luego infieren su tiempo de viaje en un futuro próximo.

Las trayectorias de entrada utilizadas son particionadas en viajes, removiendo brechas largas y datos GPS registrados cuando el vehículo no se mueve en un período largo de tiempo. Luego, se aplica *map-matching* para computar un camino coincidente con el mapa de cada trayectoria. Finalmente, se combinan las predicciones basadas en patrones de coincidencia y

[1]Denota un procedimiento que asigna objetos geográficos a ubicaciones en un mapa digital. https://link.springer.com/referenceworkentry

de datos históricos de tiempos de viaje, para obtener el tiempo de viaje de una trayectoria completa.

Para realizar la predicción utilizando datos históricos se tiene que t_i es el tiempo cuando el viaje predicho ingresa a r_i, que corresponde a la i-ésima descomposición del camino P. Se computa el tiempo de viaje de r_i en t_i, $d_{ti}^H(r_i)$ es la mediana de todos las observaciones de tiempos de viaje en el mismo intervalo de tiempo de un día como t_i. Recursivamente se calcula utilizando la siguiente fórmula.

$$d_{ti}^H(P_i) = d_t^H(P_{i-1}) + d_{ti}^H(r_i) \qquad (1)$$

$$t_{i+1} = t_i + d_{ti}^H(r_i) \qquad (2)$$

El tiempo de viaje de P es $d_t^H(P) = d_t^H(P_m)$

Sanaullah et al.(2016) desarrollan dos métodos para estimar tiempos de viaje a través de arcos obtenidos mediante la coincidencia en el mapa de puntos fijos de GPS, velocidad de los vehículos y la información de conectividad entre redes con foco especial en las frecuencias de la muestra, tasas de penetración de los vehículos y largos de ventanas de tiempo.

Los autores definen el tiempo de viaje en un arco como el tiempo de viaje promedio obtenido de todos los vehículos que viajaron en un arco durante un largo de ventana de tiempo particular. Desarrollan 2 métodos para la obtención de tiempo de viaje.

En el método 1 crean trayectorias de los vehículos considerando los puntos generados con *map-matching* tanto dentro como fuera de los arcos. Un arco se particiona en relación al tiempo total T_{AB}, donde las partes son TT_1(parte inicial) y TT_2(parte final) en las ecuaciones (3) y (4) se sintetiza la forma en que se calculan los tiempos de viaje, el tiempo de viaje de la partición del medio ($\triangle t_{i+2,i+1}$) fue obtenido de las diferencias en el tiempo t_{i+2} y t_{i+1}, d_i corresponde a la distancia en el punto i, como se muestra en la ecuación (5).

$$TT_1 = \frac{(t_{i+1} - t_i)d_{i+1}}{(d_i + d_{i+1})} \qquad (3)$$

$$TT_2 = \frac{(t_{i+3} - t_{i+2})d_{i+3}}{(d_{i+3} + d_{i+4})} \qquad (4)$$

$$TT_{AB} = \triangle t_{i+2,i+1} + \sum_{i=1}^{2} TT_i \qquad (5)$$

En el método 2, los autores asumen que el tiempo de viaje en un arco entre arcos adyacentes puede estar correlacionado entre sí durante un período de tiempo determinado. También suponen que el tiempo de viaje de un arco puede estar correlacionado en el tiempo. En este método consideran dos componentes, espacial y temporal.

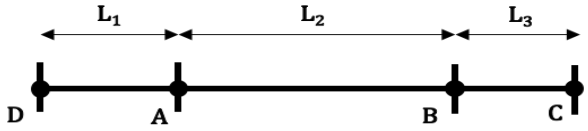


Figura 2.1: Tiempo de viaje en un arco usando la conectividad espacio-temporal.

Considerando 3 arcos distintos, DA, AB y BC como se muestra en la Figura 2.1, la componente espacial considera que el promedio ponderado del tiempo de viaje en el arco de estos tres arcos proporciona una buena estimación del tiempo de viaje del arco para AB, definido en la ecuación (6).

$$T^s_{AB} = \frac{(T_{DA} \times L_1 + T_{AB} \times L_2 + T_{BC} \times L_3)}{(L_1 + L_2 + L_3)} \qquad (6)$$

Donde L_1, L_2, L_3 corresponden al largo de los arcos DA, AB, BC, respectivamente y s representa al tiempo promedio de la componente espacial.

La componente temporal corresponde al tiempo promedio de viaje de un arco para AB sobre n-ésimas ventanas de tiempo consecutivas. En la ecuación (7) se describe el cálculo.

$$T^t_{AB} = \frac{1}{n} \sum_{k=1}^{n} (T_{AB})_{t-k} \qquad (7)$$

Donde n es el número total de ventanas de tiempo consideradas.

El tiempo de viaje total en el arco se obtiene del promedio ponderado de las componentes espacial y temporal como sale en la ecuación (8)

$$TT_{AB} = \alpha(T^s_{AB}) + \beta(T^t_{AB}) \qquad (8)$$

Donde $\alpha + \beta = 1$, $0 < \alpha < 1$, $0 < \beta < 1$.

Forum et al. (2006) usan un modelo basado en viajes donde proveen estimaciones de tiempo de traslado con un 0.3 % de error. Los autores explicitan que este modelo no considera predicciones de tiempo de traslado y solo dan un estimado aplicable fuera de las horas de mayor tráfico.

Un viaje es definido como una secuencia ordenada de observaciones de GPS. Los viajes corresponden a una lista de tramos transitados junto con marcas de tiempos que indican cuándo se produjo la transición de un segmento de camino al siguiente. Cada observación fue registrada a no más de un número específico de segundos después de la observación previa. Solo se forman viajes si son obtenidos de observaciones del mismo GPS. Se considera que el número máximo de segundos entre cada observación es de 10 segundos.

Los autores definen que el modelo basado en viajes utiliza un estilo desde-hasta, es decir, que en una intersección el tiempo de ir a la derecha, de ir a la izquierda o seguir derecho es diferente. Para cada segmento del camino se calcula el tiempo de traslado a todos sus vecinos. Para realizar el cálculo se utiliza el Algoritmo 1.

Algorithm 1 Trip-Based Algorithm

```
input: A list of road segments, R
input: A list of intervarls, I
begin
    avg = 0 , len = 0 , tt = 0;
     foreach r in R do
        neighbours = getNeighbours(r);
        foreach n in neighbours do
            foreach <interval>in I do
                avg = getAvg(r, n, <interval>);
                 len = getLength(r);
                 tt = (len/avg) ×3,6;
                 store(tt, r, n, < interval >); end
            end
        end
    end
```

En el Algoritmo 2 se sintetiza el cálculo de los tiempos de traslado.

Algorithm 2 Trip-Summation Algorithm

```
input  : A list of road segments, R
input  : An interval, <interval>
output: The total travel time, total
begin
    total = 0;
     for r = 0 ; r / sizeof(R) - 1; r++ do
        from = R[r];
         to = R[r+1];
         tt = getTT(from , to, <interval>);
         total += tt;
    end
end
```

Este modelo es eficiente para el cálculo de tiempos de traslado utilizando una componente espacial.

Woodard et al. (2016) desarrollan un método para predecir la distribución de probabilidad en una ruta en un tiempo arbitrario, los autores utilizan datos de GPS de teléfonos móviles o insertos en los vehículos.

Las mediciones de GPS contienen datos registrados periódicamente de ubicación, velocidad y marcas de tiempo. Se define inicio de un viaje con la primera lectura de GPS y termina con la última lectura de GPS en la secuencia. El tiempo total del viaje es medido con las marcas de tiempo registradas, definido como la diferencia entre la marca inicial y la final.

En primer lugar, se estima la ruta utilizada en cada viaje $i \in I$, donde I corresponde al conjunto de viajes de distintos vehículos. Se define la distancia d_{ik} para cada arco y el tiempo de viaje T_{ik} en cada arco. Donde $k \in \{1, ..., n_i\}$ es un elemento del conjunto de arcos e i un viaje en el conjunto de viajes I, n_i es el n-ésimo arco asociado al viaje i.

El tiempo total de cada arco es calculado como la suma de los tiempos parciales asociados a dicho arco.

Teniendo los valores de T_{ik} se modela T_{ik} como el ratio de varios factores:

$$T_{ik} = \frac{d_{ik}}{E_i S_{ik}} \quad i \in I, k \in \{1, ..., n_i\} \qquad (9)$$

donde E_i y S_{ik} son variables latentes de valor positivo asociadas con el viaje y el par arco-viaje, respectivamente.

E_i captura el hecho de que el viaje i pueda tener un cierto porcentaje de mayor velocidad que el promedio en cada link del viaje, esto podría ocurrir por las condiciones de tráfico que afectan el viaje. Esto representa la variabilidad del viaje captada por la velocidad. E_i se modela con una distribución log-normal, para un parámetro de varianza desconocido τ^2.

$$log(E_i) \sim N(0, \tau^2) \qquad (10)$$

S_{ik} representa la velocidad del vehículo en el link antes del efecto causado por E_i y captura la variabilidad en la velocidad debido a condiciones locales, como por ejemplo, situaciones locales de tráfico.

Los autores modelan S_{ik} en términos del estado de congestión discreta no observada $Q_{ik} \in \{1, ..., Q\}$. Q representa el estado de congestión del viaje i en el arco k Se modela, al igual que E_i con una distribución log-normal.

$$log(S_{ik})|Q_{ik} \sim N(\mu_{R_{ik}b_{ik}Q_{ik}}, \sigma^2_{R_{ik}b_{ik}Q_{ik}}) \qquad (11)$$

donde $\mu_{j,b,q}$ y $\sigma_{j,b,q}$ son parámetros desconocidos asociados a la velocidad de viaje en el arco j bajo las condiciones de congestión $q \in Q$ en el intervalo de tiempo b.

Este modelo considera la variabilidad del viaje, como los efectos del conductor y la variabilidad del arco que puede ser asociada a factores externos de tráfico, por ejemplo construcciones.

Hunter et al. (2009) presentan un algoritmo de maximización de expectativas que aprende simultáneamente los caminos probables tomados por los vehículos de prueba, así como las distribuciones de tiempo de viaje a través de una red.

Los autores buscan predecir el tiempo que toma un vehículo en recorrer los arcos de la red definida. Para esto consideran una red de caminos como un conjunto de arcos dirigidos. Consideran variables que influencian el cálculo del tiempo de viaje. Son las siguientes, el largo del arco, el número de carriles y la presencia de señales de tránsito. Además, consideran las características del comportamiento del tráfico, las características de los vehículos y condiciones exógenas, tales como las condiciones del clima, eventos deportivos, entre otros.

Cada día es descompuesto en intervalos de tiempo que comparten patrones en común. Las observaciones son muestras de las trayectorias recorridas por los vehículos de prueba que abarcan como máximo unos pocos arcos por observación.

La red es descrita como un grafo dirigido $\mathcal{D} = (\mathcal{L}, \mathcal{E})$, donde $\mathcal{L}$ es el conjunto de arcos en la red y $\mathcal{E}$ corresponde a los vértices de la red. Un camino es representado como una lista de arcos.

Se define un conjunto de observaciones R en un intervalo de tiempo. Para cada observación r que pertenece al conjunto R, el tiempo de viaje entre el punto de inicio y final es definido como d^r. Una observación también está asociada a una trayectoria particular entre los arcos $l_1^{(r)}, l_2^{(r)}, ..., l_{N_r}^{(r)}$ de la red, donde N_r es la cantidad de arcos por cada observación. Se puede definir la trayectoria de un vehículo como el vector w de tamaño L.

Con lo descrito anteriormente, el tiempo de viaje D^r es la variable aleatoria correspondiente a la suma de los tiempos de viaje en cada arco encontrados a lo largo del tiempo.

$$D^r = X_{l_1}^{(r)} + X_{l_2}^{(r)} + ... + X_{l_{N_r}}^{(r)} \qquad (12)$$

Donde X^r corresponde a los tiempos de viaje de cada arco. Usando el vector w, el tiempo de viaje del camino completo puede ser expresado en la forma vectorial $D^r = (X^T)w^r$, con T representando al total de observaciones.

Jiménez-Meza et al. (2013) proponen un marco de referencia donde el GPS registra 3 valores, Latitud, Longitud y Marca de tiempo. Con dichos valores proponen un cálculo simple para determinar el tiempo de traslado, la distancia y la velocidad del vehículo. Utilizan una segmentación de las calles con los valores del GPS.

Para calcular la distancia entre dos puntos requieren funciones de geometría esférica y trigonometría. Para calcular la distancia entre dos latitudes y longitudes consecutivas, definen 2 catetos, A y B y una hipotenusa d.

$$A = 69{,}1 \times (lat2 - lat1) \qquad (13)$$

$$B = 69{,}1 \times (lon2 - lon1) \times \cos(lat1/57{,}3) \qquad (14)$$

Donde cos corresponde a la función coseno de trigonometría. Las constantes 69.1 y 57.3 son utilizadas para transformar las coordenadas a distancia en millas. Con esto la distancia se calcula utilizando la ecuación (15)

$$d = \sqrt{A^2 + B^2} \times 1609{,}344 \qquad (15)$$

Conociendo las marcas de tiempo, se sabe el tiempo inicial y final en dos GPS consecutivos, definidos por ct_1 y ct_2. El tiempo t, en segundos, se calcula utilizando la ecuación (16)

$$t = (ct_2 - ct_1 \times 86400) \qquad (16)$$

La velocidad instantánea entre dos coordenadas de GPS en km/h se determina de la siguiente manera.

$$v = \frac{d}{t} \times 3{,}6 \qquad (17)$$

Luego, los autores definen la división del segmento de largo L como la suma de todas las distancias entre las coordenadas, como se muestra en la ecuación (18). Por otro lado, calculan el tiempo total del segmento, T, como la suma de todos los tiempos calculados previamente, tal como se muestra en la ecuación (19).

$$L = \sum_{i=1}^{n} d \qquad (18)$$

$$T = \sum_{i=1}^{n} t \qquad (19)$$

Utilizando lo calculado con las ecuaciones (18) y (19) se puede obtener la velocidad promedio del segmento, la que corresponde a la siguiente.

$$V = \frac{L}{T} \times 3{,}6 \qquad (20)$$

Wang et al. (2017) proponen un enfoque que establece relaciones entre la confiabilidad de los tiempos de viaje y la densidad de tráfico en carretera para predecir la confiabilidad de futuras condiciones de tráfico.

La confiabilidad de tiempos de viaje representa el nivel de consistencia en tiempos de viaje para un mismo recorrido por un período de tiempo.

Los autores registran datos de señales de GPS en intervalos de 20 segundos. Las condiciones de tráfico son cuantificadas utilizando la densidad de tráfico, que se define como la división entre el volumen de tráfico (cantidad de vehículos en un tramo al mismo tiempo) por la velocidad.

Se concluye de los estudios realizados por los autores que una mayor densidad de la carretera se asocia con una velocidad de desplazamiento más baja y, a veces, una confiabilidad más baja en los tiempos de traslado.

En la tesis de González (2020) se proponen modelos de estimación de tiempos de respuesta de carros de bomberos a partir de simulaciones en el software PARAMICS, considerando distintos escenarios y condiciones iniciales de tráfico. Considera variables que influyen en la estimación de tiempos de viaje, de las que se destacan las siguientes, bloque horario, número de pistas de una calle, tipo de pistas, velocidad de vehículos de bomberos o particulares.

La tesis presenta un modelo de regresión lineal a partir de las variables mencionadas, para capturar el impacto sobre el tiempo acumulado de viaje del vehículo de bomberos. A continuación se describe la regresión lineal implementada.

$$\begin{aligned} Y = \alpha + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_4 \cdot X_4 + \beta_5 \cdot X_5 + \beta_6 \cdot X_6 + \\ \beta_7 \cdot X_7 + \beta_8 \cdot X_8 + \beta_9 \cdot X_9 + \beta_{10} \cdot X_{10} + \beta_{11} \cdot X_{11} + \\ \beta_{13} \cdot X_{13} + \beta_{14} \cdot X_{14} + \beta_{15} \cdot X_{15} + \beta_{16} \cdot X_{16} + \beta_{18} \cdot X_{18} \end{aligned}$$

Donde Y representa la variable dependiente, correspondiente al tiempo acumulado. X_1 y X_2 son variables relacionadas al bloque horario. X_3 a X_8 representan los distintos números de pistas, desde 1 hasta 6 respectivamente. X_{13} y X_{17} corresponden a los tipos de pista, mixta y solo buses respectivamente. X_{10}, X_{11} y X_{12} representan la cantidad de aceras en la calle, desde 0 a 2 respectivamente. X_9, X_{14}, X_{15}, y X_{16} corresponden a interacciones entre las variables. Finalmente, X_{18} representa la velocidad media espacial de los vehículos.

El autor plantea un segundo modelo correspondiente a una linealización de modelo de regresión lineal (un modelo Log-Lineal), en este modelo se consideran dos nuevas variables, distancia total de ruta y velocidad de vehículos particulares. Se sintetiza el modelo en la siguiente regresión.

$$\begin{aligned} \ln(\mathrm{Y}) = \alpha + \beta_2 \cdot \mathrm{Z}_2 + \beta_3 \cdot \mathrm{Z}_3 + \beta_9 \cdot \mathrm{X}_9 + \beta_{11} \cdot \mathrm{X}_{11} + \beta_{13} \cdot \mathrm{X}_{13} + \beta_{15} \cdot \mathrm{X}_{15} \\ + \beta_{16} \cdot \mathrm{X}_{16} + \beta_{17} \cdot \mathrm{X}_{17} + \beta_{19} \cdot \mathrm{X}_{19} + \beta_{20} \cdot \mathrm{X}_{20} \end{aligned}$$

Donde, X_{19} es la distancia total de la ruta y X_{20} la velocidad media de los vehículos particulares medida en metros por segundo. Por otro lado, Z_2 corresponde a la normalización de la cantidad de arcos con 3 y 4 pistas y Z_3 a la normalización de la cantidad de arcos con 5 y 6 pistas. Las otras variables se mantienen.

Del trabajo realizado por González (2020) se concluye que es relevante para modelar los tiempos de viaje del carro de bomberos respecto a las velocidades de los vehículos particulares, por otro lado, la variable de distancia total hasta la emergencia es un elemento importante por considerar.

En la tesis de Vejar (2019) se propone una nueva metodología que entrega una mejor predicción de velocidades a corto plazo. Para desarrollar dicha predicción se estudian distin-

tos modelos de *machine learning*, estos correspondientes a *Random Forest*, *Support Vector Regressor*, *Artificial Neural Network* y *Long Short Term Memory Recurrent Neural Network*. Se utiliza cada modelo para realizar una predicción de velocidades, realizando modificaciones en las arquitecturas planteadas, en donde se aprovecha conocimiento sobre fenómenos del tráfico aplicado sobre los modelos.

La mayor contribución de la tesis de Vejar (2019) es la introducción de una metodología alternativa para la predicción de velocidades en corto plazo, obteniendo mejores resultados que modelos recurrentemente utilizados en la literatura. Por otro lado, se realiza una diferenciación de la predicción de las velocidades en circunstancias en las que no hay congestión con situaciones de congestión.

2.2. Random Forest

Cheng et al. (2019) desarrollan un modelo de Random Forest utilizando datos obtenidos con softwares de simulación, considerando la distribución de las velocidades, tamaño o proporción del vehículo y un detector de tiempo. Realiza estimaciones cada 300 segundos y considera 6 niveles distintos de condiciones de tráfico.

Random Forest es un modelo combinado que consiste en un conjunto de árboles de decisión de regresiones. La siguiente ecuación muestra la definición de Random Forests.

$$\{h(x, \theta_t), t = 1, 2, ..., T\}$$

donde $h(x, \theta_t)$ es un clasificador estructurado en árbol y $\{\theta_t\}$ son vectores aleatorios independientes idénticamente distribuidos. T representa el número de árboles de decisión.

El modelo de Random Forest combina muchos árboles de decisión para evitar la sobreestimación de los datos. Todos estos de computan de manera simultánea sin tener interacción entre ellos, cada uno entrega un resultado, lo que utilizando la forma regresiva de un Random Forest (*Random Forest Regressor*) se obtiene el resultado de la predicción como el promedio de los resultados de los árboles de decisión, como se muestra en la Figura 2.2.

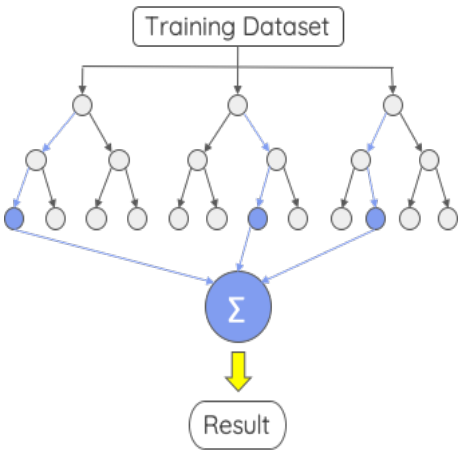


Figura 2.2: Diagrama explicativo Random Forest.

La base de entrenamiento corresponde a un porcentaje de los datos totales, lo que asegura que no depende de características individuales, este modelo utiliza todas las potenciales características predictivas.

Cada árbol utiliza una muestra aleatoria del conjunto de datos original cuando genera cada división, lo que agrega mayor aleatoriedad para evitar la sobreestimación. Esto previene que los árboles se encuentren correlacionados entre ellos[2].

Las ventajas de este modelo de predicción son las siguientes:

- Es uno de los algoritmos de aprendizaje más precisos.
- Corre eficientemente en bases de datos grandes.
- Puede soportar miles de variables de entrada.
- Entrega un estimado de qué variable es importante.
- Es efectivo para estimar datos faltantes.

2.3. Cálculo de velocidades

Cortés et al.(2011) utilizan un método de rectificación de rutas para representar las rutas de la manera lo más sencilla posible. El objetivo final es estimar velocidades comerciales de buses en espacio y tiempo.

El método de rectificación de rutas corresponde a identificar rutas como formas que son traducidas a puntos geocodificados. Los autores identifican un número mínimo de puntos requeridos para definir el camino de una ruta con un error ε. Si un punto estuviera más cerca que ε del segmento de línea, cualquier punto que no esté marcado actualmente debería descartarse sin que la curva suavizada sea peor que ε. Si el punto más alejado del segmento de línea era mayor que ε, entonces ese punto debería mantenerse en el conjunto que definió la curva modificada. Esto se realiza recursivamente.

Se realiza una proyección de los puntos sobre el mapa de Google Earth, donde se verifica que el método entrega buenos resultados.

Luego, se define una grilla de tiempo-espacio para cada ruta dada. Cada elemento de la grilla está definido por bordes D y T. Para cada elemento g de la grilla la velocidad se obtiene con la siguiente fórmula.

$$\overline{s}_g = \frac{\sum_i D_i^g}{\sum_i T_i^g}$$

En Weng et al.(2016) se establece un modelo de cálculo de velocidades basado en el tiempo de traslado estimado y la distancia entre paradas de buses. Se asume una dirección de traslado, donde los puntos GPS fueron proyectados en un arco en secuencia, y se estima la proporción de puntos GPS en el arco. Se calcula la velocidad utilizando la distancia y el tiempo entre paradas de buses consecutivas.

[2]Información obtenida de www.towardsdatascience.com

La parada de bus puede ser determinada considerando que esta se encuentra entre dos puntos consecutivos de GPS, tomando en cuenta esto, el tiempo de llegada del bus a la parada puede ser calculado.

El cálculo de la velocidad considera factores espacio-temporales que probablemente afectan la precisión del cálculo del modelo. Entre dichos factores se encuentran los diferentes períodos de tiempo, considerando mañana y tarde, las operaciones de las líneas de los buses, como buses rápidos, que no paran en todos los paraderos, caminos principales, secundarios, entre otros.

2.4. Map-matching

La correspondencia de mapas es un procedimiento que asigna objetos geográficos a ubicaciones en un mapa digital. Los objetos geográficos más típicos son las posiciones de puntos obtenidas de un sistema de posicionamiento, a menudo un receptor GPS.

En usos típicos, las posiciones GPS derivan de un receptor ubicado en un vehículo u otro objeto en movimiento que viaja en una red de carreteras, y el mapa digital modela la incrustación en el espacio geográfico de las carreteras mediante polilíneas que se aproximan a las líneas centrales de las carreteras. Las posiciones de GPS generalmente no se cruzan con las polilíneas, debido a imprecisiones.

El objetivo de la correspondencia de mapas es colocar las posiciones GPS en sus ubicaciones correctas en las polilíneas del mapa.[3]

Chicoisne et al. (2019) genera una heurística para realizar *map-matching*, es decir, proyectar puntos sobre un mapa para generar un grafo de red que tenga arcos desde un punto a otro de señales GPS. Realizan esto para encontrar la ruta más cercana en la red a una trayectoria de puntos de GPS.

Los autores desarrollan un algoritmo eficiente (MOE) para encontrar ese camino y poder detectar la presencia de ciclos y una heurística más rápida pero menos precisa (MMH) incapaz de detectar ciclos.

Se utiliza el algoritmo del camino más corto de Dijkstra para generar subrutinas. El algoritmo 3 muestra la heurística MMH.

[3]https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-39940-9_215

Algorithm 3 MMH Heuristic Outline

Data : $G = (V, E)$, $P = (p_k)_{k\in\{1,...,q\}}$, R, $d \in \mathbb{R}^m$, d_{max}
Result: A map-matching path SP
$w = d$, $p_t = p_1$;
for $k = 1, ..., q-1$ **do**
  $n_k = \lfloor d(p_k, p_{k+1})/d_{max} \rfloor$;
  for $l = 0, ..., n_k$ **do**
    $p_h = p_k + (p_{k+1} - p_k) \times l \times d_{max}/d(p_k, p_{k+1})$;
    for $e : d(p_t, e) \leq R$ **do**
      $w_e = [w_e - d(p_t, p_q)]_+$;
    end
    $S = \{v \in V : d(p_0, v) \leq R\}$;
    $T = \{v \in V : d(p_q, v) \leq R\}$;
    $\bar{G} = \{V \in \{s, t\}, E \cup \{(s, v)_{v\in S}, (v, t)_{v\in T}\}\}$;
    for $v \in S$ **do**
      $w_{(s,v)} = d(p_0, v)$;
    end
    for $v \in T$ **do**
      $w_{(v,t)} = d(p_q, v)$;
    end
    $SP = dikjstra(\bar{G}, w, s, t)$;
  end
end

Se concluye que con lo mencionado en este capítulo, el problema de la estimación y predicción de tiempos de traslado se encuentra ampliamente abordado en la literatura a lo largo del tiempo. Para continuar con este trabajo de tesis se rescatan los puntos más importantes de los modelos mencionados, se destacan los modelos basados en viaje, la implementación de un modelo de Random Forest y la utilización del cálculo de velocidades descrito anteriormente, que dan base al desarrollo del modelo que se presentará en el capítulo 4.

En el siguiente capítulo se presenta el caso de estudio y la problemática presente en la empresa, con el fin de justificar el desarrollo del trabajo de tesis.

Capítulo 3

Caso de estudio

3.1. Descripción del problema

La estimación de tiempos de traslado entre distintos puntos en una ciudad es importante para la planificación logística de una empresa, sobretodo si dicha empresa debe realizar entregas diarias a distintos puntos distribuidos en una ciudad, como Santiago.

SimpliRoute entrega de manera sencilla y fácil de usar la solución de la ruta óptima que debe seguir una empresa logística para realizar despachos o entregas en la ciudad, por medio de un software experimentado que permite personalizar las rutas de la empresa, utilizando una versión de un problema VRP (Vehicle Routing Problem[1]).

SimpliRoute es una empresa con presencia internacional en más de 13 países a lo largo del mundo, con más de 200 clientes de todos los tamaños (donde el tamaño del cliente se mide por la cantidad de vehículos correspondiente a su flota).

Actualmente, la plataforma de SimpliRoute no entrega un valor muy certero de los tiempos de traslado, se realizó una solución que corresponde a un modelo de regresión simple utilizando matrices de distancia obtenidas de OSRM y para luego ajustar este valor a la matriz de distancia obtenida de una consulta a la API de Google Maps, obteniendo nuevos valores de tiempo de traslado entre distintos puntos de la ciudad.

Utilizar Google Maps sería la mejor solución a las necesidades actuales de ruteo, ya que, según el estudio realizado por Wang and Xu (2011) los valores de tiempo de traslado entregados por Google Maps son una buena aproximación de la realidad. Sin embargo, utilizar la API de Google Maps resulta económicamente infactible para la empresa, debido al alto valor de la consulta realizada[2], lo que implica un alto valor para SimpliRoute, debido al gran volumen de consultas que se deben realizar diariamente.

[1]VRP, definición encontrada en la página web de SimpliRoute, https://www.simpliroute.com/post/como-simpliroute-resuelve-el-problema-de-ruteo-de-vehiculos

[2]Valor API Google Maps. Obtenido de https://developers.google.com/maps/documentation/distance-matrix/usage-and-billing

La solución existente aproxima los valores a los entregados por Google Maps en un 80 % de las veces, pero esto no es escalable a todas las zonas donde opera SimpliRoute, debido a que debería realizarse el ajuste para todos los países y regiones realizando nuevas consultas a OSRM y Google Maps, por lo que de nuevo se incurre en un alto costo para la empresa, ya sea económico y de tiempo de trabajo.

3.1.1. Consultas a OSRM

OSRM, Open Source Routing Machine, es un motor de enrutamiento para el algoritmo de caminos cortos en redes de caminos. Está diseñado para ser usado con los datos del proyecto OpenStreetMap[3].

A diferencia de otros servicios de enrutamiento, OSRM utiliza las jerarquías de contracción[4]. Esto da como resultado tiempos de consulta muy rápidos, normalmente por debajo de 1 milisegundo para conjuntos de datos como todo Europa. Por eso, OSRM es un buen candidato para las aplicaciones y sitios de enrutamiento receptivos basados en la web.

SimpliRoute tiene métodos para medir las consultas a OSRM desde Amazon Web Services[5] con gráficos dinámicos que muestran la frecuencia y cantidad de requerimientos realizados por nuestros clientes al momento de realizar sus rutas.

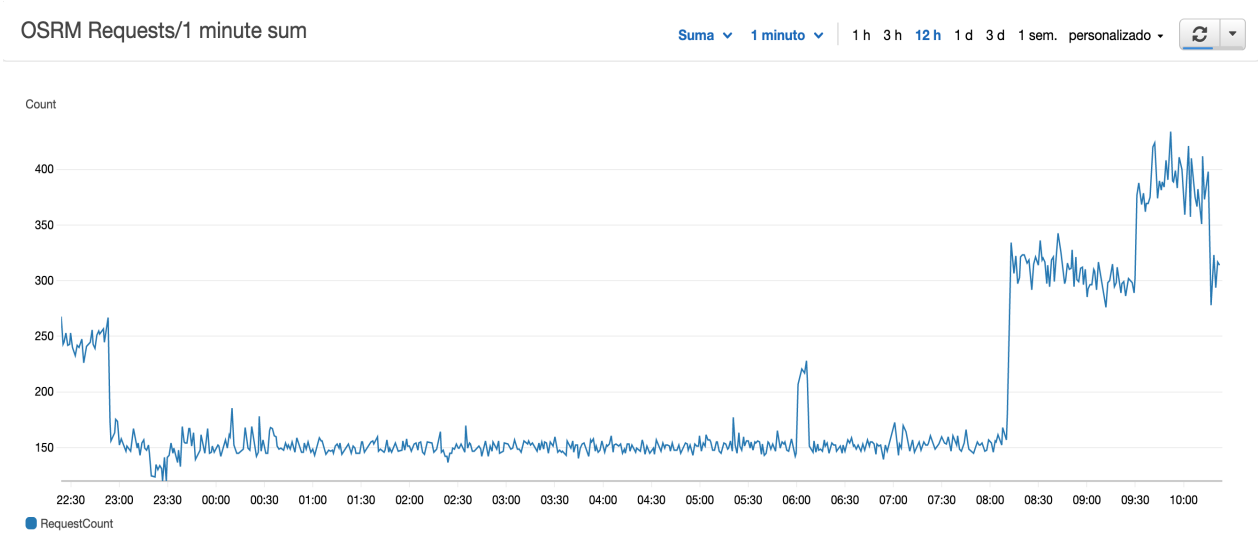


Figura 3.1: Gráfico de suma de requerimientos realizados cada minuto entre las 22:00 hrs y 10:00 hrs

[3]Proyecto colaborativo para crear mapas editables y libres. En lugar del mapa en sí, los datos generados por el proyecto se consideran su salida principal. Los mapas se crean utilizando información geográfica capturada con dispositivos GPS móviles, ortofotografías y otras fuentes libres. www.openstreetmap.org

[4]Técnica de aceleración para encontrar la ruta más corta en un gráfico.

[5]AWS, colección de servicios de computación en la nube pública que en conjunto forman una plataforma de computación en la nube, ofrecidas a través de Internet por Amazon.com

Figura 3.2: Gráfico de suma de requerimientos realizados cada minuto entre las 10:00 hrs y 22:00 hrs

En la figura 3.1 se sintetiza la suma los requerimientos realizados a OSRM cada un segundo en un período de 12 horas entre las 22:00 y 10:00 hrs. Donde el promedio por minuto es aproximadamente 150. Se puede ver que a partir de las 8:00 am este número aumenta, debido a que es la hora donde los clientes de SimpliRoute comienzan a usar el software para rutear.

En la figura 3.2 se muestra la suma los requerimientos realizados a OSRM cada un segundo en un periodo de 12 horas entre las 10:00 y 22:00 hrs. Donde el promedio de request por minuto es aproximadamente 400. Con los gráficos queda claro que las horas donde más se realizan consultas a OSRM corresponden entre las 13:00 y 14:00 horas, donde se realizan alrededor de 1.000 requerimientos por minuto.

3.1.2. Consultas a la API de Google Maps

La API de la matriz de distancia de Google maps utiliza un modelo de pago por uso. En la tabla 3.1 se muestran los valores de facturación por elemento a la API de Matriz de Distancia de Google Maps básica, que no considera las condiciones de tráfico.

Cada consulta realizada a la API de Matriz de distancia genera elementos, donde un elemento corresponde al número de puntos de origen multiplicado por el número de puntos de destino. Por ejemplo, si se requiere generar una matriz de distancia de 100 visitas de un solo cliente, se envía una consulta de 100x100, es decir, 10.000 elementos, lo que entra en la categoría de 0 - 100.000 de la tabla 3.1

Monthly Volume Range (Price per ELEMENT)		
0 - 100,000	100,001 - 500,000	500,000+
0.005 USD per each (5.00 USD per 1000)	0.004 USD per each (4.00 USD per 1000)	Contact Sales for volume pricing

Tabla 3.1: Tabla de precios mensuales por elemento de la API de Matriz de Distancia

Para obtener las información sobre el tráfico y tener una mejor estimación es necesario contratar el servicio de API de Matriz de Distancia Avanzado, el que considera los valores descritos en la tabla 3.2

Monthly Volume Range (Price per ELEMENT)		
0 - 100,000	100,001 - 500,000	500,000+
0.01 USD per each (10.00 USD per 1000)	0.008 USD per each (8.00 USD per 1000)	Contact Sales for volume pricing

Tabla 3.2: Tabla de precios mensuales por elemento de la API de Matriz de Distancia Avanzada

A pesar de que no existe un límite de máximo de elementos por día, se cuenta con limitaciones el uso de la API de Matriz de Distancia, las que se enumeran a continuación.

- Máximo de 25 puntos de origen o 25 puntos de destino por request.
- Máximo de 100 elementos por server-side request.
- Máximo de 100 elementos por client-side request.
- 1000 elementos por segundo, calculado como la suma de client-side y server-side queries.

3.1.3. Modelo actual utilizado en SimpliRoute

El modelo actual utilizado en la empresa consiste en obtener la matriz de distancia por medio de OSRM, este es un servicio de red gratuito, lo que no implica costos extras para SimpliRoute. Los valores de la matriz de distancia entregados por OSRM se alejan de la realidad de los tiempos de traslado de los vehículos, por lo que se realizó un ajuste a dichos valores.

SimpliRoute desarrolló un modelo de regresiones lineales, donde se toman los valores de la matriz de distancia obtenidos de OSRM y se ajustan por medio de regresiones lineales a los valores de la matriz de distancia de la API de Google Maps. Este modelo entrega como resultado que un 80 % de las veces los tiempos entregados se acercan a los de Google Maps en $\pm$ 5 minutos.

Por otro lado, este modelo se encuentra implementado en la Región Metropolitana sin posible escalabilidad a los países donde opera SimpliRoute, siendo estos, México, Perú, Uruguay, Brasil, Argentina, los países de Centroamérica, entre otros.

Cabe destacar que este modelo no considera tráfico en la ciudad, debido a que OSRM no tiene las condiciones de tráfico agregadas en su servicio, considerando una velocidad promedio de movimiento de los vehículos de 60 km/hr sin importar el tipo de calle, de vehículo ni dirección de movimiento.

Considerando el volumen de clientes de SimpliRoute y el volumen de consultas que realiza cada cliente por cada minuto, se llega a la conclusión de que es económicamente infactible utilizar Google Maps en el día a día, debido al alto valor que tiene el uso de la API comparado con el beneficio extra que se obtendría de utilizarlo, el error del 20 % no es justificable en base al alto valor monetario que implica cambiar el uso a la API de Google Maps. Por otro lado, con este trabajo de tesis se busca disminuir ese 20 % de error sin mayores costos monetarios.

3.2. Zona de estudio

Debido a la gran cobertura que tiene SimpliRoute en el mundo, es necesario acotar o limitar la zona a estudiar, es decir, la zona geográfica donde se obtendrán los datos, para expandir el modelo al resto de los países y regiones en las que opera SimpliRoute.

Se limita a la Región Metropolitana, debido a que corresponde a la ciudad donde se encuentra la oficina principal de SimpliRoute. Por otro lado, es conveniente limitar el trabajo a esta ciudad por la gran cantidad de datos GPS presentes en la zona, utilizando los datos de Transantiago y de los vehículos que registran datos GPS utilizando SimpliRoute. En la figura 3.3 se muestra la zona a estudiar.

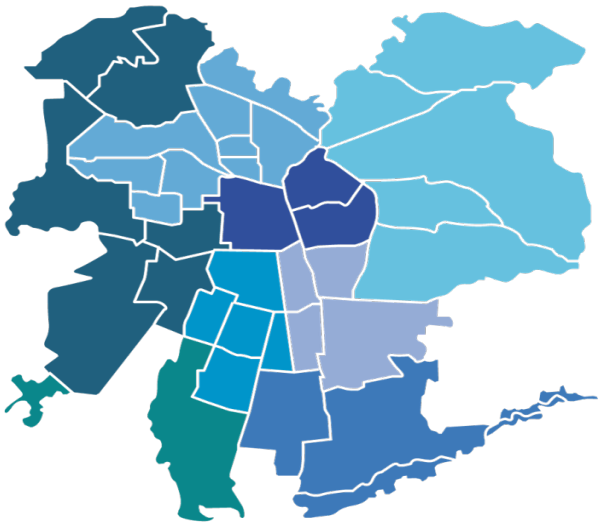


Figura 3.3: Mapa Región Metropolitana, zona de estudio

3.2.1. Tráfico en la zona de estudio

Considerando que se busca realizar estimaciones de tiempo de traslado, es relevante conocer el comportamiento del tráfico en la zona a estudiar. Es por esto que se obtiene información de TomTom, que corresponde a una plataforma que permite obtener estadísticas de congestión en distintas ciudades del mundo[6]. Se pueden obtener reportes con distintos indicadores relevantes para el entendimiento del tráfico en la ciudad.

En el año 2019 Santiago quedó calificada como la ciudad número 26 más congestionada en el mundo, con un índice de tiempo de congestión de un 44 % lo que va en aumento comparado con el año 2018, esto significa que un viaje de 30 minutos tomará un 44 % de tiempo de lo que tomaría durante las condiciones bases de Santiago sin congestión. Esto equivale a 13.2 minutos promedio extra de viaje, es decir, 43.2 minutos de tiempo de traslado en total.

En la figura 3.4 se visualiza un mapa de calor con las horas de mayor congestión, esto muestra el nivel de congestión por cada hora de cada día de la semana, en promedio a lo largo del año. El color rojo oscuro corresponde al período con mayor nivel de congestión, siendo estos entre las 18:00 y 19:00 hrs de lunes a viernes, y el viernes a las 19:00 horas el período de mayor congestión, con un 93 % de nivel de congestión.

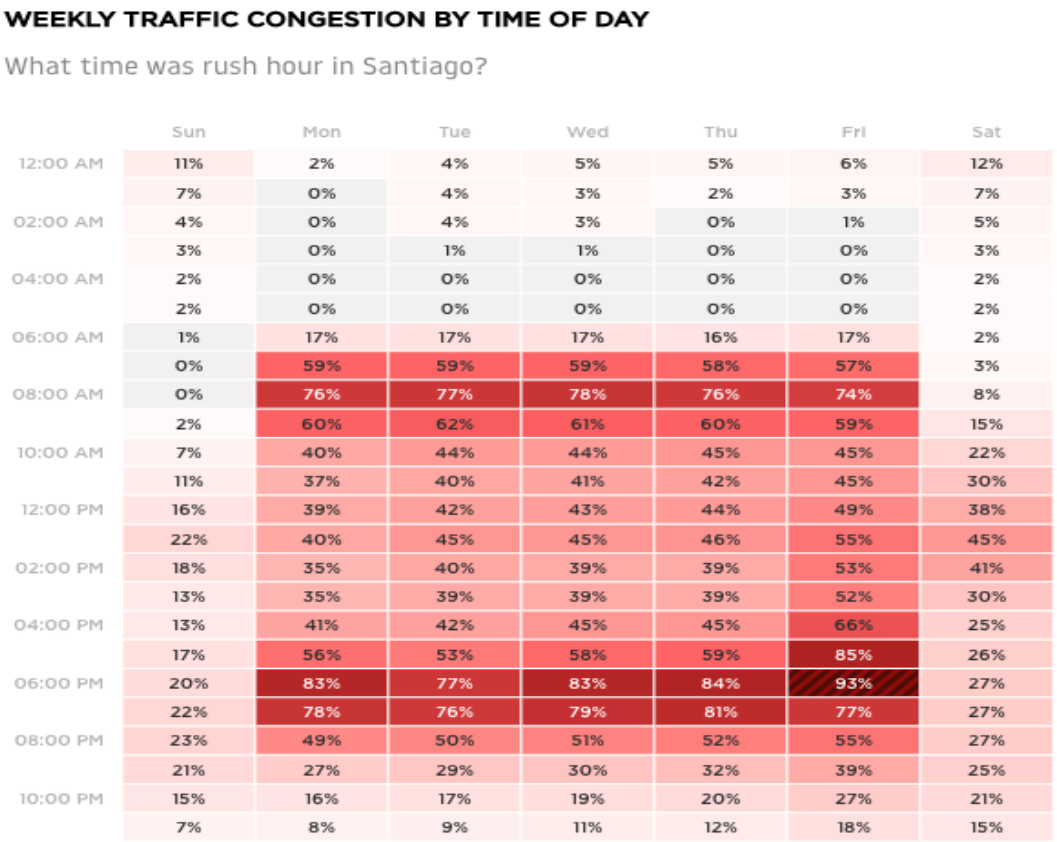


Figura 3.4: Mapa de calor - Nivel de congestión por hora del día

[6]Página web de TomTom donde se obtienen los indicadores mostrados. www.tomtom.com

En la figura 3.5 se muestran los índices de congestión en los días laborales durante la mañana y la tarde, que corresponde al período de una hora más ocupado en la mañana / tarde, según la definición de la ciudad en función de las mediciones de tráfico reales.

Se aprecia que en las tardes se produce un mayor nivel de congestión, lo que corresponde con lo mostrado en la figura 3.4, siendo estos valores de 76 % en la mañana y 84 % en la tarde, lo que implica un aumento de 23 minutos de viaje por cada 30 minutos en la mañana y 25 minutos de viaje por cada 30 minutos en la tarde con respecto a las condiciones bases de Santiago sin congestión, como se muestra en la figura 3.6

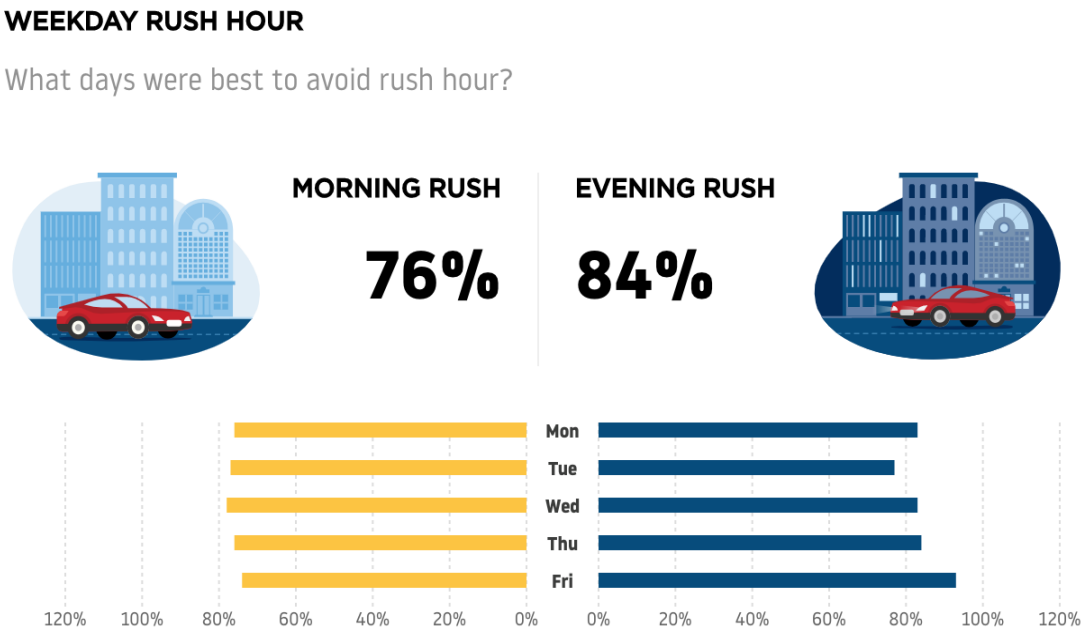


Figura 3.5: Indicador de congestión en días laborales.

Figura 3.6: Indicador de aumento de tiempo de viaje en horas peak.

Por otro lado, se analiza también la congestión en carreteras y no carreteras. Se tiene un total de 580.723.544 km de datos, que corresponde a la longitud total del sistema vial evaluado. Un 51 % de los datos son de carretera o autopista y el resto de los datos corresponden a caminos urbanos o rurales. En la figura 3.7 se puede ver que el nivel de congestión es mucho mayor en caminos que no son carreteras, correspondiente a un 50 % de nivel de congestión. En resumen, en las ciudades se tiene mayor aumento de los tiempos de viaje por la congestión.

Figura 3.7: Nivel de congestión por tipo de camino.

Se puede concluir en base a los indicadores mostrados que Santiago es una ciudad con alta congestión de tráfico en distintas horas del día. Por lo que es relevante considerar la información de tráfico al momento del cálculo de tiempos de traslado dentro de la ciudad.

Este trabajo de tesis se considera necesario para mejorar la estimación de tiempos de traslado entre distintos puntos de la ciudad debido a que no se encuentra representado de manera explícita el efecto de congestión con en el modelo actual de regresiones lineales implementado por SimpliRoute.

Capítulo 4

Modelo utilizado

En el presente capítulo se realiza una descripción del modelo propuesto para la estimación y predicción de los tiempos de traslado de la plataforma de SimpliRoute en la ciudad de Santiago.

Para mejorar la estimación de tiempos de traslado, y por ende la estimación de tráfico, se realiza un modelo basado en viajes, es decir, un modelo que considera las condiciones de GPS de los vehículos de SimpliRoute, latitud y longitud, para realizar una estimación de velocidad espacial promedio a la que se mueve el vehículo, como explican Cortés et al. (2011), y de esta manera determinar en cierto tramo (distancia entre dos coordenadas particulares) el tiempo que toma recorrerlo.

Para esto se utilizan datos históricos de una semana, sin eventualidades, de donde se obtienen los valores de las coordenadas y el tiempo exacto en que se registra dicha coordenada, además del vehículo al que corresponden los datos.

Una vez que se tiene la estimación en base a los datos históricos se desarrolla un modelo de predicción basado en Random Forest, como explican Cheng et al. (2019), para entrenar los datos y así tener una predicción personalizada en base al movimiento del vehículo de cada cliente en particular.

4.1. Modelo basado en viajes

El modelo basado en viajes que se utiliza corresponde a un modelo donde se considera la distancia entre los puntos de GPS y el tiempo registrado de cada coordenada. Considerando una tramificación[1] de un segmento generado con las coordenadas, desde un punto inicial a uno final del vehículo, por cada día, se obtiene una estimación de las velocidades de movimiento del vehículo.

[1]Separación de un segmento, es decir, del trayecto de un vehículo, en tramos equidistantes.

Este modelo considera la dirección de movimiento del vehículo, debido a que se tienen los valores ordenados por tiempo en que fueron registrados las señales de GPS. Además, considera la velocidad a la que se mueve el vehículo y la distancia entre distintas coordenadas.

4.1.1. Proyección de los puntos sobre el mapa

Se requiere tener una proyección de los datos de GPS sobre el mapa para comprobar que las mediciones son precisas, de manera que los cálculos realizados sean lo más cercano a la realidad.

OSRM Match Service

OSRM Match Service[2] es un modelo básico de proyección de datos, el que permite verificar que las mediciones de señales de GPS de los vehículos de SimpliRoute sean de buena calidad. El servicio entrega como resultado múltiples sub-tramos de un total de puntos, los puntos que no pueden ser exitosamente emparejados son removidos.

El resultado entregado por *OSRM Match Service* corresponde a una polilínea que se puede proyectar sobre un mapa de Google, y de esta forma verificar que las mediciones de las señales de GPS estén correctas y sean adecuadas para utilizar en el modelo propuesto en este capítulo. En el Algoritmo 4 se describe el método para ingresar los datos en *OSRM Match Service*.

Algorithm 4 Proyección: OSRM Match Service

input : A list of latitude values, lat
input : A list of longitude values, lon
output: code: , matchings:[distance: , confidence: , weight: , weight-name: , geometry:]
begin
  url = 'https://router.project-osrm.org/match/v1/driving/lon[1],lat[1],...,lon[final],lat[final]
  ?geometries = polyline6'
end

Figura 4.1: Diagrama de proceso de proyección de señales de GPS utilizando OSRM.

[2]http://project-osrm.org/docs/v5.5.1/api/?language=cURLmatch-service

Se entregan como datos de entrada una lista con valores de latitud y longitud de las señales de GPS, para luego ser ingresadas como se sintetiza en el Algoritmo 4, En la Figura 4.1 se observa el flujo que sigue el ingreso de datos. Luego al ingresar estos datos en el servicio de *match de OSRM*, se define el tipo de movimiento, en este caso *driving*, para señalar que corresponde a un vehículo motorizado. El servicio entrega como resultado una polilínea que es proyectada sobre un mapa, la que contiene datos de distancia, confianza de la proyección, peso asignado a cada coordenada y geometría obtenida, que corresponde a una polilínea.

4.1.2. Estimación de tiempo

Para realizar la estimación de tiempos se debe crear por cada vehículo un segmento con las coordenadas registradas en su trayectoria. A continuación, se debe dividir dicho segmento en tramos equidistantes fijos, para así, con el *timestamp* asociado a cada coordenada tener el tiempo que demora el vehículo en recorrer dicha distancia fija.

Una vez obtenida dicha velocidad, se procede a calcular el promedio espacial, obteniendo una velocidad media espacial del movimiento del vehículo. Teniendo dicha velocidad, es posible determinar para cualquier distancia entre dos coordenadas un tiempo de traslado.

Creación de tramos

Para crear los tramos se utiliza una función que permite calcular tramos de una distancia fija de 30 metros. Se debe entregar el valor de la distancia fija del tramo y un vector de coordenadas que considere todos los puntos del vehículo en un día.

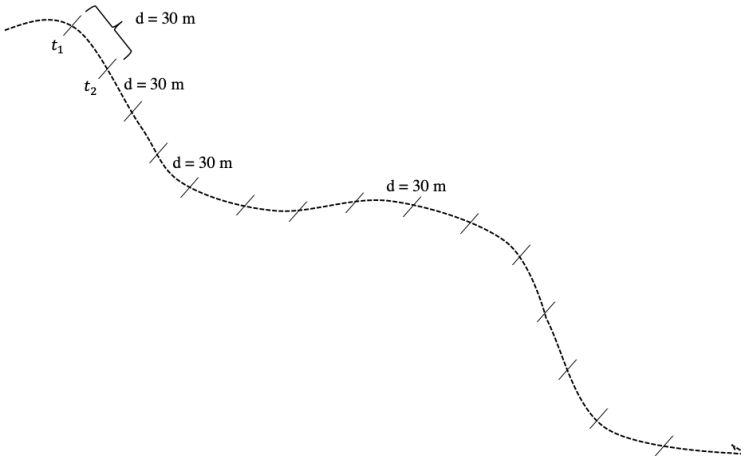


Figura 4.2: Tramificación de un segmento.

En el Algoritmo 5 se describen las entradas necesarias para generar los tramos, que corresponden a un vector de latitud, longitud y un *timestamp* asociado a dicha coordenada. Además se le debe asignar la distancia fija del largo del segmento, en metros. Este algoritmo entrega como resultado una lista de tramos; es decir, se muestran coordenadas con su respectivo timestamp a una distancia fija de 30 metros, lo que permite obtener la diferencia de tiempo por cada tramo del segmento, lo que se explica en el siguiente punto, *Cálculo de diferencia de tiempos*.

Algorithm 5 Tramificación: poly-tramado-and-normal

```
input  : A vector of latitude, longitude and timestamp, V
input  : A fixed distance of a section, d
output: A list of sections, tV
output: A list of segments, sV
output: Accumulated distance, accumulated-distance
begin
    for value in len(latitude) do
        accumulated-distance, tV, sV = poly-tramado-and-normal(V,d)
    end
end
```

Donde, *accumulated-distance* corresponde a la distancia acumulada por cada tramo que se va creando, hasta llegar al largo total del segmento. tV es un vector de tramos, que contiene la posición inicial y posición final del tramo de largo fijo. sV corresponde a un vector de segmentos, donde cada segmento será tramificado. La función *poly-tramado-and-normal* genera la división en tramos de un segmento, recibe un vector de posición y el valor de la distancia que se fija. Esto está ilustrado en la Figura 4.2.

Cálculo de diferencia de tiempos

Como se menciona en el punto anterior, se tiene un timestamp asociado a cada coordenada de inicio y fin de un tramo. Con dichos valores se puede calcular la diferencia de tiempo entre el inicio y fin de un tramo, lo que permite obtener el tiempo que el vehículo demora en recorrer 30 metros. Se aplica el Algoritmo 6 para obtener dicho valor.

Tal como se muestra en la Figura 4.3, la diferencia de tiempo se calcula con la siguiente fórmula, donde N representa a la cantidad de tramos en el segmento, t representa el tiempo.

$$\triangle t = t_{i+1} - t_i \quad \forall i \in N$$

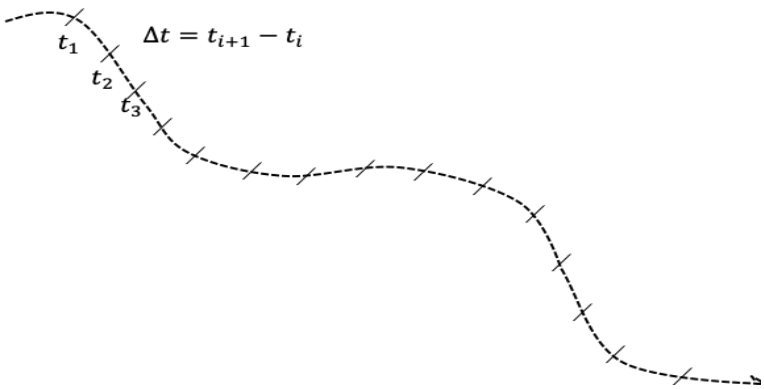


Figura 4.3: Diferencia de tiempo en cada tramo.

Algorithm 6 Diferencia de tiempo

input : A list of vectors, tV
output: A list of timestamp difference, dT
begin
  for *value in len(tV)* **do**
    dT = timestamp[i+1] - timestamp[i]
  end
end

Cálculo de velocidades

Para el cálculo de velocidad se requieren como entradas la cantidad de tramos generados en un segmento, el valor de distancia fijo de cada tramo y la diferencia de tiempo entre el punto de inicio y el punto de fin del tramo. En el Algoritmo 7 se describe el método utilizado para el cálculo de velocidad promedio.

Algorithm 7 Cálculo velocidad en un segmento

input : A list of vectors, tV
input : A fixed distance of a section, d
input : A time difference, dT[i]
output: An average velocity value per day, avg-vel
begin
  for *i in range(len(tV)-1)* **do**
    vector = tV[i]
    vel = (d/dT[i])/3.6
    avg-vel = sum(vel)/len(tV)
  end
end

A continuación, se realiza el cálculo para todos vehículos que pasen por un tramo, se utiliza la siguiente fórmula para todos los segmentos.

$$\bar{v_s} = \frac{Nd}{\sum \triangle t}$$

Donde d corresponde a la distancia fija de un tramo, $\triangle t$ corresponde al tiempo de traslado en un tramo y N la cantidad de vehículos que pasan por dicho tramo.

Determinación de tiempos entre distintos puntos de GPS

El tiempo de traslado entre dos coordenadas de GPS se calcula mediante una función que considera como valores de entrada, la posición inicial (latitud y longitud inicial), posición final (latitud y longitud final), y la velocidad media espacial calculada con lo mostrado en el apartado anterior.

Algorithm 8 Tiempo de traslado entre dos puntos

input : Initial coordinates, lat1, lon1
input : Final coordinates, lat2 , lon2
input : Velocity, v
output: Travel time, t
begin
    p1 = (lat1, lon1)
    p2 = (lat2, lon2)
    d = dist-point-to-point(p1,p2)
    t = d/v
end

En el Algoritmo 8, **v** corresponde a la velocidad media espacial, **p1** es la posición coordenada inicial, **p2** es la posición coordenada final. Teniendo ambas posiciones, se calcula la distancia **d** entre dos coordenadas de GPS utilizando la Fórmula de Haversine.

$$d = R * c$$

Donde, R es el radio de la Tierra y $c = 2 * atan2(\sqrt{a}, \sqrt{(1-a)})$[3]

$$a = sin^2(\frac{lat2 - lat1}{2}) + cos(lat1) * cos(lat2) * sin^2(\frac{long2 - long1}{2})$$

Con lo desarrollado anteriormente se obtiene el tiempo de traslado entre dos puntos de coordenadas de un vehículo. Aplicando el modelo basado en viajes descrito, se obtiene una serie de tiempos de traslado histórico de los vehículos considerados para el estudio.

[3] $atan2$: Arcotangente de dos parámetros, es una función que recibe dos parámetros y devuelve el ángulo formado entre el eje x positivo y el rayo que conecta el origen con un punto de coordenadas (x, y) distinto de (0,0) del plano euclidiano, expresado en radianes.

4.2. Predicción de tiempo

Se plantea un modelo de predicción de tiempos de traslado que utiliza la metodología de Random Forest, para aplicar dicho modelo se requiere realizar un manejo de datos de manera de que sean aplicables al modelo.

4.2.1. Random Forest

Como se menciona en Elhenawy et al. (2014) Random Forest es un modelo que permite trabajar de manera eficiente para la predicción de datos. La mayor ventaja del modelo de Random Forest es que no se sobreestiman los datos debido a la Ley de los grandes Números. Es lo que se conoce como un método de conjunto, la ventaja de este tipo de métodos es que al construir un grupo de modelos simples entrega un modelo fuerte. Random Forest utiliza la técnica de machine-learning de Árboles de Clasificación y Regresión.

Para esta tesis se utiliza el paquete de Python scikit-learn (2011), que corresponde a un paquete de *machine-learning* [4] que incluye diferentes modelos, dentro de los cuales se encuentra el de *Random Forest Regressor*, que corresponde a un estimador que ajusta cierto número de árboles de decisión de clasificación en múltiples sub-muestras del conjunto de datos y usa el promedio para mejorar la precisión predictiva y controlar el sobreajuste.

Se utilizan los siguientes módulos del paquete de scikit-learn:

- **RandomForestRegressor:** Función que pertenece al módulo *Ensemble*. Dados ciertos parámetros permite estimar valores relacionados con el conjunto de datos entregados. Se le entregan los parámetros de número de estimadores, es decir, número de árboles de decisión en el modelo y *random_state* que controla tanto la aleatoriedad del arranque de las muestras utilizadas al construir árboles como el muestreo de las características a tener en cuenta al buscar el mejor modo de exploración.

  Para el modelo implementado en esta tesis se decide utilizar los siguientes parámetros:
  - Nº de estimadores = 1000, debido a que se considera un número lo suficientemente grande para evitar la sobre-estimación de los datos, esto por la independencia que existe entre los árboles de decisión.
    Al disminuir la cantidad de estimadores el modelo sobre-estima los valores predichos, los resultados se ajustan de mejor manera a los reales, pero no es generalizable para otras predicciones.
  - random_state = 42, debido a que se probaron diferentes combinaciones de este parámetro y se llegó a la conclusión de que es un número que permite obtener una buena predicción de los datos.
    Al disminuir el valor de random_state, a cualquier número menor a 42, la precisión del modelo, disminuye a valores menores a 50 %, se hicieron pruebas con

[4] Machine-learning es el campo de estudio que brinda a las computadoras la capacidad de aprender sin ser programado explícitamente, de acuerdo con Arthur Samuel en 1959.

diferentes valores y en todo momento la precisión disminuye. Se prueba igualmente, aumentar el valor de random_state, lo que provoca que la precisión disminuya, en rangos de entre 60 % y 70 %.

- **train_test_split:** Función que pertenece al módulo *model_selection*, permite dividir un arreglo o una matriz en datos de entrenamiento y de prueba de manera aleatoria. Se le entrega el arreglo a dividirse, junto con el porcentaje que se desea de datos de entrenamiento y de prueba.

  Para esta tesis se utiliza el 80 % de los datos como entrenamiento y el 20 % como datos de prueba, debido a que mientras más datos de entrenamiento se consideren, mejor puede ser la predicción de los datos con la base de prueba.

Utilizando las funciones anteriores es posible programar la predicción de tiempos de traslado utilizando los valores históricos obtenidos del cálculo de estimación de tiempos. Se desarrolla el siguiente código, descrito en el Algoritmo 9. Para programar se utiliza el paquete de *Pandas*[5] de Python, que permite manejar los datos como *dataframes* y *numpy*[6] que es un soporte para vectores y matrices, constituyendo una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices.

Algorithm 9 Random Forest

input : A dataframe with historical time estimation, df
output: A list with time prediction, predictions
begin

Preparation of dataframe

```
labels = np.array(df['Time (Minutes)'])  Values I want to predict
features = df.drop('Time (Minutes)',axis = 1)
feature_list = list(features.columns)
features = np.array(features)
```

Definition of train and test dataset

```
train_features, test_features, train_labels, test_labels = train_test_split(features,
labels, test_size = 0.20, random_state = 42)
```

Random Forest Prediction

```
rf = RandomForestRegressor(n_estimators = 1000, random_state = 42)
rf.fit(train_features, train_labels)
predictions = rf.predict(test_features)
```

end

[5]Pandas es una herramienta de análisis y manipulación de datos de código abierto rápida, potente, flexible y fácil de usar, construido sobre el lenguaje de programación Python.

[6]Paquete fundamental para la informática científica con Python.

El Algoritmo 9 se divide en 3 partes fundamentales.

La primera corresponde a la preparación de los datos, ésta se realiza por medio de la separación de los valores que se buscan predecir (*labels* en el algoritmo) y los valores que determinan o explican dicha predicción (*features* en el algoritmo). En el caso del modelo desarrollado en esta tesis, corresponden al tiempo de traslado en minutos y la posición inicial y final, junto con la distancia respectivamente.

Una vez preparados los datos, se procede a la segunda parte, que corresponde a la definición de la base de datos de entrenamiento y de prueba. Es en esta parte del algoritmo donde se decide utilizar el 80 % de los datos como entrenamiento y los datos restantes como prueba, la distribución de los datos se realiza de manera aleatoria.

La tercera parte corresponde a la realización del modelo de Random Forest utilizando la función descrita anteriormente (*RandomForestRegressor*). Una vez realizado el modelo, se procede a ajustar el modelo con los datos de entrenamiento (*rf.fit* en el algoritmo) y calcular las predicciones con la base de prueba (*predictions* en el algoritmo).

En este capítulo se desarrolló un modelo de estimación y predicción de tiempos de traslado utilizando datos obtenidos por medio de señales de GPS. Se obtiene una estimación de los tiempos de traslado por medio del cálculo de la velocidad media espacial de los vehículos. Por otro lado, se desarrolla una predicción de dichos tiempos utilizando datos de tiempos históricos obtenidos por los cálculos de estimación.

En el siguiente capítulo se realiza una descripción de los datos utilizados para implementar el modelo en SimpliRoute y un análisis de los resultados obtenidos.

Capítulo 5

Implementación del modelo y resultados obtenidos

5.1. Obtención de los Datos

Para la implementación del modelo presentado en el Capítulo 4, se utilizan datos de pulso de GPS de una semana, la que corresponde a la semana del 13 de Mayo de 2019. Se decide trabajar con esta semana debido a que no se presentan anomalías en el tráfico, por ejemplo producto de marchas en las calles, lluvia u otras condiciones o eventos que puedan afectar el tráfico significativamente en la región. En la Figura 5.1 se observan los registros de pulsos de GPS obtenidos de la base de datos de SimpliRoute graficados en Tableau Desktop[1].

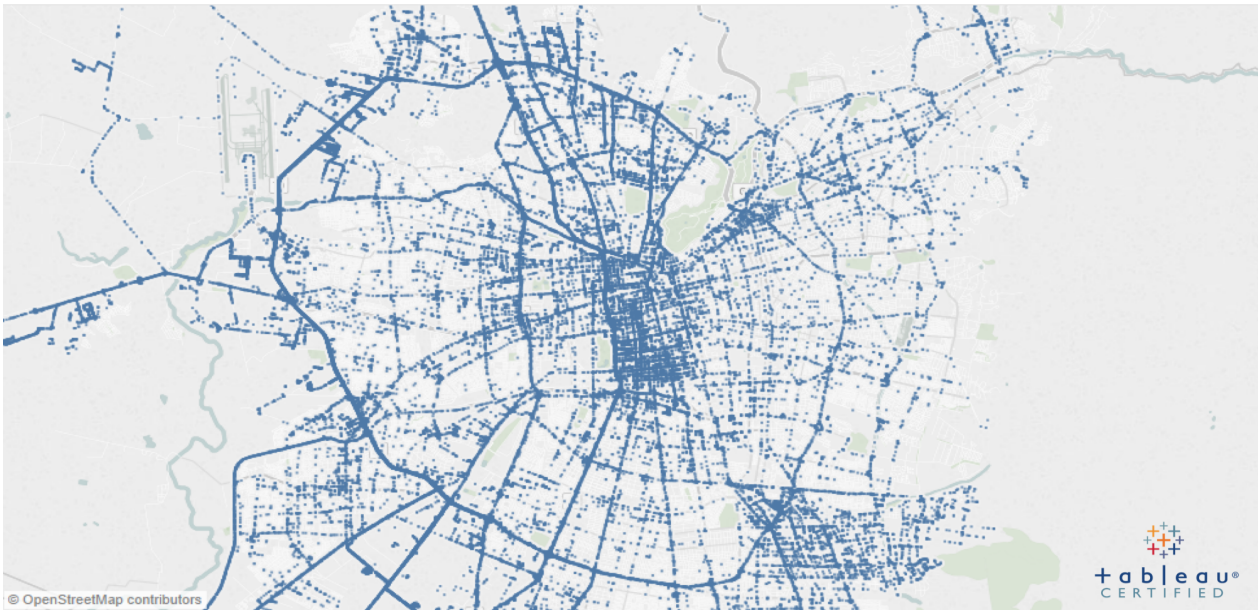


Figura 5.1: Mapa Región Metropolitana GPS SimpliRoute.

[1]Software de visualización de datos interactivos.

Realizando una ampliación a una zona específica de la Región Metropolitana (Figura 5.2) se observa que los datos de señales de GPS de SimpliRoute no cubren en su totalidad las calles de la Región Metropolitana, lo que se observa en la Figura 5.3

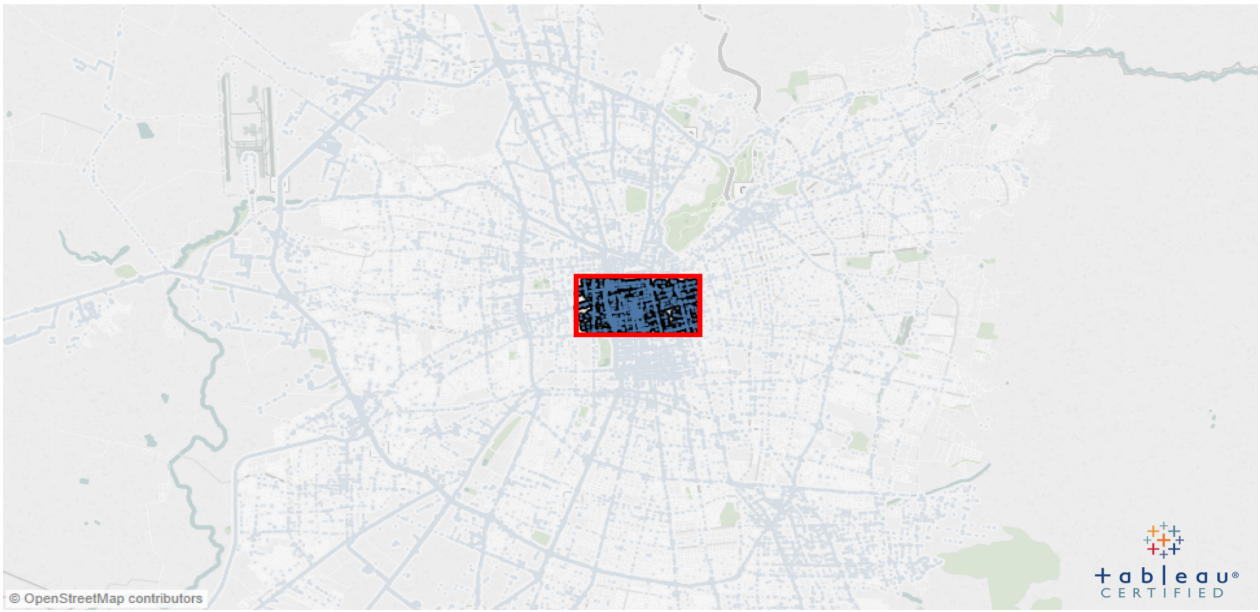


Figura 5.2: Zona elegida para testear puntos.

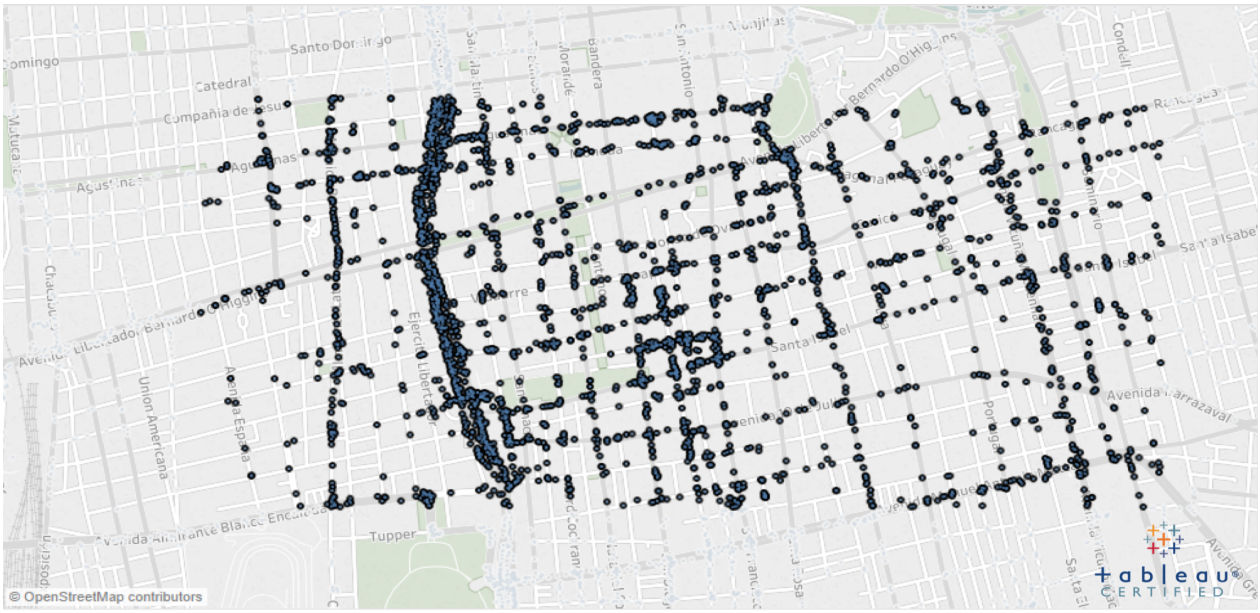


Figura 5.3: Ampliación zona de testeo.

Para aumentar los valores de datos GPS y tener valores significativos para el cálculo realizado, se decide utilizar datos de GPS de Transantiago, correspondientes a la misma semana estudiada. En la Figura 5.4 generada con QGIS[2] se observa que se tiene mayor

[2]Sistema de Información Geográfica de software libre.

volumen de datos de señales de GPS en la región, lo que permite que los cálculos a realizar sean significativos.

Figura 5.4: Mapa Región Metropolitana GPS Transantiago combinado con SimpliRoute.

Una vez obtenidos los datos se deben proyectar en un mapa para comprobar que las mediciones sean precisas. Para esto se utiliza el servicio de *match* de OSRM[3] lo que permite determinar de forma visual, como se observa en la Figura 5.5, que los datos de GPS son medidos de manera correcta, es decir, tomando un conjunto de los datos de GPS, al ser proyectados en el mapa, estos se encuentran ubicados en calles (es decir, no se ubican señales de GPS en medio de una casa o edificio, por ejemplo) lo que permite identificar claramente la trayectoria seguida por un vehículo.

[3]http://project-osrm.org/docs/v5.15.2/api/?language=JavaScriptmatch-service

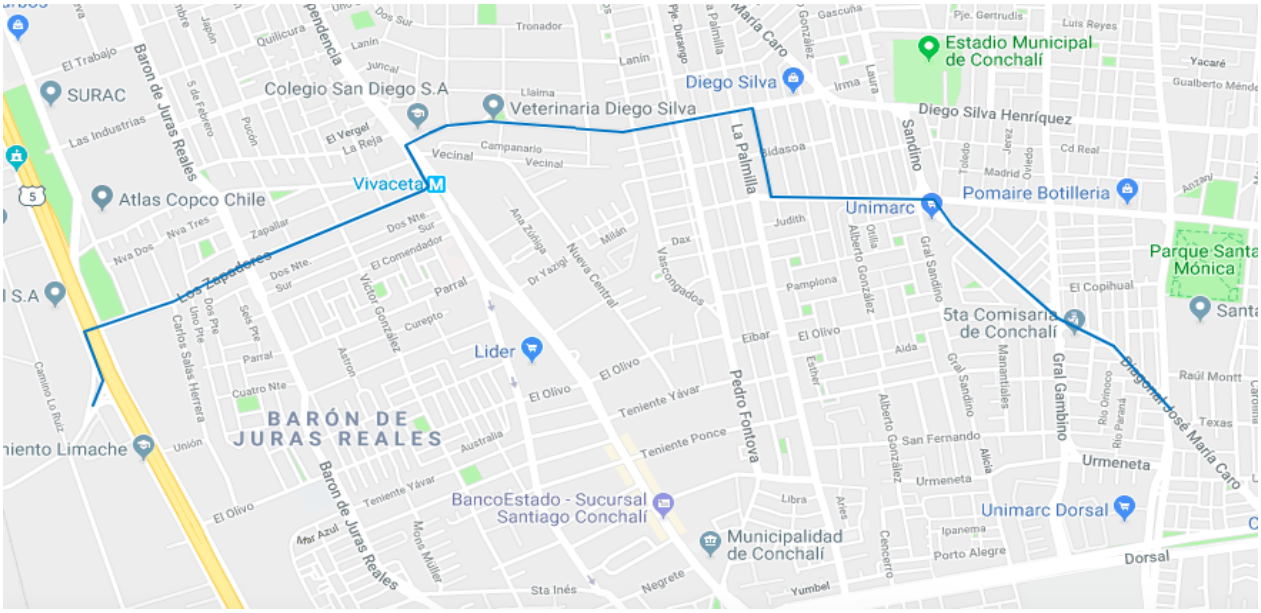


Figura 5.5: Proyección sobre un mapa de una muestra de datos de señales de GPS.

5.2. Caracterización de los Datos

En esta sección se realiza una descripción de los datos mencionados en la sección anterior, con el foco en la cantidad de vehículos utilizados, cantidad de puntos de GPS por vehículo, cantidad de puntos de pulsos de GPS en general, entre otros.

5.2.1. Datos SimpliRoute

Se presenta en la Tabla 5.1 las primeras 10 filas de los datos de GPS de SimpliRoute, obtenidos de la base de datos. Estos corresponden a coordenadas de 139 vehículos utilizados en la semana de estudio.

	Vehicle Id	Timestamp	Latitude	Longitude	Fecha
0	939	05/13/19 12:16:42	-33.377100	-70.757389	05/13/19
1	939	05/13/19 12:17:02	-33.375979	-70.757000	05/13/19
2	939	05/13/19 12:18:02	-33.374558	-70.754933	05/13/19
3	939	05/13/19 12:19:02	-33.375770	-70.749545	05/13/19
4	939	05/13/19 12:21:22	-33.372015	-70.742537	05/13/19
5	939	05/13/19 12:22:02	-33.371940	-70.735609	05/13/19
6	939	05/13/19 12:23:02	-33.373331	-70.722961	05/13/19
7	939	05/13/19 12:24:02	-33.370436	-70.706807	05/13/19
8	939	05/13/19 12:25:02	-33.366208	-70.699737	05/13/19
9	939	05/13/19 12:26:02	-33.366908	-70.699049	05/13/19

Tabla 5.1: Muestra de datos de SimpliRoute.

En la Figura 5.6 se observa la distribución de los datos de latitud y longitud graficados, en donde se puede determinar la figura de la Región Metropolitana, cabe destacar que para motivos de esta descripción los valores de latitud y longitud se encuentran redondeados, pero para motivos del modelo se utilizan los valores de coordenadas sin redondeo, debido al formato en que se guardan para el análisis. En concordancia con lo descrito en la Tabla 5.2 (a) donde se observa que se tiene un total de 717.294 señales de GPS, donde la latitud máxima es -32.9 y la mínima es -34.1, respecto a longitud, el máximo es -70.5 y el mínimo -71.6. En cuanto a tiempo el conteo de valores únicos es 358.686 datos, en la semana completa.

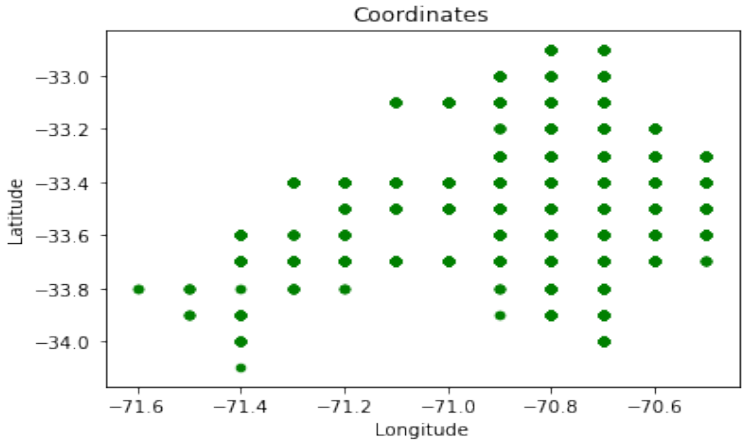


Figura 5.6: Gráfico de Latitud y Longitud datos de GPS vehículos SimpliRoute.

	Latitud	Longitud
count	717294	717294
mean	-33.494907	-70.726682
std	0.112874	0.103753
min	-34.100000	-71.600000
25 %	-33.600000	-70.800000
50 %	-33.500000	-70.700000
75 %	-33.400000	-70.700000
max	-32.900000	-70.500000

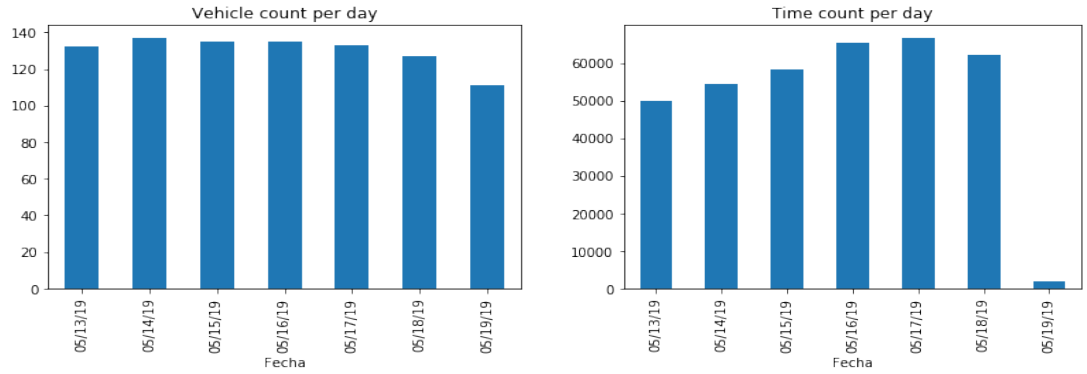
(a) Tabla descriptiva latitud y longitud vehículos SimpliRoute.

	Timestamp
count	717294
unique	358686
top	05/17/19 23:42:16
freq	4639

(b) Tabla descriptiva timestamp vehículos SimpliRoute.

Tabla 5.2: Tablas desciptivas datos SimpliRoute.

En la Figura 5.7 (a) se observa el total de vehículos utilizados para el análisis en cada día de la semana considerada, siendo el día 14/05/2019 donde se utilizó una mayor cantidad de vehículos. En la Figura 5.7 (b) se observa el conteo de diferentes tiempos de posición de los vehículos, donde el día 17/05/2019 es el día con mayor diferenciación en los tiempos medidos.



(a) Conteo de vehículos SimpliRoute **(b)** Conteo de tiempos vehículos SimpliRoute

Figura 5.7: Distribución datos SimpliRoute por día.

5.2.2. Datos Transantiago

Se presenta en la Tabla 5.3 las primeras 10 filas de los datos de GPS de Transantiago, obtenidos por ISCI de la Universidad de Chile. Estos corresponden a 2020 vehículos.

	Patente	Codigo	Timestamp	Latitude	Longitude	Fecha
0	AA-0006	T506 00R	2019-05-13 00:08:07	-33.436903	-70.656107	2019-05-13
1	AA-0006	T506 00R	2019-05-13 00:09:07	-33.436905	-70.656109	2019-05-13
2	AA-0006	T506 00R	2019-05-13 00:11:37	-33.436917	-70.656124	2019-05-13
3	AA-0006	T506 00R	2019-05-13 00:15:07	-33.436924	-70.656131	2019-05-13
4	AA-0006	T506 00R	2019-05-13 00:18:37	-33.436920	-70.656118	2019-05-13
5	AA-0006	T506 00R	2019-05-13 00:31:07	-33.436914	-70.656111	2019-05-13
6	AA-0006	T506 00R	2019-05-13 00:43:07	-33.436902	-70.656096	2019-05-13
7	AA-0006	T506 00R	2019-05-13 00:48:07	-33.436887	-70.656086	2019-05-13
8	AA-0006	T506 00R	2019-05-13 01:14:37	-33.436841	-70.656065	2019-05-13
9	AA-0006	T506 00R	2019-05-13 01:15:07	-33.436841	-70.656065	2019-05-13

Tabla 5.3: Muestra de datos de Transantiago.

Se realiza un análisis descriptivo de los datos, en la Figura 5.8 se observan graficados los datos de latitud y longitud de los buses de Transantiago, donde se obtiene una figura similar a la de la Región Metropolitana y como se observa, se tiene una mayor cantidad de datos que lo mostrado en la Figura 5.6, descrita en la sección anterior. Cómo se observa en la Tabla 5.4 (a) y (b) se cuenta con un total de 1.460.261 datos, registrados en la semana a estudiar.

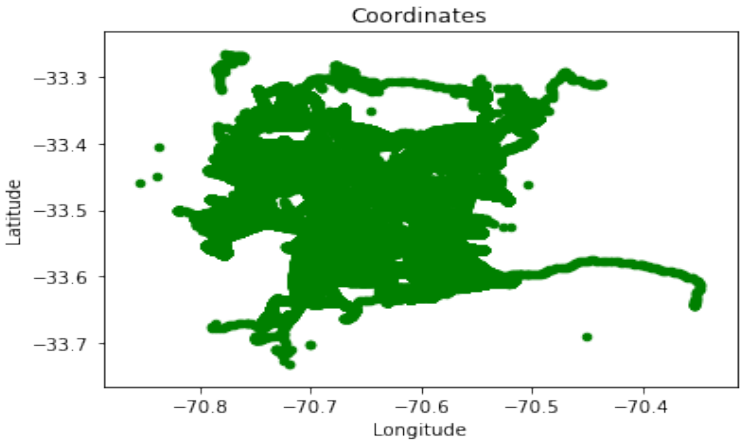


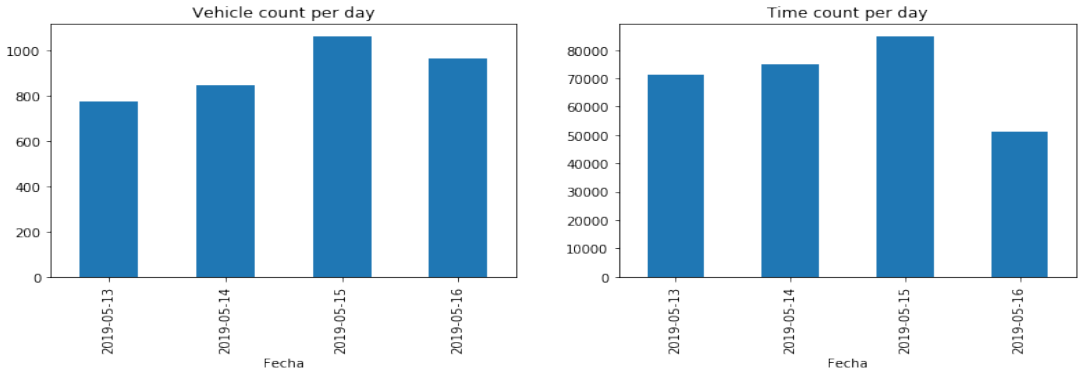
Figura 5.8: Gráfico de Latitud y Longitud datos de GPS buses Transantiago.

En la Tabla 5.4 (a) se describen los datos de latitud y longitud de las señales de GPS. Se observa que la latitud máxima es -30.56 y la mínima es -33.73, respecto a longitud el máximo es -70.34 y el mínimo -70.85.

	Latitude	Longitud
count	1.460261e+06	1.460261e+06
mean	-33.48504	-70.65457
std	6.349243e-02	7.589862e-02
min	-33.73144	-70.85522
25 %	-33.52838	-70.71769
50 %	-33.47882	-70.65488
75 %	-33.43651	-70.58681
max	-30.56231	-70.34733

Tabla 5.4: Tabla descriptiva latitud y longitud buses Transantiago.

En la Figura 5.9 (a) se observa, dentro de los primeros días de la semana la cantidad de buses utilizados, siendo el día 15/05/2019 el que cuenta con una mayor cantidad de buses en circulación en calle. Al igual que se observa en la Figura 5.9 el mismo día con la mayor cantidad de mediciones de tiempos registrados en cada posición.



(a) Conteo de buses Transantiago **(b)** Conteo de tiempos buses Transantiago

Figura 5.9: Distribución datos Transantiago por día.

5.3. Resultados

Para obtener los resultados se deben correr secuencialmente los distintos algoritmos presentados en el capítulo anterior.

Los datos presentados anteriormente se utilizan para obtener la velocidad histórica de los vehículos.

Una vez obtenida la velocidad, se genera una instancia de puntos de señales de GPS a los que se les debe calcular el tiempo de traslado entre cada señal. Estos datos se entregan como *.json* que corresponden a puntos de GPS con latitud y longitud, correspondientes a cada visita que realiza un vehículo.

Mediante el input entregado se calcula el tiempo histórico recorrido por los vehículos utilizando el Algoritmo 8 presentado en el capítulo anterior.

En la tabla 5.5 se muestran los tiempos obtenidos del cálculo histórico de tiempos de traslado, junto con la distancia entre cada punto de visita.

	Time (Minutes)	Distance (Kilometers)	Initial Latitude	Initial Longitude	Final Latitude	Final Longitude
0	12.179624	9.393359	-33.501967	-70.575230	-33.454260	-70.656147
1	18.635142	14.372083	-33.454260	-70.656147	-33.493097	-70.517773
2	39.625088	30.560275	-33.493097	-70.517773	-33.372067	-70.798388
3	26.909009	20.753183	-33.372067	-70.798388	-33.392925	-70.591907
4	2.373950	1.830874	-33.392925	-70.591907	-33.405387	-70.605320
...	...	...	...	...	...	...
399	12.694136	9.790168	-33.636791	-70.524072	-33.647429	-70.621394
400	35.223733	27.165793	-33.647429	-70.621394	-33.648890	-70.349740
401	29.158046	22.487720	-33.648890	-70.349740	-33.695889	-70.569651
402	5.371636	4.142797	-33.695889	-70.569651	-33.733600	-70.552500
403	30.178275	23.274557	-33.733600	-70.552500	-33.501967	-70.575230

Tabla 5.5: Input entregado al algoritmo.

En la tabla 5.6 se realiza una breve descripción de los valores obtenidos del cálculo de tiempo histórico. Se tiene que el tiempo promedio es de 11 minutos por trayecto entre un punto y otro del recorrido. El tiempo máximo en un trayecto es de 70 minutos.

	Time (Minutes)	Distance (Kilometers)	Initial Latitude	Initial Longitude	Final Latitude	Final Longitude
count	404	404	404	404	404	404
mean	11.525453	8.888838	-33.427474	-70.646219	-33.427474	-70.646219
std	11.056021	8.526796	0.089230	0.105079	0.089230	0.105079
min	0.001223	0.000943	-33.733600	-70.894664	-33.733600	-70.894664
25 %	2.312467	1.783457	-33.477681	-70.721538	-33.477681	-70.721538
50 %	8.139315	6.277329	-33.434154	-70.638846	-33.434154	-70.638846
75 %	19.427924	14.983505	-33.383336	-70.565057	-33.383336	-70.565057
max	70.816864	54.616479	-33.190697	-70.349740	-33.190697	-70.349740

Tabla 5.6: Descripción input entregado al modelo.

En la Figura 5.10 se observa la distribución de las coordenadas consideradas para el cálculo de tiempo de traslado histórico entre distintos puntos de la ciudad. En la Figura 5.11 se observa que a mayor distancia entre los puntos, mayor será el tiempo de traslado, además se observa que la mayor cantidad de los datos tienen una distancia menor a 30 kilómetros entre puntos coordenados de señales de GPS.

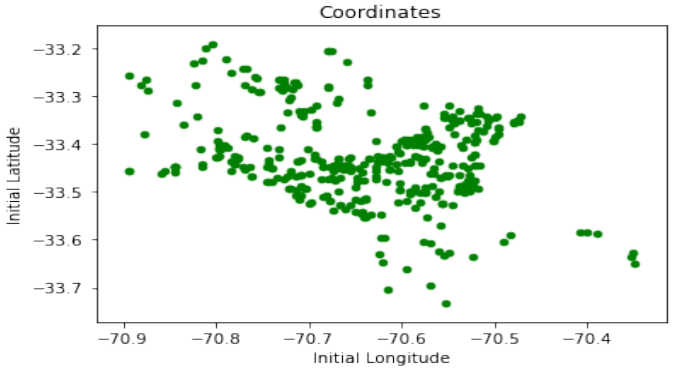


Figura 5.10: Gráfico de Latitud y Longitud datos históricos.

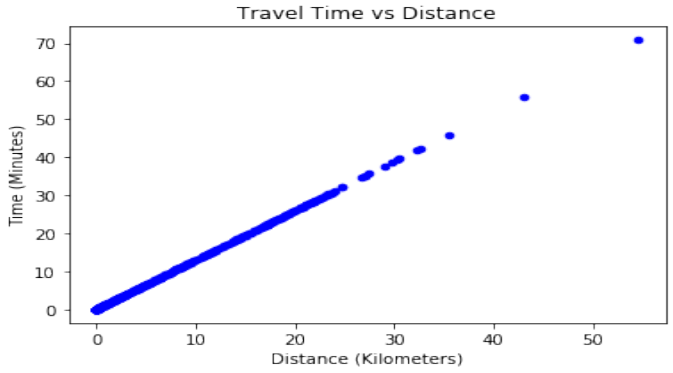


Figura 5.11: Gráfico de Distancia respecto a tiempo de traslado.

Una vez obtenidos los tiempos históricos de traslado, se corre el Algoritmo 9 donde se obtiene la predicción de tiempos de traslado.

Como se menciona en el capítulo anterior, se utiliza el 80 % de los datos como base de entrenamiento, es decir, 323 datos en la base de entrenamiento y 81 datos en la base de prueba.

Una vez ejecutado el modelo de Random Forest se obtiene una lista con la predicción de los tiempos de traslado. Se tiene como predicción de referencia los promedios históricos de tiempo de traslado, calculados en base a las variables que explican el modelo, para este valor base de referencia se calcula un error de referencia medio que corresponde a 2.67 minutos sobre la predicción, es decir, si la predicción supera los 2.67 minutos de error, entonces hay que buscar otro enfoque en la predicción de los tiempos de traslado.

La predicción del modelo se obtiene con un **96.88 %** de precisión, calculado utilizando la medida MAPE [4], la precisión se obtiene a partir de la diferencia porcentual entre el 100 % y el MAPE. En consecuencia, el modelo tiene un error porcentual de 3.12 %.

$$MAPE = \frac{\sum_{i=1}^{n} 100|\,\text{Real}_i - \text{Pronóstico}_i|}{\text{Real}_i}$$

Por otro lado, se calculan el Error Absoluto Medio (MAE, por sus siglas en inglés), que mide la magnitud promedio de los errores en un conjunto de predicciones, sin considerar su dirección; y el Error Cuadrático Medio (RMSE), que es una regla de puntuación cuadrática que mide la magnitud promedio del error. Los valores obtenidos son los siguientes,

- **MAE:** 0.11 grados
- **RMSE:** 0.32 grados

Desde el punto de vista de la interpretación, se utiliza el MAE. RMSE no describe solo el error promedio y tiene otras implicaciones que son más difíciles de descifrar y comprender. Por lo que el MAE en el modelo de predicción utilizado en esta tesis indica que el tiempo de traslado predicho puede estar calculado con una diferencia de 0.11 minutos de la realidad.

De los valores calculados se concluye que todos son menores al valor de referencia de error mencionado, es decir, Random Forest es el enfoque que se debe seguir para la predicción de los datos, lo que se complementa con el valor de precisión calculado.

A medida que se aumenta el número de estimadores a utilizar, es decir, la cantidad de árboles de decisión que tendrá el modelo, el valor del MAE y del RMSE disminuye, esto significa que mientras más grande sea el número de estimadores, mejor será la predicción.

[4]Error Porcentual Absoluto Medio que mide el tamaño del error de forma porcentual

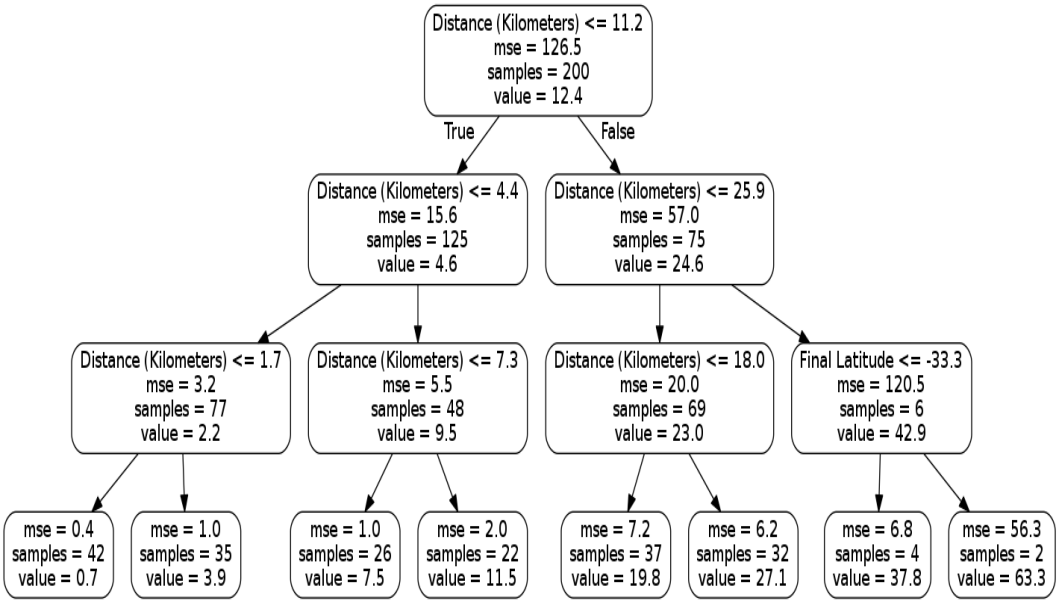


Figura 5.12: Árbol de decisión simplificado.

En la Figura 5.12 se observa uno de los árboles de decisión simplificado del modelo, esto permite tener una noción de cómo se van formando los valores predichos de tiempo de traslado en el modelo de Random Forest.

Es relevante considerar la importancia de las variables a utilizar para la predicción. En la Figura 5.13 se observa que la variable más importante a considerar es la distancia entre las posiciones inicial y final para calcular el tiempo de traslado. Esto es concordante con lo mostrado en la Figura 5.11, por lo que a mayor distancia, mayor será el tiempo de traslado.

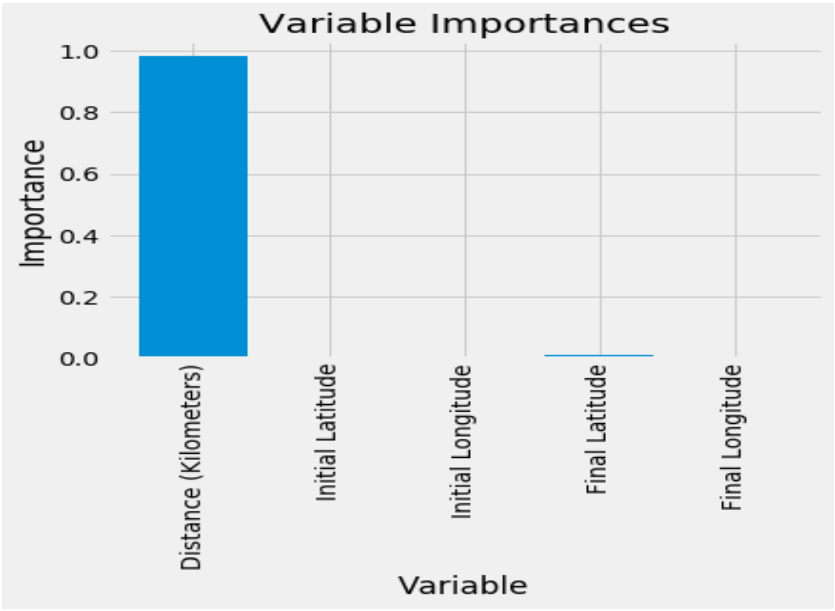


Figura 5.13: Importancia de las variables entregadas al modelo de Random Forest.

Se genera un gráfico comparativo con los valores reales y la predicción, en la Figura 5.14. Se muestra en rojo los valores predichos, y en verde los valores reales. Como se observa en el gráfico los primeros valores no se ajustan muy bien a lo real, pero a medida que se avanza en los valores se observa que los tiempos predichos se acercan cada vez más al real, esto calculado con un 96.88 % de precisión en la predicción.

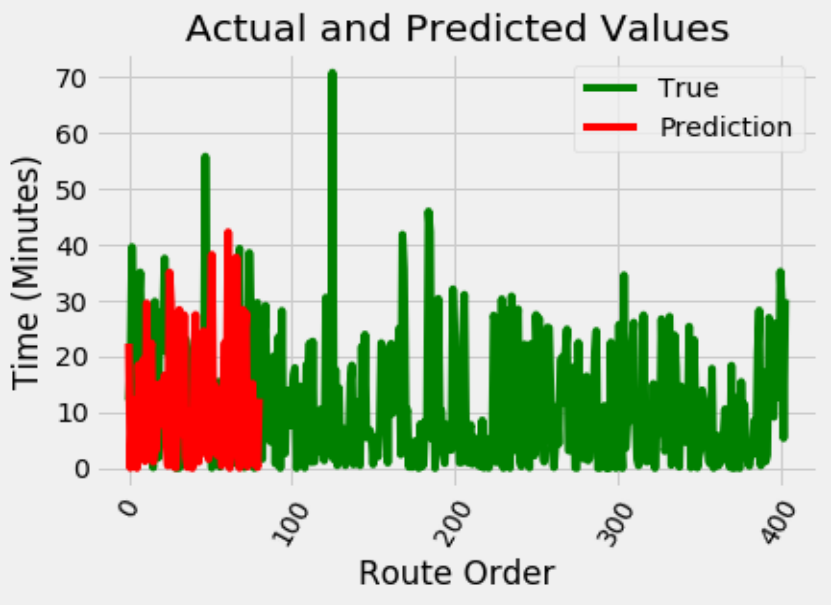


Figura 5.14: Gráfico comparativo Valores reales vs Predichos.

5.3.1. Comparación con Google Maps

Se realiza una llamada a la API de Google Maps donde se pide el tiempo de traslado entre los mismos puntos de GPS con los que se realiza la predicción utilizando el Random Forest.

Google Maps entrega valores de distintos tipos de tráfico, dentro de los que se menciona tráfico alto (pesimista), normal (mejor predicción) y bajo (optimista).[5]

- Mejor predicción: (valor predeterminado) Indica que la duración en transporte devuelta debe ser la mejor estimación del tiempo de viaje dado lo que se sabe sobre las condiciones de tráfico históricas y el tráfico en vivo. El tráfico en vivo se vuelve más importante cuanto más cerca esté ahora el tiempo de salida.
- Pesimista: Indica que la duración en transporte devuelta debe ser mayor que el tiempo de viaje real en la mayoría de los días, aunque los días ocasionales con condiciones de tráfico particularmente malas pueden exceder este valor.
- Optimista: Indica que la duración en transporte devuelta debe ser menor que el tiempo de viaje real en la mayoría de los días, aunque los días ocasionales con condiciones de tráfico particularmente buenas pueden ser más rápidos que este valor.

[5]Información obtenida de la página de documentación de la matriz de distancia de Google Maps. https://developers.google.com/maps/documentation/distance-matrix/intro?hl=estraffic-model

En la Tabla 5.7 se describen los promedios de tiempo de traslado calculados con cada método.

	Google Maps			
	Low	Normal	High	Random Forest Prediction
Mean Time Travel (Minutes)	10.34	16.43	25.51	11.72

Tabla 5.7: Tabla comparativa de valores predichos vs valores de Google Maps

Al comparar los valores entregados por Google Maps y los calculados con la predicción de Random Forest, se llega a que la metodología descrita en este trabajo de tesis entrega en promedio una predicción de 1 minuto de diferencia con los entregados por Google Maps cuando el tráfico es bajo y de 5 minutos de diferencia cuando el tráfico es normal. Cabe destacar que este modelo se aleja de la realidad cuando el tráfico es considerado alto de acuerdo a Google Maps.

Un 66 % de las veces el tiempo predicho en esta tesis es menor al entregado por Google Maps cuando el tráfico es normal, con un promedio de 15 minutos de tiempos de traslado, un 78 % el tiempo predicho es menor cuando el tráfico es alto con un promedio de 30 minutos de traslado, y un 64 % de las veces el tiempo predicho es menor cuando el tráfico es bajo con un promedio de 11 minutos de traslado. De esto se concluye que el tiempo predicho puede entregar mejores predicciones de traslado, más personalizadas a los vehículos de SimpliRoute. Esta predicción se puede catalogar como tráfico normal, debido a que la estimación promedio se encuentra entre las categorías de tráfico bajo y normal de Google Maps.

Lo que muestra que es posible, mediante los valores históricos obtener una predicción certera de los tiempo de traslado entre distintos puntos de GPS. Por otro lado, es posible con esta comparación determinar qué tipo de tráfico se está estimando, en este caso, se acercan los resultados a un tráfico optimista/mejor predicción.

Capítulo 6

Conclusiones y Comentarios

6.1. Conclusiones Generales

En este trabajo de tesis se aborda el problema de estimación y predicción de tiempos de traslado mediante la metodología de machine learning para obtener valores certeros a la realidad de los vehículos que utilizan SimpliRoute. Esto debido a que utilizar la API de Google Maps se considera insostenible económicamente en la empresa por los altos costos de utilización.

Para realizar el modelo que permite estimar los tiempos de traslado se utilizan datos de un período de 1 semana de GPS de SimpliRoute y de Transantiago, debido a la falta de cobertura de puntos en las calles por los vehículos de SimpliRoute.

Para generar el modelo de estimación de tiempos de traslado, se deben realizar los siguientes pasos:

1. Se estudian las principales metodologías de estimación de tiempos de traslado en la literatura. Considerando lo planteado por Forum Jensen et al. (Forum y Villy Larsen, 2006) de un modelo basado en viajes, es decir, una secuencia ordenada de observaciones de GPS.

2. Se trabaja con los datos de GPS de SimpliRoute y del Transantiago para obtener una base de datos de partida limpia para la modelación del problema.

3. Se realiza un algoritmo de estimación de velocidad promedio de movimiento de los vehículos utilizando la metodología desarrollada por Cortés et al. (2011) mediante la tramificación de segmentos para la obtención de velocidad media espacial.

4. Utilizando la velocidad promedio calculada, se realiza una función de cálculo de tiempo entre dos puntos de GPS mediante el cálculo de distancia entre dichos puntos.

5. Se utiliza el cálculo de tiempos históricos para entrenar un modelo de Random Forest para la predicción de los tiempos de traslado. Se obtiene una precisión de 96.88 % lo que muestra que los valores predichos se acercan a lo calculado históricamente. Además, el error absoluto medio de las predicciones es de 0.11 minutos comparado con la realidad, lo que es un buen indicador para concluir que el modelo de Random Forest es apropiado para la predicción de tiempos de traslado.

   También se concluye de esto que el número de estimadores utilizados es apropiado, debido al valor obtenido de MAE, que disminuye a medida que aumenta el número de estimadores, es decir, la cantidad de árboles de decisión a utilizar en el modelo. Por otro lado, la variable más importante considerada en el modelo es la distancia entre la posición inicial y final de traslado.

6. Se comparan los resultados obtenidos por el modelo de Random Forest con los obtenidos mediante una llamada a la API de Google Maps. Lo entregado por Google Maps presenta tres escenarios, optimista, pesimista y mejor predicción. Realizando comparaciones con lo predicho por el modelo de Random Forest, se llega a la conclusión de que el modelo desarrollado en esta tesis tiene una aproximación a los valores optimista y mejor predicción, debido a que se obtiene una diferencia promedio de 5 minutos. Es decir, los valores predichos por la metodología de Random Forest se encuentran en un rango de ± 5 minutos a los entregados por Google Maps, lo que se considera una buena predicción en base a que como empresa se busca que los valores se acerquen lo más posible a los datos de Google Maps.

   Más del 60 % de las veces los resultados de la predicción son menores a los entregados por Google Maps en el escenario optimista y de mejor predicción.

Se concluye del trabajo desarrollado en esta tesis que utilizar los datos históricos de los vehículos es una metodología robusta, debido a que es posible replicar un comportamiento de los vehículos en calle más aproximado a la realidad, de manera que es un cálculo personalizado para la empresa y con ello se puede entregar un servicio de calidad a los clientes de SimpliRoute.

Es importante destacar que este trabajo cuenta con limitaciones, previamente mencionadas en el Capítulo 1, sección 1.2. Las que serán abordadas en trabajos futuros a realizarse. Este modelo es aplicable para la Región Metropolitana, debido a que los datos históricos utilizados para entrenar el modelo de Random Forest corresponden a datos de vehículos en circulación en la Región Metropolitana, se abordará la expansión del modelo a otras regiones y países en trabajo futuro.

Finalmente, se concluye que la metodología de Random Forest es una buena aproximación a lo considerado certero, que son los valores de Google Maps, debido a que la diferencia promedio del cálculo es de 5 minutos, lo que se considera satisfactorio para estándares de SimpliRoute.

6.2. Trabajo Futuro

El trabajo de tesis se realizó con miras de continuar con un producto para SimpliRoute, donde se buscan maneras innovadoras de aplicar Machine Learning en la empresa. Es por esto que como trabajo futuro para SimpliRoute se utilizará el algoritmo y metodología presentados en esta tesis para mejorar los parámetros entregados a OSRM.

Para continuar con la mejora de los parámetros para OSRM, en primer lugar, se debe estudiar el funcionamiento de este motor de ruteo, es decir, encontrar los parámetros que pueden ser mejorables y la forma de cómo mejorarlos.

Se trabajará con datos de SimpliRoute para obtener una estimación de tiempos de traslado personalizada a la utilización de los vehículos asociados a la empresa.

En el trabajo futuro se deben abordar las limitaciones que presenta este trabajo de tesis. Se estudiarán nuevos modelos de Machine Learning que puedan ser aplicados en la predicción de los parámetros que se entregan a OSRM, en particular la velocidad, como se describe en la tesis de Vejar (2019). Además, se considerarán diferentes bloques horarios, es decir, el valor predicho dependerá de las horas del día y de los días de la semana.

Una vez mejorado esto para la Región Metropolitana, se extenderá dicho modelo a los países en los que opera SimpliRoute.

Capítulo 7

Bibliografía

Bibliografía

Cheng, J., Li, G., y Chen, X. (2019). Developing a travel time estimation method of freeway based on floating car using random forests. *Hindawi, Journal of Advanced Transportation*, *2019*, 13.

Chicoisne, R., Ordoñez, F., y Espinoza, D. (2019). Efficient algorithms to match gps data on a map. *International Journal of Operational Research*, *36*, 518 – 537.

Cortés, C. E., Gibson, J., Gschwender, A., Munizaga, M., y Zúñiga, M. (2011). Commercial bus speed diagnosis based on gps-monitored data. *Transportation Research Part C: Emerging Technologies*, *19*, 695 – 707.

Elhenawy, M., Chen, H., y Rakha, H. (2014). Random forest travel time prediction algorithm using spatiotemporal speed measurements. *21st ITS World Congress Ann Arbor, USA*, *2014*.

Forum, J. A., y Villy Larsen, T. (2006). Travel-time estimation in road networks using gps data.

González, P. (2020). *Estimación de tiempos de respuesta de vehículos de bomberos en la ciudad de santiago, a partir de un microsimulador de tráfico* (Tesis de Master no publicada). Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile. (Departamento de Ingeniería Civil Industrial)

Hunter, T., Herring, R., Abbeel, P., y Bayen, A. (2009). Path and travel time inference from gps probe vehicle data.

Jiménez-Meza, A., Arámburo-Lizárraga, J., y de la Fuente, E. (2013). Framework for estimating travel time, distance, speed, and street segment level of service (los), based on gps data. *Procedia Technology*, *7*, 61 – 70.

Li, Y., Gunopulos, D., Lu, C., y Guibas, L. (2017). Urban travel time prediction using a small number of gps floating cars. *Proceedings of SIGSPATIAL'17, Los Angeles Area, CA, USA*, *11*, 10.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Sanaullah, I., Quddus, M., y Enoch, M. (2016). Developing travel time estimation methods using sparse gps data. *Journal of Intelligent Transportation Systems, Technology, Planning and Operations*, *20*, 532 – 544.

Vejar, B. (2019). *Predicción de velocidad en una autopista urbana utilizando estados de tráfico y modelos de inteligencia artificial* (Tesis de Master no publicada). Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile.

Wang, F., y Xu, Y. (2011). Estimating o–d travel time matrix by google maps api: implementation, advantages, and implications. *Annals of GIS*, *17*, 199 – 209.

Wang, Z., Goodchild, A., y McCormark, E. (2017). A methology for forecasting freeway travel time reliability using gps data. *Transportation Research Procedia*, *25*, 842 – 852.

Weng, J., Wang, C., Huang, H., Wang, Y., y Zhang, L. (2016). Real-time bus travel speed estimation model based on bus gps data. *Advances in Mechanical Engineering*, *8*, 1 – 10.

Woodard, D., Nogin, G., Koch, P., Racz, D., Goldszmidt, M., y Horvitz, E. (2016). Predicting travel time reliability using mobile phone gps data. *Transportation Research Part C: Emerging Technologies*, *75*, 30 – 44.