



UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA



Análisis de movilidad basado en datos de telefonía móvil del Gran Concepción

POR

Javier Andrés Serón Guzmán

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción
para optar al título profesional de Ingeniero Civil en Telecomunicaciones.

Profesor Guía

Gabriel Saavedra Mondaca

Concepción,
23 de octubre de 2024

© 2024 Javier Andrés Serón Guzmán

© 2024 Javier Andrés Serón Guzmán

Ninguna parte de esta tesis puede reproducirse o transmitirse bajo ninguna forma o por ningún medio o procedimiento, sin permiso por escrito del autor.

Agradecimientos

En primer lugar, me gustaría agradecer a las personas más importantes que han formado parte de este proceso, mis papás Cecilia y Marcelo quienes siempre me han brindado apoyo en todas las decisiones que he tomado a lo largo de este tiempo, teniendo que soportar todos los momentos en los cuales he estado estresado producto del proceso universitario. También agradecer a Radharani por apoyarme y brindarme ayuda en este último tiempo, ya sea dándome ánimo o ayudándome a estudiar para ramos de otra facultad de los cuales no eran de mi gusto.

Agradecer a los compañeros y profesores que siempre han estado presentes para apoyarme en este proceso, con mención especial a Francisca, Leonardo, Kevin, Ricardo y al profesor Gabriel por siempre estar apoyando durante el proceso del proyecto. Finalmente, agradecer al CityLab por permitir la realización de este proyecto y a Diego, quien me apoyo y guío durante todo el proceso de la práctica y realización del proyecto.

Gracias por todo

Índice General

Agradecimientos	I
Índice de Figuras	IV
Índice de Tablas	VII
Resumen	1
Capítulo 1	3
1.1. Introducción.	3
1.2. Definición del problema.	4
1.2.1. Objetivos	4
1.2.1.1. Objetivo general	4
1.2.1.2. Objetivos específicos	4
1.2.2. Alcances y limitaciones	4
1.3. Metodología	5
1.3.1. Obtener y realizar un análisis exploratorio de los datos de registro móvil y de las antenas.	5
1.3.2. Diseñar sistema metodológico para el procesamiento de los datos.	5
1.3.3. Procesar los datos según la zonificación.	5
1.3.4. Predecir modo de transporte.	6
1.3.5. Analizar y visualizar de los dataframes finales.	7
Capítulo 2	8
2.1. Estado del arte	8
2.1.1. Telefonía Móvil	8
2.1.2. Patrones de movilidad	10
2.1.3. Machine Learning	10
2.1.4. Grafos	11
2.2. Marco teórico	11
2.2.1. Red Móvil	11
2.2.2. Machine Learning Supervisado	12
2.2.3. Teoría de Grafos	13

	III
2.2.4. Python	14
2.2.5. Diagrama de Voronoi	16
Capítulo 3	17
3.1. Desarrollo	17
3.1.1. Obtener y realizar un análisis exploratorio de los datos de registro móvil y de las antenas.	17
3.1.2. Diseñar sistema metodológico para el procesamiento de los datos.	20
3.1.3. Procesar los datos para la zonificación comunal.	22
3.1.4. Procesar los datos para la zonificación de interés del CityLab.	24
3.1.5. Preparar el dataset para el modelo de machine learning supervisado.	28
3.1.6. Entrenar modelo de machine learning supervisado.	30
3.1.7. Crear un grafo para representar la movilidad.	32
3.1.8. Predecir modo de transporte.	35
3.1.9. Analizar y visualizar de los dataframes finales.	36
Capítulo 4	37
4.1. Resultados	37
4.1.1. Visualización comunal	37
4.1.2. Visualización zonificación CityLab	39
4.1.3. Visualización de rutas y modo de transporte con Machine Learning	40
4.1.4. Análisis mediante el grafo	41
Capítulo 5	45
5.1. Conclusión	45

Índice de Figuras

2.1. Ejemplo de celdas de las antenas. Fuente: [1].	9
2.2. Desplazamientos totales por zona. Fuente: [3].	10
2.3. Posible ruta de una persona por cadena de Márkov. Fuente: [6].	11
2.4. Ejemplo de red Móvil. Fuente: [12].	12
2.5. Diagrama de un grafo. Fuente: [13].	14
2.6. Diagrama de Voronoi aplicado a ciudad. Fuente: [16].	16
3.7. Dataframe original datos de antenas. Fuente: Autoría propia.	17
3.8. Dataframe original datos de personas. Fuente: Autoría propia.	17
3.9. Algoritmo para limpiar y transformar datos. Fuente: Autoría propia.	18
3.10. Dataframe datos de antenas en servicio. Fuente: Autoría propia.	18
3.11. Ubicación torre de antenas. Fuente: Autoría propia.	19
3.12. Zonificaciones a trabajar	20
3.13. Sistema para procesamiento de datos de Telefonía Móvil. Fuente: Autoría propia.	22
3.14. Algoritmo para procesar datos para zonificación comunal. Fuente: Autoría propia.	23
3.15. Dataframe zonificación comunal. Fuente: Autoría propia.	23
3.16. Dataframe densidad de personas zonificación comunal. Fuente: Autoría propia.	24
3.17. Algoritmo para procesar datos para zonificación de interés del CityLab. Fuente: Autoría propia.	25
3.18. Diagrama de Voronoi en base a ubicación torre de antenas. Fuente: Autoría propia.	26

3.19. Dataframe densidad de personas zonificación micro. Fuente: Autoría propia.	27
3.20. Dataframe nueva ubicación individual. Fuente: Autoría propia.	28
3.21. Algoritmo para preparar dataset para el modelo. Fuente: Autoría propia.	28
3.22. Dataframe de combinaciones basado en Google Maps. Fuente: Autoría propia.	29
3.23. Algoritmo para entrenar modelo. Fuente: Autoría propia.	30
3.24. Dataframes con parametros para el modelo. Fuente: Autoría propia.	30
3.25. Dataframes con parámetros de forma numérica para el modelo. Fuente: Autoría propia.	31
3.26. Algoritmo para creación del grafo. Fuente: Autoría propia.	32
3.27. Dataframe zonificación micro con punto central . Fuente: Autoría propia.	33
3.28. Dataframe con todas las rutas posibles entre zonas. Fuente: Autoría propia.	33
3.29. Grafo representativo en base a zonificación micro. Fuente: Autoría propia.	34
3.30. Dataframe con rutas individuales. Fuente: Autoría propia.	34
3.31. Algoritmo para predecir modo de transporte. Fuente: Autoría propia.	35
3.32. Dataframe de entrada para el modelo. Fuente: Autoría propia.	35
3.33. Dataframe de salida del modelo. Fuente: Autoría propia.	36
4.34. Mapa densidad de personas por comuna. Fuente: Autoría propia.	37
4.35. Heatmap basado en matriz de movilidad entre las 18 y 20 hrs. Fuente: Autoría propia.	38
4.36. Gráfico flujo entrada y salida de comunas. Fuente: Autoría propia.	39
4.37. Mapa densidad de personas por zona micro. Fuente: Autoría propia.	40
4.38. Rutas obtenidas del grafo en conjunto de lo predicho del modelo. Fuente: Autoría propia.	41

	VI
4.39. Comunidades de nodos. Fuente: Autoría propia.	42
4.40. Zona más transitada. Fuente: Autoría propia.	43
4.41. Comunidades de zonas en base a los nodos. Fuente: Autoría propia.	44

Índice de Tablas

3.1. Tabla precisión modelo de Machine Learning. Fuente: Autoría propia	31
4.2. Velocidad promedio por medio de transporte. Fuente: Autoría propia	42

Siglas

API Application Programming Interface

SVM Máquinas de Vectores de Soporte

Resumen

La presente memoria de título tiene como objetivo resolver problema originado en el Citylab, centrándose en el uso de datos de telefonía móvil para visualizar patrones de desplazamiento de las personas.

Se diseñó y desarrolló un sistema capaz de procesar datos de telefonía móvil, visualizar patrones de movimiento y obtener conclusiones utilizando teoría de grafos. Para las distintas etapas del sistema, se implementaron varios algoritmos. En primer lugar, se utilizó un algoritmo para la extracción y limpieza de los datos, obteniendo un conjunto depurado, libre de errores y actualizado.

Posteriormente, se aplicó un segundo algoritmo para procesar estos datos, usando como entrada el conjunto limpio. En esta fase, se consideraron factores como la posición de la torre de antena a la que se conectaba cada dispositivo, la hora, el número de dispositivo, la tecnología utilizada y el consumo de datos. Para determinar el rango de alcance de las torres de antena, se empleó el diagrama de Voronoi, una herramienta matemática. A continuación, se distribuyeron las personas proporcionalmente al área de la zona en la que se encontraban, identificando su posible ubicación.

Las siguientes etapas del sistema se ejecutaron en paralelo. Ambas utilizaron información proporcionada por Google Maps. La primera consistió en la creación de un conjunto de datos que incluyó todas las combinaciones posibles de puntos aleatorios en el Gran Concepción, obteniendo información sobre rutas, distancias, tiempos de viaje y modos de transporte utilizados. Estos datos fueron fundamentales para entrenar un modelo de machine learning supervisado que predijo el posible medio de transporte de las personas. En la segunda etapa, se generó un grafo en el que las zonas se representaron como nodos y las posibles rutas entre ellas como aristas. Estas rutas se obtuvieron de manera similar a la creación del dataset y se utilizaron para identificar las trayectorias de las personas entre zonas, proporcionando también datos sobre las distancias recorridas.

Tras procesar los datos con el grafo, el modelo estuvo listo para predecir los modos de transporte de las personas.

Finalmente, el grafo permitió obtener conclusiones como la distancia y velocidad medias,

el modo de transporte más utilizado, así como las zonas y rutas más transitadas. Además, se visualizó la información de manera que facilitara su comprensión, logrando un sistema capaz de procesar datos de telefonía móvil, visualizar patrones de movimiento y predecir modos de transporte mediante machine learning.

Capítulo 1

1.1. Introducción.

En los últimos años, el Gran Concepción ha enfrentado problemas logísticos que se manifiestan en la saturación de calles y servicios, especialmente en horas punta. Esta situación ha evidenciado la necesidad de implementar soluciones más eficientes para gestionar la movilidad urbana. En este contexto, en el CityLab Biobío, un espacio dedicado al estudio y la investigación que utiliza la ciencia y la tecnología para apoyar la toma de decisiones relacionadas con el crecimiento y desarrollo de las ciudades, surge la necesidad de procesar y analizar datos de telefonía móvil. Gracias a un convenio con Claro, el laboratorio recibió datos que, en su estado original, no permitían una interpretación clara para contribuir eficazmente a la toma de decisiones urbanas basadas en datos y predicciones de impacto.

El uso de datos de telefonía móvil representa una oportunidad valiosa, ya que permite obtener información detallada sobre el desplazamiento de las personas, generando un análisis preciso de la movilidad a través del tráfico de datos y el registro de llamadas. Esto permite identificar patrones de desplazamiento, áreas de alta concentración y flujos de movilidad, proporcionando información crucial para la toma de decisiones. Tener información precisa sobre la movilidad es esencial para la planificación urbana y para mejorar la calidad de vida, ya que facilita decisiones que optimizan el transporte público, reducen la congestión del tráfico, mejoran la seguridad y, lo que es aún más importante, promueven la accesibilidad a los servicios básicos, que muchas veces se concentran en el centro de las ciudades, dejando de lado a la periferia.

El objetivo de esta memoria es diseñar y desarrollar un sistema que, a través del procesamiento de datos de telefonía móvil, permita visualizar los patrones de movimiento de las personas y obtener conclusiones relevantes para mejorar la gestión de la movilidad en el Gran Concepción. Para lograrlo, se combinan conocimientos en telecomunicaciones, extracción y transformación de datos, y machine learning. Estos componentes se integran para predecir los modos de transporte y explicar las dinámicas de desplazamiento observadas.

Este enfoque no solo permite un análisis detallado de la movilidad actual, sino que también establece una metodología reproducible para futuras ingestas de datos, sentando las bases para

la escalabilidad del sistema y facilitando análisis futuros en diversas áreas urbanas.

1.2. Definición del problema.

Este proyecto propone el diseño e implementación de un sistema de análisis de movilidad basado en datos de telefonía móvil. El problema surge en el CityLab Biobío debido a la incapacidad de procesar y visualizar dichos datos, lo que limita su aprovechamiento como un recurso clave para la planificación urbana.

1.2.1. Objetivos

1.2.1.1. Objetivo general

Desarrollar un sistema de análisis de movilidad capaz de procesar datos de telefonía móvil, visualizar patrones de movimiento y obtener conclusiones mediante teoría de grafos.

1.2.1.2. Objetivos específicos

1. Obtener y realizar un análisis exploratorio de los datos de registro móvil y de las antenas.
2. Diseñar sistema para el procesamiento de los datos.
3. Procesar los datos según la zonificación.
4. Predecir modo de transporte.
5. Analizar y visualizar de los dataframes finales.

1.2.2. Alcances y limitaciones

Se trabajará exclusivamente con los datos proporcionados por Claro y con aquellos que se encuentren en condiciones adecuadas para su procesamiento. Estos datos se entregan en un archivo ".csv", el cual presenta información desordenada y con posibles errores. Además, la ubicación de los individuos no es completamente precisa, por lo que siempre habrá un margen

de error en la posición registrada. El análisis se enfocará en las horas de mayor relevancia dentro del área metropolitana de Concepción.

1.3. Metodología

1.3.1. Obtener y realizar un análisis exploratorio de los datos de registro móvil y de las antenas.

Para lograr este objetivo, se trabajará en conjunto con el CityLab, que se encargará de obtener los datos de registro móvil y antenas. Posteriormente, los datos serán sometidos a un proceso de análisis para filtrar únicamente aquellos que contengan información correcta.

1.3.2. Diseñar sistema metodológico para el procesamiento de los datos.

Se establecerá un sistema diseñado para organizar el proyecto de manera que sea comprensible y se divida en distintas etapas o estaciones, donde cada una cumple una función específica. Esto permitirá que el proyecto sea lo más escalable posible, con la flexibilidad de agregar o eliminar estaciones en el futuro. El sistema no estará limitado únicamente al procesamiento de datos de telefonía móvil, sino que, en el futuro, podrá gestionar cualquier tipo de datos, siempre que estos cumplan con el estándar requerido de contener un identificador, hora y ubicación.

1.3.3. Procesar los datos según la zonificación.

Para cumplir con el objetivo, el proceso se dividirá en dos partes: una basada en la zonificación comunal o macro, y otra utilizando la zonificación de interés del CityLab o micro.

En la zonificación macro, no se requiere una precisión exacta en la ubicación de las personas, dado que las zonas son más amplias que el rango de las antenas. El sistema verificará si las personas se encuentran dentro de las áreas definidas; en caso afirmativo, se les asignará el ID o índice de la zona correspondiente. Posteriormente, se contabilizará cuántas personas están asociadas a cada ID, obteniendo así el número de personas presentes en cada zona a lo largo del

tiempo.

En la zonificación micro, correspondiente al CityLab, se utilizará un diagrama de Voronoi para simular la cobertura de las antenas. Luego, se contará cuántas personas se encuentran dentro de estas áreas, siguiendo un proceso similar al descrito en el párrafo anterior.

El siguiente paso en la zonificación micro será la intersección entre las zonas de Voronoi y las zonas micro definidas por el CityLab. Si al menos el 51 % de una zona micro está contenida dentro de una zona Voronoi, esta se considerará una "zona hija", heredando la cantidad de personas de la "zona madre". Para evitar duplicidades, las personas se distribuirán proporcionalmente según el área de cada zona micro.

Finalmente, se registrará la ubicación de las personas en intervalos de 5 minutos, asignando su nueva posición al centro de la zona a la que pertenezcan. Este proceso permitirá obtener una distribución más precisa, mejorando la exactitud con respecto a la ubicación original captada por las antenas.

1.3.4. Predecir modo de transporte.

El proceso comenzará conectándose a la API de Google Maps y definiendo los puntos de las 10 comunas del Gran Concepción. A continuación, se establecerán la hora de inicio y fin, así como el modo de transporte. Con esta información, se generarán rutas para todas las combinaciones posibles de los puntos, con intervalos de 5 minutos, simulando el proceso de búsqueda en Google Maps para determinar cómo desplazarse de un lugar a otro y cuánto tiempo tomaría según el modo de transporte.

El resultado será un *dataframe* que contendrá información sobre el tiempo, la distancia, la hora del trayecto y el modo de transporte. Las combinaciones de puntos podrán ser locales (dentro de la misma comuna) o entre comunas.

El *dataset* obtenido en el paso anterior será procesado seleccionando las columnas necesarias y transformando los datos: la distancia se convertirá a kilómetros, el tiempo a minutos, y se calculará la velocidad promedio en km/min según el modo de transporte. Además, los modos de transporte se codificarán como 1 para "driving", 2 para "walking" y 3 para "bicycling".

El *dataset* transformado se utilizará para entrenar un modelo de predicción con *Random Forest*, que empleará parámetros como el tiempo, la distancia y la velocidad para predecir el

modo de transporte.

Paralelamente, se creará un grafo a partir de un *dataframe* que contendrá las zonas micro con sus respectivos centros e identificaciones. Luego, se conectará a la API de Google Maps para obtener las rutas y combinaciones posibles entre los centros de las zonas micro, generando un nuevo *dataframe* con la distancia, la zona de inicio y la zona final. Esta información se utilizará para crear el grafo, donde los centros de las zonas serán los nodos y las rutas, distancias y tiempos serán las aristas. Finalmente, se integrará otro *dataframe* con la ubicación y el tiempo de las personas, mostrando las rutas individuales entre nodos.

Posteriormente, se importará el modelo de *Random Forest* previamente entrenado, que será alimentado con la información obtenida para predecir el modo de transporte, generando un nuevo *dataframe* con las predicciones del modelo.

1.3.5. Analizar y visualizar de los dataframes finales.

En esta etapa final, se integrarán todos los *dataframes* y los resultados del procesamiento previo para visualizar los datos de manera clara. Se crearán gráficos como mapas de densidad, matrices de movilidad y representaciones de las rutas seguidas por las personas en un periodo determinado. El uso de grafos permitirá un análisis detallado del comportamiento de las personas.

Capítulo 2

2.1. Estado del arte

Los desafíos de la movilidad urbana son especialmente relevantes en ciudades de rápido crecimiento, como las de América Latina y el Caribe. En este contexto, se destaca el potencial de los grandes datos provenientes de fuentes como GPS, registros telefónicos y aplicaciones móviles para la planificación del transporte. Los datos generados por teléfonos móviles, que capturan la ubicación y los movimientos de los usuarios, constituyen un recurso valioso para comprender los patrones de movilidad urbana.

Si bien estos datos ofrecen ventajas, como la recopilación continua de información y costos más bajos en comparación con las encuestas tradicionales, también presentan desafíos. Entre estos se incluyen las complejidades en el procesamiento de datos y la limitada información disponible sobre los viajeros.

2.1.1. Telefonía Móvil

Desde el inicio de la telefonía móvil, el comportamiento de los usuarios a lo largo del día ha sido un tema de interés para muchos investigadores y empresas. Para comprender mejor cómo funcionan las redes de telefonía móvil, es importante considerar que se basan en el concepto de celdas. Esto significa que las áreas alrededor de cada antena se dividen en zonas para cubrir una determinada área geográfica. Las redes celulares utilizan un transmisor-receptor central en cada celda, conocido como estación base. [1]

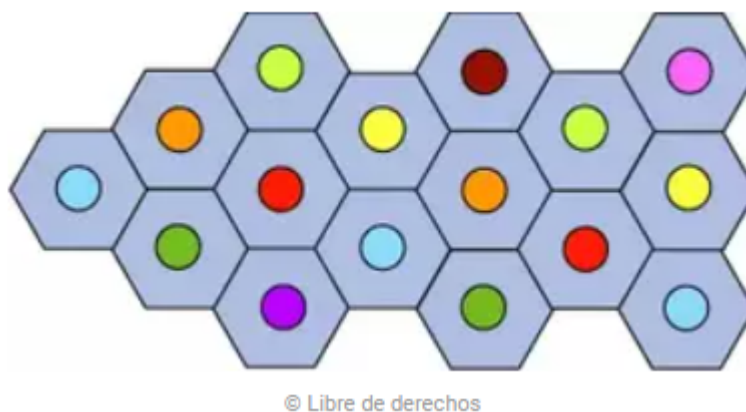


Fig. 2.1: Ejemplo de celdas de las antenas. Fuente: [1].

Entendiendo lo mencionado anteriormente, se puede explicar cómo las personas, al entrar en una zona, se conectan a la antena. Aunque sabemos que una persona se encuentra en esa área, no se puede determinar su ubicación exacta, lo que motiva revisiones de procesamiento que se describirán más adelante. Los teléfonos móviles, gracias a su capacidad para rastrear la ubicación de los usuarios, proporcionan datos valiosos sobre la movilidad.

Por ejemplo, en 2020 en España [2], se realizaron estudios sobre datos de telefonía móvil para entender cómo las restricciones afectaron los patrones de movimiento. Se observó una disminución significativa en el número de viajes y en las distancias recorridas, especialmente durante las fases más estrictas de las restricciones. Gracias al análisis de estos datos, fue posible monitorear la movilidad durante la pandemia y ajustar las medidas de restricción según la evolución de la situación epidemiológica [3].

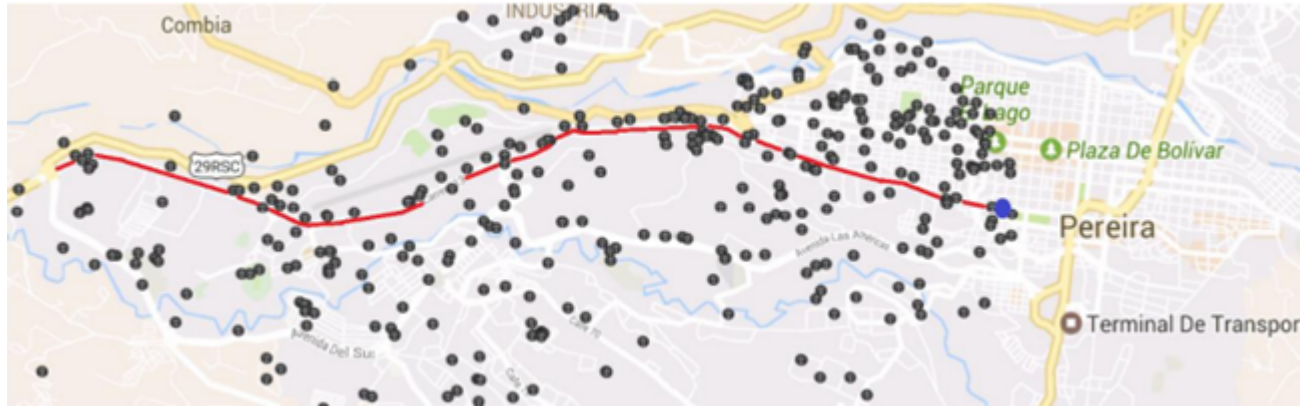


Fig. 2.3: Posible ruta de una persona por cadena de Márkov. Fuente: [6].

2.1.4. Grafos

El análisis de ciudades mediante la Teoría de Grafos es muy útil, ya que permite visualizar la ciudad de una manera diferente, considerando las esquinas más importantes como nodos y las calles que las conectan como enlaces. Esto facilita el análisis [7], ya que se pueden utilizar algoritmos para determinar qué ruta se tomó para llegar de un punto a otro.

2.2. Marco teórico

2.2.1. Red Móvil

Para entender el problema que se presenta al intentar obtener la posición exacta de un individuo que se conecta a una antena, es fundamental comprender el concepto básico de telefonía celular y los parámetros necesarios para precisar dicha posición. La telefonía celular busca conectar a dos usuarios remotos a través del equipo de red de un operador que gestiona el servicio, pero de forma inalámbrica, utilizando transmisiones de radiofrecuencia. El teléfono móvil de un usuario debe estar dentro del área de alcance de una antena, que está limitada por la potencia de la antena y la interferencia. Esta área se denomina celda, como se muestra en la Fig. 2.4, donde se puede observar la interacción entre las antenas y su alcance. A pesar de conocer todas las especificaciones de las antenas y los datos proporcionados por el usuario, sigue siendo difícil determinar la posición exacta de la persona.

Además de los factores mencionados, el tamaño de las celdas varía según si se trata de

una zona urbana o rural. En una zona urbana, donde hay una alta densidad de población y un mayor volumen de comunicaciones, las celdas tienden a ser numerosas y pequeñas, con distancias de cobertura que pueden ser de cientos o solo decenas de metros. Por el contrario, en las zonas rurales, donde la densidad es baja, las celdas son mucho más grandes, abarcando varios kilómetros, aunque rara vez superan los diez kilómetros.



Fig. 2.4: Ejemplo de red Móvil. Fuente: [12].

2.2.2. Machine Learning Supervisado

La inteligencia artificial está en boca de todos en la actualidad, ya que se ha convertido en una herramienta muy útil para optimizar procesos. El Machine Learning supervisado, que es una parte del amplio campo de la inteligencia artificial, utiliza datos etiquetados para entrenar algoritmos. Esto significa que se le proporciona al modelo la respuesta correcta, y este ajustará sus parámetros hasta obtener resultados más precisos.

Machine Learning supervisado se puede dividir en dos tipos de problemas:

- **Clasificación:** Se utiliza para asignar datos de prueba a categorías específicas. Un ejemplo sería clasificar correos electrónicos como spam o no spam. Algoritmos populares incluyen clasificadores lineales, máquinas de vectores de soporte (SVM), árboles de decisión, k vecinos más cercanos y random forest.
- **Regresión:** Este tipo de problema busca entender la relación entre variables dependientes e independientes, útil para hacer proyecciones como los ingresos futuros de una empresa.

Algoritmos comunes de Machine Learning Supervisado:

- **Naive Bayes:** Este clasificador asume que las características son independientes entre sí, un principio basado en el Teorema de Bayes. Se usa ampliamente para la clasificación de textos y detección de spam.
- **Regresión Lineal:** Utilizada para identificar relaciones entre variables, buscando trazar una línea de mejor ajuste en el conjunto de datos.
- **Máquinas de Vectores de Soporte (SVM):** Este modelo clasifica los datos separándolos mediante un hiperplano que maximiza la distancia entre las clases.
- **Gradient Bossting:** Combina múltiples árboles de decisión de manera secuencial, donde cada árbol nuevo corrige los errores del anterior, mejorando así la precisión. Ajusta los árboles en función de los errores residuales, pudiendo encontrar relaciones complejas en los datos.
- **K Vecino más Cercano (KNN):** Clasifica puntos de datos según su proximidad con otros datos. Es común en motores de recomendación.
- **Random Forest:** Combina múltiples árboles de decisión para mejorar la precisión de las predicciones, reduciendo la varianza en los resultados.

Los modelos que mejor se ajustan a la naturaleza del problema son el SVM, Random Forest y Gradient Bossting, pero donde los conjuntos de datos son pequeños el Random Forest tiende ser menos propenso al sobreajuste y tiene un menor tiempo de entrenamiento que el Gradient Bossting, también el Random Forest es más fácil de interpretar que SVM.

2.2.3. Teoría de Grafos

La teoría de grafos es un campo de las matemáticas y la informática que estudia las propiedades de los grafos. Un grafo es una pareja de conjuntos que se representa de la siguiente manera:

$$G = (V, A),$$

donde V es el conjunto de vértices y A las aristas, en el caso de los vértices es un conjunto de pares (u, v) tal que u, v pertenecen a V y las aristas se pueden escribir como ab .

Representaciones de Grafos:

- **Lista de Incidencia:** Las aristas se representan mediante un vector de pares, donde cada par indica una conexión entre dos vértices.
- **Lista de Adyacencia:** Cada vértice tiene una lista de otros vértices adyacentes, facilitando las búsquedas a costa de requerir más almacenamiento.
- **Matriz de Incidencia:** Representa las aristas en una matriz donde las filas son aristas y las columnas vértices.
- **Matriz de Adyacencia:** Es una matriz cuadrada donde se indica la presencia de una arista entre dos vértices mediante un valor de 1.

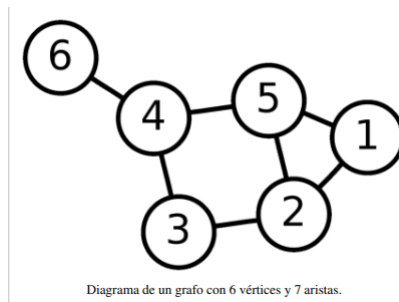


Fig. 2.5: Diagrama de un grafo. Fuente: [13].

Un grafo puede tener múltiples usos; por ejemplo, puede modelar una red de carreteras en la que los nodos son las ciudades, una red de drenaje de la ciudad o incluso una red óptica, donde las aristas son las fibras ópticas y los nodos son las estaciones. Los grafos permiten obtener conclusiones valiosas sobre la eficiencia y resiliencia de las redes que representan..

2.2.4. Python

Python es un lenguaje de programación de código abierto, creado por Guido van Rossum en 1991. Se caracteriza por estar orientado a objetos y tener una sintaxis fácil de leer, similar al inglés, lo que facilita su aprendizaje y uso. Python es ampliamente utilizado en el análisis de

datos, ya que permite la representación visual efectiva de grandes volúmenes de información, convirtiéndose en una herramienta fundamental en el campo de la ciencia de datos.

Dentro de la ciencia de datos, Python ofrece acceso a diversas bibliotecas para la visualización de datos, como gráficos de barras, líneas y circulares, lo que simplifica el análisis de patrones complejos. Además, su capacidad para gestionar grandes volúmenes de datos mediante estructuras como diccionarios, DataFrames y listas lo hace invaluable para la manipulación y análisis de información.

Python también es muy utilizado en el campo del machine learning. El aprendizaje automático utiliza algoritmos para analizar grandes cantidades de datos, destacándose por su gran capacidad para manejar volúmenes masivos de información simultáneamente y por contar con una amplia variedad de bibliotecas que abordan diferentes tipos de problemas, como por ejemplo:

- **Pandas:** Una de las librerías más utilizadas para el análisis de datos. Proporciona estructuras como DataFrames y Series que permiten manipular, limpiar y analizar datos eficientemente.
- **GeoPandas:** Amplia los tipos de datos que utiliza Pandas para trabajar con datos geoespaciales. Facilita el análisis y visualización de datos geográficos.
- **Matplotlib:** Es una librería que sirve para crear visualizaciones estáticas y dinámicas, Permitiendo la creación de gráficos variados.
- **OSMNX:** Ofrece el poder modelar, analizar y visualizar redes urbanas y otros parámetros geoespaciales utilizando OpenStreetMap.
- **Scikit-Learn:** Es una librería para crear y probar modelos de machine learning, incluyendo clasificación, regresión y clustering.
- **Seaborn:** Biblioteca basada en Matplotlib, que nos sirve para hacer visualizaciones estadísticas atractivas.
- **Networkx:** Librería para la creación y análisis de grafos y redes.
- **GoogleMaps:** Biblioteca para interactuar con la API de Google Maps, permitiendo la obtención de rutas y datos de ubicación.

2.2.5. Diagrama de Voronoi

El Diagrama de Voronoi es una herramienta matemática que permite dividir el plano en regiones según la proximidad a un conjunto de puntos. Cada región de Voronoi está formada por los puntos que están más cerca de un centro que de cualquier otro centro de otra región. A través de este diagrama, se pueden establecer los límites de las áreas de influencia de cada punto utilizado como centro, como por ejemplo en la ubicación de torres de antenas.

Este tipo de diagramas es muy útil en la geografía, donde se emplea para analizar la accesibilidad a servicios o simular áreas de alcance a partir de un punto central. También se utilizan en las ciencias de la computación, por ejemplo, en el procesamiento de imágenes, permitiendo la segmentación, identificación de patrones y su aplicación en inteligencia artificial.



Fig. 2.6: Diagrama de Voronoi aplicado a ciudad. Fuente: [16].

Capítulo 3

3.1. Desarrollo

3.1.1. Obtener y realizar un análisis exploratorio de los datos de registro móvil y de las antenas.

La obtención de datos de registro móvil y antenas se lleva a cabo en conjunto con el City-Lab, gracias al acceso que posee el laboratorio. Existen dos tipos de datos: uno que contiene información sobre la ubicación, tecnología y estado de las antenas, como se aprecia en la Fig.3.7.

cgi_hex	region	comuna	status	red	fecha_actualiza	longitud	latitud
[REDACTED]	VIII	PEMUCO	EN O&M	3 G	2018-12-19	-71.84683	-37.02167
[REDACTED]	VIII	TIRUA	EN O&M	3 G	2018-12-19	-73.50986	-38.45742
[REDACTED]	VIII	CARAHUE	EN O&M	3 G	2018-12-19	-73.48903	-38.47764
[REDACTED]	VIII	TIRUA	EN O&M	3 G	2018-12-19	-73.50986	-38.45742
[REDACTED]	VIII	CONTULMO	EN O&M	3 G	2018-12-19	-73.20928	-38.18931
--	--	--	--	--	--	--	--
[REDACTED]	VIII	0-MULCHEN	EN SERVICIO	2G	NaN	NaN	NaN
[REDACTED]	VIII	0-TOME	EN SERVICIO	2G	NaN	NaN	NaN
[REDACTED]	VIII	0-TALCAHUANO	EN SERVICIO	2G	NaN	NaN	NaN
[REDACTED]	VIII	0-QUIRIBUE	EN SERVICIO	3 G	NaN	NaN	NaN
[REDACTED]	VIII	0-SAN FABIAN	IMPLEMENTACION	3 G	NaN	NaN	NaN

Fig. 3.7: Dataframe original datos de antenas. Fuente: Autoría propia.

Otro que contiene datos sobre las personas, incluyendo la ubicación de la antenna a la que se conectaron, la hora de conexión, la tecnología de la antenna y el consumo registrado, como se muestra en la Fig.3.8.

FECHA	HORA	PCS	REGION	COMUNA	NOMBRE SITIO	CANTIDAD_CLIENTES_UNICOS	TIPO TECNOLOGIA	MB	GB	LATITUD	LONGITUD	LAC	
20231004	0		VIII	CONCEPCIÓN	Ruta Concepcion Penco	2	1	LTE	9.765815	0.009537	-36.796300	-73.040197	28382.0
20231004	0		VIII	FLORIDA	Florida		1	LTE	9.766039	0.009537	-36.819858	-72.655858	28382.0
20231004	0		VIII	LOTA	Colcura		1	LTE	9.765701	0.009537	-37.098462	-73.146252	28331.0
20231004	0		VIII	SAN PEDRO DE LA PAZ	Michahue		1	3G	0.011769	0.000011	-36.849472	-73.142750	8382.0
20231004	0		VIII	TALCAHUANO	C° David Fuentes		1	LTE	9.775058	0.009546	-36.718469	-73.114500	28382.0

Fig. 3.8: Dataframe original datos de personas. Fuente: Autoría propia.

El problema con estos datos es que contienen información errónea o desactualizada, como antenas que ya no están en funcionamiento. Por ello, los datos deben pasar por un proceso

de transformación, donde se convertirán en tipo de datos georreferenciales. En este proceso, se cambia el formato de la información de la ubicación, pasando de estar representados como latitud y longitud de forma separada a ser puntos referenciados en un mapa. Además, se lleva a cabo un proceso de limpieza, eliminando todos los registros incorrectos, como se ejemplifica en la Fig. 3.9.



Fig. 3.9: Algoritmo para limpiar y transformar datos. Fuente: Autoría propia.

Quedando solo con las antenas que están en funcionamiento y con la información correcta como se ve en la Fig. 3.10.

cgi_hex	region	comuna	status	red	fecha_actualiza	longitud	latitud
[REDACTED]	VIII	PEMUCO	EN O&M	3 G	2018-12-19	-71.846830	-37.021670
[REDACTED]	VIII	TIRUA	EN O&M	3 G	2018-12-19	-73.509860	-38.457420
[REDACTED]	VIII	CARAHUE	EN O&M	3 G	2018-12-19	-73.489030	-38.477640
[REDACTED]	VIII	TIRUA	EN O&M	3 G	2018-12-19	-73.509860	-38.457420
[REDACTED]	VIII	CONTULMO	EN O&M	3 G	2018-12-19	-73.209280	-38.189310

Fig. 3.10: Dataframe datos de antenas en servicio. Fuente: Autoría propia.

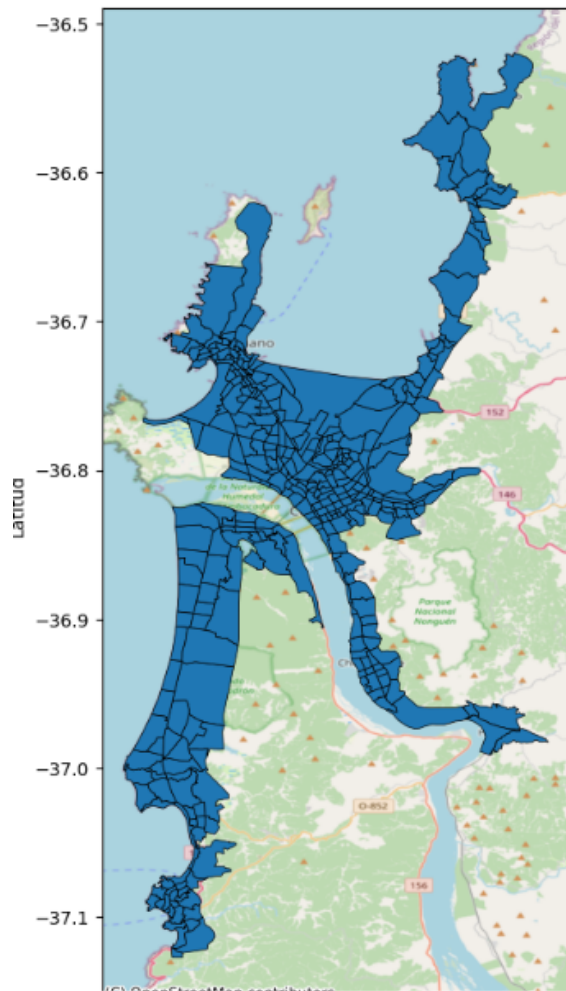
El DataFrame original de los datos de las personas contenía información en correcto estado, donde lo más importante es la ubicación y la hora en que se conectó dicha persona, como se puede ver en la Fig. 3.8.

Al visualizar los datos de las antenas, se aprecia dónde están ubicadas las torres de antenas, distribuidas en distintos puntos del Gran Concepción, como se ve en la Fig. 3.11.

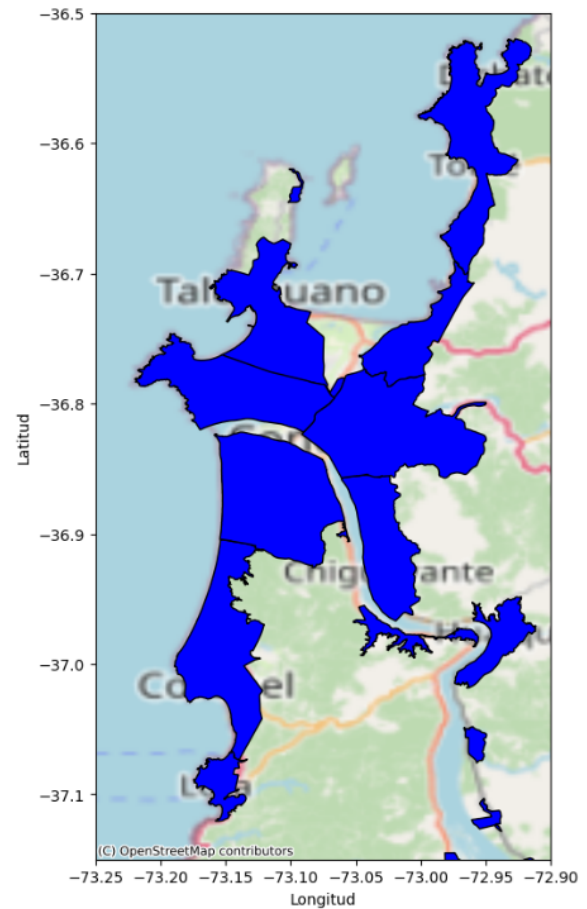


Fig. 3.11: Ubicación torre de antenas. Fuente: Autoría propia.

Además, dado que este es un trabajo en conjunto con el CityLab, ellos proporcionaron una zonificación de interés específica para sus investigaciones. Adicionalmente, se seleccionó una zonificación comunal, como se muestra en la Fig. 3.12.



(a) Zonificación CityLab



(b) Zonificación Comunal

Fig. 3.12: Zonificaciones a trabajar

3.1.2. Diseñar sistema metodológico para el procesamiento de los datos.

La idea principal detrás del diseño de este sistema es definir claramente las funciones de las distintas estaciones o etapas durante el procesamiento de los datos. Al tener estas funciones bien definidas, se garantiza un sistema escalable, que puede seguir creciendo para procesar otros tipos de datos o mejorar la exactitud del procesamiento, sin necesidad de comenzar desde cero. Además, este enfoque facilita la comprensión del flujo de trabajo del proyecto.

De manera general, en la Fig. 3.13 se presentan las partes necesarias para llevar a cabo este procesamiento de datos, las cuales se explican en mayor detalle a lo largo del presente informe.

En el bloque identificado como ETL, se representa el proceso mencionado anteriormente, donde se llevan a cabo pasos para la transformación y limpieza de los datos. Siguiendo el flujo del esquema, el sistema se divide en dos partes: una correspondiente a la zonificación proporcionada por el CityLab (micro) y otra para la zonificación comunal o macro.

Los datos que salen de los bloques *MacroZone* y *MicroZone*, representados por una línea azul, contienen información sobre la densidad de personas por sector y se dirigen al bloque *Plot*, que se encarga de la visualización de los datos. Por otro lado, desde el bloque *MicroZone* sale una flecha roja que contiene los datos individuales con las ubicaciones asignadas, los cuales entran al bloque *Grafo*, donde se obtiene la ruta que sigue cada persona para desplazarse de un lugar a otro, y esta información se transfiere al bloque *Predict*. También se observa que los bloques *Grafo* y *Maps*, donde se genera el *dataset* para entrenar el modelo de *machine learning*, están alimentados por la API de Google Maps. En el bloque *Machine Learning*, se entrena el modelo con el *dataset* que pasa del bloque *Maps* al bloque *Machine Learning*. Finalmente, en el bloque *Predict*, se obtiene el modo de transporte probable para cada persona, y esta información se dirige al bloque *Plot* para su visualización.

Es importante destacar que todos los datos procesados quedan disponibles para ser almacenados en una base de datos o como archivos, permitiendo su visualización dinámica. No obstante, el bloque *Plot* ofrece una visualización estática. Este esquema tiene el propósito de organizar el procedimiento del proyecto, hacerlo escalable y facilitar la comprensión del flujo del mismo. Lo que se realiza dentro de cada bloque se explica en mayor detalle en los apartados correspondientes de esta sección de desarrollo.

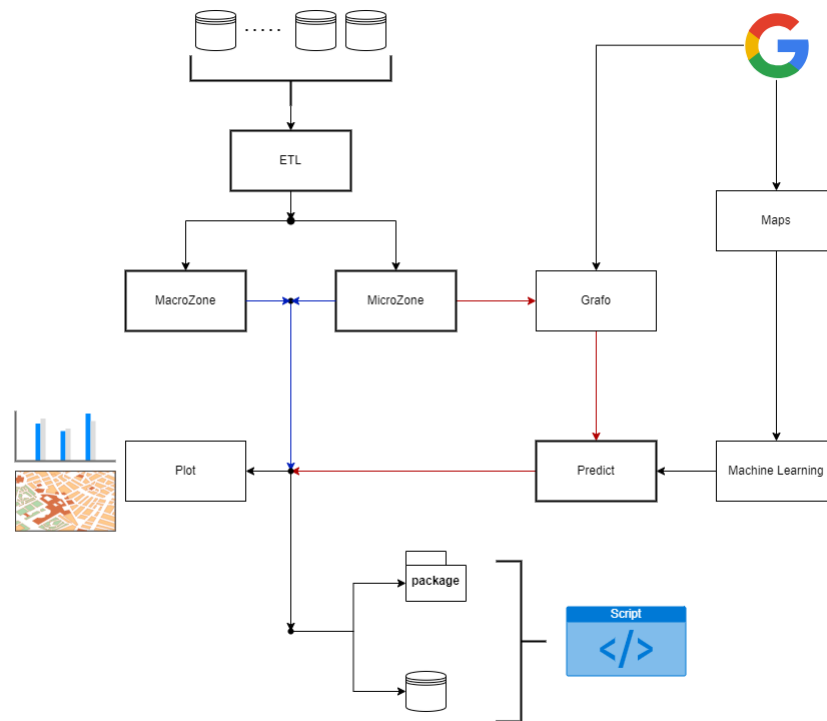


Fig. 3.13: Sistema para procesamiento de datos de Telefonía Móvil. Fuente: Autoría propia.

3.1.3. Procesar los datos para la zonificación comunal.

El algoritmo que se ve en la Fig.3.14 ayuda a entender el flujo para poder procesar los datos para la zonificación comunal.

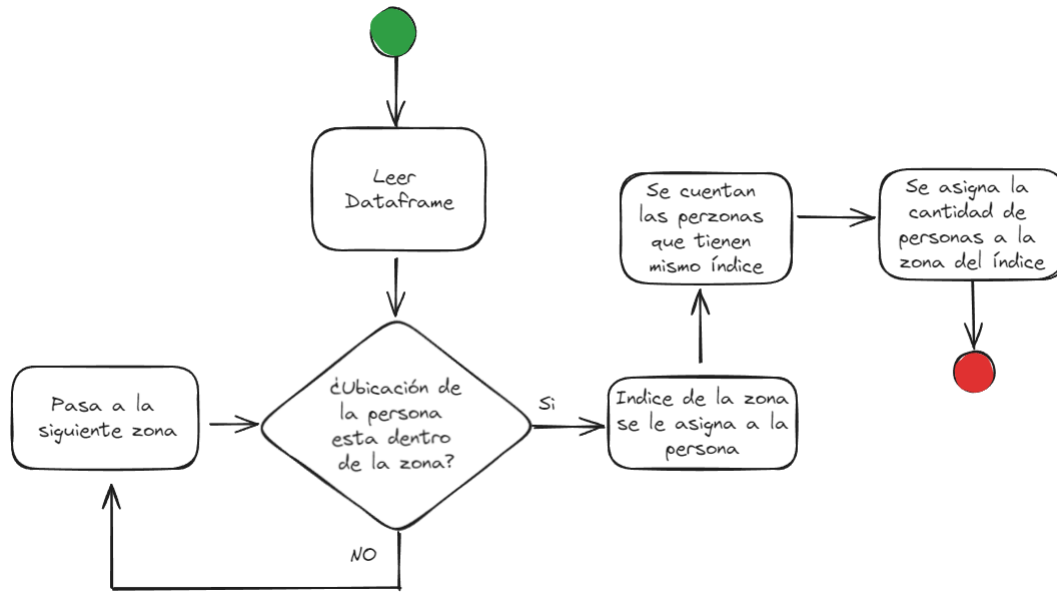


Fig. 3.14: Algoritmo para procesar datos para zonificación comunal. Fuente: Autoría propia.

Siguiendo el algoritmo lo primero es leer el DataFrame correspondiente a la zonificación, como se ve en la Fig. 3.15, que contiene la información de los polígonos que conforman las comunas. Luego, se lee el DataFrame de las personas de la Fig. 3.8, que incluye la ubicación de las antenas a las que se conectaron.

	COM	geometry
18	Coronel	POLYGON ((-73.12223 -36.91114, -73.12211 -36.9...
19	Coronel	POLYGON ((-73.52500 -36.97123, -73.52500 -36.9...
20	Coronel	POLYGON ((-73.51356 -37.03377, -73.51125 -37.0...
21	Coronel	POLYGON ((-73.03362 -36.96458, -73.03360 -36.9...
22	Lota	POLYGON ((-73.14034 -37.08569, -73.14088 -37.0...
23	Penco	POLYGON ((-72.96346 -36.70036, -72.96348 -36.7...
24	San Pedro de La Paz	POLYGON ((-73.13430 -36.82244, -73.12825 -36.8...
25	Talcahuano	POLYGON ((-73.10376 -36.68692, -73.10371 -36.6...
26	Tomé	POLYGON ((-72.79945 -36.63494, -72.79892 -36.6...
39	Talcahuano	POLYGON ((-73.09067 -36.63192, -73.09071 -36.6...
40	Tomé	POLYGON ((-72.92464 -36.54388, -72.92528 -36.5...
42	Hualqui	POLYGON ((-72.94809 -36.98381, -72.94789 -36.9...
43	Chiguayante	POLYGON ((-73.02432 -36.85572, -73.02407 -36.8...
44	Concepción	POLYGON ((-73.02333 -36.78681, -73.02242 -36.7...
59	Hualqui	POLYGON ((-72.96025 -37.04790, -72.96014 -37.0...
60	Hualqui	POLYGON ((-72.95158 -37.10316, -72.95136 -37.1...
61	Hualqui	POLYGON ((-72.92820 -37.15369, -72.92792 -37.1...

Fig. 3.15: DataFrame zonificación comunal. Fuente: Autoría propia.

Luego de haber leído los dataframes correspondientes seguimos con el siguiente paso del algoritmo, dado que las zonas comunales son amplias, no es necesario realizar un procesamiento adicional para aumentar la precisión de la ubicación de las personas. Por lo tanto, se intersectan

las zonas con la ubicación de la antena a la que se conectó cada persona, verificando si dicha ubicación se encuentra dentro del polígono de la zona correspondiente, haciéndose siempre la misma pregunta si es que la persona se encuentra dentro de la zona, de no ser así, pasa a la siguiente zona volviendo a repetir el proceso hasta finalizar con todas las personas.

Si la persona se encuentra dentro de una zona pasamos al siguiente paso en donde se le asigna el índice o ID de esa zona. Luego en el siguiente bloque se cuentan las personas con el mismo índice y en el último paso se le asigna el número total de personas relacionadas con ese índice a la zona correspondiente. El resultado es un DataFrame en el que se muestra la cantidad de personas en la columna "PCS" y la hora en que se registró esa cantidad de personas.

Cabe destacar que algunas comunas aparecen repetidas debido a la zonificación utilizada, que divide las comunas en sectores. Sin embargo, estos sectores siguen siendo relativamente grandes. Los resultados finales se muestran en la Fig. 3.16.

	COM	geometry	Indice	PCS	Generated_Time
18	Coronel	POLYGON ((-73.12223 -36.91114, -73.12211 -36.9...	18	44.0	18:00:00
19	Coronel	POLYGON ((-73.52500 -36.97123, -73.52500 -36.9...	19	0.0	18:00:00
20	Coronel	POLYGON ((-73.51356 -37.03377, -73.51125 -37.0...	20	2.0	18:00:00
21	Coronel	POLYGON ((-73.03362 -36.96458, -73.03360 -36.9...	21	7.0	18:00:00
22	Lota	POLYGON ((-73.14034 -37.08569, -73.14088 -37.0...	22	10.0	18:00:00
...	...	...	...	...	...
748	Concepción	POLYGON ((-73.02333 -36.78681, -73.02242 -36.7...	44	132.0	18:55:00
763	Hualqui	POLYGON ((-72.96025 -37.04790, -72.96014 -37.0...	59	0.0	18:55:00
764	Hualqui	POLYGON ((-72.95158 -37.10316, -72.95136 -37.1...	60	0.0	18:55:00
765	Hualqui	POLYGON ((-72.92820 -37.15369, -72.92792 -37.1...	61	0.0	18:55:00
767	Hualpén	POLYGON ((-73.11523 -36.77579, -73.11489 -36.7...	63	26.0	18:55:00

216 rows × 5 columns

Fig. 3.16: Dataframe densidad de personas zonificación comunal. Fuente: Autoría propia.

3.1.4. Procesar los datos para la zonificación de interés del CityLab.

Para esta zonificación la cual se denomina zonificación micro para diferenciarla de la comunal, el procesamiento es más largo como se puede ver en la Fig.3.17, donde se ve el algoritmo para entender el flujo de procesamiento de estos datos.

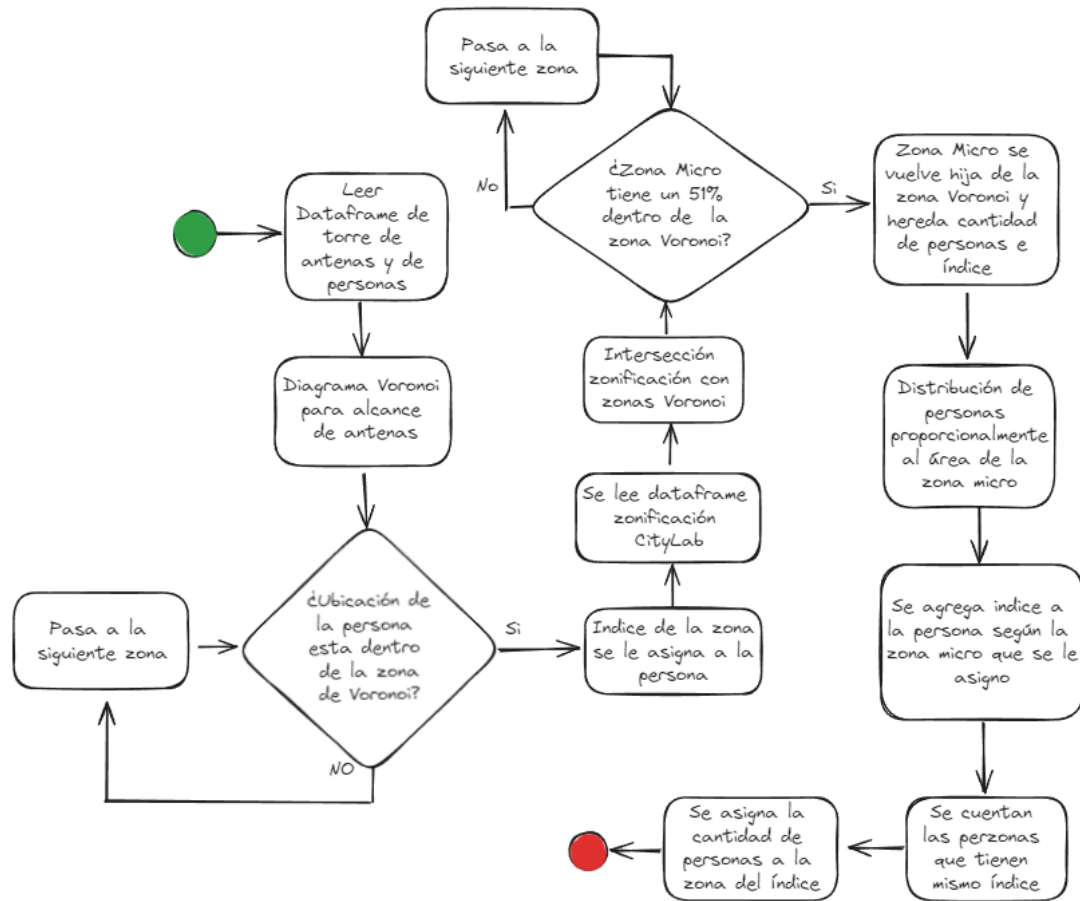


Fig. 3.17: Algoritmo para procesar datos para zonificación de interés del CityLab. Fuente: Autoría propia.

Siguiendo el flujo del algoritmo, el proceso comienza en el primer bloque, donde se leen los *dataframes* de entrada: el correspondiente a las antenas, como se muestra en la Fig. 3.10, y el que contiene la información de las personas, ilustrado en la Fig. 3.8. Por el momento, no se incluye la lectura del *dataframe* de zonificación proporcionado por el CityLab.

Dado que no se conoce con precisión el alcance de las antenas, en el segundo bloque se simula el rango de cobertura de las torres mediante un diagrama de Voronoi. Este diagrama se genera midiendo las distancias entre todos los puntos, lo que permite calcular y delimitar las zonas resultantes, tal como se muestra en la Fig. 3.18.

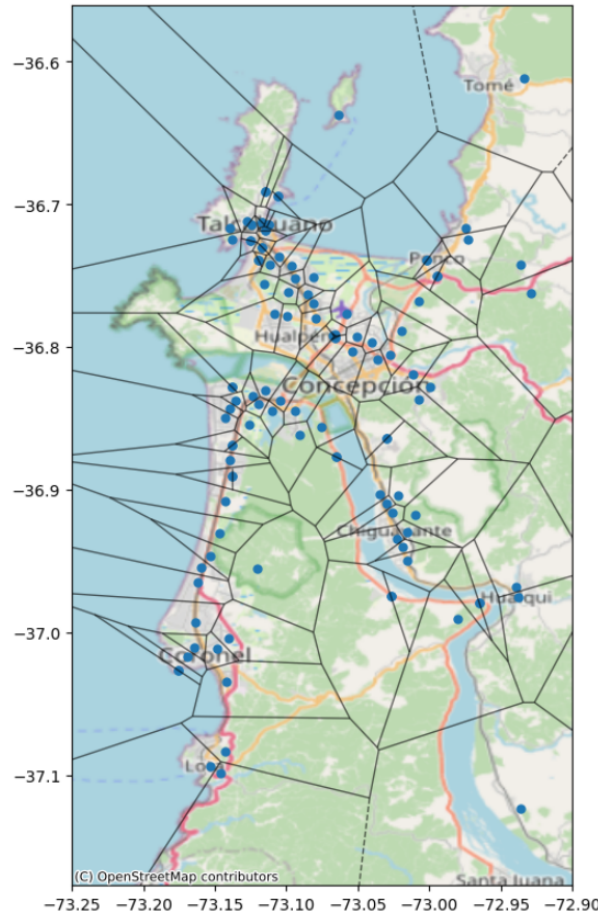


Fig. 3.18: Diagrama de Voronoi en base a ubicación torre de antenas. Fuente: Autoría propia.

Al igual que en el caso anterior, en el tercer bloque se cuenta la cantidad de personas que se encuentran dentro de las zonas generadas por el diagrama de Voronoi. A cada persona se le asigna un índice si está dentro de una zona; en caso contrario, se evalúa la siguiente zona, repitiendo este proceso hasta procesar a todas las personas. Posteriormente, en el siguiente bloque, se procede a leer el *DataFrame* de la zonificación proporcionada por el CityLab. En el bloque posterior, se intersectan las zonas micro, definidas por el CityLab, con las zonas generadas por el diagrama de Voronoi. Si al menos el 51 % del área de una zona micro se encuentra dentro de una zona del diagrama de Voronoi, esta se convierte en una "zona hija" de la zona de Voronoi, heredando la cantidad de personas de la "zona madre". Si no cumple con el porcentaje necesario, se pasa a la siguiente zona y se repite el proceso hasta que todas las zonas queden asignadas.

En el bloque *"Distribución proporcional de personas según el área de la zona micro"*, se reciben los datos del bloque anterior, pero con duplicaciones. La función de este bloque es resolver este problema dividiendo el total de personas en la zona de Voronoi entre las zonas hijas que contiene, asignando las personas de manera proporcional al área de cada zona micro.

Por ejemplo, si una zona de Voronoi tiene 100 personas y tres zonas hijas, donde dos miden $25m^2$ cada una y la tercera mide $50m^2$, se asignarán 50 personas a la zona más grande y 25 a cada una de las zonas más pequeñas. Se asume que, a mayor área de una zona micro, mayor será la cantidad de personas que la ocupen.

El resultado es un *DataFrame* que indica cuántas personas se encuentran en cada zona micro, como se observa en la Fig. 3.19.

ZONA		geometry	pcs_zone	Generated_Time
0	842	POLYGON ((-73.03576 -36.82071, -73.03580 -36.8...	10	18:00:00
1	272	POLYGON ((-73.10782 -36.72671, -73.10778 -36.7...	0	18:00:00
2	301	POLYGON ((-73.13148 -36.72311, -73.13142 -36.7...	0	18:00:00
3	240	MULTIPOLYGON (((-73.11570 -36.71893, -73.11571...	0	18:00:00
4	511	POLYGON ((-73.09561 -36.74974, -73.09539 -36.7...	0	18:00:00
...	...	...	...	...
4435	410	POLYGON ((-73.08325 -36.73274, -73.07974 -36.7...	0	18:55:00
4436	50	POLYGON ((-72.95561 -36.61593, -72.95553 -36.6...	0	18:55:00
4437	73	POLYGON ((-72.95305 -36.62311, -72.95305 -36.6...	0	18:55:00
4438	290	POLYGON ((-73.11826 -36.72200, -73.11813 -36.7...	0	18:55:00
4439	181	POLYGON ((-72.99852 -36.75543, -72.99715 -36.7...	0	18:55:00

4440 rows × 4 columns

Fig. 3.19: Dataframe densidad de personas zonificación micro. Fuente: Autoría propia.

En el antepenúltimo bloque del algoritmo, a las personas se les asigna un ID correspondiente a la zona a la que pertenecen. Es importante destacar que este ID solo identifica la zona, ya que el identificador único para cada persona sigue siendo su número de teléfono. En el siguiente bloque, se procede a contar cuántas personas hay en cada zona. Sin embargo, en este punto, la nueva ubicación de las personas se asigna al centroide de la zona micro a la que pertenecen. Además, se verifica en qué zona estaban en intervalos de 5 minutos, como se ilustra en la Fig. 3.20. En este nuevo enfoque, la columna *centroid* representa la nueva posición de cada persona, lo que permite obtener una distribución más precisa y realista en comparación con la ubicación original proporcionada por la antena.

	PCS	ID_ZONA	Generated_Time	largest_polygon_id	geometry	centroid
0		2130.0	18:00:00	48.0	POLYGON ((-73.12400663343671 -36.8410324689550...	POINT (-73.12551 -36.84988)
1		671.0	18:00:00	22.0	POLYGON ((-73.07405022652297 -36.8103390741980...	POINT (-73.07224 -36.81381)
2		572.0	18:00:00	27.0	POLYGON ((-73.07781700224884 -36.7790448206534...	POINT (-73.07854 -36.78131)
3		1300.0	18:00:00	59.0	POLYGON ((-73.13545240427806 -36.8382546119750...	POINT (-73.14212 -36.84542)
4		1660.0	18:00:00	39.0	POLYGON ((-73.1428316267848 -37.00169349338941...	POINT (-73.14840 -37.01503)
...		...	...	...	...	...
4838		240.0	18:55:00	69.0	MULTIPOLYGON (((-73.11570168679218 -36.7189339...	POINT (-73.11342 -36.71828)
4839		2060.0	18:55:00	75.0	POLYGON ((-73.11061511379299 -36.7417740067135...	POINT (-73.11565 -36.75255)
4840		910.0	18:55:00	21.0	POLYGON ((-73.04916940523563 -36.8224064533439...	POINT (-73.05047 -36.82533)
4841		632.0	18:55:00	22.0	POLYGON ((-73.05124239274758 -36.7998218759576...	POINT (-73.05200 -36.80188)
4842		2060.0	18:55:00	75.0	POLYGON ((-73.11061511379299 -36.7417740067135...	POINT (-73.11565 -36.75255)

4843 rows × 6 columns

Fig. 3.20: Dataframe nueva ubicación individual. Fuente: Autoría propia.

3.1.5. Preparar el dataset para el modelo de machine learning supervisado.

Siguiendo lo planteado en el algoritmo de la Fig.3.21.



Fig. 3.21: Algoritmo para preparar dataset para el modelo. Fuente: Autoría propia.

El primer paso es conectarse a la API de Google Maps mediante una API-KEY la cuál sirve para poder hacer las solicitudes a la API. Luego, se definen los puntos correspondientes a la

ubicación de las 10 comunas que conforman el Gran Concepción. A cada uno de estos puntos se le asigna un radio de 1 km, dentro del cual se generan puntos aleatorios. A continuación, se define la hora de inicio y final, así como el medio de transporte. A partir de esto, se manda la solicitud a la API con los respectivos puntos generados y los parámetros indicados anteriormente, devolviendonos las rutas para todas las combinaciones posibles entre los puntos generados, con intervalos de 5 minutos.

El resultado es un DataFrame que contiene la información de estas rutas, como se puede apreciar en la Fig. 3.22.

	Combinación Comuna	Modo de Transporte	Tiempo Estimado	Distancia	Hora de Solicitud
0	Chiguayante-Chiguayante	bicycling	2 mins	0.6 km	2024-07-30 18:00:00
1	Chiguayante-Chiguayante	bicycling	2 mins	0.6 km	2024-07-30 18:05:00
2	Chiguayante-Chiguayante	bicycling	2 mins	0.6 km	2024-07-30 18:10:00
3	Chiguayante-Chiguayante	bicycling	2 mins	0.6 km	2024-07-30 18:15:00
4	Chiguayante-Chiguayante	bicycling	2 mins	0.6 km	2024-07-30 18:20:00
...	...	...	...	...	...
2095	Lota-Lota	walking	1 hour 1 min	4.2 km	2024-07-30 18:35:00
2096	Lota-Lota	walking	1 hour 1 min	4.2 km	2024-07-30 18:40:00
2097	Lota-Lota	walking	1 hour 1 min	4.2 km	2024-07-30 18:45:00
2098	Lota-Lota	walking	1 hour 1 min	4.2 km	2024-07-30 18:50:00
2099	Lota-Lota	walking	1 hour 1 min	4.2 km	2024-07-30 18:55:00

2100 rows × 5 columns

Fig. 3.22: Dataframe de combinaciones basado en Google Maps. Fuente: Autoría propia.

El resultado del proceso incluye datos sobre el tiempo que tomó cada ruta, la distancia recorrida, la hora en que se realizó el trayecto y el modo de transporte utilizado. En el DataFrame, si una combinación de puntos es, por ejemplo, Lota-Lota, esto indica que el trayecto se realizó dentro de la misma comuna. Por otro lado, si la combinación es Lota-Concepción, significa que se tomaron puntos de diferentes comunas para realizar el cálculo de la ruta.

Estos datos permiten observar tanto desplazamientos locales dentro de una comuna como aquellos que ocurren entre comunas, lo cual es fundamental para el análisis de movilidad.

3.1.6. Entrenar modelo de machine learning supervisado.

En la Fig.3.23. se ven los pasos a seguir para poder entrenar el modelo

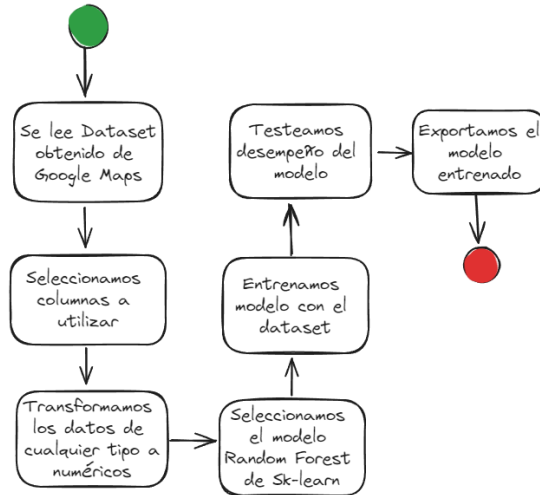


Fig. 3.23: Algoritmo para entrenar modelo. Fuente: Autoría propia.

Se lee el dataset que se obtuvo en el punto anterior que se ve en la Fig. 3.22, luego se seleccionan las columnas que se van a necesitar, se les hace una transformación a la data para que la distancia quede en km y el tiempo en minutos, y además se les agrega la velocidad promedio en Km/min dependiendo el modo de transporte como se ve en la Fig. 3.24.

	Modo de Transporte	Tiempo Estimado	Distancia	speed
0	driving	0	17.8	0.500
1	bicycling	0	17.5	0.250
2	walking	226	16.7	0.083
5	driving	0	17.8	0.500
6	bicycling	0	17.5	0.250
...	...	...	...	...
9595	walking	61	4.2	0.083
9596	walking	61	4.2	0.083
9597	walking	61	4.2	0.083
9598	walking	61	4.2	0.083
9599	walking	61	4.2	0.083

6960 rows x 4 columns

Fig. 3.24: Dataframes con parametros para el modelo. Fuente: Autoría propia.

Además se cambia el modo de transporte por driving como 1, walking a 2 y bicycling en 3, y sacando las combinaciones que se mantuvieron en el mismo lugar para que el modelo lo pueda

leer, quedando el siguiente dataframe de la fig. 3.25.

	Modo de Transporte	Tiempo Estimado	Distancia	speed
2	2	226	16.7	0.083
7	2	226	16.7	0.083
12	2	226	16.7	0.083
17	2	226	16.7	0.083
22	2	226	16.7	0.083
...	...	...	...	...
9595	2	61	4.2	0.083
9596	2	61	4.2	0.083
9597	2	61	4.2	0.083
9598	2	61	4.2	0.083
9599	2	61	4.2	0.083
2688 rows × 4 columns				

Fig. 3.25: Dataframes con parámetros de forma numérica para el modelo. Fuente: Autoría propia.

Se le entrega al modelo el dataset ya transformado, en el que se le proporcionarán como parámetros el tiempo, la distancia y la velocidad, para que pueda predecir el modo de transporte. Dependiendo de estos factores, el modelo debería ser capaz de hacer esta predicción. Por ejemplo, una vez entrenado, si se le pide que realice la predicción de una distancia corta y un tiempo largo, debería concluir que el modo de transporte es a pie. Esto se debe a que cada árbol de decisión en el modelo aprende utilizando una porción distinta de los datos, lo que significa que los árboles que votan no han visto los mismos ejemplos durante el entrenamiento. Cada árbol toma su decisión basada en la similitud de los patrones aprendidos, y el resultado final es el que más árboles han votado.

Se utilizaron 2688 datos, de los cuales un 20 % se reservó para evaluar el funcionamiento del modelo Random Forest, logrando una precisión de 0.92. Este 20 % se utiliza para probar el modelo y verificar la corrección de los resultados. Finalmente, se exporta el modelo para su uso futuro, cuando se disponga de nuevos datos listos para realizar predicciones.

Modelo	Precisión
Random Forest	0.92

Tabla 3.1: Tabla precisión modelo de Machine Learning. Fuente: Autoría propia

3.1.7. Crear un grafo para representar la movilidad.

Al igual que los pasos anteriores se sigue el algoritmo que se ve en la Fig. 3.26.

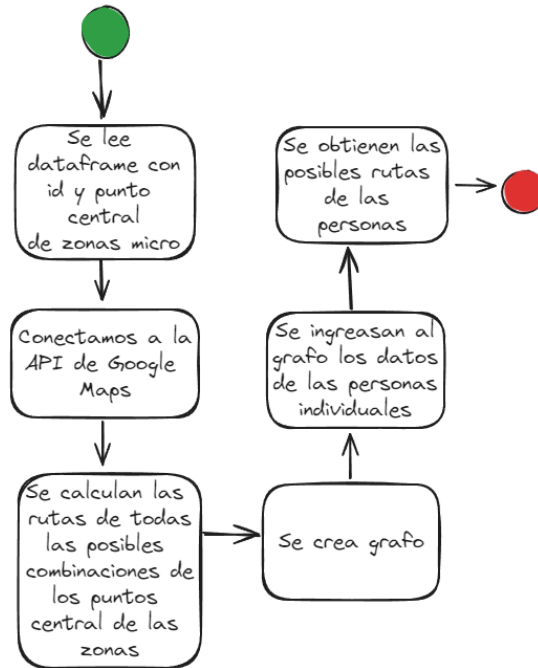


Fig. 3.26: Algoritmo para creación del grafo. Fuente: Autoría propia.

Lo primero que se hace es leer el dataframe en donde se tienen todas las zonas micro con sus respectivos centros e id como se ve en la siguiente Fig.3.27.

	ZONA		centroid
0	842	POINT (-73.03481 -36.81737)	
1	272	POINT (-73.10749 -36.72945)	
2	301	POINT (-73.13744 -36.72794)	
3	240	POINT (-73.11342 -36.71828)	
4	511	POINT (-73.09602 -36.75360)	
...	...		...
365	410	POINT (-73.08670 -36.73618)	
366	50	POINT (-72.95611 -36.61920)	
367	73	POINT (-72.95293 -36.62744)	
368	290	POINT (-73.12247 -36.72174)	
369	181	POINT (-73.00495 -36.77004)	

Fig. 3.27: Dataframe zonificación micro con punto central . Fuente: Autoría propia.

Luego al igual que en la construcción del dataset se tiene que conectar a la API de Google Maps para encontrar todas las rutas y posibles combinaciones entre todos los centros de las zonas micro, quedando el siguiente dataframe 3.28 con la distancia la zona inicial y final y la ruta.

	Nodo_Origen	Nodo_Destino	Distancia (metros)	Duración	LineString
0	842	272	13737	21 mins	LINESTRING (-36.81737 -73.03481, -36.81809 -73...
1	842	301	18296	25 mins	LINESTRING (-36.81737 -73.03481, -36.81809 -73...
2	842	240	18504	27 mins	LINESTRING (-36.81737 -73.03481, -36.81809 -73...
3	842	511	10827	16 mins	LINESTRING (-36.81737 -73.03481, -36.81809 -73...
4	842	523	10440	15 mins	LINESTRING (-36.81737 -73.03481, -36.81809 -73...
...	...	...	...	...	...
68260	50	290	31753	37 mins	LINESTRING (-36.6192 -72.95612, -36.6186 -72.9...
68261	50	181	26797	34 mins	LINESTRING (-36.6192 -72.95612, -36.6186 -72.9...
68262	73	290	31779	36 mins	LINESTRING (-36.62761 -72.95309, -36.62813 -72...
68263	73	181	26823	34 mins	LINESTRING (-36.62761 -72.95309, -36.62813 -72...
68264	290	181	23275	31 mins	LINESTRING (-36.72175 -73.12249, -36.72284 -73...

Fig. 3.28: Dataframe con todas las rutas posibles entre zonas. Fuente: Autoría propia.

Después se agrega esta información para la creación del grafo en donde los nodos serían los centros de las zonas y las aristas serían las rutas, teniendo como especificaciones tiempo, la distancia que toma llegar de una ruta a otra, a continuación en la fig. 3.29 se vería una parte del grafo, ya que el visualizar el grafo real tomaría mucho tiempo de procesamiento.

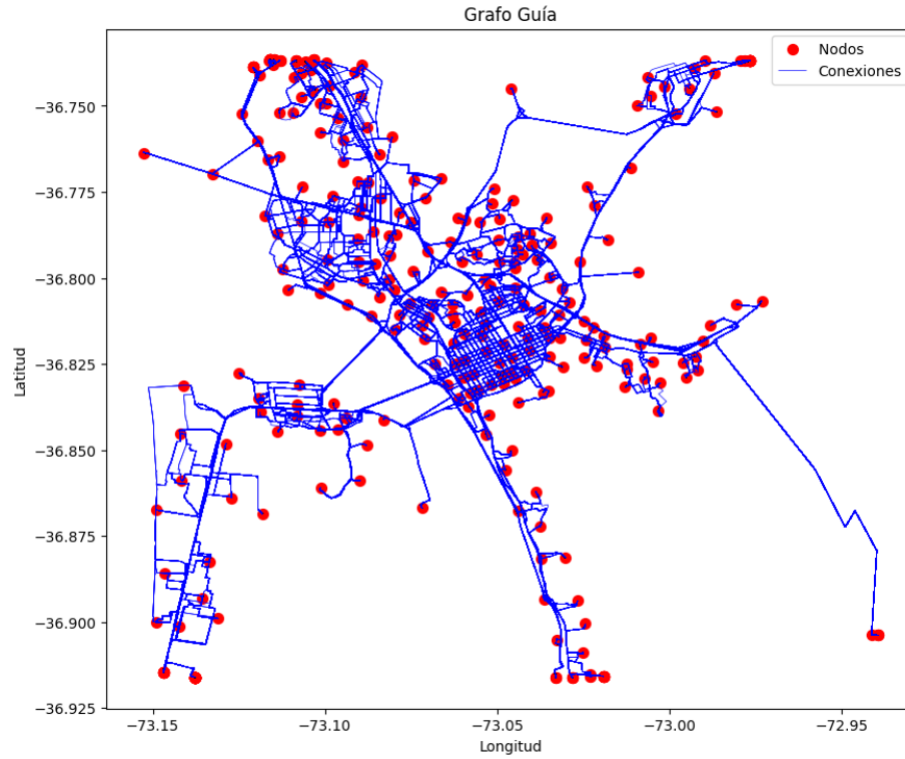


Fig. 3.29: Grafo representativo en base a zonificación micro. Fuente: Autoría propia.

Ahora se ingresa el dataframe de la Fig. 3.20 que contiene la ubicación individual y el tiempo, obteniendo la ruta que tomo para llegar de un nodo a otro en el lapsus del tiempo que cualquier persona quiera, entregando el siguiente dataframe 3.30.

	PCS	Distancia	Nodo_Inicial	Nodo_Final	Tiempo Estimado	Ruta_Completa	LineString_Completo
80		2.099	1071	1074	15	[1071, 1074]	LINESTRING (-36.829 -72.99549, -36.82858 -72.9...
122		25.030	1660	1660	20	[1660, 1651, 1640, 2870, 1640, 1651, 1660]	MULTILINESTRING ((-36.99852 -73.15208, -36.998...
177		6.866	960	1071	25	[960, 1022, 1071]	MULTILINESTRING ((-36.81993 -73.019, -36.82037...
199		5.621	1871	841	30	[1871, 651, 870, 880, 841]	MULTILINESTRING ((-36.79872 -73.07832, -36.799...
294		63.426	1680	1680	20	[1680, 1380, 1422, 2170, 990, 1680]	LINESTRING (-37.01884 -73.17542, -37.01846 -73...
347		15.106	1311	1871	30	[1311, 2150, 1871]	LINESTRING (-36.88597 -73.14659, -36.8859 -73...
385		2.051	1970	1960	10	[1970, 1960]	LINESTRING (-36.79478 -73.09924, -36.79504 -73...
445		3.834	1600	1631	10	[1600, 1620, 1631]	LINESTRING (-37.01462 -73.16387, -37.0145 -73...
646		20.713	2290	734	35	[2290, 791, 742, 734]	LINESTRING (-36.67873 -72.96838, -36.67868 -72...
694		1.692	900	890	5	[900, 890]	LINESTRING (-36.82684 -73.04186, -36.82781 -73...
753		1.148	640	631	10	[640, 631]	LINESTRING (-36.8097 -73.05481, -36.80985 -73...
817		6.840	500	2590	15	[500, 310, 2590]	LINESTRING (-36.76308 -73.08369, -36.76273 -73...
861		24.227	1422	1600	15	[1422, 1380, 1592, 1600]	LINESTRING (-36.85881 -73.09021, -36.85881 -73...
1087		14.456	1600	1810	30	[1600, 2730, 1810]	LINESTRING (-37.09869 -73.15288, -37.09844 -73...
1169		12.684	1692	1570	40	[1692, 1631, 1592, 1570]	LINESTRING (-36.95708 -73.16235, -36.95701 -73...
1170		11.135	565	172	10	[565, 172]	LINESTRING (-36.79021 -73.06588, -36.79004 -73...
1304		31.025	741	841	15	[741, 172, 181, 842, 841]	LINESTRING (-36.78703 -73.01131, -36.78719 -73...
1315		7.131	900	2150	15	[900, 2150]	LINESTRING (-36.83625 -73.1082, -36.83632 -73...
1392		5.048	2870	1570	20	[2870, 1570]	LINESTRING (-36.94802 -73.13415, -36.94812 -73...
1546		2.166	1220	990	10	[1220, 990]	LINESTRING (-36.83223 -73.05486, -36.83303 -73...
1599		9.559	990	1290	15	[990, 1290]	LINESTRING (-36.83125 -73.14103, -36.83127 -73...

Fig. 3.30: Dataframe con rutas individuales. Fuente: Autoría propia.

3.1.8. Predecir modo de transporte.

Lo primero es leer el dataframe de la Fig.3.30. que obtenimos del grafo, siguiendo el algoritmo de la Fig. 3.31.

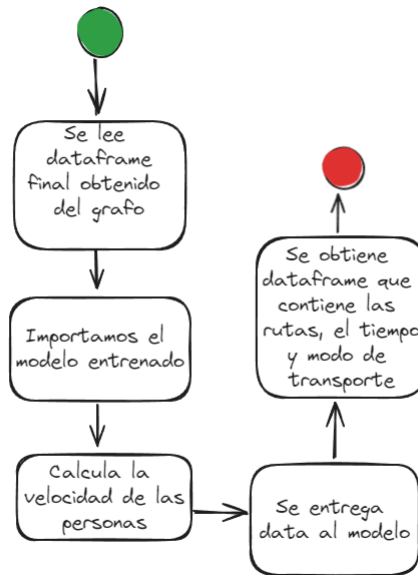


Fig. 3.31: Algoritmo para predecir modo de transporte. Fuente: Autoría propia.

Ahora se importa el modelo Random Forest entrenado, se calcula la velocidad a la que se movió cada persona, entregandole como input al modelo el dataframe de la Fig.3.32.

	PCS	Distancia	Nodo_Inicial	Nodo_Final	Tiempo Estimado	Ruta_Completa	LineString_Completo	speed
80		2.099	1071	1074	15	[1071, 1074]	LINESTRING (-36.829 -72.99549, -36.82858 -72.9...	0.139933
122		25.030	1660	1660	20	[1660, 1651, 1640, 2870, 1640, 1651, 1660]	MULTILINESTRING ((-36.99852 -73.15208, -36.998...	1.251500
177		6.866	960	1071	25	[960, 1022, 1071]	MULTILINESTRING ((-36.81993 -73.019, -36.82037...	0.274640
199		5.621	1871	841	30	[1871, 651, 870, 880, 841]	MULTILINESTRING ((-36.79872 -73.07832, -36.799...	0.187367
294		63.426	1680	1680	20	[1680, 1380, 1422, 2170, 990, 1680]	LINESTRING (-37.01884 -73.17542, -37.01846 -73...	3.171300
347		15.106	1311	1871	30	[1311, 2150, 1871]	LINESTRING (-36.88597 -73.14659, -36.8859 -73...	0.503533
385		2.051	1970	1960	10	[1970, 1960]	LINESTRING (-36.79478 -73.09924, -36.79504 -73...	0.205100
445		3.834	1600	1631	10	[1600, 1620, 1631]	LINESTRING (-37.01462 -73.16387, -37.0145 -73...	0.383400
646		20.713	2290	734	35	[2290, 791, 742, 734]	LINESTRING (-36.67873 -72.96838, -36.67868 -72...	0.591800
694		1.692	900	890	5	[900, 890]	LINESTRING (-36.82684 -73.04186, -36.82781 -73...	0.338400
753		1.148	640	631	10	[640, 631]	LINESTRING (-36.8097 -73.05481, -36.80985 -73...	0.114800

Fig. 3.32: Dataframe de entrada para el modelo. Fuente: Autoría propia.

Como output del modelo tendríamos el siguiente dataframe que se puede apreciar en la Fig.3.33.

	PCS	Distancia	Nodo_Inicial	Nodo_Final	Tiempo Estimado	Ruta_Completa	LineString_Completo	Nivel de Tráfico	speed	Modo de Transporte Predicho
80		2.099	1071	1074	15	[1071, 1074]	LINESTRING (-72.99549 -36.82900, -72.99587 -36...	5	0.139933	walking
122		25.030	1660	1660	20	[1660, 1651, 1640, 2870, 1640, 1651, 1660]	MULTILINESTRING ((-73.15208 -36.99852, -73.152...	5	1.251500	driving
177		6.866	960	1071	25	[960, 1022, 1071]	MULTILINESTRING ((-73.01900 -36.81993, -73.017...	5	0.274640	bicycling
199		5.621	1871	841	30	[1871, 651, 870, 880, 841]	MULTILINESTRING ((-73.07832 -36.79872, -73.080...	5	0.187367	bicycling
294		63.426	1680	1680	20	[1680, 1380, 1422, 2170, 990, 1680]	LINESTRING (-73.17542 -37.01884, -73.17507 -37...	5	3.171300	driving
347		15.106	1311	1871	30	[1311, 2150, 1871]	LINESTRING (-73.14659 -36.88597, -73.14658 -36...	5	0.503533	driving
385		2.051	1970	1960	10	[1970, 1960]	LINESTRING (-73.09924 -36.79478, -73.09946 -36...	5	0.205100	bicycling
445		3.834	1600	1631	10	[1600, 1620, 1631]	LINESTRING (-73.16387 -37.01462, -73.16376 -37...	5	0.383400	driving
646		20.713	2290	734	35	[2290, 791, 742, 734]	LINESTRING (-72.96838 -36.67873, -72.96837 -36...	5	0.591800	driving
694		1.692	900	890	5	[900, 890]	LINESTRING (-73.04186 -36.82684, -73.04421 -36...	5	0.338400	bicycling
753		1.148	640	631	10	[640, 631]	LINESTRING (-73.05481 -36.80970, -73.05471 -36...	5	0.114800	walking
817		6.840	500	2590	15	[500, 310, 2590]	LINESTRING (-73.08369 -36.76308, -73.08370 -36...	5	0.456000	driving
861		24.227	1422	1600	15	[1422, 1380, 1592, 1600]	LINESTRING (-73.09021 -36.85881, -73.09035 -36...	5	1.615133	driving

Fig. 3.33: Dataframe de salida del modelo. Fuente: Autoría propia.

3.1.9. Analizar y visualizar de los dataframes finales.

En esta etapa final, donde convergen todos los DataFrames y el procesamiento realizado, se procede a graficar los datos para visualizarlos de manera más clara. Esto puede incluir mapas de densidad, matrices de movilidad o representaciones de las rutas seguidas por las personas durante un periodo determinado. Es importante recordar que parte del procesamiento final se llevó a cabo mediante un grafo, lo que permite obtener información más detallada sobre el comportamiento de las personas a través de un análisis de grafos. Los resultados obtenidos se reflejarán en el siguiente capítulo.

Capítulo 4

4.1. Resultados

4.1.1. Visualización comunal

En los resultados de los dataframes enfocados en la zonificación macro, se puede visualizar la densidad de personas por comuna en una hora específica como se muestra en la fig. 4.34.

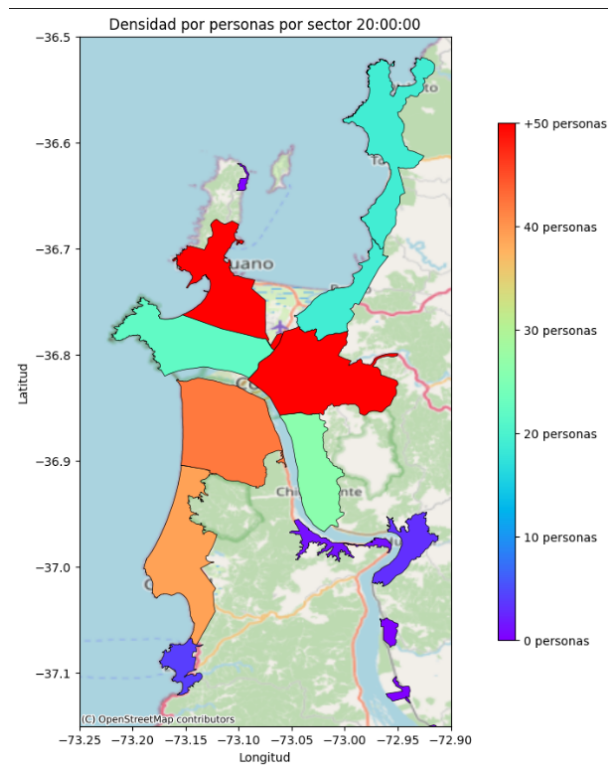


Fig. 4.34: Mapa densidad de personas por comuna. Fuente: Autoría propia.

En el mapa se puede apreciar que la mayor concentración de personas se encuentra en Concepción y Talcahuano, seguido de San Pedro de la Paz. Esto tiene sentido, ya que son los lugares donde se cuenta con un mayor número de servicios básicos, como universidades en el caso de Concepción y Talcahuano, así como comercio. Por el contrario, en la periferia se pueden observar menos personas, dado que muchos estudiantes y trabajadores residen en el centro.

Durante las horas pico, aún hay muchas personas que no regresan a sus hogares, ya que, en general, si viven lejos del centro, los trayectos para llegar a su vivienda pueden durar casi 3 horas debido a la congestión vehicular que ocurre en esos horarios.

Otra forma de visualizar esta información es tomando dos DataFrames en distintas horas y utilizando un heatmap basado en una matriz de movilidad para observar cuántas personas se movieron de una zona a otra, como se muestra en la Fig. 4.35.

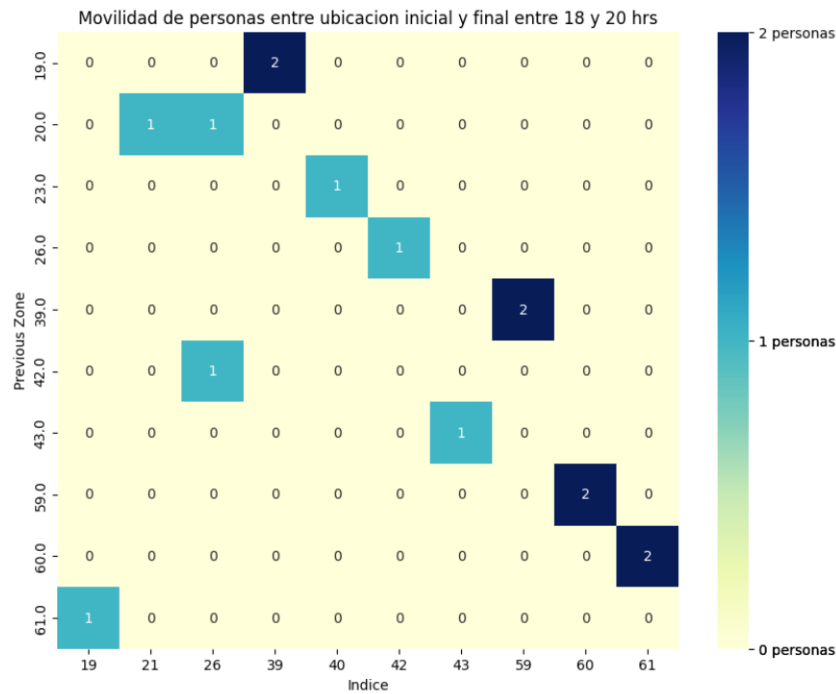


Fig. 4.35: Heatmap basado en matriz de movilidad entre las 18 y 20 hrs. Fuente: Autoría propia.

También se implementaron otras formas de visualizar la actividad de las personas, como se puede ver en la Fig. 4.36 donde hay un gráfico de barras que tiene comunas repetidas, pero eso producto del archivo que se usó para la zonificación comunal, representando el naranjo las personas que salen de la comuna y el azul las personas que entran.

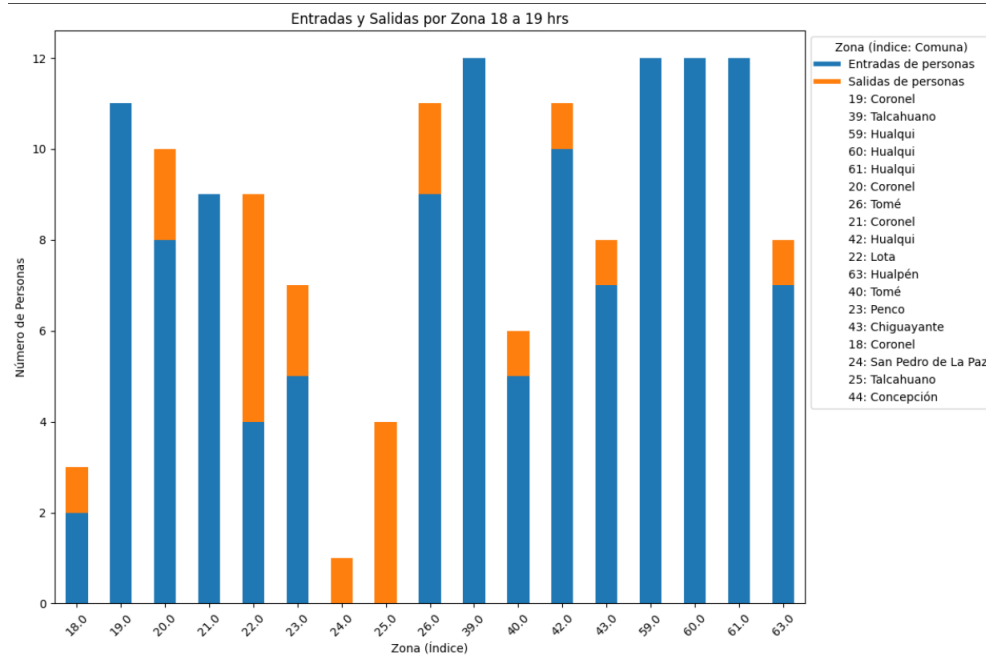


Fig. 4.36: Gráfico flujo entrada y salida de comunas. Fuente: Autoría propia.

Al observar el gráfico de la Fig. 4.36, se puede visualizar un alto flujo de entrada a las zonas de la periferia, lo que indica que hay una gran cantidad de personas que regresan a sus hogares después de una jornada laboral o de estudios. Esto sugiere que, debido a problemas de accesibilidad, muchas de estas personas no pueden llevar a cabo su vida cotidiana en la zona donde residen.

4.1.2. Visualización zonificación CityLab

En este caso, para visualizar la densidad de personas por sector la mejor forma es como se ve en la Fig. 4.37, ya que, si lo hacemos de las otras formas como se muestra en la zonificación macro, serian mucha información porque son muchas más zonas y al momento de visualizarlas no se podría diferenciar nada en la imagen.

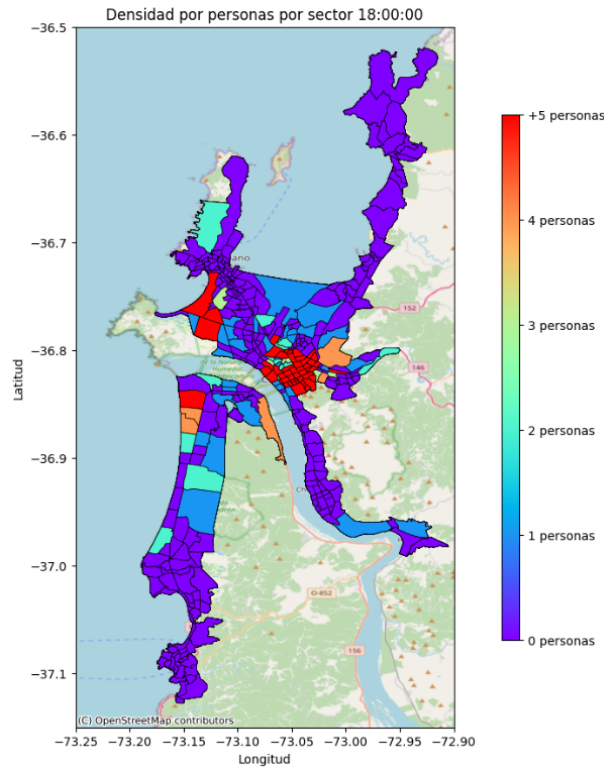


Fig. 4.37: Mapa densidad de personas por zona micro. Fuente: Autoría propia.

A diferencia del resultado macro, en la Fig. 4.37 se puede visualizar de una manera un poco más discernida la ubicación de las personas, observándose una mayor concentración en la zona del centro, donde se encuentran todos los locales de comida, el mall, el barrio universitario y los tribunales. Esto tiene sentido, ya que es donde hay mayor accesibilidad al comercio. Además, está la Universidad de Concepción, que siempre será un foco de concentración de personas.

4.1.3. Visualización de rutas y modo de transporte con Machine Learning

Como se puede apreciar en la Fig. 3.33 este es el dataframe final que entrega el modelo, se puede ver que el dataframe es parecido al que se mostró anteriormente en la metodología, pero se le agrego la columna de lo que predijo el modelo.

Y también como en los casos anteriores se pueden visualizar las rutas como se ve en la Fig. 4.38, mostrando claramente el predominio de autos.

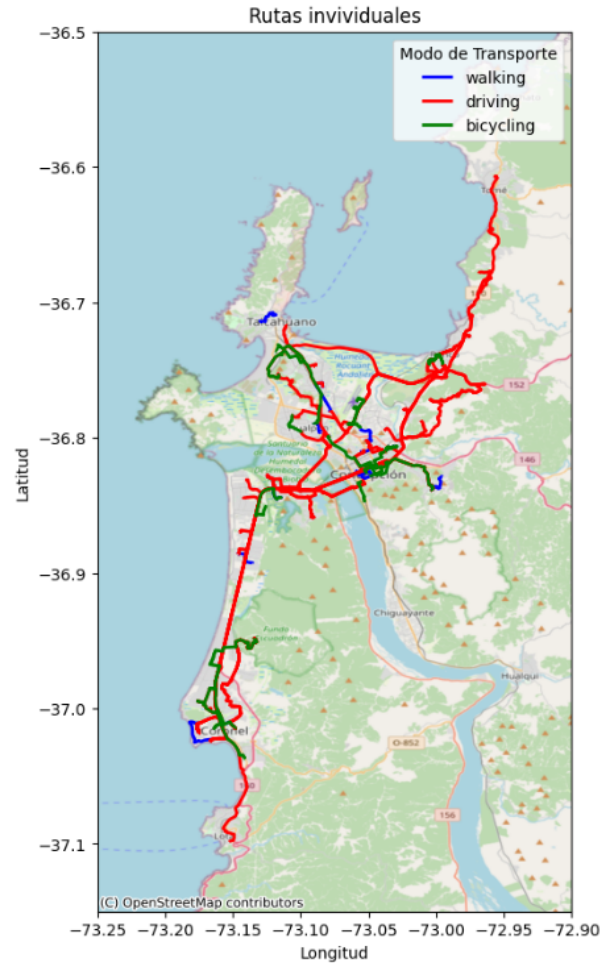


Fig. 4.38: Rutas obtenidas del grafo en conjunto de lo predicho del modelo. Fuente: Autoría propia.

Pudiendo apreciar que lo que más se utiliza es el modo de transporte *"driving"*, viendo que generalmente las distancias cortas son *"walking"*, también podríamos pensar que los modo *"bicycling"* sean falsos positivos de autos que entraron en un taco y recorrieron una distancia no larga, entonces no se demoraron mucho, pudiendo así confundir al modelo y arrojar un falso positivo de este modo de transporte.

4.1.4. Análisis mediante el grafo

Mediante un análisis de grafo se nos entrega una serie de resultados en los cuales se puede destacar:

- El nodo más transitado es 761 con 8 ocurrencias.

- La ruta más transitada es (760,761) con 2 ocurrencias.
- La distancia media recorrida es 10.01 km.
- El tiempo medio estimado es 21.02 minutos.
- El modo de transporte más común es driving con 25 ocurrencias.
- Velocidad promedio por modo de transporte:

Modo de transporte	Speed
Bicycling	0.253417 [km/min]
Driving	1.058077 [km/min]
Walking	0.093954 [km/min]

Tabla 4.2: Velocidad promedio por medio de transporte. Fuente: Autoría propia

- Se encontraron 11 comunidades de nodos, donde una comunidad es un grupo de nodos en las cuales las personas que se encuentran en algunos de esos nodos tienden a moverse entre esos mismos nodos de la comunidad como se aprecia en la Fig. 4.39.

```
Se encontraron 11 comunidades de nodos.
Comunidad 1: ['1380', '1570', '1592', '1600', '1620', '1631', '1640', '1651', '1660', '1692', '1810', '2730', '2870']
Comunidad 2: ['1300', '1320', '1332', '1360', '2072', '2130', '631', '640', '651', '841', '870', '880']
Comunidad 3: ['122', '160', '2590', '3002', '310', '480', '500', '572', '750', '761']
Comunidad 4: ['13', '1960', '1970', '561', '565', '606', '607', '620', '632', '92']
Comunidad 5: ['1311', '1871', '2060', '2080', '2150', '403', '524', '890', '900']
Comunidad 6: ['1022', '1071', '1074', '1841', '910', '940', '960']
Comunidad 7: ['1220', '1290', '1422', '1680', '2100', '2170', '990']
Comunidad 8: ['1891', '1912', '380', '392', '394', '461']
Comunidad 9: ['2290', '734', '742', '791']
Comunidad 10: ['172', '181', '741', '842']
Comunidad 11: ['212', '221']
```

Fig. 4.39: Comunidades de nodos. Fuente: Autoría propia.

Con los resultados obtenidos del grafo, se observa que la forma en que la gente más se moviliza es en automóvil. También se puede visualizar la zona más transitada, como se muestra en la Fig. 4.40. Lo más probable es que la zona más transitada se encuentre en el centro de Concepción, debido al alto flujo de personas en ese sector.

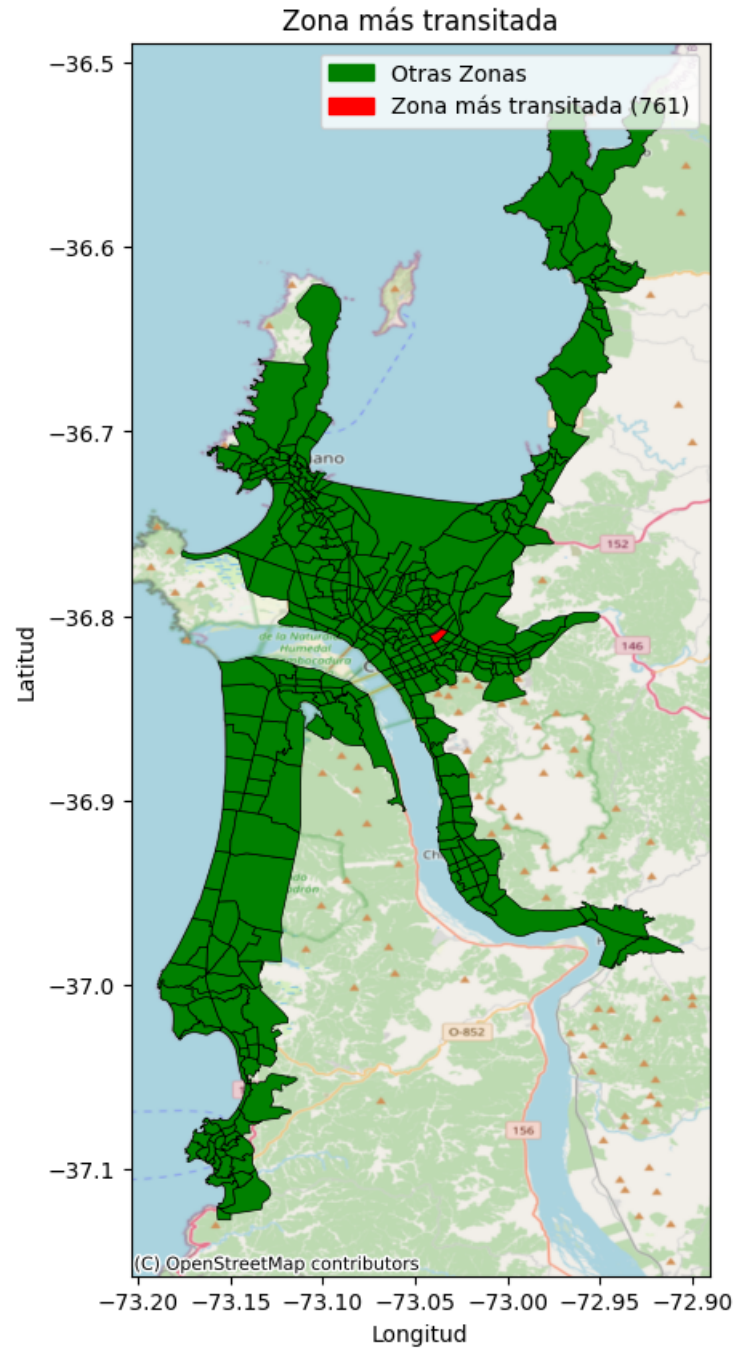


Fig. 4.40: Zona más transitada. Fuente: Autoría propia.

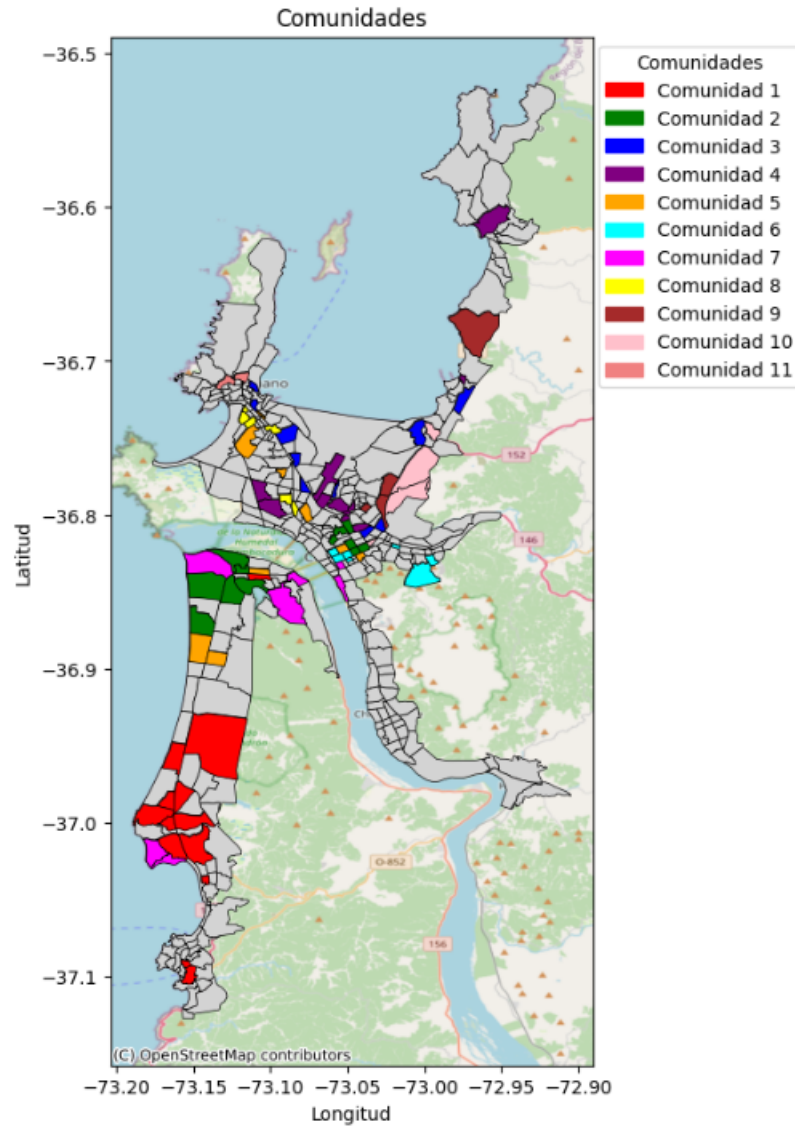


Fig. 4.41: Comunidades de zonas en base a los nodos. Fuente: Autoría propia.

En la Fig. 4.41 se pueden observar las comunidades que tienden a mostrar desplazamientos entre zonas cercanas. Sin embargo, también es posible ver algunas zonas periféricas que pertenecen a la misma comunidad que una del centro. Esto podría deberse a personas que deben movilizarse lejos de sus hogares para trabajar o estudiar.

Capítulo 5

5.1. Conclusión

En la presente Memoria de Título se presenta un sistema para utilizar datos de telefonía móvil en el análisis de movilidad en el Gran Concepción, empleando técnicas de procesamiento de datos, teoría de grafos y algoritmos de Machine Learning. Este sistema no solo permite visualizar el comportamiento de las personas, sino que también proporciona la capacidad de predecir el modo de transporte utilizado.

El análisis basado en grafos ayudó a identificar las rutas y nodos más transitados, generando trayectorias bastante exactas, ya que se crearon en función de las mismas que ofrecería Google Maps para llegar de un lugar a otro. Esta información es importante para la toma de decisiones en la planificación urbana, ya que se evidenció una centralización de las personas, atribuida a una posible mala distribución de los servicios básicos, lo que demuestra una deficiente accesibilidad. Además, se observó una tendencia de las personas a utilizar más vehículos en comparación con otros modos de transporte, lo que podría provocar congestión en los horarios pico del día. Este tipo de análisis no solo mejora la comprensión del flujo de movilidad, sino que también contribuye a la optimización de recursos, constituyendo un aporte significativo para la planificación de infraestructuras viales y transporte público.

Los mayores desafíos presentados consistieron en lograr una mayor precisión en la ubicación de las personas. Sin embargo, gracias a la simulación del rango de las torres de las antenas y la distribución por zonas según su tamaño, se logró una representación más precisa de los desplazamientos y ubicaciones de las personas.

El sistema está diseñado para futuras ingestas de datos, lo que lo hace flexible y escalable para futuros estudios de movilidad. No necesariamente se limita a datos de telefonía móvil, ya que podría ampliarse a todo tipo de datos, convirtiéndose así en un aporte no solo para una solución inmediata en el análisis de movilidad del Gran Concepción, sino también como base

para investigaciones futuras relacionadas con el comportamiento de las personas en la ciudad, constituyendo un valioso insumo para la toma de decisiones en la logística urbana.

Bibliografía

- [1] Museo de Internet. (s.f.). ¿Qué es la red GSM y cómo funciona? En CCM - La Comunidad de los Usuarios de CCM. Recuperado el 10 de mayo de 2024, de <https://es.ccm.net/aplicaciones-e-internet/museo-de-internet/enciclopedia/11304-que-es-la-red-gsm-y-como-funciona/>
- [2] Fernández de Andalucía, C. (2020). Movilidad Urbana a Partir de Datos Móviles. Instituto de Estadística y Cartografía de Andalucía.
- [3] Sierra, J. (2020, 29 de diciembre). Movilidad y Big Data: Así se mueve Sevilla [Entrada de blog]. Recuperado el 10 de mayo de 2024, de <https://jaimesierra.wordpress.com/2020/12/29/movilidad-y-big-data-asi-se-mueve-sevilla/>
- [4] Moleres, C. J., & Fulponi, J. I. (2022). Metodología para el estudio de la movilidad con datos de Facebook: generación de matrices origen-destino en ciudades de América Latina y análisis para Buenos Aires. Monografía del BID, 1058.
- [5] Marcos Reinaldo Catalan Painen, “Diseño de Esquemas de Clustering para la Identificación Espectral de Carga en Hornos Flash”, Memoria de Título, Ingeniero Civil Metalúrgico, Abril 2021, Departamento de Ingeniería Metalúrgica. Facultad de Ingeniería, Universidad de Concepción
- [6] Cristian Stiven Vargas Rojo, “Análisis Big Data De Los Patrones de Movilidad a Partir de Registros de Llamada”, Trabajo de Grado, Ingeniero de Sistemas y Computación, 2017, Facultad de Ingenierías, Universidad Tecnológica de Pereira.
- [7] Marcelino Felipe Álvarez Núñez, Jonathan Alejandro Parra Muñoz, “Teoría de Grafos”, Seminario para optar al título de profesor de enseñanza media en educación matemática, Chillán 2013, Escuela de Pedagogía en Educación Matemática, Faculta de Educación y Humanidades, Universidad del Bío-Bío.
- [8] Martínez Llario, J. C. (2020).” PostGIS Análisis Espacial Avanzado” (2nd ed.).
- [9] PostGIS. (s.f.). PostGIS 2.0 Manual. Recuperado el 10 de mayo de 2024, de <https://postgis.net/docs/manual-2.0/>

- [10] Adrián Gilabert Pérez, “Utilización de OpenStreetMap en el cálculo de coberturas mediante un trazador de rayos”, Trabajo Fin de Grado, Escuela Técnica Superior de Ingeniería de Telecomunicación, Universidad Politécnica de Cartagena.
- [11] OpenStreetMap. (s.f.). Mapa de la ubicación [Mapa]. Recuperado el 10 de mayo de 2024, de <https://www.openstreetmap.org/#map=13/-36.8815/-73.1008&layers=TD>
- [12] Orange. (s.f.). ¿Cómo funciona una red móvil? Radio Waves. <https://radio-waves.orange.com/es/como-funciona-una-red-movil/>
- [13] Autor desconocido. (2012). Grafo 3. Universidad de Pamplona. https://www.unipamplona.edu.co/unipamplona/portallIG/home_23/recursos/genera-1/11072012/graf3.pdf
- [14] IBM. (s.f.). ¿Qué es el aprendizaje supervisado? IBM. <https://www.ibm.com/es-es/topics/supervised-learning>
- [15] Universitat Oberta de Catalunya. (s.f.). Diagramas de Voronoi en R. <https://datascience.recursos.uoc.edu/es/diagramas-de-voronoi-en-r/>
- [16] Gadea, A. (2014, mayo 29). El mundo perfecto de Voronoi: donde matemática y geografía se unen. Geografía Infinita. <https://www.geografiainfinita.com/2014/05/el-mundo-perfecto-de-voronoi-donde-matematica-y-geografia-se-unen/>
- [17] Scikit-learn developers. (s.f.). Scikit-learn: Machine learning in Python. <https://scikit-learn.org/stable/>
- [18] FreeCodeCamp. (2020). Algoritmo de la ruta más corta de Dijkstra: Introducción gráfica. <https://www.freecodecamp.org/espanol/news/algoritmo-de-la-ruta-mas-corta-de-dijkstra-introduccion-grafica/>
- [19] Gehlke, C. (2023). OSMnx: Python for urban street networks. <https://osmnx.readthedocs.io/en/stable/>
- [20] Hunter, J. D. (n.d.). Matplotlib: Python plotting. <https://matplotlib.org/>
- [21] Gillies, S. (n.d.). GeoPandas: Python geospatial analysis. <https://geopandas.org/en/stable/>
- [22] McKinney, W. (n.d.). Pandas documentation. <https://pandas.pydata.org/docs/>
- [23] HubSpot. (n.d.). ¿Qué es Python?. <https://blog.hubspot.es/website/que-es-python>

- [24] Google. (n.d.). GoogleMaps (Python package). <https://pypi.org/project/googlemaps/>
- [25] Plotly. (n.d.). Python graphing library. <https://plotly.com/python/>
- [26] Waskom, M. (n.d.). Seaborn: statistical data visualization. <https://seaborn.pydata.org/>
- [27] Hagberg, A. A., Schult, D. A., & Swart, P. J. (n.d.). NetworkX documentation. <https://networkx.org/documentation/stable/tutorial.html>