ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELECTRÍCA Y ELECTRÓNICA

DETECCIÓN DE PATRONES DE COMPORTAMIENTO DE PARÁMETROS DE RF EN REDES DE COMUNICACIÓN MÓVIL MEDIANTE MEDICIONES DE CAMPO Y TÉCNICAS DE MACHINE LEARNING

ESTUDIO DE LAS REGIONES LIMÍTROFES DE HANDOVER EN LA PARROQUIA RURAL DE CALDERÓN CON LA AYUDA DE LA HERRAMIENTA R APLICANDO TECNICAS DE MACHINE LEARNING

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN TELECOMUNICACIONES

NAVARRETE LUZURIAGA HENRY PAUL

henry.navarrete@epn.edu.ec

LUPERA MORILLO PABLO ANIBAL

pablo.lupera@epn.edu.ec

DMQ, Febrero 2022

CERTIFICACIONES

Yo, Navarrete Luzuriaga Henry Paul declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

HENRY NAVARRETE

Certifico que el presente trabajo de integración curricular fue desarrollado por Navarrete Luzuriaga Henry Paul, bajo mi supervisión.

PABLO LUPERA
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Navarrete Luzuriaga Henry Paul

Lupera Morillo Pablo Aníbal

DEDICATORIA

Esta tesis va dedicada a la memoria de mi abuelo José Luzuriaga, que siempre me inculcó la importancia de los estudios en el éxito personal del ser humano.

A mis padres que siempre me brindaron su apoyo incondicional, motivacional y económico a lo largo de toda mi carrera estudiantil. A mis hermanos que son mi principal fuente de motivación para ser un ejemplo de hijo y estudiante.

A mis amigos personales y compañeros académicos que me brindaron consejos e intercambiaron ideas con respecto al gran esfuerzo y trabajo que conllevó mi carrera universitaria.

A mis abuelos, tíos y primos que siempre estuvieron predispuestos a brindarme una ayuda o consejo.

AGRADECIMIENTO

Agradezco infinitamente el amor y apoyo brindado por parte de mi madre María Luzuriaga a lo largo de todas mis etapas educativas. A mi padre Washington Iza que puso a disposición mía toda la experiencia obtenida por el a lo largo de su vida. Ambos me brindaron confianza, estabilidad y seguridad, les agradezco el que siempre me hayan dado la libertad de poder decidir y escoger las decisiones mas importantes de mi vida.

Agradezco también a mi padre Rodrigo Navarrete, por formar parte y estar presente en el proceso de mi etapa estudiantil y deportiva.

Agradezco a mi abuelita María Betancourt y a mi abuelo José Luzuriaga, por criarme como un hijo más y sentar las bases para el desarrollo y crecimiento para volverme en una persona con gran criterio y excelentes valores.

Agradezco a Melissa Boada y a gran parte de su familia, ya que siempre me brindó un apoyo emocional en los momentos que más sentía derrotado o cansado. A mis amigos incondicionales, que me ha dado la universidad, el deporte y la vida.

Agradezco a la Escuela Politécnica Nacional, a toda su planta docente, en especial a la Doctora Diana Navarro por ser mi tutora académica a lo largo de mi carrera estudiantil y al Doctor Pablo Lupera por dar la apertura al desarrollo de este trabajo de titulación.

ÍNDICE DE CONTENIDO

RESUMEN

Este trabajo se divide en tres partes detalladas a continuación:

Dentro del capítulo uno relacionado al marco teórico, se realizó un estudio previo de las herramientas de recolección de datos con el fin de determinar cuáles serían las que mejor compatibilidad tendrían con el móvil usado, tomando en cuenta que el estudio se plantea para determinar el comportamiento de las zonas limítrofes en los procesos de handover en una conexión con la tecnología móvil LTE. Además, se ahondo en el funcionamiento de la herramienta R, ya que esta se la utiliza para desarrollar el código del modelo propuesto en este trabajo. Para finalizar con este capítulo se profundizó en la teoría de Machine Learning y se optó por trabajar con el modelo de aprendizaje supervisado específicamente con la técnica de árbol de decisión

En el capítulo dos, se definen la zona y rutas seleccionadas, donde se va a realizar las mediciones, posterior a eso se elabora un cronograma, en base al tiempo disponible para realizar esta tarea. Una vez que se obtuvieron los archivos de los datos recolectados en rutas, se procede a crear el código para aplicar este modelo en las tres rutas, gracias a la librería “rpart” y sus funciones que posee Rstudio se facilitó en gran medida el trabajo de programación.

Para finalizar en el capítulo tres, se realizó un análisis de los resultados obtenidos, comparando las estadísticas de evaluación del modelo como lo son el accuracy, sensibilidad y evolución del error, lo que permitió plantear las conclusiones y recomendaciones del presente estudio.

PALABRAS CLAVE: LTE, Handover, RStudio, Machine Learning, árbol de decisión, accuracy, sensibilidad.

ABSTRACT

This work is divided into three parts detailed below:

Within chapter one related to the theoretical framework, a preliminary study of the data collection tools was carried out in order to determine which ones would have the best compatibility with the mobile used, taking into account that the study is proposed to determine the behavior of border areas in handover processes in a connection with LTE mobile technology. In addition, the operation of the R tool is delved into, since it is used to develop the code of the model proposed in this work. To end this chapter, the theory of Machine Learning was deepened, and it was decided to work with the supervised learning model specifically with the decision tree technique.

In chapter two, the area and selected routes are defined, where the measurements will be carried out, after that a schedule is elaborated, based on the time available to carry out this task. Once the files of the data collected in routes were obtained, the code to apply this model in the three routes is created, thanks to the "rpart" library and its functions that Rstudio has, the work of programming.

To conclude in chapter three, an analysis of the results obtained was carried out, comparing the evaluation statistics of the model such as the accuracy, sensitivity and evolution of the error, which allowed the conclusions and recommendations of this study to be drawn up.

KEYWORDS: LTE, Handover, RStudio, Machine Learning, Decision Tree, accuracy, sensitivity.

1 INTRODUCCIÓN

Se realizarán mediciones en un sector determinado de la ciudad de Quito, específicamente en la zona nororiente, parroquia rural de Calderón, debido al alto porcentaje poblacional de la zona y a la gran proyección urbana que tiene dicho sector, durante las mediciones no se ejecutará la conexión de ningún tipo de servicio, los datos móviles estarán apagados y de igual forma cualquier tipo de conexión inalámbrica como radio o servicios de Streaming. Estas mediciones se realizarán a cabo con la finalidad de detectar y determinar las características de las regiones limítrofes entre celdas servidas por diferentes sectores de una misma estación base y/o las regiones limítrofes entre celdas servidas por diferentes sectores de diferentes estaciones base.

Este primer estudio tiene la finalidad de detectar las características de este tipo de zonas limítrofes con la finalidad de establecer si los parámetros involucrados tienen un mismo patrón de comportamiento o existe algún tipo de diferencia. Los resultados servirán para estudios futuros que se encuentren direccionados en mejorar los procesos de handover

El trabajo se centra en la recolección de datos en tres rutas ubicadas en la parroquia rural de Calderón, dicha zona ha tenido un crecimiento poblacional y estructural a lo largo de esta última década, además, el desarrollo de la tecnología móvil obliga a los aperadores de telefonía celular a realizar una reestructura de los diseños y ubicación de sus estaciones base para lograr una cobertura total a sus usuarios.

El estudio se centra en analizar el comportamiento de las señales de la tecnología móvil Long Term Evolution (LTE), por sus siglas en inglés, en el momento de realizar el proceso de handover, se escogió este sector ya que su ubicación geográfica se presta también para considerar zonas geográficas donde se encuentran lugares altos y planicies, además de sectores de vivienda, comercial y de tránsito vehicular.

En la actualidad, existen varios softwares de simulación y lenguajes de programación que permiten realizar e implementar modelos de predicción usando Machine Learning enfocado en el algoritmo de aprendizaje supervisado dependiendo del tipo técnica a utilizar, uno de los modelos utilizados es el de regresión y árbol de decisión, debido a su sencillo análisis de resultados. Matlab y Phyton son los más utilizados comúnmente, sin embargo, existen herramientas como R, que tienen librerías y funciones que facilitan la programación del modelo y que será la utilizado en el presente proyecto.

1.1 Objetivo general

Detectar y determinar las características de las regiones limítrofes entre celdas de una o varias radios bases en rutas establecidas en la parroquia de Calderón, utilizando herramientas de recolección de muestras en un dispositivo móvil enfocado en el comportamiento de la tecnología 4G y con la ayuda de la aplicación R para el análisis de los datos con técnicas de Machine Learning.

1.2 Objetivos específicos

1. Consultar, especificar y determinar las herramientas que mejor se acoplen al terminal móvil, tomando en cuenta el tipo de archivo que entreguen estas aplicaciones con el fin de que sean compatibles con la herramienta de manipulación de los datos.
2. Realizar un cronograma que cumpla con el tiempo establecido para la etapa de recolección de datos y su posterior análisis
3. Implementar un código en la herramienta R aplicando la técnica de Machine Learning de aprendizaje supervisado para entender el comportamiento de las muestras tomadas y su relación con el handover.

1.3 Alcance

En el proyecto se realiza el estudio de las herramientas necesarias para la recolección y manipulación de datos las cuales son “Net Monitor Lite”, “GPS Logger”, “4G Force Lte”, “CellMapper” y “Network Cell Info Lite”, dentro de las que se escogen las de mayor compatibilidad con el modelo de terminal móvil utilizado, posterior a eso, se identifica las zonas dentro de la parroquia rural escogida en la que se va a realizar la etapa de medición, para observar los procesos de handover, el cual se define como la transición de conexión del terminal móvil de una radio base a otra [1], tomando en cuenta si la zona es comercial, residencial o de tránsito vehicular, puesto que son zonas donde los distribuidores de servicios de telefonía móvil optan por reducir costos y brindan una cobertura celular solo en las vías principales o lugares concurridos, similar al caso presentado en la “Comuna los Cerritos” ubicada en una parroquia rural de Guayaquil [2].

Haciendo énfasis a lo anterior escogen tres rutas, las cuales son: Ruta 1 denominada Calderón con punto de Inicio: Calle De los Arrieros y Atahualpa. punto final: Avenida

Cacha y Atahualpa. Dicha ruta cuenta con una extensión de 2 kilómetros de distancia y se la considera una zona residencial. La Ruta 2 llamada San Juan con punto de Inicio: Agustín Guerrero & Amalia Uriguen. punto final: Carlos Mantilla & Santander, la distancia que tiene esta ruta es de 2.2 kilómetros, es una ruta de conexión vehicular con bajo número de viviendas y locales comerciales a su alrededor. Por último, la Ruta 3, a la cual se la conocerá como Marianas con punto de Inicio: Giovanni Calles & Francisco de Albornoz y punto final: Capitán Giovanni Calles & Avenida Cacha, esta ruta se caracteriza por ser una de las zonas más comerciales de dicha parroquia, aquí se pueden encontrar colegios, fábricas y locales comerciales a lo largo del 1.7 kilómetros.

Cabe mencionar que el estudio se lo realiza sobre el comportamiento de la tecnología móvil LTE, ya que es la tecnología con mayor proyección dentro de las rutas determinadas, por lo que es necesario saber la frecuencia en la que trabaja y si el móvil es compatible o no con esta tecnología [3][4].

Las muestras son recolectadas con la aplicación “Net Monitor Lite”, ya que esta permite exportar los datos recolectados en un archivo tipo csv, por lo que facilita en gran manera su estudio y análisis con la herramienta de programación “R” , además se utiliza en un modelo de terminal móvil “Samsung A20S”, con un servidor de telefonía móvil.

En la etapa de medición se realiza un cronograma con el fin de distribuir los días para la recolección, análisis y comparación de los datos en las tres diferentes rutas escogidas, cabe resaltar que existen algunas variables que diferencian un conjunto de muestras de otras, aún si estas se encuentran en la misma zona señalada para el experimento, tales como velocidad, altura, hora del día y flujo de personas.

Para la etapa de programación, predicción y estudio del handover con la ayuda de las librerías descargables en R, como lo son dplyr,rpart, se utiliza la técnica de machine learning de “Aprendizaje Supervisado” y para la clasificación de datos se utiliza la técnica de inducción de reglas o mejor conocida como árbol de decisión [5][6].

1.4 Marco teórico

1.4.1 Herramientas para la recolección de datos

En el presente capitulo se realiza una descripción individual de cada una de las herramientas digitales que se utilizarán en la recepción y monitoreo de las señales,

además de definir las características técnicas del dispositivo móvil donde se va a descargar y manipular las aplicaciones.

1.4.1.1 Net Monitor Lite

Se trata de una aplicación gratuita desarrollada por la empresa “PARIZENE”, se dispone de una versión actual 1.11.2, la misma que solicitará al terminal móvil varios permisos de acceso tales como: ubicación, teléfono, almacenamiento y entre otros.

Una de las principales funciones que tiene esta aplicación es el poder monitorear redes CDMA, GSM, WCDMA, LTE, TD-SCDMA, 5G NR con el fin de entregar información de la celda actual, vecina y la intensidad de la señal. Posee un soporte multi SIM en el caso de que el terminal móvil a utilizar sea un dispositivo multibanda además de utilizar el GPS integrado en el dispositivo para la geolocalización. Permite generar una base de datos y exportarla en un archivo CLF/ KML con información personalizada sobre las celdas [7].

1.4.1.1.1 Funcionamiento de la aplicación

Una vez instalada la aplicación se procede a realizar la configuración inicial que consiste en ceder los permisos de acceso solicitados por parte de la app. La interfaz de la aplicación se abre y se presentan al usuario los tres íconos en la parte inferior, como se muestra en la figura 1.1, cada uno representa una funcionalidad diferente de la aplicación.

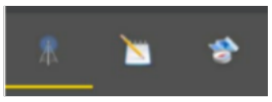


Figura 1.1 Iconos de la interfaz de la aplicación Net Monitor

En la primera pestaña se refleja una lista de antenas disponibles cercanas al terminal móvil, como se muestra a continuación en la figura 1.2. En base a esta lista el usuario puede conocer la ubicación y señal que se recibe de cada antena de las estaciones base y el tipo de enlace habilitado. En la segunda pestaña se observa otra lista similar con la diferencia que aquí el usuario puede editar a su gusto las diferentes antenas, lo que permite que sean identificadas muy fácilmente, la pestaña en mención se muestra en la figura 1.3. En la última pestaña el usuario tiene la posibilidad de observar un mapa de geolocalización, como se ve en la figura 1.4 en donde se despliega la ubicación de las antenas de las estaciones base [8].

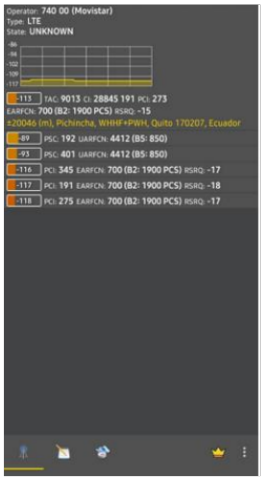


Figura 1.2 Pestaña de antenas cercanas **Figura 1.3** Pestaña para identificar las antenas

Figura 1.4 Ubicación de las antenas en el mapa.

1.4.1.2 GPS Logger

GPS Logger es una aplicación diseñada por “BASICAIRDATA” con el objetivo de registrar la posición y ruta del usuario. Lo que la convierte como una herramienta útil en varios escenarios cómo:

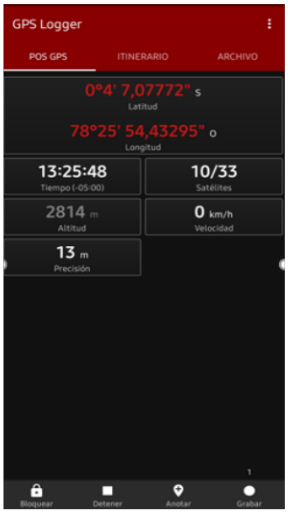
- Registro de viajes
- Colocación de marcas de posición a lo largo del camino.
- Consultar de la posición, velocidad, altitud y dirección de viajes realizados [9].

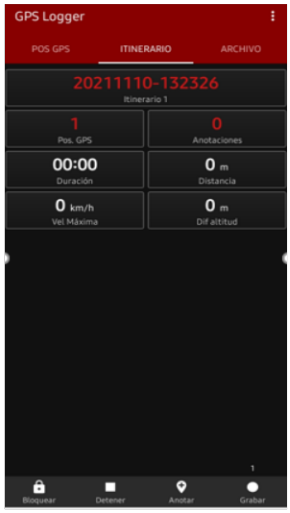
1.4.1.2.1 Funcionamiento de la aplicación

Esta app funciona sin conexión a internet, lo que conlleva a no trabajar necesariamente con mapas integrados. Una de las ventajas es que el usuario puede acceder en cualquier momento a las rutas de medición grabadas, utilizando la lista disponible en la aplicación,

además de visualizarlas guardarlas y compartirlas en diferentes tipos de formatos como lo son GPX, KML y TXT.

La interfaz de esta aplicación es muy fácil de utilizar, al instalar la aplicación y entregar los permisos de acceso solicitados, se procede a esperar una cantidad corta de segundos hasta encontrar en la primera pestaña una imagen similar a la de la figura 2.1, el nombre de esta pestaña es “POS GPS” y aquí se encuentran básicamente las coordenadas de ubicación del usuario en tiempo real, altitud, cantidad de satélites referenciados, velocidad y el margen de error de la precisión. En la pestaña dos o “ITINERARIO”, el usuario puede observar las estadísticas en tiempo real de su trayecto. La interfaz de esta herramienta se la puede visualizar en la figura 2.2. Dentro de la tercera pestaña “ARCHIVO” , el usuario puede revisar el historial de todos los viajes grabados cuando considere necesario, esta información puede ser exportada y compartida, así se observa en la figura 2.3 [10].





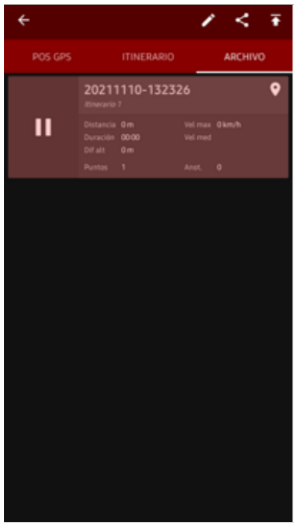


Figura 2.1 Pestaña POS GPS **Figura 2.2** Pestaña Itinerario **Figura 2.3** Pestaña Archivo

En la parte inferior de las pestañas se identifican cuatro íconos, los cuales son:

- **Bloqueo:** Cumple con la funcionalidad de bloquear la barra inferior, para evitar marcas de puntos accidentales durante la grabación del trayecto.
- **Detener:** Icono que detiene la grabación o finaliza la pista que está siendo grabada.
- **Anotar:** Permite agregar una posición a la muestra que está siendo grabada.
- **Grabar**: Icono que permite iniciar, activar o desactivar la grabación.

1.4.1.3 Force LTE

Es una aplicación para teléfonos móviles diseñada por “Xsquare Studio”. Esta aplicación ayuda al usuario a cambiar la red a 2G, 3G o 4G y permanecer en la red elegida. Una de las principales desventajas de esta aplicación es que no es compatible con todos los teléfonos, ya que depende de la marca y diseño del terminal móvil. Esto debido a que algunos fabricantes de teléfonos bloquean la oportunidad de forzar el cambio de red. Al momento de ingresar a la aplicación el usuario debe seleccionar le método que sea compatible con su modelo de teléfono móvil. A continuación, se observa que por defecto el terminal se encuentra configurado con la opción LTE/UMTS auto (PRL), esto se observa en las figuras 3.1 y 3.2 [11].

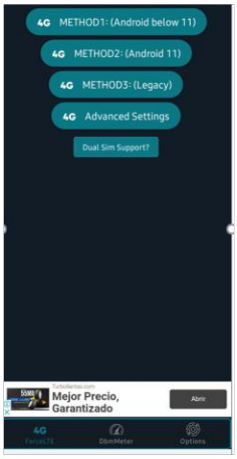


Figura 3.1 Métodos para forzar al terminal

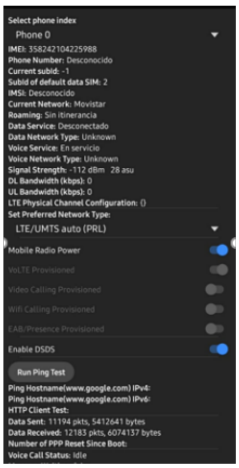


Figura 3.2 Configuración predeterminada

En la figura 3.3, se observa como el usuario puede forzar a su dispositivo a estar conectado a una red LTE con la opción “LTE only”, por último, se verifica la estabilidad de la conexión en la pestaña “DbmMeter” tal como se refleja en la figura 3.4.

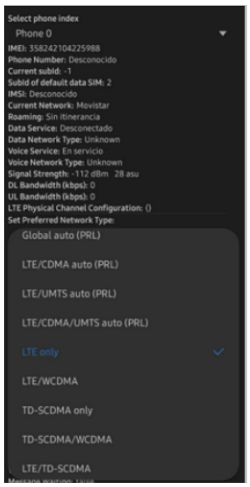


Figura 3.3 Opción de LTE only

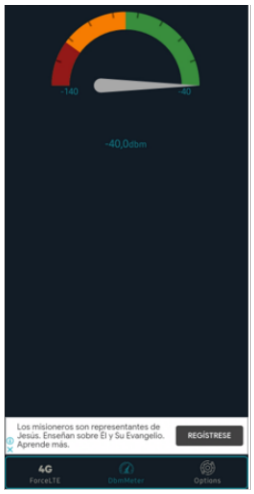


Figura 3.4 Estabilidad de la conexión

1.4.1.4 CellMapper

CellMapper es una aplicación diseñada por “cellmapper.net” que tiene como objetivo principal indicar información avanzada de las redes celulares. Además de permitir grabar y aportar con datos para contribuir a mapas de cobertura celular. Para utilizar esta aplicación es necesario trabajar con un dispositivo inteligente con sistema operativo Android 4.0 o superior. Las redes que actualmente soporta esta aplicación en su última versión 5.5.3 son: GSM, UMTS, CDMA, LTE, LTE-A, NR. [12].

Algunas de las características principales de esta app son las siguientes [13]:

- Muestra datos de información de bajo nivel de la red celular junto con las bandas de frecuencia de operación
- Indica las frecuencias celulares
- Muestra un mapa con la cobertura total, cobertura por sectores y por bandas de operación.
- Soporta dispositivos con tecnología Dual SIM

La Interfaz de la aplicación se la visualiza en la figura 4.1, en donde se encuentra cuatro pestañas las cuales son: GPS, subir datos, cuenta y grabar. Mientras que, en la figura 4.2, se encuentran algunas de las opciones principales que ya fueron nombradas.

Figura 4.1 Interfaz principal de CellMapper.

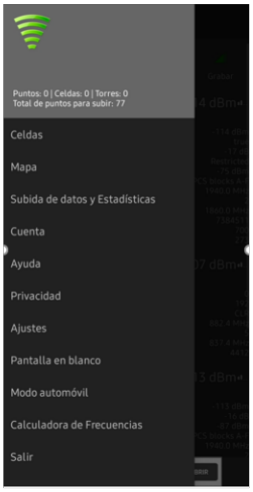


Figura 4.2 Estabilidad de la conexión

1.4.1.5 Network Cell Info Lite

Network Cell Info lite es una aplicación diseñada por “M2Catalyst, LLC” la cual provee la ubicación de la señal en el mapa y mide la potencia de la red para registrar la señal y señales vecinas. Una gran ventaja de esta aplicación es que cubre un gran número de las tecnologías de las redes celulares, tales como: LTE, HSPA+, HSPA, WCDMA, EDGE,

GSM, CDMA, EVDO. En la figura 5, se muestra un ejemplo de la información entregada de la red celular y WIFI en donde el terminal móvil está conectado [14].

Figura 5 Ejemplo de conexión de la red celular y WiFi

1.4.1.6 Características técnicas del terminal móvil

El modelo de terminal móvil que se va a utilizar para este trabajo es de la serie Galaxy A20S del fabricante Samsung, el cual presenta varias especificaciones técnicas que se resumen en la siguiente tabla 1, las mismas que fueron recopiladas de la página oficial del fabricante [15].

Tabla 1. Características técnicas del terminal móvil

Descripción	**Características**			
Procesador	Velocidad CPU de 1.8 GHz		CPU tipo Octa-Core	
Memoria	RAM de 3 GB	Interna de 32 GB	Externa Micro SD hasta 512 GB	
Redes / Bandas	**2G GSM** GSM850, GSM900, DCS1800, PCS1900	**3G UMTS** B1(2100), B2(1900), B4(AWS), B5(850), B8(900)	**4G FDD LTE** B1(2100), B2(1900), B3(1800), B4(AWS), B5(850), B7(2600), B8(900), B12(700), B13(700), B17(700), B20(800), B28(700), B66(AWS-3)	**4G TDD LTE** B38(2600), B40(2300), B41(2500)
Sistema Operativo	Android			
Conectividad	**Wi-Fi** 802.11 b/g/n 2.4GHz	**Localización** GPS, Glonass, Beidou, Galileo	**Versión Bluetooth** Bluetooth v4.2	**Versión USB** USB 2.0

1.4.2 Herramienta RStudio

Rstudio es una herramienta digital, definida como software de programación en un entorno de desarrollo integrado (IDE), la interfaz gráfica es muy amigable con el usuario , en la cual se puede diferenciar claramente, una consola, un editor donde se ejecuta directamente el código, por último, se observa una barra de herramientas para el trazado, historial de archivos ejecutados, conexiones y el tutorial, en la figura 6 se indica dicha interfaz gráfica [10].

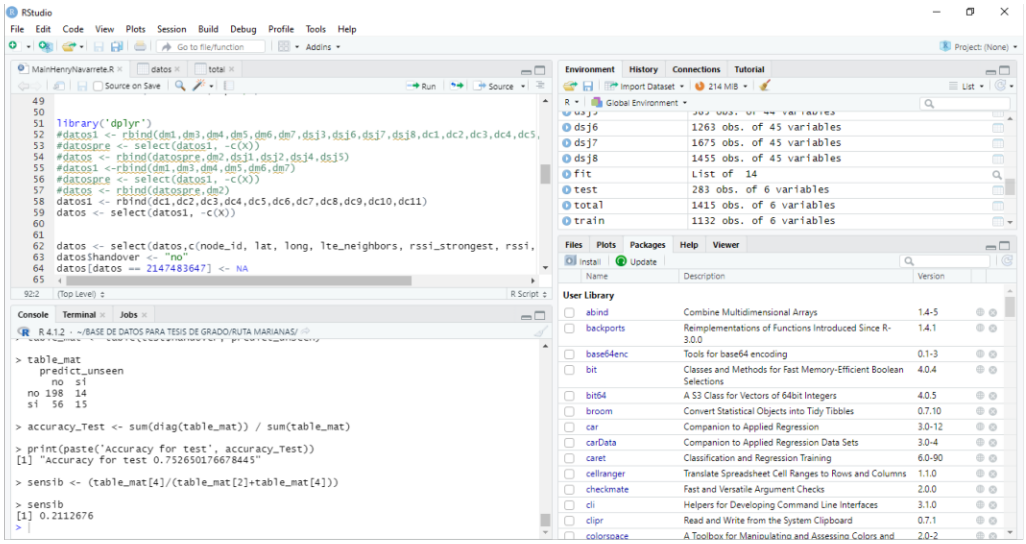


Figura 6. Interfaz gráfica de RStudio.

Esta herramienta presenta varias ediciones, comerciales y de código abierto, sus versiones de escritorio son compatibles con sistemas operativos Windows, Linux y Mac, mientras que las versiones de navegador deben mantener una conexión a Rstudio Server o Workbench. [16]. Para fines de este trabajo se ocupó la edición de código abierto, fácil de descargar e instalar en el ordenador.

1.4.3 Machine Learning- Aprendizaje de máquina

En las últimas décadas el aprendizaje de máquina o mejor conocido como Machine Learning, ha sido una gran herramienta para trabajar con grandes cantidades de datos, realizar análisis, estudios y posteriores modelos que ayuden a predecir el comportamiento de cualquier tipo de variable en base a un número determinado de variables que se deseen comparar.

Algunas definiciones de Machine de acuerdo con algunos autores son las siguientes Tom Mitchell, tiene una definición técnica relacionada donde menciona que "Se dice que un

programa de computadora aprende de la experiencia E con respecto a alguna clase de tareas T y medida de desempeño P, si su desempeño en las tareas T, medido por P, mejora con la experiencia E". Mientras que Arthur Samuel tiene una idea más genérica de Machine Learning, la cual se declara de la siguiente manera ”El campo de estudio que brinda a las computadoras la capacidad de aprender sin ser programadas explícitamente”[17].

Al mencionar Machine Learning se entiende una idea general de la herramienta a utilizar, la cual se deriva de la inteligencia artificial por lo que es necesario diferenciar los tipos de técnicas comúnmente utilizadas y que han sido aplicadas en diferentes ciencias humanas a lo largo del tiempo, dentro de las que destacan la aplicación en medicina, desarrollo tecnológico, educación, construcción, finanzas, telecomunicaciones entre otras [17].

En la actualidad grandes empresas multinacionales, utilizan Machine Learning en muchos de sus proyectos más exitosos y prometedores, fueron IBM y Microsoft las primeras empresas reconocidas mundialmente a inicios del nuevo milenio en poner en práctica este tipo de inteligencia artificial. En el caso de IBM lo hizo con su ordenador Watson, mientras que Microsoft lo implementó con la versión beta de Azure Machine Learning. En base al éxito de estos modelos otras empresas como Google, Amazon, DeepMind siguieron sus pasos.[16]

Se pueden mencionar tres tipos de técnicas o algoritmos marcados con la finalidad de alcanzar un aprendizaje autónomo de máquina y estos son:

- Aprendizaje supervisado.
- Aprendizaje no supervisado.
- Aprendizaje de refuerzo.

Para el fin de este trabajo se utilizará la técnica de aprendizaje supervisado debido que su función y modelo es una de las técnicas más acorde a lo que se espera obtener, sin embargo, es necesario mencionar varias diferencias entre esta técnica y las otras dos para entenderlas [18].

1.4.3.1 Aprendizaje no supervisado

Esta técnica es utilizada cuando un conjunto de datos o variables no poseen una etiqueta que ayuda a diferenciar una de otra, por lo que el algoritmo recurre a una revisión de características similares o diferentes, las cuales servirán para sacar una conclusión del comportamiento de los datos [18].

1.4.3.2 Aprendizaje de refuerzo

Al igual que con el aprendizaje no supervisado el conjunto de datos no posee una etiqueta, con la diferencia que en este algoritmo el modelo con el pasar del tiempo es retroalimentado mediante actualizaciones de los datos [18].

1.4.3.3 Aprendizaje supervisado

Es un modelo de aprendizaje donde el algoritmo necesita de una cierta cantidad de datos etiquetados de la variable que se desea predecir, una de las características de este modelo es que mientras mayor sea el conjunto de datos etiquetados el aprendizaje del algoritmo será mejor. Al conjunto de datos se lo denomina datos de entrenamiento y por lo general se utiliza alrededor del ochenta por ciento de una data, mientras que el restante veinte por ciento se utilizará como test para definir la confiabilidad del modelo. La finalidad de este modelo es que en un futuro se pueda ingresar otra base de datos sin la necesidad que estos estén etiquetados para que el modelo pueda predecir la variable de salida en función del comportamiento de los datos de entrada. Haciendo una analogía se puede decir qué este modelo de aprendizaje es muy utilizado en la educación convencional ya que se plantean problemas y una forma para solucionarlo, sin embargo, esta misma forma de solución en un futuro servirá para problemas similares [19].

1.4.3.3.1 Árboles de decisión

Es un método del aprendizaje supervisado, considerado como un algoritmo de clasificación ya que se obtiene una función de valores discretos, su análisis e interpretación es muy simple ya que muy comúnmente se lo asocia con un diagrama de flujo, aunque presentan varias diferencias.[18]. Un ejemplo claro se presentado a continuación con la figura 7.

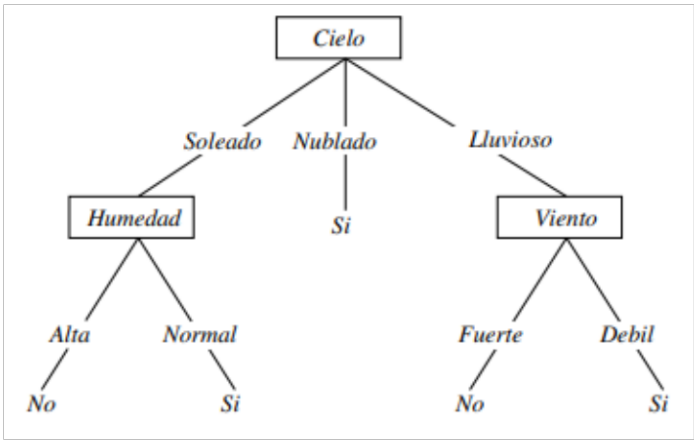


Figura 7. Ejemplo de un árbol de decisión [18].

Con este ejemplo se pretende deducir cuando hacer deporte en función de las condiciones del cielo, humedad y viento. Como se observa la raíz del árbol es el nivel más alto, del cual empiezan a nacer ramas de criterios diferenciados, estas a su vez de ser necesario entregan valores discretos en base a la variable que se está analizando.

Así se encuentra un modelo que pueda predecir una variable en función de un grupo de variables de entrada que tengan relación entre sí y la variable de salida. Por último, el árbol de decisión se lo considera un modelo de clasificación con técnica de aprendizaje supervisado y en resumen su característica es que entrega valores discretos y etiqueta a los datos de salida obtenidos [19].

2 METODOLOGÍA

2.1 Etapa de recolección de los datos

En el presente trabajo la etapa de recolección de datos tiene la finalidad de detectar y determinar las características de las regiones limítrofes de las celdas servidas por diferentes sectores de una misma estación base o regiones limítrofes entre celdas servidas por diferentes estaciones base.

Para lograr este cometido es necesario plantearse algunos objetivos principales como lo son:

- Definir la funcionalidad y aporte que brinda cada parámetro obtenido en las mediciones con las aplicaciones anteriormente descritas
- Determinar una ruta o zona donde se presenten los procesos de handover para realizar el proceso la recolección de datos
- Exportar y analizar las mediciones de los parámetros obtenidos en un formato csv. y ayudarse del uso de la herramienta Rstudio

2.1.1 Identificación y definición de los parámetros

Actualmente la tecnología celular vigente que soporta el terminal móvil con el que se realizará la recolección de datos es LTE ,sin embargo, es necesario mencionar que existen dentro de la zona escogida para la medición, donde dicha tecnología no tiene cobertura suficiente, por lo cual como se indicó en el apartado anterior, se cuenta con otra aplicación la cual exige al terminal a mantenerse en el rango de frecuencias de la tecnología LTE, por lo que al momento de realizar las mediciones garantiza una conexión permanente con este tipo de red..

En base a esto, se enumeran y describen los parámetros que serán tomados en cuenta en el desarrollo de este trabajo visibles en la tabla 2:

Tabla 2. Cronograma de actividades para la recolección de datos

Variable	Significado
report	Número de muestra
Sys_time	Indica la hora en la que comienza la medición
lte_neighbors	Establece el número de estaciones base LTE vecinos
rssi_strongest	Indica el número de estaciones base LTE vecinos
rssi	Indicador de fuerza de la señal, mide el nivel de potencia de las señales inalámbricas recibidas en el terminal móvil.

rsrq	Mide la calidad de la señal de referencia recibida parámetro relacionado con el rssi
band	La banda de frecuencia en la que está el terminal móvil
lat	Entrega la coordenada de latitud
long	Entrega la coordenada de longitud
node_id	Número de nodo al que está conectado el terminal móvil
cid	Número de celda en el que se conecta el terminal móvil

2.1.2 Zonas y rutas para la recolección de datos

La zona identificada en la figura 8.1 donde se van a realizar las mediciones se ubica en el nororiente de la ciudad de Quito, en la parroquia rural de Calderón, una de las parroquias con mayor índice poblacional de la capital y que por el relieve de esta existen varias zonas “ciegas”, tanto de cobertura celular como de otros servicios inalámbricos.

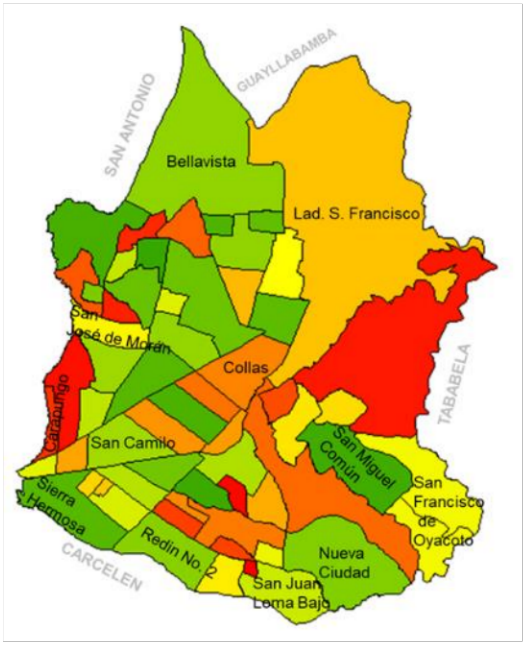


Figura 8.1 Parroquia Calderón con su división de barrios

2.1.2.1 Rutas seleccionadas.

Se han seleccionado tres rutas que servirán como base de estudio para este trabajo, las cuales se describen a continuación.

2.1.2.1.1 Primera Ruta Calderón

Se escoge la ruta, que se representa en la figura 8.2, ya que es una de las principales vías de tránsito que comunica varios barrios de la parroquia, además por su relieve y la cantidad de domicilios ubicados alrededor de esta vía. Es una ruta que tienen una longitud aproximada de 2 kilómetros, donde se presentan muchos procesos de handover

Punto de Inicio: Calle De los Arrieros y Atahualpa

Punto final: Avenida Cacha y Atahualpa

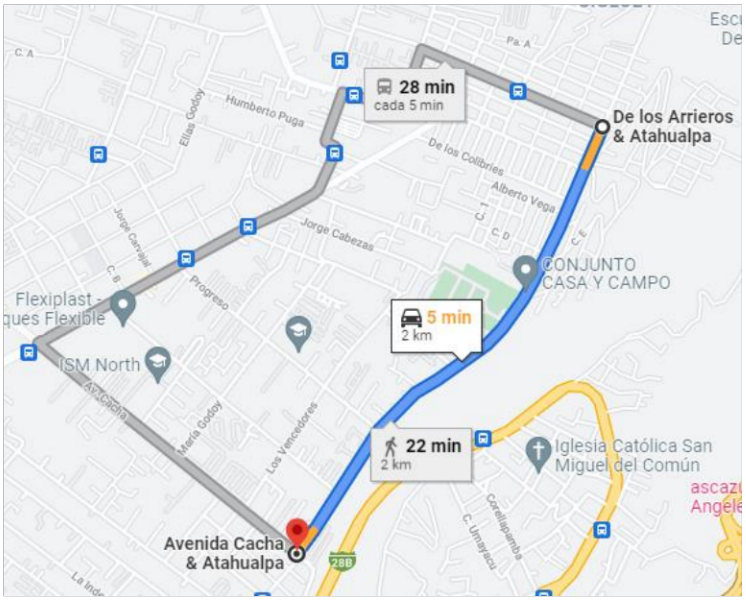


Figura 8.2 Calle Atahualpa. Tomado de Google Maps.

2.1.2.1.2 Segunda Ruta San Juan

La segunda ruta considerada y que se presenta en la Figura 8.3 tiene una longitud aproximada de 2.2 kilómetros, en donde se identificaron varias antenas base. Esta ruta, se trata de una vía medianamente transitada, a diferencia de la ruta anterior, no presenta gran índice poblacional a sus alrededores, parámetro a tomar en cuenta para su estudio más adelante.

Punto de Inicio: Carlos Mantilla & Santander

Punto final: Agustín Guerrero & Amalia Uriguen

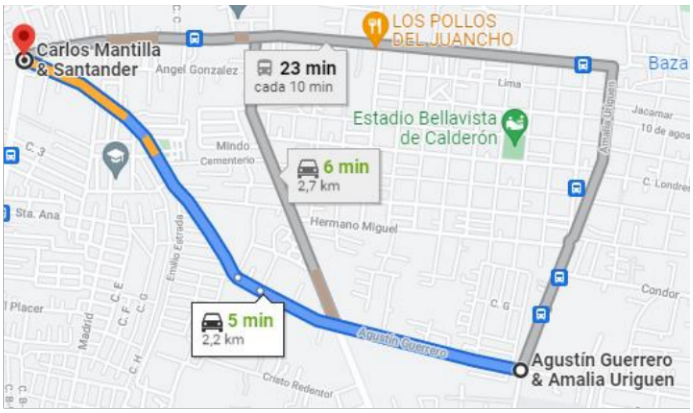


Figura 8.3 Trayecto ruta 2. Tomado de Google Maps.

2.1.2.1.3 Tercera Ruta Marianas

Ruta de 1.7 kilómetros, que se muestra en la figura 8.4, principal vía de conexión, alto tránsito y zona comercial de la parroquia.

Punto de Inicio: Giovanni Calles & Francisco de Albornoz

Punto final: Capitán Giovanni Calles & Avenida Cacha

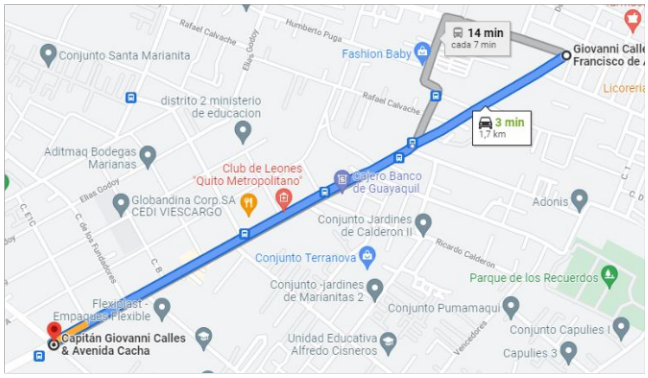


Figura 8.4 Trayecto ruta 3. Tomado de Google Maps.

2.1.2.2 Cronograma de la etapa de recolección datos

Para la etapa de recolección, manipulación y observación de los datos se tiene previsto un tiempo estimado de 38 días, el mismo que empezó el 1 de diciembre finalizó el 7 de enero. A continuación, se presenta en la tabla 3 el cronograma con todas las actividades realizadas en este lapso.

Cabe recalcar que al tener tres rutas se decidió, realizar las mediciones a lo largo del mismo trayecto en varias horas del día y a diferentes velocidades, para esto, se transportó en un bus urbano, automóvil y caminando.

Tabla 3. Cronograma de actividades para la recolección de datos

Cronograma de actividades							
ACTIVIDADES	**Días Asignados**						
	Lunes	**Martes**	**Miércoles**	**Jueves**	**Viernes**	**Sábado**	**Domingo**
Entrega de cronograma escrito			1-dic				
Manipulación y familiarización con herramientas				2-dic	3-dic		
Mediciones en la ruta 1 caminando	6-dic						
Mediciones en la ruta 1 en bus o automóvil		7-dic					
análisis de datos ruta 1			8-dic				

Omisión y corrección de datos ruta 1				9-dic			
Resultados ruta 1 y elaboración de material escrito					10-dic	11-dic	
Mediciones en la ruta 2 caminando	13-dic						
Mediciones en la ruta 2 en bus o automóvil		14-dic					
análisis de datos ruta 2			15-dic				
Omisión y corrección de datos ruta 2				16-dic			
Resultados ruta 2 y elaboración de material escrito					17-dic	18-dic	
Mediciones en la ruta 3 caminando	20-dic						
Mediciones en la ruta 3 en bus o automóvil		21-dic					
análisis de datos ruta 3			22-dic				
Omisión y corrección de datos ruta 3				23-dic			
Resultados ruta 3 y elaboración de material escrito	27-dic	28-dic					
Entrega borrador de etapa de recolección de datos			29-dic				
Correcciones borrador				30-dic			2-ene
Nuevas mediciones de ser necesario	3-ene	4-ene					
Análisis nuevas mediciones			5-ene				
Resultados nuevos mediciones y elaboración de material escrito				6-ene	7-ene		

2.2 Etapa de manipulación de los datos

Una vez finalizada la etapa de recolección de datos en las tres rutas establecidas y definidas previamente, es necesario analizar los datos obtenidos, los cuales en este estudio deben tener una relación directa con el estudio del handover. Se tendrán datos que serán utilizados en las entradas del algoritmo de Machine Learning con técnica de aprendizaje supervisado y modelo de árbol de decisión.

Los datos obtenidos dentro de las tres rutas en total se dividen en 26 sesiones, las cuales presentan sus respectivos números de muestras dentro de cada sesión. Se debe mencionar que las mediciones se realizaron en un horario rotativo. El número total de muestras recolectadas para la base de datos es de 25830. Cabe mencionar que aun utilizando la aplicación ”Force4G”, se presentaron caídas de esta tecnología móvil principalmente en la ruta denominada Calderón. De esta forma se obtuvo la cantidad de muestras por tecnología móvil presentadas en detalle en la figura 9, esta información estadística es proporcionada por la misma aplicación utilizada para la recolección de datos

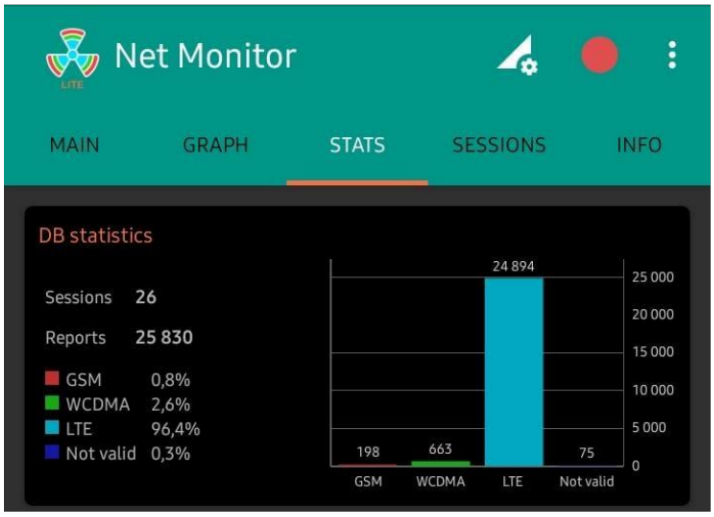


Figura 9. Resumen estadístico de la cantidad de muestras por tecnología de la recolección de datos.

Los archivos obtenidos por cada sesión son de tipo csv, los cuales facilitan su estudio y posterior importación a la herramienta R, es así como a continuación se presenta el número de archivos obtenidos por ruta.

Ruta Calderón: 11 sesiones, Anexo digital I, en esta ruta fue la que más cambios y caídas de servicio presento.

Ruta San Juan: 8 sesiones, Anexo digital II, no se registraron grandes cambios u observaciones significativas.

Ruta Marianas: 7 sesiones, Anexo digital III, al igual que en la ruta San Juan el comportamiento de la tecnología móvil fue estable.

En total la aplicación Net Monitor entrega un archivo csv con 44 variables por muestra, por lo que es necesario una depuración de esta, estas con el fin de relacionarlas con el comportamiento del handover. En la tabla 4 se indica la lista de las variables a utilizar proporcionadas por la aplicación en los archivos csv.

Tabla 4 Variables seleccionadas para el estudio del handover

Variable	Significado
lte_neighbors	Establece el número de estaciones base LTE vecinos
rssi_strongest	Indica el número de estaciones base LTE vecinos
rssi	Indicador de fuerza de la señal, mide el nivel de potencia de las señales inalámbricas recibidas en el terminal móvil.
rsrq	Mide la calidad de la señal de referencia recibida parámetro relacionado con el rssi
band	La banda de frecuencia en la que está el terminal móvil

lat	Entrega la coordenada de latitud
long	Entrega la coordenada de longitud
node_id	Número de nodo al que está conectado el terminal móvil
cid	Número de celda en el que se conecta el terminal móvil

2.3 Código implementado.

2.3.1 Primera Etapa: Importar los archivos csv.

El código implementado en el lenguaje de la herramienta R para importar los datos desde la carpeta creada en el escritorio del autor es el que está en la tabla 5.1 :

Tabla 5.1 Sección del código para importar datos

```
#ESCUELA POLITECNICA NACIONAL
#TRABAJO DE TITULACION
#
#COMPONENTE 1
#CREACION DE UN MODELO PREDICTIVO DE HANDOVER Y EL ESTUDIO EN LAS ZONAS LIMITROFES
#CON LA TECNICA DE APRENDIZAJE SUPERVISADO DE ARBOL DE DESICIÓN
#Autor: Henry Paul Navarrete Luzuriaga
#Librerias a utilizar: dplyr,rpart, rpart.plot (Librería para el árbol de decisión)



# Importar los archivos csv de los datos tomados
# Ruta Zabala - Calderón: Se importa los 11 archivos de la ruta Calderón
setwd("/Users/henry/Documents/BASE DE DATOS PARA TESIS DE GRADO/RUTA CALDERON")
dc1<- read.csv("RZC1.csv",sep=";")
dc2<- read.csv("RZC2.csv",sep=";")
dc3<- read.csv("RZC3.csv",sep=";")
dc4<- read.csv("RZC4.csv",sep=";")
dc5<- read.csv("RZC5.csv",sep=";")
dc6<- read.csv("RZC6.csv",sep=";")
dc7<- read.csv("RZC7.csv",sep=";")
dc8<- read.csv("RZC8.csv",sep=";")
dc9<- read.csv("RZC9.csv",sep=";")
```

```
dc10<- read.csv("RZC10.csv",sep=";")
dc11<- read.csv("RZC11.csv",sep=";")

# RUTA MARIANAS
# Ruta Marianas: Se importa los 7 archivos de la ruta Marianas
setwd("/Users/henry/Documents/BASE DE DATOS PARA TESIS DE GRADO/RUTA MARIANAS")
dm1<- read.csv("RMJ1.csv",sep=";")
dm2<- read.csv("RMJ2.csv",sep=";")
dm3<- read.csv("RMJ3.csv",sep=";")
dm4<- read.csv("RMJ4.csv",sep=";")
dm5<- read.csv("RMJ5.csv",sep=";")
dm6<- read.csv("RMJ6.csv",sep=";")
dm7<- read.csv("RMJ7.csv",sep=";")


#RUTA SAN JUAN
# Ruta San Juan: Se importa los 8 archivos de la ruta San Juan
setwd("/Users/henry/Documents/BASE DE DATOS PARA TESIS DE GRADO/RUTA SAN JUAN")
dsj1<- read.csv("RSJ1.csv",sep=";")
dsj2<- read.csv("RSJ2.csv",sep=";")
dsj3<- read.csv("RSJ3.csv",sep=";")
dsj4<- read.csv("RSJ4.csv",sep=";")
dsj5<- read.csv("RSJ5.csv",sep=";")
dsj6<- read.csv("RSJ6.csv",sep=";")
dsj7<- read.csv("RSJ7.csv",sep=";")
dsj8<- read.csv("RSJ8.csv",sep=";")
```

Con la función **setwd,** Rstudio accede a la dirección de la carpeta creada en el ordenador, en donde se encuentran los archivos que se desean importar

Como segundo punto se debe asignar un nombre a la matriz que se generará una vez importados cada uno de los archivos, para esto se utiliza la función **read.csv,** por último al ser un archivo csv, es necesario incluir en la línea de código el comando **sep=";",** que ayudará al lector diferenciar el criterio de separación de las variables y los datos.

2.3.2 Segunda Etapa: Concatenación de los datos.

Tabla 5.2 Sección del código para concatenar los datos

```
library('dplyr')

# Se concatenan los archivos en una sola base de datos ruta Calderón
datoscalderoninicial <- rbind(dc3,dc6,dc7,dc8,dc9,dc10)
datosprecalderon <- select(datoscalderoninicial, -c(X))
datoscalderon <- rbind(datosprecalderon,dc11,dc1,dc2,dc4)

# Se concatenan los archivos en una sola base de datos ruta marianas
datosmarianasinicial <-rbind(dm1,dm3,dm4,dm5,dm6,dm7)
datospremarianas <- select(datosmarianasinicial, -c(X))
datosmarianas <- rbind(datospremarianas,dm2)

# Se concatenan los archivos en una sola base de datos ruta San Juan
datossanjuaninicial <-rbind(dsj3,dsj6,dsj7)
datospresanjuan <- select(datossanjuaninicial, -c(X))
datossanjuan <- rbind(datospresanjuan,dsj1,dsj2,dsj4)
```

En esta etapa, el código de la tabla 5.2 procede a unir los datos de todas las sesiones en una sola matriz por ruta establecida (datoscalderon, datosmarianas, datossanjuan), lo más importante de esta etapa es identificar que Rstudio en algunas sesiones entrega una columna adicional denominada “X”, la cual si no es eliminada no permitirá unir todos los datos, ya que las dimensiones de las matrices previas serán diferentes, esta columna X es eliminada con la línea de comando ***datospreRUTA <- select(datosRUTA, -c(X))***, aclarando que *datosRUTAinicial* (donde RUTA puede ser, calderon, marianas o sanjuan), es la concatenación de todas las sesiones por ruta que presentan esta columna adicional.

2.3.3 Tercera Etapa: Filtro de variables e inclusión de la columna Handover.

Tabla 5.3 Sección del código para el filtro de variables seleccionadas e inclusión de handover

```
# Creación de la columna handover en ruta calderón

datoscalderon <- select(datoscalderon,c(node_id, lat, long, lte_neighbors,
```

```
rssi_strongest, rssi, rsrq, band, cid)) # se selecciona las variables con las que se trabajaran
datoscalderon$handover <- "no"    # se crea una columna llamada handover con todos los espacios en sus filas rellenados con el carácter "no"
datoscalderon[datoscalderon == 2147483647] <- NA #se identifica las filas que presenten este error determinado por la aplicación cuando existía problemas al tomar la muestra
datoscalderon <- datoscalderon %>%
  na.omit()  # Una vez identificado el error y sustituido por el carácter "NA" se procede a eliminar estas filas de la base de datos"
datoscalderon <- filter(datoscalderon, rssi >= -100)  #filtramos los datos de las filas que posean un rssi >= a -100 dB

# Se procede a identificar las filas donde existe un handover para su posterior estudio y predicción
j<-length(datoscalderon)
for (m in 1:nrow(datoscalderon)) {
  if(m>1){
    x <- datoscalderon$node_id[m]
    y <- datoscalderon$node_id[m-1] #Se crean las variables "X" y "Y" para almacenar el valor de la fila donde el valor del node_id es diferente
  }
  else{
    x=1
    y=1 # en el caso de no existir una variación en los valores de node_id "X" y "Y" se les asigna un valor igual a 1
  }
  if (x != y) {
    datoscalderon$handover[m]<-"si"
    datoscalderon$handover[m-1]<-"si"
    datoscalderon$handover[m-2]<-"si"
    datoscalderon$handover[m+1]<-"si"    # si los valores de "X" y "Y" son diferentes se procede a rellenar los espacios de las filas de la columna handover con el carácter "si", que identifica que hubo un handover en esta muestra, adicional se utiliza el artificio de señalar una muestra antes y una muestra despues para el estudio de las zonas limítrofes
```

```
  }
}

# Creación de la columna handover en ruta Marianas
datosmarianas <- select(datosmarianas,c(node_id, lat, long, lte_neighbors, rssi_strongest, rssi, rsrq, band, cid)) # se selecciona las variables con las que se trabajaran
datosmarianas$handover <- "no"    # se crea una columna llamada handover con todos los espacios en sus filas rellenados con el carácter "no"
datosmarianas[datosmarianas == 2147483647] <- NA  # se identifica las filas que presenten este error determinado por la aplicación cuando existía problemas al tomar la muestra
datosmarianas <- datosmarianas %>%
  na.omit()  # Una vez identificado el error y sustituido por el carácter "NA" se procede a eliminar estas filas de la base de datos"
datosmarianas <- filter(datosmarianas, rssi >= -100)  # filtramos los datos de las filas que posean un rssi >= a -100 dB

# Se procede a identificar las filas donde existe un handover para su posterior estudio y predicción
j<-length(datosmarianas)
for (m in 1:nrow(datosmarianas)) {
  if(m>1){
    x <- datosmarianas$node_id[m]
    y <- datosmarianas$node_id[m-1] #Se crean las variables "X" y "Y" para almacenar el valor de la fila donde el valor del node_id es diferente
  }
  else{
    x=1
    y=1 # en el caso de no existir una variación en los valores de node_id "X" y "Y" se les asigna un valor igual a 1
  }
  if (x != y) {
    datosmarianas$handover[m]<-"si"
    datosmarianas$handover[m-1]<-"si"
    datosmarianas$handover[m-2]<-"si"
```

```
   datosmarianas$handover[m+1]<-"si"    # si los valores de "X" y "Y" son diferentes se procede a rellenar los espacios de las filas de la columna handover con el carácter "si", que identifica que hubo un handover en esta muestra, adicional se utiliza el artificio de señalar una muestra antes y una muestra despues para el estudio de las zonas limítrofes
  }
}

# Creación de la columna handover en ruta San Juan
library('dplyr')
datossanjuan <- select(datossanjuan,c(node_id, lat, long, lte_neighbors, rssi_strongest, rssi, rsrq, band, cid)) # se selecciona las variables con las que se trabajaran
datossanjuan$handover <- "no"    # se crea una columna llamada handover con todos los espacios en sus filas rellenados con el carácter "no"
datossanjuan[datossanjuan == 2147483647] <- NA #se identifica las filas que presenten este error determinado por la aplicación cuando existe problemas al tomar la muestra
datossanjuan <- datossanjuan %>%
  na.omit()  # Una vez identificado el error y sustituido por el carácter "NA" se procede a eliminar estas filas de la base de datos"
datossanjuan <- filter(datossanjuan, rssi >= -100)  #filtramos los datos de las filas que posean un rssi >= a -100 dB

# Se procede a identificar las filas donde existe un handover para su posterior estudio y predicción
j<-length(datossanjuan)
for (m in 1:nrow(datossanjuan)) {
  if(m>1){
    x <- datossanjuan$node_id[m]
    y <- datossanjuan$node_id[m-1] #Se crean las variables "X" y "Y" para almacenar el valor de la fila donde el valor del node_id es diferente
  }
  else{
    x=1
    y=1 # en el caso de no existir una variación en los valores de node_id "X" y "Y" se les asigna un valor igual a 1
```

```
  }
  if (x != y) {
    datossanjuan$handover[m]<-"si"
    datossanjuan$handover[m-1]<-"si"
    datossanjuan$handover[m-2]<-"si"
    datossanjuan$handover[m+1]<-"si"     # si los valores de "X" y "Y" son diferentes se procede a rellenar los espacios de las filas de la columna handover con el carácter "si", que identifica que hubo un handover en esta muestra, adicional se utiliza el artificio de señalar una muestra antes y una muestra despues para el estudio de las zonas limítrofes
  }
}
```

En esta etapa el código de la tabla 5.3, selecciona las variables consideradas para el análisis en Rstudio, estas variables se encuentran en la tabla 4, esto se logra con el comando **select.** Posterior a esto se crea una columna llamada handover en donde se la rellena automáticamente con el carácter “no” en todas sus filas.

Es necesario completar en un inicio, todas las filas de la columna handover con el carácter “no”, con el fin de optimizar el proceso de la creación manual de esta columna, además, debido a que posterior a esto se realiza una identificación de las filas donde ocurre handover y se realiza el cambio por el carácter “si” en las casillas como se indicará posteriormente.

En la revisión de los datos obtenidos por la aplicación se observó que cuando la señal era muy baja la muestra arrojaba el valor 2147483647, que se lo considera como error, por lo que se filtran estas muestras para que no sean parte del estudio. Para esto, se aplica la línea de comando *datosRUTA[datosRUTA == 2147483647] <- NA*, que reemplaza este error por el carácter NA y posterior a esto se lo elimina con el comando na.omit.

Por último en esta parte se crea un lazo for con el fin de identificar la fila donde las muestras cambian de valor de node_id y señalar que hubo un handover, como las muestras son tomadas con un tiempo de 1 segundo y tomando en cuenta que el componente consiste en estudiar el comportamiento del handover en zonas limítrofes, se opta por escoger dos muestras antes del cambio de celda y dos muestras después de

dicho proceso para de esta manera conseguir que el modelo cuente con los datos para el análisis.

2.3.4 Cuarta Etapa: Diseño del modelo de árbol de decisión.

Tabla 5.4 Sección del código para la creación del árbol de decisión

```
# Etapa de creación de árbol de decisión y estadísticas del modelo en ruta Calderón
totalcalderon<-select(datoscalderon, -c(lat, long, cid)) # se crea la variable del conjunto total
totalcalderon$node_id <- as.character(totalcalderon$node_id)  # se define a los valores de la variable node_id como caracteres

# Creación de los conjuntos de entrenamiento y prueba
create_train_test <- function(data, size =0.8, train=TRUE) {
  n_row = nrow(data)
  total_row = size * n_row
  train_sample <-1: total_row
  if (train == TRUE) {
    return (data[train_sample, ])
  } else {
    return (data[-train_sample, ])
  }
}
totalcalderon <- totalcalderon[sample(1:nrow(totalcalderon)), ]

traincalderon <- create_train_test(totalcalderon, 0.8, train = TRUE) # Conjunto de entrenamiento equivalente al 80% de la base total
testcalderon <- create_train_test(totalcalderon, 0.8, train = FALSE) # Conjunto de prueba equivalente al 20% de la base total

library(rpart)
library(rpart.plot)
fitcalderon <- rpart(handover ~ ., data=traincalderon,
            method='class',
            minsplit = 1,
```

```
        minbucket = 1)  # se indica al modelo la variable que deseamos predecir, el método del árbol es de tipo de clasificación

rpart.plot(fitcalderon, extra = 106) # comando para graficar el árbol de decisión
print(fitcalderon)          # comando para entregar en formato texto el árbol de decisión
plotcp(fitcalderon)         # comando para plotear la evolución del error del modelo

predict_unseen <-predict(fitcalderon, testcalderon, type = 'class')
table_matcalderon <- table(testcalderon$handover, predict_unseen)
table_matcalderon # se crea la matriz de incertidumbre para la ruta Calderón

accuracy_Testcalderon <- sum(diag(table_matcalderon)) / sum(table_matcalderon)  # fórmula para calcular el accuracy del modelo
print(paste('Accuracy for test', accuracy_Testcalderon))  # comando para imprimir el valor del accuracy del modelo

sensibilidadcalderon <- (table_matcalderon[4]/(table_matcalderon[2]+table_matcalderon[4])) # fórmula para calcular la sensibilidad del modelo
print(paste('Sensibilidad del test', sensibilidadcalderon)) # comando para imprimir el valor de la sensibilidad del modelo

# Etapa de creación de árbol de decisión y estadísticas del modelo en ruta Marianas
totalmarianas<-select(datosmarianas, -c(lat, long, cid)) # se crea la variable del conjunto total
totalmarianas$node_id <- as.character(totalmarianas$node_id)  # se define a los valores de la variable node_id como caracteres

# Creación de los conjuntos de entrenamiento y prueba
create_train_test <- function(data, size =0.8, train=TRUE) {
  n_row = nrow(data)
  total_row = size * n_row
  train_sample <-1: total_row
  if (train == TRUE) {
    return (data[train_sample, ])
```

```
  } else {
    return (data[-train_sample, ])
  }
}
totalmarianas <- totalmarianas[sample(1:nrow(totalmarianas)), ]
trainmarianas <- create_train_test(totalmarianas, 0.8, train = TRUE) # Conjunto de entrenamiento equivalente al 80% de la base total
testmarianas <- create_train_test(totalmarianas, 0.8, train = FALSE) # Conjunto de prueba equivalente al 20% de la base total

fitmarianas <- rpart(handover ~ ., data=trainmarianas,
                     method='class',
                     minsplit = 1,
                     minbucket = 1)  #se indica al modelo la variable que deseamos predecir, el metodo del árbol es de tipo de clasificación

rpart.plot(fitmarianas, extra = 106) #comando para graficar el árbol de decisión
print(fitmarianas)          # comando para entregar en formato texto el árbol de decisión
plotcp(fitmarianas)         # comando para plotear la evolución del error del modelo

predict_unseen <-predict(fitmarianas, testmarianas, type = 'class')
table_matmarianas <- table(testmarianas handover, predict_unseen)
table_matmarianas # se crea la matriz de incertidumbre para la ruta marianas

accuracy_Testmarianas <- sum(diag(table_matmarianas)) / sum(table_matmarianas) # fórmula para calcular el accuracy del modelo
print(paste('Accuracy del test', acuracy_Testmarianas))  # comando para imprimir el valor del accuracy del modelo

sensibilidadmarianas <- (table_matmarianas[4]/(table_matmarianas[2]+table_matmarianas[4])) # fórmula para calcular la sensibilidad del modelo
print(paste('Sensibilidad del test', sensibilidadmarianas)) # comando para imprimir el valor de la sensibilidad del modelo

# Etapa de creación de árbol de decisión y estadísticas del modelo en ruta San
```

Juan

```
totalsanjuan<-select(datossanjuan, -c(lat, long, cid)) # se crea la variable del conjunto total
totalsanjuan$node_id <- as.character(totalsanjuan$node_id)  # se define a los valores de la variable node_id como caracteres

# Creación de los conjuntos de entrenamiento y prueba
create_train_test <- function(data, size =0.8, train=TRUE) {
  n_row = nrow(data)
  total_row = size * n_row
  train_sample <-1: total_row
  if (train == TRUE) {
    return (data[train_sample, ])
  } else {
    return (data[-train_sample, ])
  }
}
totalsanjuan <- totalsanjuan[sample(1:nrow(totalsanjuan)), ]

trainsanjuan <- create_train_test(totalsanjuan, 0.8, train = TRUE) # Conjunto de entrenamiento equivalente al 80% de la base total
testsanjuan <- create_train_test(totalsanjuan, 0.8, train = FALSE) # Conjunto de prueba equivalente al 20% de la base total

fitsanjuan <- rpart(handover ~ ., data=trainsanjuan,
                    method='class',
                    minsplit = 1,
                    minbucket = 1)  # se indica al modelo la variable que deseamos predecir, el metodo del árbol es de tipo de clasificación

rpart.plot(fitsanjuan, extra = 106) # comando para graficar el árbol de decisión
print(fitsanjuan)          # comando para entregar en formato texto el árbol de decisión
plotcp(fitsanjuan)         # comando para plotear la evolución del error del modelo

predict_unseen <-predict(fitsanjuan, testsanjuan, type = 'class')
table_matsanjuan <- table(testsanjuan$handover, predict_unseen)
```

```
table_matsanjuan # se crea la matriz de incertidumbre para la ruta san juan

accuracy_Testsanjuan <- sum(diag(table_matsanjuan)) / sum(table_matsanjuan) # fórmula para calcular el accuracy del modelo
print(paste('Accuracy del test', accuracy_Testsanjuan)) # comando para imprimir el valor del accuracy del modelo

sensibilidadsanjuan <- (table_matsanjuan[4]/(table_matsanjuan[2]+table_matsanjuan[4])) # fórmula para calcular la sensibilidad del modelo
print(paste('Sensibilidad del test', sensibilidadsanjuan)) # comando para imprimir el valor de la sensibilidad del modelo
```

El código de la tabla 5.5, como primer punto elimina las variables que no serán parte del estudio del handover pero que, si se utilizaron para identificar previamente estos cambios en las muestras.

Posterior a eso se crean los conjuntos de entrenamiento y test para el modelo del árbol de decisión en base a la técnica de Machine Learning de aprendizaje supervisado. En esta parte también se aprovecha, la ventaja de que Rstudio ya posee una librería y funciones que permite simplificar el proceso de programación para la aplicación del modelo y es lo que se usa en la parte final del código, dicha librería es rpart.

Por último, es necesario plotear el árbol de decisión, para su posterior análisis, generar una matriz de incertidumbre para verificar la cantidad de falsos positivos y falsos negativos en el modelo, el porcentaje de accuracy del modelo y la sensibilidad de éste. El modelo debe ser replicado para las matrices unidas de las tres distintas rutas y hacer el respectivo análisis tomando en cuenta los patrones que se utilizaron previamente para la elección de dichas rutas. Es necesario mencionar que un modelo con sensibilidad más alta presentará mejores resultados, esto también se pondrá en comparación con la evolución del error del modelo que se lo obtendrá gráficamente con la función ***plotcp***, perteneciente a la librería rpart.

Para una mejor comprensión del código que se utilizó para el desarrollo de este estudio, estará adjunto como anexo II.

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1 Resultados

En este capítulo se pretende analizar y desglosar la información obtenida en los árboles de decisión, en base al modelo y código propuesto en el anterior capitulo. Para obtener una mejor comprensión de los resultados se va a realizar un análisis por cada ruta.

3.1.1 Ruta Calderón.

Para entender el comportamiento de los datos recolectados en esta ruta se debe mencionar las siguientes observaciones que se presentaron al momento de recolectar estos datos.

3.1.1.1 Observaciones

3.1.1.1.1 Horario de recolección

Los datos fueron recolectados en dos horarios distintos, entre las 10 am y 1 pm de lunes a miércoles de la semana previamente establecida dentro del cronograma propuesto, y entre las 5 pm y 7 pm de los días ya mencionados, esto con el fin de observar si existía alguna variación en los datos recolectados acerca del proceso de handover. Se debe mencionar que al ser una zona residencial la gran parte de las personas se encuentran llegando a sus hogares en el horario vespertino, mientras que en el horario matutino el índice poblacional disminuye en gran porcentaje, si bien el contexto de la pandemia por Covid-19, obliga a muchas personas y estudiantes a realizar actividades telemáticas, las condiciones y restricciones al momento de realizar las mediciones se mantenían flexibles para cumplir con este tipo de actividades de manera presencial.

3.1.1.1.2 Velocidad de movimiento

Alrededor del 75 por ciento de los datos recolectados en esta ruta se la recolectó a una velocidad promedio de 5.8 Km/h lo que equivale a 1.6 m/s que es la velocidad que se trató de mantener al momento de caminar por la zona, mientras que el otro 25 por ciento de los datos se la obtuvo a una velocidad promedio de 40 Km/h, al transportarse en varias unidades de transporte público que circulan por la zona

3.1.1.1.3 Altura

La altura de recolección de muestras en el 75 por ciento de la base de datos es de un 1.5 metros, mientras que la altura promedio cuando los datos eran recolectados en la unidad de transporte fue de alrededor de 3 metros

3.1.1.1.4 Cantidad de estaciones base (node_id) en la zona

En la tabla 6.1 se presentan todos los node_id, que son también conocidos como la identificación del radio base que fueron reconocidos por la aplicación Net Monitor Lite y que será la principal variable para el estudio del modelo utilizado.

Tabla 6.1 Tabla identificación de CID y NODE_ID identificados en la ruta Zabala-Calderón

cid	Node-id	Tecnología
190	28247	LTE
190	28251	LTE
190	28353	LTE
190	28451	LTE
191	28474	LTE
190	28686	LTE
191	28759	LTE
191	28811	LTE
191	28816	LTE
190	28864	LTE
190	28934	LTE
55474	10	WCDMA
37038	10	WCDMA

En la tabla 6.1, se identifica como la gran mayoría de los node_id pertenecen a la tecnología LTE, sin embargo, dentro de esta ruta existían zonas muy marcadas donde, la señal decaía, por lo que el terminal móvil al estar funcionando con una conexión forzada LTE, gracias a la aplicación Force 4G, los datos de las variables en estas muestras no otorgan información relevante para el estudio planteado.

3.1.1.2 Base de datos, conjunto de entrenamiento y prueba.

El conjunto de datos de esta ruta se divide en, una parte de entrenamiento perteneciente al 80 por ciento y el de prueba correspondiente al 20 por ciento restante. Esta división se realiza una vez que se filtran las señales con muy poca intensidad y se hayan definido las celdas en donde existe handover. En la tabla 6.2 se indica la cantidad de datos que disponibles en la ruta Zabala Calderón.

Tabla 6.2 Dimensiones de conjuntos de datos para ruta Zabala – Calderón

Conjunto	Dimensión	Variables ingresadas
Base de datos	1490	7
Entrenamiento	1192	7
Prueba	298	7

3.1.1.3 Árbol de decisión obtenido

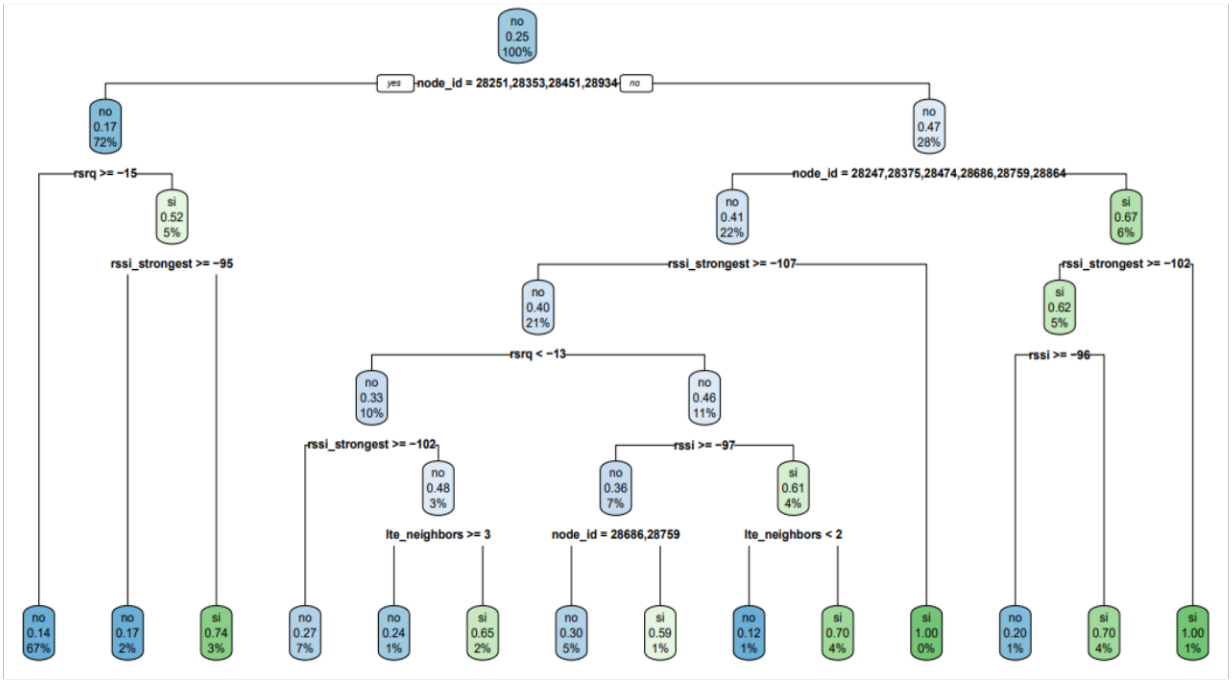


Figura 10.1 Árbol de decisión ruta Zabala – Calderón

Para entender el comportamiento y los resultados del árbol de la figura 10.1 se utiliza la función “print” de la herramienta RStudio para poder imprimir el árbol de decisión en un formato texto, el cual se lo observa en la figura 10.2, lo que permitirá sacar conclusiones de la funcionalidad del modelo en esta ruta.

En base a lo obtenido para esta ruta se puede concluir lo siguiente:

Al modelo para el entrenamiento han ingresado 1192 muestras de datos cada una con sus 7 variables de entrada. En la raíz del árbol, (parte superior del gráfico) se observa que, de las 1192 muestras iniciales 298 no presenta handover lo que equivale al 25% de la base de datos inicial

```
n= 1192

node), split, n, loss, yval, (yprob)
      * denotes terminal node

  1) root 1192 298 no (0.7500000 0.2500000)
    2) node_id=28251,28353,28451,28934 861 144 no (0.8327526 0.1672474)
      4) rsrq>=-15.5 799 112 no (0.8598248 0.1401752) *
      5) rsrq< -15.5 62  30 si (0.4838710 0.5161290)
       10) rssi_strongest>=-95.5 24   4 no (0.8333333 0.1666667) *
       11) rssi_strongest< -95.5 38  10 si (0.2631579 0.7368421) *
    3) node_id=28247,28276,28375,28474,28609,28686,28759,28811,28816,28864 331 154 no
 (0.5347432 0.4652568)
      6) node_id=28247,28375,28474,28686,28759,28864 259 106 no (0.5907336 0.4092664)
       12) rssi_strongest>=-107.5 254 101 no (0.6023622 0.3976378)
         24) rsrq< -13.5 123  41 no (0.6666667 0.3333333)
           48) rssi_strongest>=-102.5 83  22 no (0.7349398 0.2650602) *
           49) rssi_strongest< -102.5 40  19 no (0.5250000 0.4750000)
             98) lte_neighbors>=2.5 17   4 no (0.7647059 0.2352941) *
             99) lte_neighbors< 2.5 23   8 si (0.3478261 0.6521739) *
         25) rsrq>=-13.5 131  60 no (0.5419847 0.4580153)
           50) rssi>=-97.5 80  29 no (0.6375000 0.3625000)
            100) node_id=28686,28759 63  19 no (0.6984127 0.3015873) *
            101) node_id=28474 17   7 si (0.4117647 0.5882353) *
           51) rssi< -97.5 51  20 si (0.3921569 0.6078431)
            102) lte_neighbors< 1.5 8   1 no (0.8750000 0.1250000) *
            103) lte_neighbors>=1.5 43  13 si (0.3023256 0.6976744) *
       13) rssi_strongest< -107.5 5   0 si (0.0000000 1.0000000) *
      7) node_id=28276,28609,28811,28816 72  24 si (0.3333333 0.6666667)
       14) rssi_strongest>=-102.5 63  24 si (0.3809524 0.6190476)
         28) rssi>=-96.5 10   2 no (0.8000000 0.2000000) *
         29) rssi< -96.5 53  16 si (0.3018868 0.6981132) *
       15) rssi_strongest< -102.5 9   0 si (0.0000000 1.0000000) *
```

Figura 10.2. Árbol de decisión ruta Zabala – Calderón en formato texto

El primer parámetro de comparación del árbol se lo hace en base a que, si la muestra pertenece o no a los nodos 28251, 28353, 28451, 28934, en el caso de pertenecer a estos nodos las muestras pasan al ramal de la parte izquierda, para el resto de las comparaciones se seguirá esta misma lógica, es decir, si cumple con la condición al lado izquierdo, caso contrario al derecho.

Ramal izquierdo: Son 861 muestras que cumplen con la primera condición, de este subconjunto de muestras 144 no presentan handover, mientras que el restante de muestras pasa a la comparación de rsrq (calidad de la señal de referencia recibida). De las 799, su RSRQ es mayor o igual a -15.5 dBm, 112 no presentan handover, mientras que en las 62 muestras, su señal es menor a -15.5 dBm, 30 de estas si presentan handover significando así el 52% de este subconjunto.

La última comparación de este ramal es si el rssi_strongest es mayor o igual a -95.5 dBm, de los cuales 4 de las 24 muestras que cumplen con esta condición no presentan handover, mientras que 10 de las 38 muestras que no cumplen con esta condición presentan handover.

Ramal derecho: El procedimiento es similar, primero se organizan subconjuntos para saber en qué nodo se encuentra la muestra, para este nivel se realiza la comparación con

los nodos 28247, 28276, 28375, 28474, 28609, 28686, 28759, 28811, 28816, 28864, generando una subdivisión de los que pertenecen o no.

El programa propone varias condiciones para este conjunto de datos, la primera es que si estos tienen un rssi_strongest mayor o igual a -107 dBm, posterior a esto si cumplen con esta condición se compara si el RSRQ es menor a -13 dBm. En conclusión, dentro de este sub-ramal y en base a las condiciones propuestas por el modelo se obtienen 21 muestras en donde se ejecutó el handover. Mientras que en el ramal que no cumplen con la segunda condición de comparación de los nodos se obtienen 49 muestras donde se produjo handover.

Al realizar la comprobación del modelo con el conjunto de prueba se obtiene que en total el modelo arroja 80 predicciones afirmativas de que se producirá handover en un conjunto de prueba de 298 muestras.

Una vez entendido el principio de funcionamiento de este árbol para la manipulación de las muestras, es fácil comprenderlo con simplemente observar la figura 1. El modelo arroja en la parte del ramal derecho después de poner varias condiciones a las muestras que no cumplieron

Se puede observar claramente como la variable principal que utiliza el modelo para las comparaciones es la numerología del node_id al que se mantiene conectado el terminal, mientras que el resto de los valores utilizados para las comparaciones de este modelo se encuentran, en base al algoritmo de la función “rpart” y obviamente los datos proporcionados por muestra.

3.1.1.4 Evaluación del modelo.

Además de esto el modelo arroja varias estadísticas muy interesantes para ser analizadas, como lo son la matriz de incertidumbre visualizada en la tabla 6.3, accuracy y sensibilidad en la tabla 6.4 y la evolución del error figura 10.3.

Tabla 6.3 Matriz de incertidumbre para ruta Zabala – Calderón

Existe Handover	NO	SI
NO	193	25
SI	54	26

Esta matriz proporciona una idea, de cuantas veces el modelo arroja un falso positivo, además de encontrar el accuracy que es el porcentaje de validación del modelo en base a los aciertos y errores y se lo calcula en base a la ecuación 1, mientras que para

encontrar la sensibilidad del modelo nos centramos netamente en las predicciones positivas otorgadas ya que el principal objetivo es predecir el handover, la sensibilidad será calculada en base a la ecuación 2.

El cálculo del accuracy y sensibilidad se realiza a continuación:

$$\%\ acurracy = \frac{aciertos}{errores+aciertos} * \ 100\% \qquad \text{ec.1}$$

$$\%\ acurracy = \frac{193+26}{(25+54)+(193+26)} * \ 100\%$$

$$\%\ acurracy = \frac{219}{296} * \ 100\%$$

$$\%\ acurracy = 73.98\ \%$$

$$\%sensibilidad = \frac{Aciertos\ en\ la\ prediccion\ positiva}{(Falsos\ positivos+Aciertos)\ en\ la\ prediccion\ positiva} * \ 100\% \qquad \text{ec.2}$$

$$\%sensibilidad = \frac{26}{54+26} * \ 100\%$$

$$\%sensibilidad = 32.5\ \%$$

Tabla 6.4 Accuracy y Sensibilidad en la ruta Zabala – Calderón

Estadística	Valor en %
Accuracy	73.98%
Sensibilidad	32.5 %

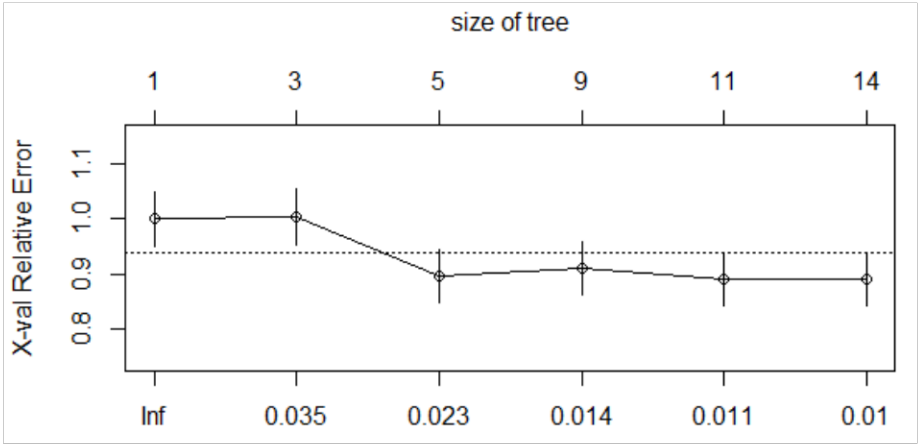


Figura 10.3. Evolución del error en la ruta Zabala-Calderón

Este gráfico permite observar cómo evoluciona el error, es decir, este podría aumentar o disminuir en base a los ramales que se empiezan a crear, como motivo de estudio en un siguiente trabajo, se podría aplicar la técnica de “Podar el árbol” con el fin de analizar qué cambios existirían, en comparación con un árbol no podado. Para fines de este trabajo no se lo realizará, sin embargo, se deducirá su comportamiento: Se puede observar, que casi en todo momento el error permanece semi constante ya que existen subidas y bajadas, a excepción del intervalo 3 y 5 donde el error claramente disminuye, pero luego vuelve al comportamiento inicial de una constante.

Este modelo de predicción presenta un accuracy de 74% y una sensibilidad de 32.5% en la ruta Zabala - Calderón, medidas que pueden ser mejoradas en futuros trabajos, cabe resaltar que esta zona presentó tramos donde la señal era nula y que solo se recolectaron y utilizaron muestras con tecnología móvil 4G.

3.1.2 Ruta Marianas.

Dentro del estudio de esta ruta es necesario mencionar algunas observaciones que se encontraron al momento de recolectar los datos.

3.1.2.1 Observaciones

3.1.2.1.1 Horario de recolección

Los datos fueron recolectados en dos horarios distintos, el primer horario fue de 11 am a 1 pm, mientras que el segundo horario fue entre las 5 pm y 6 pm de lunes a miércoles de la semana previamente establecida dentro del cronograma propuesto, con el fin de observar si existía alguna variación en el proceso de handover. Esta ruta a diferencia de la anterior se presenta como una zona comercial y de gran tránsito de personas, debido que dentro de sus zonas aledañas existen, colegios, escuelas, locales comerciales, bares, gimnasios, veterinarias y bodegas de empresas. La justificación del primer horario se basa en que es una hora pico debido a que las personas consumen sus alimentos y los estudiantes salen de su jornada estudiantil (gran mayoría de las instituciones cumplían con actividades presenciales, en la fecha que se tomaron los datos), mientras que la justificación del segundo horario va ligado a que la jornada laboral termina, los trabajadores se dirigen a sus casas y existe mayor número de afluencia de personas a locales de distracción y de recreación como bares, gimnasios y locales comerciales.

3.1.2.1.2 Velocidad de movimiento

Alrededor del 80 por ciento de los datos de esta ruta se recolectaron a una velocidad promedio de 5 Km/h lo que equivale a 1.3 m/s correspondiente a la velocidad del peatón que tomó las muestras en esta ruta, mientras que el 20 por ciento restante de los datos se obtuvieron a una velocidad promedio de 35 Km/h, en varias unidades de transporte público que circulan por la zona.

3.1.2.1.3 Altura

La altura de recolección de muestras en el 80 por ciento de la base de datos es de un 1 metro, mientras que la altura promedio cuando los datos eran recolectados en la unidad de transporte fue de alrededor de 3.5 metros

3.1.2.1.4 Cantidad de estaciones base (node_id) en la zona

En la tabla 7.1 se presentan todos los node_id identificados dentro de esta ruta

Tabla 7.1 Tabla identificación de CID y NODE_ID identificados en la ruta Marianas

cid	Node-id	Tecnología
191	28474	LTE
191	28759	LTE
191	28816	LTE
191	28759	LTE

Se observa en la tabla 7.1, que todos los node_id pertenecen al cid 191, en comparación con la ruta anterior se observa que estos node_id también estaban presentes para el respectivo estudio de la ruta y por ende todos tienen una tecnología móvil LTE de gran cobertura que no genera inconveniente alguno, esto resulta ser muy obvio ya que al ser una zona de gran afluencia los operadores móviles tienden a tener bien cubiertas estas áreas.

3.1.2.2 Base de datos, conjunto de entrenamiento y prueba.

Al igual que con el caso pasado el conjunto de datos de esta ruta, se dividieron en entrenamiento y prueba con la división de 80% y 20% respectivamente, la dimensión de estos conjuntos se los observa en la tabla 7.2.

Tabla 7.2 Dimensiones de conjuntos de datos para ruta Marianas

Conjunto	Dimensión	Variables ingresadas
Base de datos	2867	7
Entrenamiento	2293	7
Prueba	574	7

3.1.2.3 Árbol de decisión obtenido

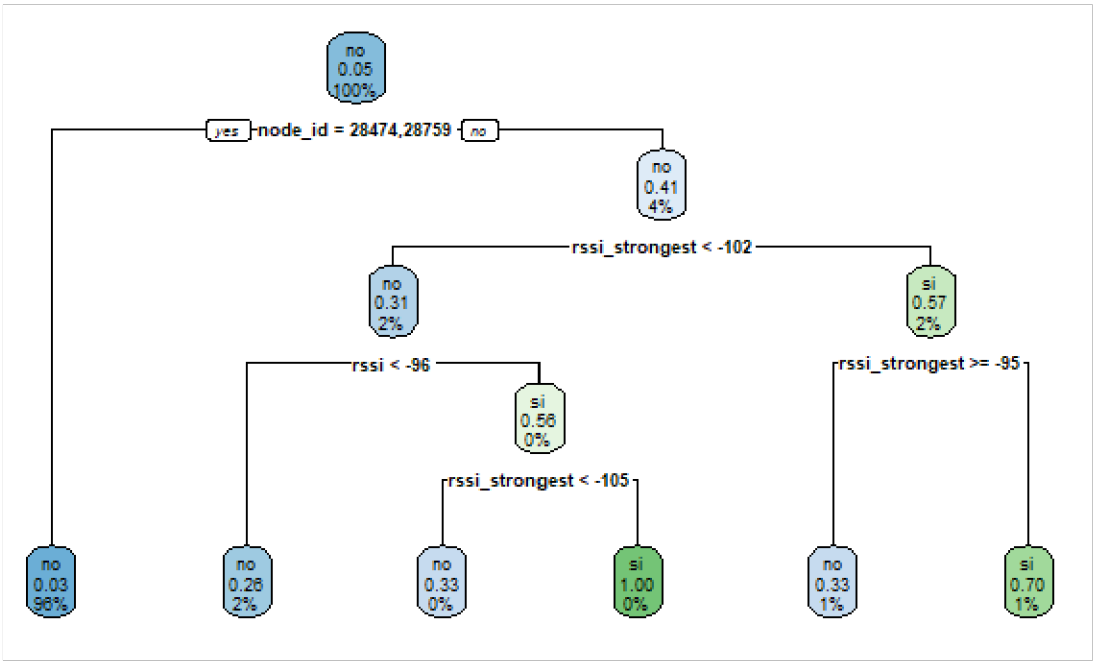


Figura 11.1 Árbol de decisión de la ruta Marianas

En la figura 11.1 se observa a simple vista que es un árbol menos complejo en comparación con el árbol obtenido en la ruta Zabala-Calderón figura 10.1, esto debido a que, se puede deducir a simple vista que la variable de mayor importancia utilizada por el algoritmo de la función de rpart para la predicción del handover de esta ruta es el rssi_strongest (celda vecina con mayor indicador de fuerza de la señal) , de cierto modo esto era predecible, ya que en esta zona el handover va a depender mucho de la saturación de usuarios conectados a una radio base. En la figura 11.2 se detalla el mismo árbol en formato texto con una descripción de las condiciones presentes:

```
n= 2293

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 2293 107 no (0.95333624 0.04666376)
   2) node_id=28474,28759 2203  70 no (0.96822515 0.03177485) *
   3) node_id=28816 90  37 no (0.58888889 0.41111111)
     6) rssi_strongest< -102.5 55  17 no (0.69090909 0.30909091)
      12) rssi< -96.5 46  12 no (0.73913043 0.26086957) *
      13) rssi>=-96.5 9   4 si (0.44444444 0.55555556)
        26) rssi_strongest< -105 6   2 no (0.66666667 0.33333333) *
        27) rssi_strongest>=-105 3   0 si (0.00000000 1.00000000) *
     7) rssi_strongest>=-102.5 35  15 si (0.42857143 0.57142857)
      14) rssi_strongest>=-95.5 12   4 no (0.66666667 0.33333333) *
      15) rssi_strongest< -95.5 23   7 si (0.30434783 0.69565217) *
```

Figura 11.2 Árbol de decisión ruta Marianas en formato texto

En base a lo obtenido para esta ruta se puede concluir lo siguiente:

Al modelo han ingresado 2293 muestras cada una con sus 7 variables de entrada para el modelo. En la raíz del árbol, se observa que, de las 2293 muestras iniciales 107 indican que no tienen handover equivalente al 4.6% de la base de datos inicial.

Se realiza la primera comparación donde se pregunta si pertenecen a los nodos 28474,28759, se observa que 2203 muestras cumplen con esta condición y además 70 muestras de este subconjunto no presentan handover, lo que le corresponde el 3% de estos datos.

Los datos que no cumplan con la condición inicial pasan al siguiente nivel donde se revisa la condición de que si el rssi_strongest es menor a -102.5 dB, observamos que 55 de las 90 muestras que quedaban cumplen con la condición y a la vez 17 de estas no presentan handover. A estos 55 datos se les propone una nueva condición para saber si el RSSI es menor que -96.5 dB, 46 muestras cumplen con esta condición y a la vez 12 de estas no presentan handover, pero de las 9 muestras que no cumplen con esta condición 5 si tienen handover, equivalente al 55% de este pequeño subconjunto de datos. El modelo profundiza el análisis con estas 9 muestras y se propone otra condición para rssi_strongest, esta vez se comprueba si es menor a -105 dB, 6 muestras cumplen con esta condición, y 2 de estas no presentan handover, mientras que las 3 muestras que no cumplen la condición tienen handover.

En el caso de las 35 muestras que el rssi_strongest era mayor o igual a -102.5 dB, 15 de estas presentan handover, lo que corresponde al 57.1 % de este subconjunto, por último se propone una condición más a estas 35 que consisten en conocer si el rssi_strongest es mayor o igual a -95.5 dB, 12 muestras cumplen con esta condición y a la vez 4 de este subgrupo no tienen handover, mientras que de las 23 restantes que no cumplen con esta condición 7 presentan handover.

Si sumamos todas las muestras que dentro del modelo afirman tener handover se obtienen 28 muestras un dato importante a tomar en cuenta más adelante.

3.1.2.4 Evaluación del modelo.

Ahora se analizarán las estadísticas arrojadas por el modelo y serán presentadas en la tabla 7.3 que corresponde a la matriz de incertidumbre, mientras que en la tabla 7.4 se muestra el accuracy y sensibilidad, por último, la evolución del error en la figura 11.3.

Tabla 7.3 Matriz de incertidumbre para la ruta Zabala – Calderón

Existe Handover	NO	SI
NO	545	0
SI	21	8

Se observa claramente como de los 29 datos que el modelo predijo que existía handover, 21 son falsos positivos y solo 8 cumplen con la variable deseada. Es necesario observar el accuracy y la sensibilidad de este modelo para poder obtener conclusiones de este.

El cálculo del accuracy y sensibilidad se realiza a continuación:

$$\%\, acurracy = \frac{aciertos}{errores+aciertos} * \ 100\% \qquad \text{ec.1}$$

$$\%\, acurracy = \frac{545+8}{(21+0)+(545+8)} * \ 100\%$$

$$\%\, acurracy = \frac{553}{574} * \ 100\%$$

$$\%\, acurracy = 96.34\ \%$$

$$\%sensibilidad = \frac{Aciertos\ en\ la\ prediccion\ positiva}{(Falsos\ positivos+Aciertos)\ en\ la\ prediccion\ positiva} * \ 100\% \qquad \text{ec.2}$$

$$\%sensibilidad = \frac{8}{21+8} * \ 100\%$$

$$\%sensibilidad = 27.6\ \%$$

Tabla 7.4 Accuracy y Sensibilidad en la ruta Marianas

Estadística	Valor en %
Accuracy	96.34 %
Sensibilidad	27.6 %

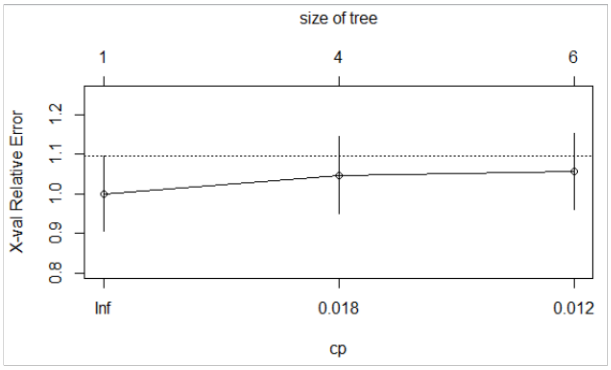


Figura 11.3 Evolución del error en la ruta Marianas

En este caso el error del árbol aumenta apenas su dimensión empieza a crecer hasta el intervalo 4, donde se presenta un comportamiento constante hasta su ramificación final.

Si bien el modelo en esta ruta obtuvo un mejor accuracy y mayor sensibilidad, la evolución del error es mayor al que se obtuvo con la ruta pasada, en esta base de datos se ingresaron 574 datos de prueba al modelo el cual pudo predecir solo 28 muestras en comparación con las 80 muestras que predijo en la anterior ruta con un número menor de datos ingresados, son resultados esperados, ya que desde el momento de la recolección de datos se observó que el número de node_id reconocidos por la aplicación fueron pocos, sin embargo se observa que son los necesarios para cubrir y dar una buena cobertura a esta zona.

3.1.3 Ruta San Juan.

Dentro del estudio de esta ruta es necesario mencionar algunas observaciones que se presentaron durante la recolección de datos.

3.1.3.1 Observaciones

3.1.3.1.1 Horario de recolección

Para esta ruta se recolectaron los datos en el horario de 8 am a 10 am y en el horario de 7 pm a 9 pm, la justificación de estos horarios netamente va ligada a la disponibilidad del medio de transporte y la accesibilidad a esta ruta ya que es una sola con bajo movimiento comercial y peatonal.

3.1.3.1.2 Velocidad de movimiento

Al igual que con los casos pasados la mayoría de los datos recolectados fue a una velocidad promedio de 1.5 m/s, correspondiente a un 75% de la base de datos, mientras que el otro 25% fue recolectado en un automóvil a una velocidad promedio de 30 Km/h

3.1.3.1.3 Altura

La altura de recolección de las muestras obtenidas a pie fue de 1.5 metros, mientras que las recolectadas en automóvil fue aproximadamente a 1 metro de altura.

3.1.3.1.4 Cantidad de estaciones base (node_id) en la zona

En la tabla 8.1 se presentan todos los node_id que la aplicación identifico dentro de esta ruta

Tabla 8.1 Tabla identificación de CID y NODE_ID identificados en la ruta Marianas

cid	Node-id	Tecnología
191	28001	LTE
190	28474	LTE
190	28611	LTE
192	28783	LTE
192	28811	LTE
191	28816	LTE
191	28845	LTE

Se observa en la tabla 8.1, la presencia de 3 cid en la ruta establecida, la aplicación reconoció 7 distintos node_id que le servirán para el estudio de esta ruta, el terreno básicamente se lo define como una planicie donde la cobertura móvil es regular con presencia total de la tecnología LTE, por lo que se entiende que el operador móvil tiene una cobertura dentro de esta ruta.

3.1.3.2 Base de datos, conjunto de entrenamiento y prueba.

En la tabla 8.2 se presentan las dimensiones correspondientes para los conjuntos de la base de datos, dividida en, entrenamiento y prueba. Estas dimensiones se obtuvieron una vez aplicados los filtros necesarios a las muestras y la respectiva división de 80% y 20% respectivamente

Tabla 8.2 Dimensiones de conjuntos de datos para la ruta Marianas

Conjunto	Dimensión	Variables ingresadas
Base de datos	1854	7
Entrenamiento	1483	7
Prueba	371	7

3.1.3.3 Árbol de decisión obtenido

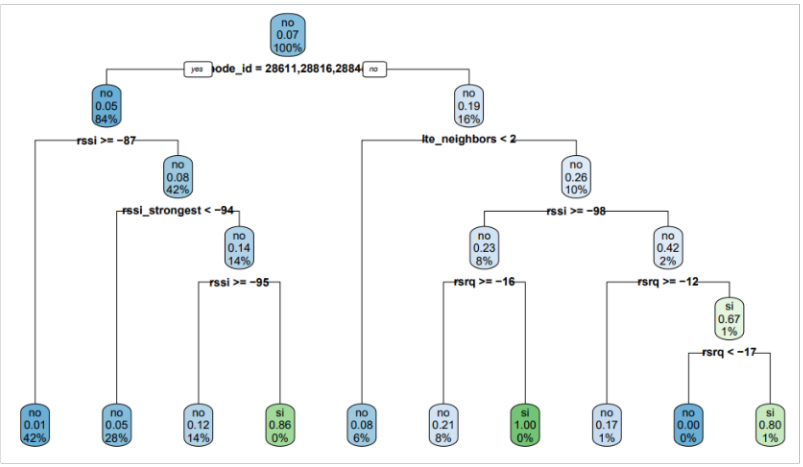


Figura 12.1 Árbol de decisión ruta San Juan

En la figura 12.1 el árbol mantiene un comportamiento similar a los ya explicados anteriormente en las rutas pasadas, se hace énfasis en la comparación inicial de los nodos, las comparaciones de los valores de RSSI, rssi_strongest, RSRQ y una nueva variable lte_neighbors, que no ha sido utilizada por el algoritmo en las condiciones de comparación de los árboles pasados.

```
n= 1483

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 1483 102 no (0.931220499 0.068779501)
   2) node_id=28611,28816,28845 1250  57 no (0.954400000 0.045600000)
     4) rssi>=-87.5 628   5 no (0.992038217 0.007961783) *
     5) rssi< -87.5 622  52 no (0.916398714 0.083601286)
      10) rssi_strongest< -94.5 409  22 no (0.946210269 0.053789731) *
      11) rssi_strongest>=-94.5 213  30 no (0.859154930 0.140845070)
        22) rssi>=-95.5 206  24 no (0.883495146 0.116504854) *
        23) rssi< -95.5 7   1 si (0.142857143 0.857142857) *
   3) node_id=28001,28474,28783,28811 233  45 no (0.806866953 0.193133047)
     6) lte_neighbors< 1.5 89   7 no (0.921348315 0.078651685) *
     7) lte_neighbors>=1.5 144  38 no (0.736111111 0.263888889)
      14) rssi>=-98.5 120  28 no (0.766666667 0.233333333)
        28) rsrq>=-16.5 117  25 no (0.786324786 0.213675214) *
        29) rsrq< -16.5 3   0 si (0.000000000 1.000000000) *
      15) rssi< -98.5 24  10 no (0.583333333 0.416666667)
        30) rsrq>=-12.5 12   2 no (0.833333333 0.166666667) *
        31) rsrq< -12.5 12   4 si (0.333333333 0.666666667)
          62) rsrq< -17.5 2   0 no (1.000000000 0.000000000) *
          63) rsrq>=-17.5 10   2 si (0.200000000 0.800000000) *
```

Figura 12.2 Árbol de decisión de la ruta San Juan en formato de texto

La figura 12.2, muestra este nuevo árbol en forma de texto, donde lo más relevante es que el modelo arroja 33 predicciones de que existirá handover en el grupo de 371 muestras ingresadas en el conjunto de prueba, como ya ha sido mencionado anteriormente, estos datos servirán para encontrar las estadísticas y comportamiento del modelo

3.1.3.4 Evaluación del modelo.

Ahora se analizarán las estadísticas arrojadas por el modelo y serán presentadas en la tabla 8.3 que representa la matriz de incertidumbre, mientras que en la tabla 8.4 el accuracy y sensibilidad, por último, la evolución del error se puede observar en la figura 12.3.

Tabla 8.3 Matriz de incertidumbre para la ruta San Juan

Existe Handover	NO	SI
NO	336	2
SI	23	10

Se observa claramente como de los 33 datos que el modelo predijo que existía handover, 23 son falsos positivos y solo 10 cumplen con la predicción de la variable deseada.

El cálculo del accuracy y sensibilidad se realiza a continuación:

$$\%\, acurracy = \frac{aciertos}{errores+aciertos} * \ 100\% \qquad \text{ec.1}$$

$$\%\, acurracy = \frac{336+10}{(23+2)+(336+10)} * \ 100\%$$

$$\%\, acurracy = \frac{346}{371} * \ 100\%$$

$$\%\, acurracy = 93.26\ \%$$

$$\%sensibilidad = \frac{Aciertos\ en\ la\ prediccion\ positiva}{(Falsos\ positivos+Aciertos)\ en\ la\ prediccion\ positiva} * \ 100\% \qquad \text{ec.2}$$

$$\%sensibilidad = \frac{10}{23+10} * \ 100\%$$

$$\%sensibilidad = 30.3\ \%$$

Tabla 8.4 Accuracy y Sensibilidad en ruta Marianas

Estadística	Valor en %
Accuracy	93.26 %
Sensibilidad	30.3 %

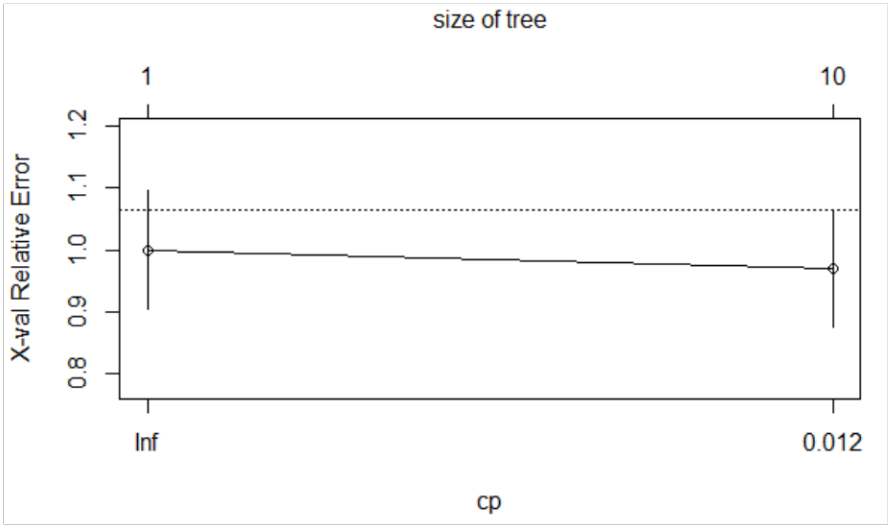


Figura 12.3. Evolución del error en ruta San Juan

En este caso de acuerdo con la figura 12.3, se observa que la evolución del error del árbol disminuye desde que empieza la dimensión del árbol, este decrecimiento mantiene una pendiente a lo largo del intervalo formado del 1 al 10.

Este modelo arroja una evolución del error idónea ya que mantiene su pendiente, por lo que no sería necesario aplicar otra técnica, como por ejemplo podar el árbol para mejorar sus resultados.

3.1.4 Resumen Global de las Tres rutas

En la tabla 9 se resume todos los datos relevantes obtenidos en base a los análisis de las tres rutas

Tabla 9 Resumen de los modelos de las 3 rutas

Características	Zabala-Calderón	Marianas	San Juan
Base de Datos	9061 datos	6978 datos	9849 datos
cid encontrados	190 191	191	190 191 192
Node_id encontrados	28247 28251 28353 28451 28474 28686 28759 28811 28816 28864 28934 10	28474 28759 28816 28759	28001 28474 28611 28783 28811 28816 28845
Base de Datos filtrada	1791 datos	2867 datos	1854 datos
Entrenamiento	1432 datos	2293 datos	1483 datos
Test	359 datos	574 datos	371 datos
Predicciones	80	29	33
Falsos positivos	54	21	23
Predicciones positivas	26	8	10
Accuracy	73.98%	96.34 %	93.26 %
Sensibilidad	32.5 %	27.6 %	30.3 %

Realizando una comparación entre los tres escenarios en que se empleó el modelo de árbol de decisión, se concluye que en base a estadísticas otorgadas el modelo resultó ser más óptimo es el caso de la ruta Zabala-Calderón, ya que otorgó 80 predicciones dentro de un conjunto de prueba de 359 datos, de las cuales 26 fueron predicciones positivas y

54 falsos positivos y aun así se alcanzó la sensibilidad del modelo más alta de los 3 escenarios estudiados, este comportamiento está ligado a que en esta ruta la señal del teléfono móvil tiende a cambiar y conectarse a mas node_id en comparación con las rutas Marianas y San Juan que poseen alrededor de la mitad de la cantidad de node_id.

3.2 Conclusiones

En este trabajo se propone una metodología de datos y análisis para detectar condiciones de handover en regiones limítrofes entre celdas de diferentes estaciones base o de una misma estación base. Se establecieron ciertas condiciones que se producen previo al proceso de ejecución del handover y durante su ejecución. Estos resultados permiten tener datos referenciales para el estudio detallado del proceso de handover.

Si bien uno de los objetivos específicos de este trabajo fue analizar el comportamiento del handover entre radio bases pertenecientes a una misma celda o de diferentes celdas, al momento de hacer la recolección de datos y posterior diseño del programa, se obtuvieron árboles de decisión tan simples que no se prestaban para su análisis, por lo que se optó por trabajar únicamente con la condición de que la señal cambie o no de node_id, lo cual resultó positivo para el estudio.

Las muestras recolectadas fueron tomadas con un intervalo de un segundo de diferencia, sin embargo, se debe mencionar que con la base inicial establecida, los datos no eran adecuados para el modelo, por lo que haciendo énfasis en la parte del componente de interés se decidió tomar una muestra anterior y posterior a las dos donde ocurrió el handover y así aumentar la base de datos para el modelo, el artificio resultó positivo, ya que el modelo pudo aumentar su sensibilidad la cual paso de un intervalo de 5 a 10 por ciento a 27 y 32 por ciento.

Se concluye en base a los datos recolectados, que las tres rutas están dentro del área de cobertura de las celdas 190, 191, y 192, de las cuales solo la 191 está presente en las tres rutas y solo dos de sus node_id se presentan en las tres rutas, los cuales son 28474 y el 28816.

El modelo con mayor accuracy de los tres es el de la ruta Marianas con un valor del 96.34%, es decir, la zona más comercial, mientras que el modelo con mayor sensibilidad, tanto en porcentaje como en aciertos positivos a la predicción, fue el de la Ruta Zabala-Calderón con el 32.5% y 24 predicciones positivas. Sin embargo, el modelo con menor evolución de error en base a la gráfica obtenida es el encontrado en la ruta donde se presenta menor cantidad de población, es decir la ruta San Juan.

En las zonas limítrofes el modelo considera que las variables con mayor impacto, para poder predecir el handover son “node_id”, “rssi_strongest”, “RSSI”, las cuales aparecen en las condiciones de los tres árboles de decisión.

Los tres modelos al tener un accuracy alto permiten concluir que, la técnica utilizada se podría utilizar de referencia para predecir las zonas limítrofes en las cuales exista una baja probabilidad de que se ejecute el proceso de handover. Además, por los datos recolectados y utilizados para la generación de estos modelos es pertinente indicar que estos modelos se podrían aplicar en rutas cortas y donde no existen grandes cantidades de procesos de handover, lo cual se podría probar en un próximo estudio.

3.3 Recomendaciones

Como motivo de estudio en un trabajo futuro, se recomienda aplicar la técnica de “Podar el árbol” con el fin de analizar qué cambios existirían, en comparación con un árbol no podado.

Se recomienda investigar e implementar otra técnica de clasificación de aprendizaje supervisado, tales como “Clasificación de Bayes” o “Ensamble”. y comparar las estadísticas arrojadas entre los modelos obtenidos con las nuevas técnicas y el que se presenta en este documento.

Al momento de escoger la herramienta de recolección de datos para el desarrollo del trabajo, se recomienda asegurarse que esta no posea versiones premium, ya que esto puede influir al momento de querer descargar o acceder a los archivos generados donde se almacenó la información de las muestras.

Si bien la herramienta Rstudio demostró ser muy útil para el desarrollo de este modelo, se recomienda trabajar en el diseño y creación de un programa que simule esta técnica de aprendizaje supervisado en un software de simulación con interfaz más amigable con el estudiante, tales como Matlab o Python.

Si se desea generalizar este modelo, para cualquier ruta se recomienda no trabajar con la variable de entrada node_id ya que está limita al modelo a trabajar netamente con las radio-bases establecidas en dicha ruta.

Se recomienda que antes de escoger un área a estudiar, se tome en cuenta algunos parámetros tales como: posibilidad de acceso, crecimiento poblacional, crecimiento comercial para realizar una comparativa en escenarios diferentes

4 REFERENCIAS BIBLIOGRÁFICAS

[1] F. C. Suarez. Telefonía, teoría y ejemplos prácticos. Jorge Sarmiento Editor, 2020.

[2] M. Silva, D. De la Cruz, A. Amat y M, Freire, “Análisis e implementación de un sistema de radiofrecuencia para mejorar la cobertura de telefonía móvil en la comuna Cerrito de los Morreños”, Buenos Aires, 18 th LACCEI International Multi-Conference for Engineering, Education, and Technology, 29-31 July 2020.

[3] C. D. Radicelli, M. Pomboza y L. Cepeda, Conectividad a Internet en zonas rurales mediante tecnologías de TDT (DVB-RCT2), o telefonía móvil (4G-LTE)” Dyna, vol.85, n.204, pp.319-324. December 20th, 2017.

[4] M. Labib, V. Marojevic y J. Reed. “Analizar y mejorar la resistencia de los sistemas LTE / LTE-A a la suplantación de radiofrecuencia” IEEE Conference on Standards for Communications and Networking (CSCN)

[5] L. V. Villa, “Introducción a Machine Learning Introducción a Machine Learning,” 2018.

[6] J. A. A. Jiménez and M. A. G. Naranjo, “Tema 12: Arboles de decisión,” 2000.

[7] “Netmonitor - Apps en Google Play.” https://play.google.com/store/apps/details?id=com.parizene.netmonitor&hl=es_EC&gl=US (accessed Nov. 17, 2021).

[8] “Obtén más detalles sobre las antenas disponibles con NetMonitor.” https://www.adslzone.net/moviles/obten-mas-detalles-sobre-las-antenas-disponibles-con-netmonitor/ (accessed Nov. 17, 2021).

[9] “GPS Logger - Apps en Google Play.” https://play.google.com/store/apps/details?id=eu.basicairdata.graziano.gpslogger&hl=es_EC&gl=US (accessed Nov. 17, 2021).

[10] “Registrador GPS BasicAirData - Guía de inicio - Datos aéreos básicos.” https://www.basicairdata.eu/projects/android/android-gps-logger/getting-started-guide-for-gps-logger/ (accessed Nov. 17, 2021).

[11] “¿Como tener LTE siempre? 4G Force LTE Only - YouTube.” https://www.youtube.com/watch?v=MHJmOeRLxBc (accessed Nov. 17, 2021).

[12] “CellMapper - Apps en Google Play.” https://play.google.com/store/apps/details?id=cellmapper.net.cellmapper&hl=es_EC&gl=US (accessed Nov. 17, 2021).

[13] “Network Cell Info Lite - Apps en Google Play.” https://play.google.com/store/apps/details?id=com.wilysis.cellinfolite&hl=es_EC&gl=US (accessed Nov. 17, 2021).

[14] “Network Cell Info - Manual.” https://m2catalyst.com/apps/network-cell-info/manual (accessed Nov. 17, 2021).

[15] “Samsung Galaxy A20s | Samsung Latinoamérica.” https://www.samsung.com/latin/smartphones/galaxy-a/galaxy-a20s-blue-32gb-sm-a207mzbdgto/ (accessed Nov. 17, 2021).

[16] “RStudio - RStudio.” https://www.rstudio.com/products/rstudio/ (accessed Jan. 16, 2022).

[17] S. Bayar, “EL MACHINE LEARNING A TRAVÉS DE LOS TIEMPOS, Y LOS APORTES A LA HUMANIDAD,” 2018.

[18] L. V. Villa, “Introducción a Machine Learning Introducción a Machine Learning,” 2018.

[19] J. A. A. Jiménez and M. A. G. Naranjo, “Tema 12: Arboles de decisión,” 2000.

5 ANEXOS

ANEXO I DATOS RECOLECTADOS

ANEXO DIGITAL I. Archivos csv, ruta Calderón

Link para acceder a los archivos:

https://drive.google.com/drive/folders/1k9sb-W9BizoFidg0opc1aKfEXdeiJxhg?usp=sharing

ANEXO DIGITAL II. Archivos csv, ruta Marianas

Link para acceder a los archivos:

https://drive.google.com/drive/folders/1-tFDYjetqtrTK7GJZYiN3luVgJOjMvjg?usp=sharing

ANEXO DIGITAL III. Archivos csv, ruta San Juan

Link para acceder a los archivos:
https://drive.google.com/drive/folders/1XpDtM_2hvUlFj_Z_7QIOgiUloZFFRdLw?usp=sharing

ANEXO II. Código implementado en el desarrollo de este trabajo completo.

```
#ESCUELA POLITECNICA NACIONAL
#TRABAJO DE TITULACION
#
#COMPONENTE 1
#CREACION DE UN MODELO PREDICTIVO DE HANDOVER Y EL ESTUDIO EN LAS ZONAS LIMITROFES
#CON LA TECNICA DE APRENDIZAJE SUPERVISADO DE ARBOL DE DESICIÓN
#Autor: Henry Paul Navarrete Luzuriaga
#Librerias a utilizar: dplyr,rpart, rpart.plot (Librería para el árbol de decisión)



# Importar los archivos csv de los datos tomados

# Ruta Zabala - Calderon: Se importa los archivos de la ruta Calderon
setwd("/Users/henry/Documents/BASE DE DATOS PARA TESIS DE GRADO/RUTA CALDERON")
```

```
dc1<- read.csv("RZC1.csv",sep=";")
dc2<- read.csv("RZC2.csv",sep=";")
dc3<- read.csv("RZC3.csv",sep=";")
dc4<- read.csv("RZC4.csv",sep=";")
dc5<- read.csv("RZC5.csv",sep=";")
dc6<- read.csv("RZC6.csv",sep=";")
dc7<- read.csv("RZC7.csv",sep=";")
dc8<- read.csv("RZC8.csv",sep=";")
dc9<- read.csv("RZC9.csv",sep=";")
dc10<- read.csv("RZC10.csv",sep=";")
dc11<- read.csv("RZC11.csv",sep=";")
library('dplyr')
# Se concatenan los archivos en una sola base de datos
datoscalderoninicial <- rbind(dc3,dc6,dc7,dc8,dc9,dc10)
datosprecalderon <- select(datoscalderoninicial, -c(X))
datoscalderon <- rbind(datosprecalderon,dc11,dc1,dc2,dc4)

# Creación de la columna handover

datoscalderon <- select(datoscalderon,c(node_id, lat, long, lte_neighbors, rssi_strongest, rssi, rsrq, band, cid)) # se selecciona las variables con las que se trabajaran
datoscalderon$handover <- "no"    # se crea una columna llamada handover con todos los espacios en sus filas rellenados con el carácter "no"
datoscalderon[datoscalderon == 2147483647] <- NA  # se identifica las filas que presenten este error determinado por la aplicación cuando existía problemas al tomar la muestra
datoscalderon <- datoscalderon %>%
  na.omit()  # Una vez identificado el error y sustituido por el carácter "NA" se procede a eliminar estas filas de la base de datos"
datoscalderon <- filter(datoscalderon, rssi >= -100)  # filtramos los datos de las filas que posean un rssi >= a -100 dBm

# Se procede a identificar las filas donde existe un handover para su posterior estudio y predicción
j<-length(datoscalderon)
```

```
for (m in 1:nrow(datoscalderon)) {
  if(m>1){
    x <- datoscalderon$node_id[m]
    y <- datoscalderon$node_id[m-1] #Se crean las variables "X" y "Y" para almacenar el valor de la fila donde el valor del node_id es diferente
  }
  else{
    x=1
    y=1 # en el caso de no existir una variación en los valores de node_id "X" y "Y" se les asigna un valor igual a 1
  }
  if (x != y) {
    datoscalderon$handover[m]<-"si"
    datoscalderon$handover[m-1]<-"si"
    datoscalderon$handover[m-2]<-"si"
    datoscalderon$handover[m+1]<-"si"     # si los valores de "X" y "Y" son diferentes se procede a rellenar los espacios de las filas de la columna handover con el carácter "si", que identifica que hubo un handover en esta muestra, adicional se utiliza el artificio de señalar una muestra antes y una muestra despues para el estudio de las zonas limítrofes

  }
}



#Se eliminan las variables sobrantes para el estudio.
totalcalderon<-select(datoscalderon, -c(lat, long, cid)) # se crea la variable del conjunto total
totalcalderon$node_id <- as.character(totalcalderon$node_id)  # se define a los valores de la variable node_id como caracteres

# Creación de los conjuntos de entrenamiento y prueba
create_train_test <- function(data, size =0.8, train=TRUE) {
  n_row = nrow(data)
  total_row = size * n_row
  train_sample <-1: total_row
```

```
  if (train == TRUE) {
    return (data[train_sample, ])
  } else {
    return (data[-train_sample, ])
  }
}
totalcalderon <- totalcalderon[sample(1:nrow(totalcalderon)), ]

traincalderon <- create_train_test(totalcalderon, 0.8, train = TRUE) # Conjunto de entrenamiento equivalente al 80% de la base total
testcalderon <- create_train_test(totalcalderon, 0.8, train = FALSE) # Conjunto de prueba equivalente al 20% de la base total

library(rpart)
library(rpart.plot)
fitcalderon <- rpart(handover ~ ., data=traincalderon,
            method='class',
            minsplit = 1,
            minbucket = 1)  # se indica al modelo la variable que deseamos predecir, el metodo del árbol es de tipo de clasificación

rpart.plot(fitcalderon, extra = 106) # comando para graficar el árbol de decisión
print(fitcalderon)          # comando para entregar en formato texto el árbol de decisión
plotcp(fitcalderon)         # comando para plotear la evolución del error del modelo

predict_unseen <-predict(fitcalderon, testcalderon, type = 'class')
table_matcalderon <- table(testcalderon$handover, predict_unseen)
table_matcalderon # se crea la matriz de incertidumbre para la ruta calderon

accuracy_Testcalderon <- sum(diag(table_matcalderon)) / sum(table_matcalderon)  # fórmula para calcular el accuracy del modelo
print(paste('Accuracy for test', accuracy_Testcalderon))  # comando para imprimir el valor del accuracy del modelo

sensibilidadcalderon <- (table_matcalderon[4]/(table_matcalderon[2]+table_matcalderon[4]))  # fórmula para
```

```
calcular la sensibilidad del modelo
print(paste('Sensibilidad del test', sensibilidadcalderon)) # comando para imprimir el valor de la sensibilidad del modelo

#RUTA SAN JUAN
# Ruta San Juan: Se importa los archivos de la ruta San Juan
setwd("/Users/henry/Documents/BASE DE DATOS PARA TESIS DE GRADO/RUTA SAN JUAN")
dsj1<- read.csv("RSJ1.csv",sep=";")
dsj2<- read.csv("RSJ2.csv",sep=";")
dsj3<- read.csv("RSJ3.csv",sep=";")
dsj4<- read.csv("RSJ4.csv",sep=";")
dsj5<- read.csv("RSJ5.csv",sep=";")
dsj6<- read.csv("RSJ6.csv",sep=";")
dsj7<- read.csv("RSJ7.csv",sep=";")
dsj8<- read.csv("RSJ8.csv",sep=";")

# Se concatenan los archivos en una sola base de datos
datossanjuaninicial <-rbind(dsj3,dsj6,dsj7)
datospresanjuan <- select(datossanjuaninicial, -c(X))
datossanjuan <- rbind(datospresanjuan,dsj1,dsj2,dsj4)

# Creación de la columna handover
library('dplyr')
datossanjuan <- select(datossanjuan,c(node_id, lat, long, lte_neighbors, rssi_strongest, rssi, rsrq, band, cid)) # se selecciona las variables con las que se trabajaran
datossanjuan$handover <- "no"    # se crea una columna llamada handover con todos los espacios en sus filas rellenados con el carácter "no"
datossanjuan[datossanjuan == 2147483647] <- NA  # se identifica las filas que presenten este error determinado por la aplicación cuando existía problemas al tomar la muestra
datossanjuan <- datossanjuan %>%
  na.omit()  # Una vez identificado el error y sustituido por el carácter "NA" se procede a eliminar estas filas de la base de datos"
datossanjuan <- filter(datossanjuan, rssi >= -100)  # filtramos los datos de las filas que posean un rssi >= a -100 dBm
```

```
# Se procede a identificar las filas donde existe un handover para su posterior estudio y predicción
j<-length(datossanjuan)
for (m in 1:nrow(datossanjuan)) {
  if(m>1){
    x <- datossanjuan$node_id[m]
    y <- datossanjuan$node_id[m-1] #Se crean las variables "X" y "Y" para almacenar el valor de la fila donde el valor del node_id es diferente
  }
  else{
    x=1
    y=1 # en el caso de no existir una variación en los valores de node_id "X" y "Y" se les asigna un valor igual a 1
  }
  if (x != y) {
    datossanjuan$handover[m]<-"si"
    datossanjuan$handover[m-1]<-"si"
    datossanjuan$handover[m-2]<-"si"
    datossanjuan$handover[m+1]<-"si"     # si los valores de "X" y "Y" son diferentes se procede a rellenar los espacios de las filas de la columna handover con el caracter "si", que identifica que hubo un handover en esta muestra, adicional se utiliza el artificio de señalar una muestra antes y una muestra despues para el estudio de las zonas limítrofes

  }
}



#Se eliminan las variables sobrantes para el estudio.
totalsanjuan<-select(datossanjuan, -c(lat, long, cid)) # se crea la variable del conjunto total
totalsanjuan$node_id <- as.character(totalsanjuan$node_id)  # se define a los valores de la variable node_id como caracteres

# Creación de los conjuntos de entrenamiento y prueba
```

```
create_train_test <- function(data, size =0.8, train=TRUE) {
  n_row = nrow(data)
  total_row = size * n_row
  train_sample <-1: total_row
  if (train == TRUE) {
    return (data[train_sample, ])
  } else {
    return (data[-train_sample, ])
  }
}
totalsanjuan <- totalsanjuan[sample(1:nrow(totalsanjuan)), ]
trainsanjuan <- create_train_test(totalsanjuan, 0.8, train = TRUE) # Conjunto de entrenamiento equivalente al 80% de la base total
testsanjuan <- create_train_test(totalsanjuan, 0.8, train = FALSE) # Conjunto de prueba equivalente al 20% de la base total

fitsanjuan <- rpart(handover ~ ., data=trainsanjuan,
                    method='class',
                    minsplit = 1,
                    minbucket = 1)  # se indica al modelo la variable que deseamos predecir, el metodo del árbol es de tipo de clasificación

rpart.plot(fitsanjuan, extra = 106) # comando para graficar el árbol de decisión
print(fitsanjuan)          # comando para entregar en formato texto el árbol de decisión
plotcp(fitsanjuan)         # comando para plotear la evolución del error del modelo

predict_unseen <-predict(fitsanjuan, testsanjuan, type = 'class')
table_matsanjuan <- table(testsanjuan$handover, predict_unseen)
table_matsanjuan # se crea la matriz de incertidumbre para la ruta san juan

accuracy_Testsanjuan <- sum(diag(table_matsanjuan)) / sum(table_matsanjuan)  # fórmula para calcular el accuracy del modelo
print(paste('Accuracy del test', accuracy_Testsanjuan))  # comando para imprimir el valor del accuracy del modelo

sensibilidadsanjuan                                                    <-
```

```
(table_matsanjuan[4]/(table_matsanjuan[2]+table_matsanjuan[4])) # fórmula para calcular la sensibilidad del modelo
print(paste('Sensibilidad del test', sensibilidadsanjuan)) # comando para imprimir el valor de la sensibilidad del modelo



# RUTA MARIANAS
# Ruta Marianas: Se importa los archivos de la ruta Marianas
setwd("/Users/henry/Documents/BASE DE DATOS PARA TESIS DE GRADO/RUTA MARIANAS")
dm1<- read.csv("RMJ1.csv",sep=";")
dm2<- read.csv("RMJ2.csv",sep=";")
dm3<- read.csv("RMJ3.csv",sep=";")
dm4<- read.csv("RMJ4.csv",sep=";")
dm5<- read.csv("RMJ5.csv",sep=";")
dm6<- read.csv("RMJ6.csv",sep=";")
dm7<- read.csv("RMJ7.csv",sep=";")

# Se concatenan los archivos en una sola base de datos
datosmarianasinicial <-rbind(dm1,dm3,dm4,dm5,dm6,dm7)
datospremarianas <- select(datosmarianasinicial, -c(X))
datosmarianas <- rbind(datospremarianas,dm2)

# Creación de la columna handover
datosmarianas <- select(datosmarianas,c(node_id, lat, long, lte_neighbors, rssi_strongest, rssi, rsrq, band, cid)) # se selecciona las variables con las que se trabajaran
datosmarianas$handover <- "no"    # se crea una columna llamada handover con todos los espacios en sus filas rellenados con el carácter "no"
datosmarianas[datosmarianas == 2147483647] <- NA  # se identifica las filas que presenten este error determinado por la aplicación cuando existía problemas al tomar la muestra
datosmarianas <- datosmarianas %>%
  na.omit()  # Una vez identificado el error y sustituido por el carácter "NA" se procede a eliminar estas filas de la base de datos"
datosmarianas <- filter(datosmarianas, rssi >= -100)  # filtramos los datos de las filas
```

```
que posean un rssi >= a -100 dBm

# Se procede a identificar las filas donde existe un handover para su posterior estudio y predicción
j<-length(datosmarianas)
for (m in 1:nrow(datosmarianas)) {
  if(m>1){
    x <- datosmarianas$node_id[m]
    y <- datosmarianas$node_id[m-1] #Se crean las variables "X" y "Y" para almacenar el valor de la fila donde el valor del node_id es diferente
  }
  else{
    x=1
    y=1 # en el caso de no existir una variación en los valores de node_id "X" y "Y" se les asigna un valor igual a 1
  }
  if (x != y) {
    datosmarianas$handover[m]<-"si"
    datosmarianas$handover[m-1]<-"si"
    datosmarianas$handover[m-2]<-"si"
    datosmarianas$handover[m+1]<-"si"    # si los valores de "X" y "Y" son diferentes se procede a rellenar los espacios de las filas de la columna handover con el carácter "si", que identifica que hubo un handover en esta muestra, adicional se utiliza el artificio de señalar una muestra antes y una muestra despues para el estudio de las zonas limítrofes

  }
}



#Se eliminan las variables sobrantes para el estudio.
totalmarianas<-select(datosmarianas, -c(lat, long, cid)) # se crea la variable del conjunto total
totalmarianas$node_id <- as.character(totalmarianas$node_id)  # se define a los valores de la variable node_id como caracteres
```

```
# Creación de los conjuntos de entrenamiento y prueba
create_train_test <- function(data, size =0.8, train=TRUE) {
  n_row = nrow(data)
  total_row = size * n_row
  train_sample <-1: total_row
  if (train == TRUE) {
    return (data[train_sample, ])
  } else {
    return (data[-train_sample, ])
  }
}
totalmarianas <- totalmarianas[sample(1:nrow(totalmarianas)), ]

trainmarianas <- create_train_test(totalmarianas, 0.8, train = TRUE) # Conjunto de entrenamiento equivalente al 80% de la base total
testmarianas <- create_train_test(totalmarianas, 0.8, train = FALSE) # Conjunto de prueba equivalente al 20% de la base total

fitmarianas <- rpart(handover ~ ., data=trainmarianas,
                     method='class',
                     minsplit = 1,
                     minbucket = 1)  # se indica al modelo la variable que deseamos predecir, el metodo del árbol es de tipo de clasificación

rpart.plot(fitmarianas, extra = 106) # comando para graficar el árbol de decisión
print(fitmarianas)          # comando para entregar en formato texto el árbol de decisión
plotcp(fitmarianas)          # comando para plotear la evolución del error del modelo

predict_unseen <-predict(fitmarianas, testmarianas, type = 'class')
table_matmarianas <- table(testmarianas$handover, predict_unseen)
table_matmarianas # se crea la matriz de incertidumbre para la ruta marianas

accuracy_Testmarianas <- sum(diag(table_matmarianas)) / sum(table_matmarianas)  # fórmula para calcular el accuracy del modelo
print(paste('Accuracy del test', accuracy_Testmarianas))  # comando para imprimir el valor del accuracy del modelo
```

```
sensibilidadmarianas <- (table_matmarianas[4]/(table_matmarianas[2]+table_matmarianas[4])) # fórmula para calcular la sensibilidad del modelo
print(paste('Sensibilidad del test', sensibilidadmarianas)) # comando para imprimir el valor de la sensibilidad del modelo
```