

# **Reto Datos**

David Guzmán Leyva - A01706417  
Enrique Santos Fraire - A01705746  
Leonardo Alvarado Menéndez - A01705998  
Oscar Enrique Delgadillo Ochoa - A01705935

# Índice

Índice	2
Bitácora de cambios	3
Herramientas y tecnologías	3
Modelo de almacenamiento	4
Ajuste de datos	4
Muestreo	4
Cross validation	4
Enfoque Big Data	5

# Bitácora de cambios

Versión	Fecha	Autor(es)	Modificaciones
1.0	19/10/2022	David Guzmán Leyva Enrique Santos Fraire Leonardo Alvarado Menéndez Oscar Enrique Delgadillo Ochoa	Línea base
1.1	28/11/2022	David Guzmán Leyva Enrique Santos Fraire Leonardo Alvarado Menéndez Oscar Enrique Delgadillo Ochoa	Ajuste al modelo de almacenamiento, cross validation y enfoque

## Herramientas y tecnologías

La herramienta primaria que se va a utilizar para llevar a cabo el modelado de minería de datos será el framework de Apache Spark, asumiendo que nuestros datos son Big Data. La elección de este fue por sus diversas cualidades, ya que dicho framework tiene las siguientes ventajas:

- Rapidez: Ejecuta las cargas de trabajo 100 veces más rápido que con Hadoop MapReduce. Con Spark, se tiene un alto rendimiento con los datos por lotes y de streaming gracias al programador de grafos acíclicos dirigidos de última generación, al optimizador de consultas y al motor físico de ejecución.
- Manejo de datos: Spark puede hacer uso de Resilient Distributed Datasets (RDD).
- Facilidad de uso: Spark cuenta con más de 80 operadores generales que facilitan el desarrollo de aplicaciones en paralelo. Puede ser utilizado de forma interactiva desde el shell de Scala, Python, R y SQL para escribir aplicaciones rápidamente.
- Uso general: Spark permite usar una pila de bibliotecas que incluye SQL, DataFrame, MLlib y Spark ML para aprendizaje automático, GraphX y Spark Streaming.
- Framework de código abierto: Además de ser gratis, cuenta con potencial colectivo para aportar más ideas, desarrollarlas más rápido y solucionar los problemas en cuanto aparecen.

Asimismo emplearemos algunas librerías de python necesarias para realizar ETL:

- Pandas: Sirve para la manipulación y análisis de datos, además ofrece estructuras de datos y operaciones para manipular tablas numéricas y series temporales.
- Numpy: Para realizar cálculos u operaciones con los datos.
- Geopy: Será utilizado para la obtención de comunas.

Entornos de ejecución del proyecto:

- Google Colab: Permite a cualquier usuario escribir y ejecutar código arbitrario de Python en el navegador. Es especialmente adecuado para tareas de aprendizaje automático, análisis de datos y educación.

- Jupyter notebook: Es un entorno de desarrollo interactivo basado en la web para cuadernos, códigos y datos.
- Anaconda: Es una distribución libre y abierta de los lenguajes Python y R, utilizada en ciencia de datos, y aprendizaje automático. En ella podemos crear Jupyter notebooks.


Manejo de versiones del proyecto:

- Github: es una forja para alojar proyectos utilizando el sistema de control de versiones Git.
- Link de repositorio: <https://github.com/Davidguzley/Movilidad-Chile>

Administración del proyecto:

- Google Drive: Es un servicio de alojamiento de archivos en el cual realizamos planes de trabajo y documentación del proyecto de manera colaborativa

## Modelo de almacenamiento

 Data Understanding V2

## Ajuste de datos

 Data Preparation V2

## Muestreo

- [Repositorio](#)
- [Acceso a drive](#)

## Cross validation

Para realizar validación cruzada utilizaremos CrossValidator de Spark ML, esta validación cruzada de K-fold realiza la selección del modelo al dividir el conjunto de datos en un conjunto de pliegues particionados aleatoriamente que no se superponen y que se utilizan como conjuntos de datos de entrenamiento y prueba separados.

Por ejemplo, con  $k = 3$  pliegues, la validación cruzada de K-fold generará 3 (entrenamiento, prueba) pares de conjuntos de datos, cada uno de los cuales usa  $2/3$  de los datos para entrenamiento y  $1/3$  para prueba. Cada pliegue se utiliza como conjunto de prueba exactamente una vez.

Cabe mencionar que la ventaja de utilizar validación cruzada con Spark ML es que todas las validaciones se realizan de manera más rápida, ya que se pueden correr con paralelismo, definiendo un número de threads.

# Enfoque Big Data

- Debido a la cantidad de datos con los que contamos, y que el número de datos no aumentará a lo largo del reto, el dataset puede ser considerado como normal data, ya que no cuenta con ninguna de las 5 características (volumen, velocidad, variedad, veracidad y valor).
- En caso de que nuestro dataset contara con más días, realizaremos un modelo de regresión de tiempo y tendríamos una cantidad mayor de datos, por lo que cumpliría con la característica de volumen.
- Otro caso sería tomar los registros de forma continua, por lo que obtenemos datos a una gran velocidad, ya que recibimos datos de varios usuarios por segundo.
- Sin embargo nuestra normal data cuenta con un número de registros un poco alto y por cuestiones de tiempo, debemos extraer, leer, procesar y obtener datos nuevos de la manera más rápida posible, esto a raíz que actualmente nuestros equipos de cómputo cuentan con entornos de ejecución un poco cortos para lograr sacar el proyecto adelante en el tiempo acordado. Por lo que haremos uso de Apache Spark para solucionar este problema, nos será de gran ayuda principalmente con sus modelos para generación de predicciones, ya que es un framework de computación en clúster capaz de dividir y ejecutar las tareas de los algoritmos de aprendizaje automático de manera eficiente gracias a su MapReduce.