



DOI: <http://dx.doi.org/10.23857/dc.v8i3>

Ciencias Técnicas y Aplicadas
Artículo de Investigación

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil utilizando las cadenas de Márkov con Python

Logistics analysis of the loading and unloading points of a transport company in the city of Guayaquil using Markov chains with Python

Análise logística dos pontos de carga e descarga de uma empresa de transporte na cidade de Guayaquil usando cadeias de Markov com Python

Juan Carlos Yturralde-Villagómez ^I
juan.yturralde@ug.edu.ec
<https://orcid.org/0000-0001-9948-177X>

Renzo Rogelio Padilla-Gómez ^{II}
renzo.padillag@ug.edu.ec
<https://orcid.org/0000-0003-4301-1335>

Correspondencia: juan.yturralde@ug.edu.ec

***Recibido:** 29 de junio del 2022 ***Aceptado:** 12 de julio de 2022 * **Publicado:** 10 de agosto de 2022

- I. Universidad UNIR, Universidad de Guayaquil, Guayaquil, Ecuador.
- II. Universidad UNIR, Universidad de Guayaquil, Guayaquil, Ecuador.

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil utilizando las cadenas de Márkov con Python

Resumen

Existen muchas compañías de transporte pesado que se dedican a la transportación desde el puerto marítimo de la ciudad de Guayaquil hasta diferentes lugares dentro y fuera ella. Este artículo se enfocó en hacer un análisis logístico de la ruta que recorre una unidad de transporte de una compañía dedicada al traslado de trigo y pasta de soya para abastecer a la industria de alimentos balanceados para camarones. Una de las problemáticas que enfrenta este tipo de compañías es que generalmente desconocen el comportamiento de las rutas de transporte de la materia prima. Por esa razón, se examinó la información de un conjunto de datos correspondientes a las rutas realizadas por la unidad de transporte pesado correspondiente al periodo del 4 al 31 de octubre del año 2021, utilizando las cadenas de Markov.

La metodología utilizada para realizar el análisis logístico de las rutas establecidas por la unidad de transporte fue CRISP-DM, ya que se estableció el entendimiento del negocio, la representación de los puntos geográficos utilizando Python, la exploración de los datos, la construcción de la matriz de transición de probabilidades, la verificación de la distribución estacionaria y por último el análisis de los resultados.

El resultado del análisis fue de mucha ayuda para la toma de decisiones de la compañía con respecto a la logística del transporte ya que se detectó que tres destinos, de nueve en total, cubren el 75% de visitas de la unidad de transporte. El análisis se efectuó por medio de la matriz de transición de probabilidad con distribución estacionaria, y con la ayuda de un script de Python desarrollado se calculó la probabilidad de ocurrencia de una ruta determinada.

Palabras clave: Análisis logístico; matriz de transición; distribución estacionaria y cadena de Márkov.

Abstract

There are many heavy transport companies that are dedicated to the transportation from the seaport of the city of Guayaquil to different places inside and outside it. This research focused on a logistical analysis of the route that a transport unit of a company dedicated to the transportation of wheat and soybean paste to supply the shrimp feed industry. One of the problems faced by this type of company is that they are generally unaware of the behavior of the raw material transportation routes. For this reason, the information of a set of data corresponding to the routes performed by the heavy transport unit corresponding to the period from October 4 to October 31, 2021, was examined using Markov chains.

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil utilizando las cadenas de Márkov con Python

The methodology used to perform the logistic analysis of the routes established by the transportation unit was CRISP-DM, since it established the understanding of the business, the representation of the geographical points using Python, the exploration of the data, the construction of the probability transition matrix, the verification of the stationary distribution and finally the analysis of the results.

The result of the analysis was very helpful for the company's decision making with respect to transportation logistics since it was detected that three destinations, out of nine in total, cover 75% of the visits of the transportation unit. The analysis was performed by means of the probability transition matrix with stationary distribution, and with the help of a developed Python script the probability of occurrence of a given route was calculated.

Keywords: Logistic analysis, transition matrix, stationary distribution and Markov chain.

Resumo

Existem muitas empresas de transporte pesado que se dedicam ao transporte do porto marítimo da cidade de Guayaquil para diferentes lugares dentro e fora dela. Este artigo teve como foco fazer uma análise logística da rota percorrida por uma unidade de transporte de uma empresa dedicada ao transporte de pasta de trigo e soja para abastecer a indústria de rações para camarões. Um dos problemas enfrentados por esse tipo de empresa é que geralmente desconhecem o comportamento das rotas de transporte da matéria-prima. Para isso, foi examinada a informação de um conjunto de dados correspondente às rotas realizadas pela unidade de transporte pesado correspondente ao período de 4 a 31 de outubro de 2021, utilizando cadeias de Markov. A metodologia utilizada para realizar a análise logística das rotas estabelecidas pela unidade de transporte foi a CRISP-DM, uma vez que se estabeleceu o entendimento do negócio, a representação dos pontos geográficos utilizando Python, a exploração dos dados, a construção do matriz de transição de probabilidade, a verificação da distribuição estacionária e por fim a análise dos resultados. O resultado da análise foi muito útil para a tomada de decisão da empresa em relação à logística de transporte, pois detectou-se que três destinos, do total de nove, abrangem 75% das visitas da unidade de transporte. A análise foi realizada por meio da matriz de transição de probabilidade com distribuição estacionária e, com

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil utilizando las cadenas de Márkov con Python

o auxílio de um script desenvolvido em Python, foi calculada a probabilidade de ocorrência de uma determinada rota.

Palavras-chave: Análise logística; matriz de transição; distribuição estacionária e cadeia de Markov.

Introducción

En términos de toma de decisiones de una compañía, la cadena de Markov ha sido muy útil en la solución de problema en los cuales, ayudado con la predicción del consumo de ciertos productos, el estado del clima de una determinada ciudad, los valores de la bolsa de valores de un determinado mercado financiero, entre otras situaciones o aplicaciones de la vida real. En este apartado, se busca observar algunos artículos importantes en las tomas de decisiones utilizando la cadena de Markov, enfocados a la transportación y los negocios.

En 1988, se desarrolló un modelo de predicción para el rendimiento de puentes de hormigón y de aceros en el Departamento de Carretera de Indiana (Jiang et al., 1988). El modelo predecía las calificaciones de los puentes según sus condiciones y su curva de rendimiento, para lo cual se utilizó las cadenas de Márkov basada en la probabilidad estocásticas de las condiciones del puente. El procedimiento, aunque parece sencillo, permitió un alto nivel de precisión en la predicción de las condiciones del puente.

En el artículo “Un modelo de cadena de Markov del cambio en las redes de transporte y el uso del suelo” (Levinson & Chen, 2005) se analiza la evolución de las vías de transportes y el uso del suelo por medio de una matriz de transición que registra la interacción entre el transporte y el uso de suelo para predecir el futuro del crecimiento de las zonas del sector. El estudio se basa en la red de carreteras y el uso del suelo en Twin Cities Metropolitan en el periodo de 1958 a 1990. A través de las cadenas de Márkov se concluye que las carreteras afectan en el desarrollo de las zonas agrícolas y recreativas. Estas zonas con carreteras tienen muchas más probabilidades de convertirse en zonas de empleo y residenciales que si no hubiera carretera. Mientras que las zonas urbanizadas están menos influenciadas por carreteras que las zonas de empleo ya que tienen menos probabilidades de cambiar sus atributos de uso de la tierra. En cambio, las zonas residenciales es más probable que

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil utilizando las cadenas de Márkov con Python

cambien a zonas de uso mixto (residencial y empleo) si contienen carreteras, es lo que concluye este estudio.

En el artículo uso de la cadena de Márkov para un modelo de negocio (Viñamagua, 2017) se puede apreciar claramente el análisis de tres productos de sábila de la industria “Talea” en el cantón Loja. Con la ayuda de la cadena de Márkov y el algoritmo ALOA-MARKOV, se puede predecir la distribución de los compradores de productos de sábila en un determinado mes. Este análisis puede ser de gran importancia para planificación de la producción de la industria “Talea” y su logística de distribución en el cantón Loja, pues se estudia el comportamiento de la compra de dichos productos. El desarrollo de un modelo enfocado en la cadena de Markov permite un breve periodo de entrenamiento de las técnicas de modelado y permite una fácil implementación del módulo de propósito de viaje. (Cory M. Krause, 2018). Al incluir el propósito del viaje en el enfoque del modelo Markov permite generar una precisión de la predicción del destino.

En el Ecuador, con la suscripción del Decreto 1024, se estableció una banda de precio (USD/galón) para el diésel, lo que ha ocasionado un aumento del 75.92% por galón. Esta alza también repercute en los accesorios y repuestos para la operación del transporte pesado.

En ese sentido, debido al excesivo incremento del diésel, uso casi exclusivo de la transportación pesada, es fundamental plantear un análisis de los registros de la posición geográfica y los viajes realizados, a través de la plataforma del sistema de rastreo, utilizando el modelo de Márkov. De esta manera, obtendremos información relevante para que la Gerencia de la compañía tome decisiones para controlar su flota de transporte de carga.

Los resultados del presente trabajo se fundamentan en las cadenas de Márkov con series de tiempo discreto donde, por medio de las probabilidades de transición, será posible encontrar información determinante sobre el desplazamiento diario que efectúa una unidad de transporte de carga en la ciudad de Guayaquil y Durán.

Metodología

La metodología de trabajo es la Cross-Industry Standard Process for Data Mining (CRISP-DM, por sus siglas en inglés). Es un método probado para los trabajos de minería de datos que incluyen de 6 fases que se describen en la figura 1:

Elaborado por: (IBM, 2021)

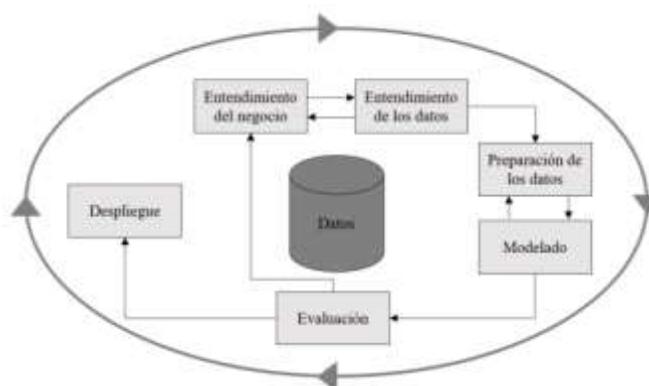


Figura 1: Metodología CRISP-DM

El ciclo vital del modelo está compuesto de seis fases mediante el uso de flechas que indican los lugares más importantes y frecuentes entre fases. La secuencia de las fases no es necesariamente estricta. De hecho, la mayor cantidad de proyectos avanzan y retroceden entre fases de ser necesario.

El modelo de CRISP-DM es flexible y se puede personalizar fácilmente. Por ejemplo, si una organización intenta detectar actividades de lavado de dinero, es probable que necesite realizar una criba de grandes cantidades de datos sin un objetivo de modelado específico. En lugar de realizar el modelado, el trabajo se centrará en explorar y visualizar datos para descubrir patrones sospechosos en datos financieros. CRISP-DM permite crear un modelo de minería de datos que se adapte a las necesidades concretas del usuario.

En tal situación, las fases de modelado, evaluación y despliegue pueden ser menos relevantes que las fases de preparación y comprensión de datos. Sin embargo, es muy fundamental considerar

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil utilizando las cadenas de Márkov con Python

algunas cuestiones que surgen durante fases posteriores para la planificación a largo plazo y objetivos futuros de minería de datos.

Las fases del trabajo de fin de máster se ajustan a la metodología CRISP-DM, ya que se conforma de seis etapas: entendimiento del negocio, representación de los datos, exploración de los datos, generación de la matriz de transición, determinación de la distribución estacionaria de la matriz de transición y análisis de resultados como se puede apreciar en la figura 2.

Elaborado por: Elaboración propia



Figura 2: Metodología del Trabajo

Identificación de requisitos

1. Los requisitos necesarios para culminar con éxito el desarrollo del análisis logístico de las rutas de carga y descarga de una unidad de transporte serían los siguientes:
2. Depurar correctamente el dataset correspondiente a la información de la ubicación geográfica y la ruta en el periodo de estudio.
3. Determinar correctamente cuando la plataforma Gurtam establece que desplazamiento con una ruta.
4. Explorar adecuadamente los datos para detectar el comportamiento del patrón de los datos en análisis para cubrir todos los destinos de la unidad en su ruta de trabajos.

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil utilizando las cadenas de Márkov con Python

5. Detectar que la matriz de transición de probabilidades del número de visitas de todos los destinos de la unidad tenga una distribución estacionaria.

Resultados

El análisis realizado fue a un conjunto de datos que corresponden a las ubicaciones geográficas en el periodo del 4 al 31 de octubre del 2021, de una unidad de transporte de una compañía dedicada a transportar productos al granel en la ciudad de Guayaquil, utilizando las cadenas de Márkov.

El análisis de la información se dividirá en seis etapas: entendimiento del negocio, representación de los datos, exploración de los datos, generación de la matriz de transición de probabilidades, determinar si la matriz de transición es o no una distribución estacionaria y análisis de los resultados. Se realizó una investigación previa sobre el modelo de negocio de la compañía de transportes en la cual se indica que los puntos de carga correspondían a los nodos A, B y D. No se colocan los nombres de los lugares para respetar la privacidad de la información de la compañía de la que se hace el análisis. Los restantes 6 nodos en estudio corresponden a los puntos de descarga.

El conjunto de datos de la ubicación geográfica corresponde al periodo previamente mencionado, de la unidad de transporte en análisis, para la representación y exploración de los datos. De esta manera, se detectó el patrón de comportamiento de la data en análisis, para lo cual se utilizó como lenguaje de programación Python y como entorno de desarrollo integrado (IDE) spyder y jupyter notebook.

La matriz de transición se generó previa exploración visual de la representación, en dos dimensiones, de la información generada por la unidad de transporte en sus recorridos. A partir de esto, se obtuvo un patrón de comportamiento que abarca, durante un periodo determinado, todos los nodos en análisis. Luego, se realizó un cálculo sencillo de multiplicar 100 veces a la matriz de transición por sí mismo y se encontró que la matriz de probabilidad de transición de la unidad de análisis tenía una distribución estacionaria. Además, se corroboró la información realizando cálculos de manera analítica, a través de la librería numpy de Python.

Como aporte para el análisis se desarrolló un script que calcula la probabilidad de ocurrencia de una ruta determinada.

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil utilizando las cadenas de Márkov con Python

Entrada de datos

La información de la unidad de transporte se obtiene desde una plataforma web llamada GURTAM, utilizada para el monitoreo en tiempo real de unidades de transporte liviano y pesado. Es importante indicar que la ruta del vehículo pesado analizada consta de un equipo instalado llamado gps tracker de la marca teltonika, modelo FMB 920, que genera su ubicación cada 30 segundos cuando está en movimiento y 10 minutos cuando se encuentra estacionado o parado a una velocidad de 0 Km/h.

La descarga del dataset se realiza desde la siguiente página web: <https://hosting.wialon.us/> . La información es accesible únicamente para los usuarios registrados. Es decir, la misma no es de acceso público.

Una vez que se ingresa a la plataforma, se genera un informe para obtener las ubicaciones geográficas en el periodo establecido o personalizado de máximo tres meses. En la gráfica 3 y 4 podemos verificar cómo se accede a la información analizada.

Fuente: (GURTAM, 2021)

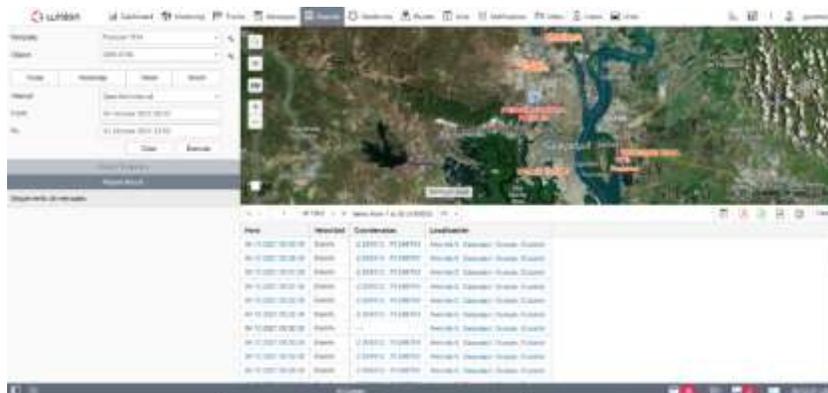


Figura 3: Informe de la unidad de transporte en Gurtam

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil
 utilizando las cadenas de Márkov con Python

Fuente: (GURTAM, 2021)

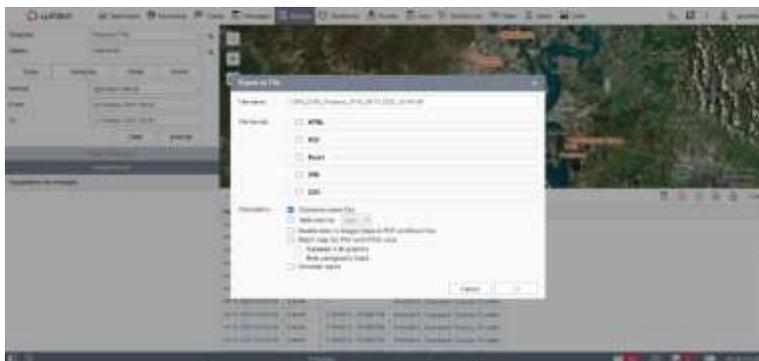


Figura 4: Exportación del dataset

En la figura 3 se puede visualizar que una vez que se da clic izquierdo en el ícono que se encuentra del lado izquierdo del símbolo de la impresión, se procede a exportar el archivo generado en el periodo establecido de la figura 4 en los siguientes formatos: HTML, PDF, EXCEL, XML Y CSV. Para la presente investigación se descargó un archivo con formato EXCEL. En la tabla 2 se puede observar los primeros 10 datos de un total de 95.096 disponibles del periodo de tiempo analizado. Los registros de las ubicaciones geográficas son enviados en grados decimales cada 30 segundos cuando el vehículo se encuentra en movimiento, y cada 10 minutos cuando el vehículo se encuentra estacionado. Es importante indicar que el gps tracker registra al vehículo como estacionado o parado cuando reporta una velocidad de 0 km/h por un lapso de 300 segundos.

Figura 5: Formato original del Dataset en Excel.

Hora	Velocidad	Coordenadas	Localización
04.10.2021 00:00:08	0 km/h	-2.264813, -79.896764	Avenida 5, Guayaquil, Guayas, Ecuador
04.10.2021 00:00:38	0 km/h	-2.264813, -79.896764	Avenida 5, Guayaquil, Guayas, Ecuador
04.10.2021 00:01:08	0 km/h	-2.264813, -79.896764	Avenida 5, Guayaquil, Guayas, Ecuador
04.10.2021 00:01:38	0 km/h	-2.264813, -79.896764	Avenida 5, Guayaquil, Guayas, Ecuador
04.10.2021 00:02:08	0 km/h	-2.264813, -79.896764	Avenida 5, Guayaquil, Guayas, Ecuador
04.10.2021 00:02:38	0 km/h	-2.264813, -79.896764	Avenida 5, Guayaquil, Guayas, Ecuador
04.10.2021 00:02:52	0 km/h	----	Avenida 5, Guayaquil, Guayas, Ecuador
04.10.2021 00:03:08	0 km/h	-2.264813, -79.896764	Avenida 5, Guayaquil, Guayas, Ecuador
04.10.2021 00:03:38	0 km/h	-2.264813, -79.896764	Avenida 5, Guayaquil, Guayas, Ecuador
04.10.2021 00:04:08	0 km/h	-2.264813, -79.896764	Avenida 5, Guayaquil, Guayas, Ecuador

Representación y Exploración de la Data

Una vez descargado el dataset se tuvo que dividir la columna de coordenadas para separar la longitud y latitud, para poder convertir las coordenadas geográficas decimales a sistemas de coordenadas geográficas UTM y lograr la representación gráfica de la latitud en el eje de la “x” y la longitud en el eje de la “y” en dos dimensiones.

A continuación, en la tabla 3, se podrá observar el archivo en Excel de los 10 primeros valores de un total de 17.593, correspondientes al periodo del 11 al 17 de octubre del 2021, donde se convirtió las coordenadas geográficas decimales a UTM.

Figura 6: Formato transformado del Dataset

Fecha	Fecha	Velocidad	Latitud (Decimales)	Longitud (Decimal)	Latitud(UTM)	Longitud(UTM)
11.10.2021	7:11:31	13	-2.264613	-79.896871	622663.0161	9749644.249
11.10.2021	7:11:35	16	-2.264547	-79.897022	622646.229	9749651.558
11.10.2021	7:11:38	15	-2.26459	-79.89712	622635.3269	9749646.813
11.10.2021	7:11:59	15	-2.264851	-79.897511	622591.8221	9749617.992
11.10.2021	7:14:20	16	-2.26545	-79.898267	622507.6975	9749551.835
11.10.2021	7:14:26	15	-2.265649	-79.898427	622489.8874	9749529.848
11.10.2021	8:25:22	1	-2.26524	-79.898347	622498.8185	9749575.058
11.10.2021	9:00:52	9	-2.264903	-79.897378	622606.6085	9749612.232
11.10.2021	9:01:22	12	-2.264858	-79.897947	622543.3343	9749617.255
11.10.2021	9:01:53	10	-2.265663	-79.898427	622489.8862	9749528.301

Fuente: (GURTAM, 2021)

En primera instancia se realizó un análisis diario del comportamiento de la unidad de transporte durante el periodo que comprende del 11 al 17 de octubre del 2021. Para el efecto, se graficaron las coordenadas geográficas de la latitud en el eje de las ‘x’ y la longitud en el eje de la ‘y’ en el sistema de coordenadas UTM, utilizando Python como herramienta de análisis de la información.

Así también, se generó el siguiente código para graficar las ubicaciones geográficas:

```
import pandas as pd
import matplotlib.pyplot as plt

df2 = pd.read_excel('dataset.xlsx')
```

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil utilizando las cadenas de Márkov con Python

```
#datos=17593  
plt.scatter(df2['Latitud(UTM)'],df2['Longitud(UTM)'],color='g')  
plt.title('Ubicación Geográfica UTM del GBN 6785 11-17OCT21')  
plt.xlabel('Latitud')  
plt.ylabel('Longitud')  
plt.show()
```

El citado código, que corresponde a los datos de los 17.953 puntos, correspondiente a las ubicaciones geográficas de la unidad de transporte pesado durante el periodo del 11 al 17 de octubre del 2021, fue graficado con el entorno de desarrollo spyder de Phython, según se observa en la figura 5.

Elaborado por: Elaboración propia



Figura 7: Representación de los datos de la Unidad (11 AL 17 OCT 21)

El análisis desprende que el patrón de comportamiento de la unidad de transporte es el mismo si se lo visualiza desde la herramienta web Google Earth Pro o cualquiera otra herramienta de visualización de posiciones geográficas. Por supuesto, esta acción es posible realizarla con Python, con un mejor rendimiento y ventajas comparativas. En ese sentido, al utilizar Python se obtiene el mismo resultado con un menor costo computacional ya que si graficamos los 17.593 puntos geográficos, siempre será mucho más rápido que si se ejecuta la misma acción en la herramienta de visualización de Google Earth Pro.

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil utilizando las cadenas de Márkov con Python

En tal virtud, se procedió a analizar el comportamiento del patrón de la unidad de transporte, diariamente, desde el lunes 11 al domingo 17 de octubre del 2021, de lo cual se obtuvo los resultados que se muestran en la figura 6.

Elaborado por: Elaboración propia

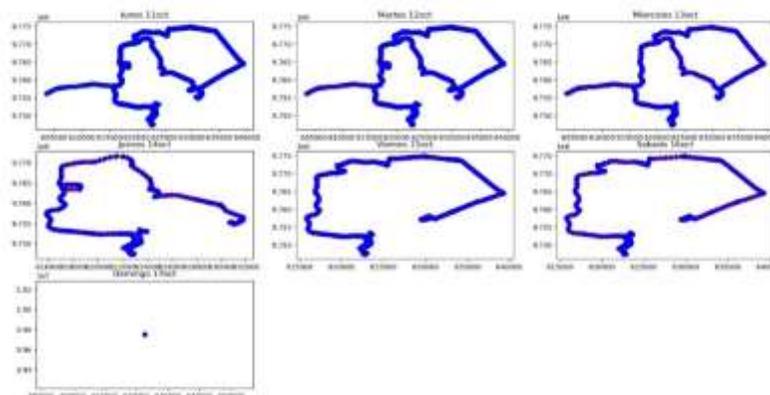


Figura 8: Posición geográfica de la unidad de transporte del 11 al 17 octubre 2021.

En la figura 6 se puede visualizar que existe una similitud en el comportamiento del patrón de movilización de lunes a miércoles. No obstante, de jueves a sábado hay una variación considerable en la conducta de la unidad de transporte. Finalmente, el domingo la unidad no tiene movimiento. Por lo tanto, al haber una notable variación de comportamiento del patrón en los días mencionados, fue necesario ampliar el periodo de investigación del 4 al 31 de octubre de 2021, a fin de tener un análisis más profundo y, de esta manera, las conclusiones puedan ser más exactas.

La figura 7 revela la conducta de la unidad de transporte en el periodo de tiempo ampliado utilizado para la presente investigación:

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil utilizando las cadenas de Márkov con Python

Elaborado por: Elaboración propia

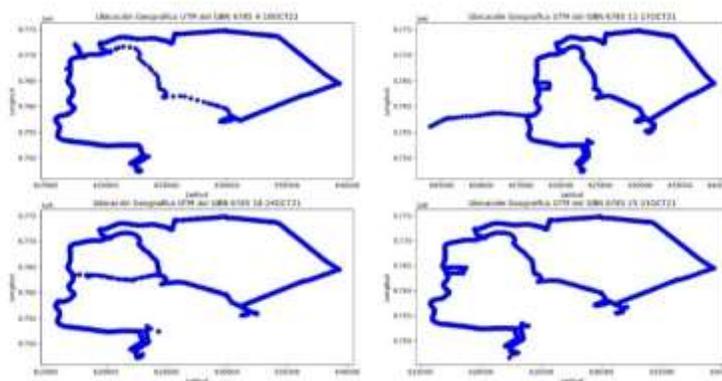


Figura 9: Posición geográfica de la unidad semanal del 4 al 31 de octubre 2021

En la figura 78 se puede visualizar que el patrón de recorrido de la unidad de transporte en las dos primeras semanas cubre los 9 puntos de destinos que tiene que recorrer, graficados en la figura 8 y, a la vez, se aprecia claramente una similitud del patrón cada 2 semanas.

Como resultado de este análisis, se procedió analizar la información cada 2 semanas.

Elaborado por: Elaboración propia



Figura 10: Ubicación Geográfica del GBN 6785 11 al 17 octubre 2021.

La grafica de la figura 8 fue descargada de la plataforma web Google Earth Pro. En esta, se pueden visualizar las rutas y los puntos de destinos de la unidad transporte analizado en la semana del 11 al

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil utilizando las cadenas de Márkov con Python

17 de octubre del 2021, donde se puede apreciar un comportamiento similar a la gráfica realizada en Python en la figura 7 en la mencionada semana.

Extracción de las rutas

La información previamente analizada dio cabida a que el estudio se lo realice cada catorce días, en función del patrón de comportamiento del recorrido de la unidad de transporte. La data analizada en la presente investigación corresponde a un tráiler de una compañía privada dedicada a la transportación de productos al granel de la ciudad de Guayaquil. En el análisis realizado durante un periodo de 14 días, desde el 4 de octubre del 2021 al 17 de octubre del 2021, se pudo extraer la información de los puntos de interés visitados por la unidad de transporte por medio de la plataforma gurtam, lo que se aprecia en la tabla 4, en la que se visualizan los primeros 10 desplazamientos de un total de 245.

Figura 11: Desplazamiento de la unidad de transporte del 4 al 17 de octubre 2021.

Posición inicial	Coordenadas iniciales	Fin	Posición final	Coordenadas finales	Duración	Tiem
Avenida 5, Guayaquil, Guayas, Ecuador	-2.264813, -79.896764	04.10.2021 07:07:57	Avenida 25 De Julio, Guayaquil, Guayas, Ecuador	-2.264873, -79.898062	0:03:00	
Avenida 25 De Julio, Guayaquil, Guayas, Ecuador	-2.265105, -79.898196	04.10.2021 08:21:38	Avenida 25 De Julio, Guayaquil, Guayas, Ecuador	-2.264898, -79.897982	0:03:44	
Avenida 25 De Julio, Guayaquil, Guayas, Ecuador	-2.264898, -79.897982	04.10.2021 08:30:08	Avenida 25 De Julio, Guayaquil, Guayas, Ecuador	-2.265762, -79.898489	0:01:00	
Avenida 25 De Julio, Guayaquil, Guayas, Ecuador	-2.265762, -79.898489	04.10.2021 10:17:19	E49, Eloy Alfaro, Guayas, Ecuador	-2.195324, -79.830409	1:25:41	
E49, Eloy Alfaro, Guayas, Ecuador	-2.195324, -79.830409	04.10.2021 10:25:49	E49, Eloy Alfaro, Guayas, Ecuador	-2.194673, -79.831733	0:02:30	
E49, Eloy Alfaro, Guayas, Ecuador	-2.194673, -79.831733	04.10.2021 10:38:19	E49, Eloy Alfaro, Guayas, Ecuador	-2.197083, -79.835911	0:06:00	
E49, Eloy Alfaro, Guayas, Ecuador	-2.197083, -79.835911	04.10.2021 11:17:59	E49, Eloy Alfaro, Guayas, Ecuador	-2.194951, -79.831760	0:13:40	
E49, Eloy Alfaro, Guayas, Ecuador	-2.194951, -79.831760	04.10.2021 12:57:40	Avenida 25 De Julio, Guayaquil, Guayas, Ecuador	-2.265339, -79.898364	1:33:11	
Avenida 25 De Julio, Guayaquil, Guayas, Ecuador	-2.265339, -79.898364	04.10.2021 13:08:01	Avenida 25 De Julio, Guayaquil, Guayas, Ecuador	-2.265102, -79.898142	0:01:21	
ella Maris, Guayaquil, Guayas, Ecuador	-2.266485, -79.897938	04.10.2021 14:28:14	Avenida 25 De Julio, Guayaquil, Guayas, Ecuador	-2.264775, -79.897840	0:01:30	

Fuente: (GURTAM, 2021)

En primera instancia, se planteó la posibilidad de sacar los puntos visitados por la unidad por medio de los nodos que tenían mayor frecuencia con ayuda de python. Sin embargo, debido a la extensa área de los puntos de carga y descarga existieron complicaciones ya que la información se reflejaba según se puede apreciar en la tabla 4, con lo cual se dificultaba la visualización de rutas falsas generadas el desplazamiento se producía en una distancia mínima de 100 metros, ya que la

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil
 utilizando las cadenas de Márkov con Python

plataforma web registra como una ruta cuando la unidad tiene un desplazamiento mínimo de 100 metros.

Con la ayuda de geocercas en la plataforma Gurtam se generó un informe de geocercas que ayudó a depurar el número de veces que la unidad visitaba un lugar, ya sea para cargar o descargar. En la tabla 5 se puede ver la hora de entrada y salida de la unidad en una geocerca, lo que facilitó detectar el número de veces en el día que la unidad visitó un punto y, a su vez, las rutas realizadas, en virtud de que la unidad de transporte no ejecuta muchos viajes durante el día, debido a la logística de la carga y descarga de grandes cantidades de grano y pasta. En tal sentido, se logró realizar la depuración de forma manual. Luego de la depuración, se generó un archivo en Excel con 101 rutas creadas por la unidad de transporte como se puede visualizar en la tabla 5.

Figura 12: Ingreso y salida de los puntos de descarga y carga de la unidad de transporte.

C	Agrupación	Geocerca	Hora de entrada	Hora de salida	Duración en
1	04.10.2021	----	00:00:08	2021-10-05 09:02:12	1 días 1:34:40
1.1	2021-10-04 00:00:08	Cargomar	00:00:08	08:51:54	8:51:46
1.2	2021-10-04 10:36:50	Gisis 4.5	10:36:50	11:12:50	0:36:00
1.3	2021-10-04 12:56:40	Cargomar	12:56:40	14:37:32	1:40:52
1.4	2021-10-04 16:26:08	Gisis 4.5	16:26:08	16:56:39	0:30:31
1.5	2021-10-04 19:06:41	Cargomar	19:06:41	2021-10-05 09:02:12	13:55:31
2	2021-10-05	----	10:52:20	2021-10-06 07:27:29	18:25:27
2.1	2021-10-05 10:52:20	Gisis 4.5	10:52:20	11:26:21	0:34:01
2.2	2021-10-05 13:21:34	Cargomar	13:21:34	15:31:46	2:10:12
2.3	2021-10-05 15:46:15	Cargomar	15:46:15	2021-10-06 07:27:29	15:41:14
3	2021-10-06	----	09:19:08	23:23:03	7:01:21
3.1	2021-10-06 09:19:08	Gisis 4.5	09:19:08	09:55:08	0:36:00
3.2	2021-10-06 12:02:00	Cargomar	12:02:00	12:48:14	0:46:14
3.3	2021-10-06 12:52:28	Andipuerto	12:52:28	14:49:42	1:57:14
3.4	2021-10-06 16:07:25	Brentag	16:07:25	16:13:55	0:06:30

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil utilizando las cadenas de Márkov con Python

3.5	2021-10-06 17:12:12	Andipuerto	17:12:12	18:34:21	1:22:09
3.6	2021-10-06 18:37:39	Cargomar	18:37:39	19:30:04	0:52:25
3.7	2021-10-06 22:02:14	Andipuerto	22:02:14	23:23:03	1:20:49

Fuente: (GURTAM, 2021)

En la figura 9 se puede visualizar un ejemplo, en un nodo o punto de destino, de las geocercas realizadas para generar el informe de geocercas indicado en la tabla 5. Claramente se puede observar que es un perímetro bastante extenso que ayuda a verificar que la unidad llegó al punto analizado.

Fuente: (GURTAM, 2021)



Figura 13: Creación de geocercas en Andipuerto.

Con la información del número de veces que la unidad visita los nodos y las rutas realizadas, se procede a realizar la matriz de transición de probabilidades de la cadena de Márkov de las diferentes rutas o destinos. En la matriz de transición, se etiquetó cada destino con una letra para hacer más sencillo el análisis, como se visualiza en la tabla 6. Con el informe de geocercas, se constató el número de veces que la unidad de transporte recorría una ruta de los nodos en análisis, para posteriormente calcular la frecuencia de cada recorrido en mención.

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil utilizando las cadenas de Márkov con Python

Figura 14: Destino de la Unidad de Transporte

Numero	Etiqueta	Destino
1	A	ANDIPUERTO
2	B	BRENTAG
3	C	CARGOMAR
4	D	ECUABULK
5	E	EMPAGRAN
6	F	GISIS 4.5
7	G	GISIS 6.5
8	H	MOCHASA
9	I	TLD

Elaborado por: (Juan Carlos Yturalde V, 2022)

Las 101 rutas generadas por la unidad de transporte del 4 al 17 de octubre del 2021, se especifican detalladamente en la tabla 7. Cabe mencionar que se realiza la correspondiente depuración de las rutas, según se describió en párrafos anteriores.

Figura 15: Rutas etiquetadas de la unidad de transporte del 4 al 17 de octubre del 2021

Número	Origen	Destino	Ruta	Número	Origen	Destino	Ruta	Número	Origen	Destino	Ruta
1	CARGOMAR	GISIS 4.5	CF	35	ANDIPUERT	CARGOMAR	AC	69	ANDIPUERT	ECUABULK	AD
2	GISIS 4.5	CARGOMAR	FC	36	CARGOMAR	TLD	CI	70	ECUABULK	ANDIPUERT	DA
3	CARGOMAR	GISIS 4.5	CF	37	TLD	O	IA	71	O	ECUABULK	AD
4	GISIS 4.5	CARGOMAR	FC	38	ANDIPUERT	TLD	AI	72	ECUABULK	CARGOMAR	DC
5	CARGOMAR	GISIS 4.5	CF	39	O	CARGOMAR	AC	73	CARGOMAR	GISIS 4.5	CF
6	GISIS 4.5	CARGOMAR	FC	40	CARGOMAR	TLD	CI	74	GISIS 4.5	CARGOMAR	FC
7	CARGOMAR	GISIS 4.5	CF	41	ANDIPUERT	O	IA	75	CARGOMAR	GISIS 4.5	CF
8	GISIS 4.5	CARGOMAR	FC	42	O	TLD	AI	76	GISIS 4.5	CARGOMAR	FC
9	CARGOMAR	O	CA	43	TLD	CARGOMAR	IC	77	CARGOMAR	GISIS 4.5	CF
10	ANDIPUERT	BRENTAG	AB	44	CARGOMAR	O	CA	78	GISIS 4.5	O	FA
11	ANDIPUERT	O	BA	45	ANDIPUERT	GISIS 6.5	AG	79	O	ECUABULK	AD
12	O	CARGOMAR	AC	46	GISIS 6.5	TLD	GI	80	ECUABULK	O	DA

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil
 utilizando las cadenas de Márkov con Python

13	CARGOMAR ANDIPUERT	ANDIPUERT O	CA	47	TLD	EMPAGRAN	IE	81	ANDIPUERT O	ECUABULK ANDIPUERT	AD
14	O	CARGOMAR ANDIPUERT	AC	48	EMPAGRAN	CARGOMAR ANDIPUERT	EC	82	ECUABULK ANDIPUERT	O	DA
15	CARGOMAR ANDIPUERT	O	CA	49	CARGOMAR ANDIPUERT	O	CA	83	O	ECUABULK	AD
16	O	CARGOMAR	AC	50	O	GISIS 6.5 ANDIPUERT	AG	84	ECUABULK	CARGOMAR ANDIPUERT	DC
17	CARGOMAR	BRENTAG ANDIPUERT	CB	51	GISIS 6.5 ANDIPUERT	O	GA	85	CARGOMAR ANDIPUERT	O	CA
18	BRENTAG ANDIPUERT	O	BA	52	O	GISIS 6.5	AG	86	O	ECUABULK ANDIPUERT	AD
19	O	BRENTAG ANDIPUERT	AB	53	GISIS 6.5	CARGOMAR ANDIPUERT	GC	87	ECUABULK ANDIPUERT	O	DA
20	BRENTAG ANDIPUERT	O	BA	54	CARGOMAR ANDIPUERT	O	CA	88	O	ECUABULK ANDIPUERT	AD
21	O	CARGOMAR ANDIPUERT	AC	55	O	ECUABULK ANDIPUERT	AD	89	ECUABULK ANDIPUERT	O	DA
22	CARGOMAR ANDIPUERT	O	CA	56	ECUABULK ANDIPUERT	O	DA	90	O	ECUABULK ANDIPUERT	AD
23	O	CARGOMAR	AC	57	O	ECUABULK ANDIPUERT	AD	91	ECUABULK ANDIPUERT	O	DA
24	CARGOMAR	GISIS 4.5	CF	58	ECUABULK ANDIPUERT	O	DA	92	O	ECUABULK ANDIPUERT	AD
25	GISIS 4.5	CARGOMAR	FC	59	O	ECUABULK	AD	93	ECUABULK ANDIPUERT	O	DA
26	CARGOMAR	GISIS 4.5 ANDIPUERT	CF	60	ECUABULK	MOCHASA	DH	94	O	ECUABULK	AD
27	GISIS 4.5 ANDIPUERT	O	FA	61	MOCHASA	CARGOMAR ANDIPUERT	HC	95	ECUABULK	CARGOMAR ANDIPUERT	DC
28	O	TLD	AI	62	CARGOMAR ANDIPUERT	O	CA	96	CARGOMAR ANDIPUERT	O	CA
29	TLD	CARGOMAR ANDIPUERT	IC	63	O	ECUABULK ANDIPUERT	AD	97	O	ECUABULK ANDIPUERT	AD
30	CARGOMAR ANDIPUERT	O	CA	64	ECUABULK ANDIPUERT	O	DA	98	ECUABULK ANDIPUERT	O	DA
31	O	TLD ANDIPUERT	AI	65	O	ECUABULK ANDIPUERT	AD	99	O	ECUABULK ANDIPUERT	AD
32	TLD ANDIPUERT	O	IA	66	ECUABULK ANDIPUERT	O	DA	100	ECUABULK ANDIPUERT	O	DA
33	O	TLD ANDIPUERT	AI	67	O	ECUABULK ANDIPUERT	AD	101	O	ECUABULK	AD
34	TLD	O	IA	68	ECUABULK	O	DA				

Elaborado por: (Juan Carlos Yturalde V, 2022)

En la tabla 7, se puede visualizar que existen 101 rutas de destino de la unidad de transporte analizada y se puede observar el origen y el destino de cada ruta. Para facilitar el análisis se etiquetó con una letra mayúscula a cada destino. De esta forma, se puede trabajar con mayor facilidad la matriz de transición que se explicará en la siguiente sección.

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil utilizando las cadenas de Márkov con Python

Generación de la matriz de probabilidad de transición

En esta sección se estableció, en base al número de ocurrencia de las rutas de la sección anterior, una matriz de transición de probabilidad con una dimensión 9X9, debido a que se tiene 9 destinos o lugares importantes en el modelo del negocio para carga, descarga o estación de la unidad de transporte.

Se contabilizó la frecuencia de ocurrencia de cada ruta según la información obtenida en la tabla 7 y se la colocó en un matriz, con lo cual se obtuvo una matriz de ocurrencia de las rutas de la unidad de transporte, como se puede apreciar en la tabla 7.

Figura 16: Matriz de ocurrencia de las rutas

(Origen/Destino)	A	B	C	D	E	F	G	H	I
A	0	2	7	19	0	0	3	0	5
B	3	0	0	0	0	0	0	0	0
C	11	1	0	0	0	9	0	0	2
D	14	0	3	0	0	0	0	1	0
E	0	0	1	0	0	0	0	0	0
F	2	0	7	0	0	0	0	0	0
G	1	0	1	0	0	0	0	0	1
H	0	0	1	0	0	0	0	0	0
I	4	0	2	0	1	0	0	0	0

Elaborado por: (Juan Carlos Yturalde V, 2022)

Luego se realizó la sumatoria de la ocurrencia de cada renglón para, de esta manera, normalizarla dividiendo cada elemento de cada renglón por la sumatoria de ocurrencia de cada renglón. En ese sentido, se obtuvo una matriz normalizada que se utilizará en el análisis de la información, dando como resultado la matriz de transición de probabilidades de las rutas de la unidad de transporte según se puede visualizar en la tabla 9.

Figura 17: Matriz de Transición de Probabilidad

(Origen/Destino)	A	B	C	D	E	F	G	H	I
A	0.00000	0.05556	0.19444	0.52778	0.00000	0.00000	0.08333	0.00000	0.13889
B	1.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
C	0.47826	0.04348	0.00000	0.00000	0.00000	0.39130	0.00000	0.00000	0.08696

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil
 utilizando las cadenas de Márkov con Python

D	0.77778	0.00000	0.16667	0.00000	0.00000	0.00000	0.00000	0.05556	0.00000
E	0.00000	0.00000	1.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
F	0.22222	0.00000	0.77778	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
G	0.33333	0.00000	0.33333	0.00000	0.00000	0.00000	0.00000	0.00000	0.33333
H	0.00000	0.00000	1.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
I	0.57143	0.00000	0.28571	0.00000	0.14286	0.00000	0.00000	0.00000	0.00000

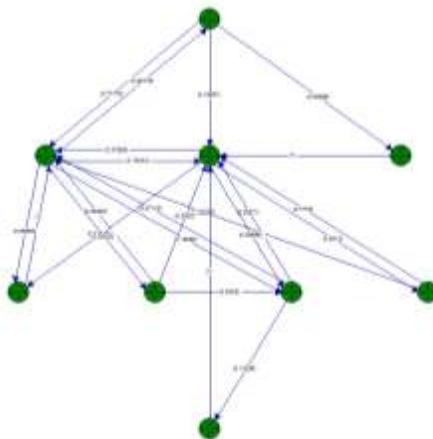
Elaborado por: (Juan Carlos Yturalde V, 2022)

Con la matriz de transición de probabilidad y utilizando las cadenas de Márkov, se pretende verificar si el sistema de rutas generadas es estable, y luego determinar la probabilidad de ocurrencia de un destino en particular o rutas de máximo tres pasos.

En la figura 11 se puede visualizar cómo se distribuye la ruta de transporte pesado por los puntos de descargas y carga por medio de un diagrama de grafos. Los nodos etiquetados de la letra A hasta a la I se pueden visualizar con los números desde 0 al 8 respectivamente. Es decir, que el nodo A se grafica como el nodo 0 y así sucesivamente.

En este diagrama de grafo, figura 10, se puede visualizar claramente que todos los nodos llegan a ANDIPUERTO a excepción de los nodos MOCHASA y EMPAGRAN que tiene que pasar previamente por el nodo CARGOMAR para después arribar al nodo ANDIPUERTO.

Figura 18: Diagrama de grafo de la unidad de transporte



Elaborado por: Elaboración propia

Matriz estacionaria

En esa sección se realizan los análisis matemáticos para verificar si la matriz de transición de probabilidad es estacionaria o no, lo cual arrojó como resultado que es estacionario, ya que luego de multiplicar varias veces la matriz por sí misma se detectó que la matriz de transición de probabilidad se estabilizó en el paso 100 con la ayuda de python y la librería sympy, como se puede observar en la figura 11.

El código para generar la multiplicación de la matriz de transición 100 veces, es el que se detalla a continuación:

```
from sympy import *  
A=Matrix([[0,0.0556,0.1944,0.5278,0,0,0.0833,0,0.1389],[1,0,0,0,0,0,0,0,0],[0.4783,0.0435,0,0,0,0,  
3913,0,0,0.0869],[0.7778,0,0.1667,0,0,0,0,0.0555,0  
],[0,0,1,0,0,0,0,0,0],[0.2222,0,0.7778,0,0,0,0,0,0],[0.3333,0,0.3333,0,0,0,0,0,0.3334],[0,0,1,0,0,0,0,  
0,0  
],[0.5714,0,0.2857,0,0.1429,0,0,0,0]])  
A**100
```

El análisis previo se lo realizó multiplicando la matriz de transición por sí mismo, un número elevado de veces. El producto de la matriz de transición se obtuvo al multiplicar cien veces por sí misma la matriz de transición y se obtuvo como resultado renglones idénticos. A continuación, se realizará analíticamente otro método para saber si la matriz tiene una distribución estacionaria, utilizando Python y la librería numpy:

En primer lugar, hallamos la matriz transpuesta de la matriz de transición. El código de python para obtener la matriz de transición transpuesta es el siguiente:

```
import numpy as np  
P=np.array([[0,0.0556,0.1944,0.5278,0,0,0.0833,0,0.1389],[1,0,0,0,0,0,0,0,0],[0.4783,0.0435,0,0,0,  
0.3913,0,0,0.0869],[0.7778,0,0.1667,0,0,0,0,0.0555,0],[0,0,1,0,0,0,0,0,0],[0.2222,0,0.7778,0,0,0,0,0,  
0],[0.3333,0,0.3333,0,0,0,0,0,0.3334],[0,0,1,0,0,0,0,0,0],[0.5714,0,0.2857,0,0.1429,0,0,0,0]])  
#MATRIZ DE TRANSICIÓN  
P = P.transpose() #1.MATRIZ DE TRANSICIÓN TRANSPUESTA
```

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil
utilizando las cadenas de Márkov con Python

```
print(P)
```

Figura 19: Matriz de Transición Transpuesta

```
[[0.      1.      0.4783 0.7778 0.      0.2222 0.3333 0.      0.5714]
 [0.0556 0.      0.0435 0.      0.      0.      0.      0.      0.      ]
 [0.1944 0.      0.      0.1667 1.      0.7778 0.3333 1.      0.2857]
 [0.5278 0.      0.      0.      0.      0.      0.      0.      0.      ]
 [0.      0.      0.      0.      0.      0.      0.      0.      0.1429]
 [0.      0.      0.3913 0.      0.      0.      0.      0.      0.      ]
 [0.0833 0.      0.      0.      0.      0.      0.      0.      0.      ]
 [0.      0.      0.      0.0555 0.      0.      0.      0.      0.      ]
 [0.1389 0.      0.0869 0.      0.      0.      0.3334 0.      0.      ]]
```

Elaborado por: Elaboración propia

En segundo lugar, hallamos los valores y vectores propios de la matriz transpuesta. Se utiliza el método `np.linalg.eig(P)` (Numpy, 2022) para obtener los vectores y valores propios W y V respectivamente. Los valores propios son nueve, pero para encontrar la distribución estacionaria a largo plazo se escoge el valor de 1 correspondiente al primer valor del arreglo W de la figura 12. Una vez elegido el valor propio uno, el vector propio corresponde a la primera columna de la matriz de vectores propios de la figura 14. El siguiente código permite hallar los valores propios de la matriz de transición de probabilidad transpuesta:

```
W, V = np.linalg.eig(P)
```

```
print(W)
```

Figura 20: Matriz de valores propios

```
[ 1.00000000e+00+0.00000000e+00j -7.70789415e-01+0.00000000e+00j
 -5.41417585e-01+0.00000000e+00j  3.09106536e-01+0.00000000e+00j
  6.49197025e-02+0.00000000e+00j -6.18192379e-02+0.00000000e+00j
 -4.46041490e-17+1.31173332e-09j -4.46041490e-17-1.31173332e-09j
 -3.35728944e-17+0.00000000e+00j]
```

Elaborado por: Elaboración propia

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil utilizando las cadenas de Márkov con Python

Para visualizar la matriz de vectores propios se imprime V con el código `print(V)` y así mostramos la matriz de valores propios como se visualiza en la figura 13:

Figura 21: Matriz de vectores propios

```
[[ 7.47000910e-01+0.00000000e+00j  8.00755584e-01+0.00000000e+00j
 2.35732774e-01+0.00000000e+00j -3.02103828e-01+0.00000000e+00j
-3.65470066e-02+0.00000000e+00j -3.74768571e-02+0.00000000e+00j
-2.53165385e-17+1.03289720e-24j -2.53165385e-17-1.03289720e-24j
 1.12034124e-16+0.00000000e+00j]
 [ 6.18115319e-02+0.00000000e+00j -4.78278060e-02+0.00000000e+00j
-8.44467972e-02+0.00000000e+00j  1.42442068e-02+0.00000000e+00j
 6.30438229e-02+0.00000000e+00j -2.10792486e-02+0.00000000e+00j
-4.32601960e-16+1.11607975e-08j -4.32601960e-16-1.11607975e-08j
 1.91103620e-01+0.00000000e+00j]
 [ 4.66167387e-01+0.00000000e+00j -1.76019400e-01+0.00000000e+00j
 7.49752616e-01+0.00000000e+00j  4.87355178e-01+0.00000000e+00j
 1.40799995e-01+0.00000000e+00j  7.78578468e-02+0.00000000e+00j
 8.43893681e-16-1.97185514e-23j  8.43893681e-16+1.97185514e-23j
-6.30191945e-16+0.00000000e+00j]
 [ 3.94267080e-01+0.00000000e+00j -5.48319410e-01+0.00000000e+00j
-2.29803688e-01+0.00000000e+00j -5.15842863e-01+0.00000000e+00j
-2.97128750e-01+0.00000000e+00j  3.19969735e-01+0.00000000e+00j
-3.87665490e-15+1.67123518e-08j -3.87665490e-15-1.67123518e-08j
-2.80085309e-17+0.00000000e+00j]
 [ 2.35805357e-02+0.00000000e+00j  1.61336997e-02+0.00000000e+00j
 4.18291471e-02+0.00000000e+00j -4.00134777e-02+0.00000000e+00j
-2.87369591e-01+0.00000000e+00j  6.87900989e-01+0.00000000e+00j
-7.07106781e-01+0.00000000e+00j -7.07106781e-01-0.00000000e+00j
 3.39126846e-01+0.00000000e+00j]
 [ 1.82411299e-01+0.00000000e+00j  8.93582472e-02+0.00000000e+00j
-5.41870465e-01+0.00000000e+00j  6.16946131e-01+0.00000000e+00j
 8.48664365e-01+0.00000000e+00j -4.92820301e-01+0.00000000e+00j
 9.14099024e-15-9.20378970e-08j  9.14099024e-15+9.20378970e-08j
-8.60052295e-01+0.00000000e+00j]
 [ 6.22251758e-02+0.00000000e+00j -8.65384745e-02+0.00000000e+00j
-3.62687519e-02+0.00000000e+00j -8.14128656e-02+0.00000000e+00j
-4.68943253e-02+0.00000000e+00j  5.04992022e-02+0.00000000e+00j
-3.18388217e-16-1.25099497e-23j -3.18388217e-16+1.25099497e-23j
 5.47041619e-17+0.00000000e+00j]
 [ 2.18818230e-02+0.00000000e+00j  3.94812470e-02+0.00000000e+00j
 2.35568719e-02+0.00000000e+00j -9.26194551e-02+0.00000000e+00j
-2.54016038e-01+0.00000000e+00j -2.87262038e-01+0.00000000e+00j
 7.07106781e-01+7.06555492e-08j  7.07106781e-01-7.06555492e-08j
 3.29821829e-01+0.00000000e+00j]
 [ 1.65014246e-01+0.00000000e+00j -8.70236878e-02+0.00000000e+00j
-1.58481706e-01+0.00000000e+00j -8.65530263e-02+0.00000000e+00j
-1.30552473e-01+0.00000000e+00j -2.97589327e-01+0.00000000e+00j
 1.10988375e-15-6.49080141e-09j  1.10988375e-15+6.49080141e-09j
-4.59454367e-16+0.00000000e+00j]]
```

Elaborado por: Elaboración propia

Se visualiza la primera columna de la matriz de vectores propios con el siguiente comando `print(V[:,0])` y se obtienen los valores de la figura 14.

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil utilizando las cadenas de Márkov con Python

Figura 22: Vector propio de la Matriz de Transición Transpuesta

```
[0.74700091+0.j 0.06181153+0.j 0.46616739+0.j 0.39426708+0.j  
0.02358054+0.j 0.1824113 +0.j 0.06222518+0.j 0.02188182+0.j  
0.16501425+0.j]
```

Elaborado por: Elaboración propia

Finalmente, se realiza la sumatoria de la parte real de la primera columna de la matriz de vectores propios, lo cual refleja como resultado un valor de 2.12436, que luego se divide para cada posición del vector con el siguiente comando `print(np.dot(1/2.12436,V[:,0]))` y se obtiene el resultado que refleja la figura 15, que corresponde a la primera columna de matriz de vectores propios normalizados.

Figura 23: Vector propio normalizados

```
[0.35163574+0.j 0.02909654+0.j 0.21943898+0.j 0.18559335+0.j  
0.01110007+0.j 0.08586647+0.j 0.02929126+0.j 0.01030043+0.j  
0.07767716+0.j]
```

Elaborado por: Elaboración propia

Se concluye que los resultados del vector propio normalizado de la figura 16 corresponde a los reglones de la matriz con distribución estacionaria. De esta manera, se comprueba por dos métodos analíticos que la matriz de transición estudiada corresponde a una matriz con distribución estacionaria.

Script en Python de las probabilidades de ocurrencia de una ruta determinada

El script desarrollado presenta la probabilidad de ocurrencia de una determinada ruta. Esto ayudaría al departamento de logística de la compañía de transporte a que tenga información de primera mano y de manera inmediata para la toma de decisiones para ahorrar recursos sobre el recorrido de la unidad.

El diagrama de bloque del código está dividido en cuatro etapas o procesos. En la primera etapa corresponde al ingreso de la ruta de tres pasos ingresadas por medio de teclado. En este paso, el

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil utilizando las cadenas de Márkov con Python

usuario debe colocar correctamente las rutas de partida de la unidad y los tres destinos adicionales que se necesita colocar para encontrar la probabilidad de ocurrencia. La segunda etapa consta de un proceso predefinido en el cual se carga el dataframe y computa la matriz de transición de probabilidad. La tercera etapa conlleva al procesamiento del cálculo de la probabilidad de ocurrencia según la ruta ingresada. Finalmente, la cuarta etapa corresponde a la presentación o visualización de la probabilidad de ocurrencia según la ruta establecida. En la figura 16 se puede apreciar el diagrama de flujo del script desarrollado para generar la probabilidad de ocurrencia de 3 pasos o rutas.

Figura 24: Diagrama de flujo (Script probabilidad de ocurrencia de una ruta)



Elaborado por: Elaboración propia

Para que el script funcione correctamente, primero se ingresa el nodo o punto inicial de la unidad transporte y posteriormente se ingresa la secuencia de tres nodos o destinos de la unidad para obtener la ruta en análisis. Si algún nodo es escrito incorrectamente, el script solicitará que se

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil utilizando las cadenas de Márkov con Python

ingrese nuevamente el nodo o punto correctamente para posteriormente computar la probabilidad de ocurrencia de la ruta ingresada.

El script desarrollado permite calcular la probabilidad de ocurrencia de n pasos, pero en este estudio nos enfocamos en script de tres pasos como ejemplo.

Discusión

A partir de la matriz de transición de probabilidades de la ocurrencia de visitas a los nueve destinos en los que se desplaza la unidad de transporte en análisis y de la generación de un código para la obtención de la probabilidad de ocurrencia de una ruta de 3 pasos.

En el análisis se trabajó con 95.096 datos de las ubicaciones geográficas del recorrido de la unidad de transporte de trigo y pasta de soya de la ciudad de Guayaquil, extraído de la página web wialon en formato decimal, y luego se convirtieron en formato UTM para representar todos los puntos geográficos que reportaban cada 30 segundos cuando se encontraba en movimiento y cada 10 minutos cuando la unidad estaba detenida, en el periodo del 4 al 31 de octubre del 2021.

Se concluyó visualmente que el patrón del recorrido de la unidad en las dos primeras semanas cubría todos los destinos de sus visitas. Además, se confirmó que, por medio del proceso de la cadena de Markov, la matriz de transición de probabilidad del número de visitas a los nueve destinos o nodos de descargas y cargas de la unidad tenían una distribución estacionaria.

La metodología que más se acopla a la investigación es la CRISP-DM, ya que en el análisis establecido en este artículo se tuvo que entender el modelo del negocio; luego realizar una representación de los datos de la ubicación geográfica de la unidad de transporte en análisis; para, como siguiente paso, explorar la información de manera visual, detectando su comportamiento o patrón para elegir el periodo donde cubra todas las posibles rutas para que ningún nodo en análisis quede fuera.

El cuarto paso fue la generación de la matriz de transición de probabilidades del número de visitas a los nueve nodos en análisis en el periodo del 4 al 17 de octubre del 2021, para luego mediante el proceso de la cadena de Markov detectar si la matriz de transición tenía una distribución estacionaria.

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil utilizando las cadenas de Márkov con Python

El quinto paso fue la visualización de los resultados de la matriz de transición estacionaria a largo plazo, para concluir con el último paso que corresponde al análisis de los resultados obtenidos.

En la etapa de representación y exploración de los datos se logró graficar en dos dimensiones los 95.096 datos de una manera clara y eficiente de la misma manera que si los puntos geográficos fueran representados en la herramienta de Google Earth Pro, pero con la ventaja de que el costo computacional utilizando el lenguaje de programación Python es mucho menor. Esta representación y visualización de los datos de las coordenadas geográficas UTM se logró realizar de manera eficiente debido a que la zona y la banda de las coordenadas UTM son las mismas para toda la extensión geográfica en análisis, es decir 17M.

En la generación de la matriz de transición de probabilidades se realizó un filtro de información y depuración manual obteniendo 101 rutas establecidas en el periodo del 4 al 17 de octubre del 2021. Luego, con la ayuda de las herramientas Python, se multiplicó 100 veces la misma matriz de transición, concluyendo que la matriz tenía una distribución estacionaria, comprobada analíticamente utilizando el cálculo de los vectores propios de la matriz de transición transpuesta con ayuda de la librería numpy y el método `np.linalg.eig(P)` llegando al mismo resultado.

En la etapa de análisis de resultados se obtuvo que el 75,76% de las visitas de la unidad lo realizan a los siguientes puntos: Andiupuerto, Cargomar y Ecuabulk que son los lugares de cargas exclusivamente. Además, que con la información generada por el script desarrollado de la probabilidad de ocurrencia de una ruta determinada con un pequeño calculo, se podría estimar el costo de combustible utilizado en un determinado periodo.

El script de obtención de la probabilidad de ocurrencia de una ruta de tres pasos se desarrolló con el lenguaje de programación de Python, utilizando el entorno de desarrollo integrado spyder. Con este código, permite al área de logística de la empresa conocer de manera inmediata la probabilidad de ocurrencia de una determinada ruta de tres pasos y, además, en base la información de la probabilidad de ocurrencia de la ruta, en conjunto con el valor de la probabilidad obtenido en la matriz estacionario de cada punto, se podrá estimar el número de kilómetros recorridos en una ruta determinada; así como estimar el consumo de combustible en ese itinerario en particular. De esa

Análisis logístico de los puntos de carga y descarga de una compañía de transporte de la ciudad de Guayaquil utilizando las cadenas de Márkov con Python

manera, el área de logística de la compañía de transporte puede tener más información para la toma de decisiones.

Conclusiones

Se verificó que la ruta analizada del vehículo de transporte pesado de la compañía de la ciudad de Guayaquil que transporta trigo y pasta de soya tiene una distribución estacionaria como lo refleja el estudio realizado por medio de la cadena de Markov.

Se comprobó que utilizando la librería de numpy los resultados de los vectores propios son más exactos que utilizando la librería de sympy.

Se concluye que la probabilidad a largo plazo de visita de los nueve nodos analizados corresponde a los reglones de la matriz estacionaria.

Se establece que el 75,65% de las visitas corresponde a los nodos de Andipuerto, Cargomar y Ecuabulk.

Se ha comprobado que el uso de Python para graficar grandes cantidades de datos, como se lo realizó en este trabajo de fin de máster, es comparativamente más rentable al ser gratuito y por tener un costo computacional bajo.

La metodología utilizada para analizar la matriz de transición a largo plazo permitirá ser la base para el estudio de optimización de rutas y costo de combustibles promedio de una ruta determinada. Utilizar coordenadas geográficas UTM permitió graficar la latitud y longitud en dos dimensiones con jupyter notebook de python y analizar gráficamente sus rutas con un costo computacional mucho menor que si utiliza Google Earth Pro. La representación gráfica del movimiento de la ruta se realizó en dos dimensiones por encontrarse en la misma zona y banda de las coordenadas UTM en análisis.

Referencias

1. Gestión de Operaciones. (2015). Cadenas de Markov (Ejercicios Resueltos). <https://www.gestiondeoperaciones.net/cadenas-de-markov/cadenas-de-markov-ejercicios-resueltos/>
2. GURTAM. (2021). Wialon Hosting. <https://hosting.wialon.us/>
3. IBM. (2021). CRISP-DM Help Overview - IBM Documentation. <https://www.ibm.com/docs/en/spss-modeler/SaaS?topic=dm-crisp-help-overview>
4. Jiang, Y. I., Saito, M., & Sinha, K. C. (1988). Bridge Performance Prediction Model Using the Markov Chain. In TRANSPORTATION RESEARCH RECORD (Vol. 1180).
5. Levinson, D., & Chen, W. (2005). Paving New Ground: A Markov Chain Model of the Change in Transportation Networks and Land Use. In Access to Destinations (pp. 243–266). Emerald Group Publishing Limited. <https://doi.org/10.1108/9780080460550-012>
6. Numpy. (2022). numpy.linalg.eig — NumPy v1.22 Manual. <https://numpy.org/doc/stable/reference/generated/numpy.linalg.eig.html>
7. PRIMICIAS. (2021). Precios del diésel y de gasolinas Extra y Ecopaís suben desde el 12 de octubre. <https://www.primicias.ec/noticias/economia/precios-gasolinas-diesel-ecuador/>
8. Viñamagua, G. B. (2017). Uso de las cadenas de Markov para un modelo de negocios. INNOVA Research Journal, 2(9.1), 112–123. <https://doi.org/10.33890/innova.v2.n9.1.2017.507>