

UNIVERSIDAD PRIVADA BOLIVIANA



TRABAJO FINAL

“EMPRESA LOGISTICS GROUP”

DIPLOMADO EN MACHINE LEARNING Y CIENCIA DE DATOS

MÓDULO: MODELOS DE PREDICCIÓN ANALÍTICA

**DOCENTE**: Phd. Hernan Payrumani Mamani

**GRUPO**: Logistics Group

**INTEGRANTES**:

Gustavo Cadena

Gabriel Antonio Mamani Titi

Meliza Mayori Velasco Sanchez

**Fecha**: 27-12-2023

**1.INTRODUCCIÓN**

Logistic Group es una empresa líder en el sector logístico, ofreciendo soluciones integrales para la distribución y gestión eficiente de materiales. Fundada en 2015, se ha destacado por su compromiso con la excelencia en el transporte, almacenamiento y distribución de productos.

La empresa se ha consolidado a lo largo de los años gracias a su enfoque innovador y su capacidad para adaptarse a las demandas cambiantes del mercado. Su ámbito de actuación abarca en la ciudad de Cochabamba, sirviendo a una amplia gama de clientes en diversos sectores, desde minoristas hasta empresas de manufactura.

Uno de los desafíos más significativos que Logistic Group enfrenta actualmente se centra en la asignación de recursos para el proceso de "pickeo" de materiales en dos de sus clientes clave: Family Group y Home Center. El pickeo involucra la selección y recolección de productos en el almacén para satisfacer los pedidos de estos supermercados.

Con una creciente cantidad de operaciones diarias de pickeo para atender los pedidos de Family Group y Home Center, Logistic Group se encuentra ante la necesidad crítica de optimizar la asignación de encargados para este proceso. La correcta asignación de personal es crucial para garantizar la eficiencia operativa, minimizar tiempos de espera y cumplir con las demandas de los clientes.

**2. ALCANCE**

***“El alcance del estudio se limita al análisis de la asignación de recursos humanos al área de logística en las gestiones 2022 y 2023”.***

El análisis que se realizara en el presente proyecto será para tratar de predecir la cantidad de operarios que se requiere para tener un óptimo control del picking de la empresa y se trabajara con datos obtenidos en las gestiones 2022 y 2023, así también mencionar que la empresa ha tenido un incremento en la demanda de importación de productos para Family Group y Home Center en fechas específicas, lo cual exige a la empresa realizar un alto movimiento en ingresos y salidas de los productos de manera diaria, para lo cual en las fechas más significativas se requiere una mayor cantidad de Recursos Humanos (operarios) para el buen funcionamiento del área de logística.

**2.1. DELIMITACIÓN ESPACIAL.**

El presente proyecto se realizará en la Empresa “Logistic Group” en el área de Recursos Humanos personal operativo y la asignación de personal para el área de logística.

**2.2. DELIMITACIÓN TEMPORAL**

El análisis del presente proyecto se realizará con base a datos de reportes diarios del picking del área de logística en las gestiones 2022 – 2023.

**2.2. IDENTIFICACIÓN DEL PROBLEMA.**

La falta de precisión en la asignación de operarios para el pickeo de materiales. Esta ineficiencia ha generado una utilización inadecuada de los recursos humanos, resultando en tiempos de espera prolongados, sobreutilización o subutilización del personal, lo que afecta a la productividad y la calidad del servicio.

**2.3 APLICACIÓN DE MODELOS DE REGRESIÓN**

Se aplicarán modelos de regresión para predecir la cantidad optima de operarios requeridos en función de dos variables clave: la cantidad de líneas y la cantidad de artículos a pickear en el almacén diariamente. Estos modelos permitirán estimar de manera mas precisa y eficiente la cantidad de recursos humanos necesarios para el proceso de pickeo en diferentes situaciones operativas.

**3. DATOS**

**3.1 ORIGEN DE LOS DATOS**

Los datos serán obtenidos de la base de datos interna de Logistic Group, específicamente de los registros históricos de operaciones diarias de pickeo para Family Group y Home Center del año 2022-2023. Tras una ardua limpieza de los datos obtenidos, las filas y columnas de la data con las que se trabajara son los siguientes: la cantidad de líneas, artículos pickeados y la cantidad de operarios asignados en cada día de operación.

No se generarán nuevas variables a partir de los datos de la base, con el fin de no generar multicolinealidad de las variables. Los datos con los que se trabajara serán utilizados tanto para entrenamiento como para la predicción de cantidades de producción.

**3.2 PROCESO DE ANALISIS DE DATOS**

Se analizan los datos mediante visualizaciones de gráficos de dispersión entre las variables independientes. Se buscará identificar patrones o desviaciones significativas para entender los datos.

**GRAFICO Nª 1**

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

En el grafico se puede visualizar la existencia de dos datos atípicos, después de un análisis se puede determinar si es necesario imputar los datos denotando asimismo la existencia de una relación proporcional lineal entre ambas variables.

Por otro lado, el análisis de correlación de las variables muestra resultados por encima del 60%, denotando una correlación positiva en general.

**GRAFICO Nª 2**

Forma, Cuadrado

Descripción generada automáticamente

CORRELACIÓN MEDIANTE HEATMAP

**4. MODELADO**

**4.1 AJUSTE DE MODELO**

Para seleccionar el modelo más adecuado en base a los datos, se optó por utilizar inicialmente un modelo de regresión lineal debido a la naturaleza del problema de predicción de la cantidad de operarios requeridos en función de variables numéricas, como la cantidad de líneas y artículos a pickear.

**Justificación del modelo de regresión lineal**

**Cualitativo**: El modelo de regresión lineal se ajusta al problema al intentar modelar la relación lineal entre las variables de entrada (cantidad de líneas, cantidad de artículos) y la variable de salida (cantidad de operarios asignados).

**Selección de Porcentaje de Datos de Entrenamiento y Prueba:** Se optó por una división común del 70% para entrenamiento y 30% para prueba. Esta elección permite que el modelo se entrene con una cantidad significativa de datos y se pruebe con un conjunto lo suficientemente grande como para evaluar su rendimiento sin sobreajuste o subajuste excesivo.

**Porcentaje de rendimiento aceptado**

Se estableció un criterio de rendimiento aceptado en base a la evaluación del coeficiente de determinación (R²), donde un valor cercano a 1 indicará un buen ajuste del modelo a los datos. Se considera aceptable un R² superior al 0.7, aunque se buscará optimizar el modelo para obtener un R² más alto si es posible.

**4.2 ANÁLISIS DE ERRORES**

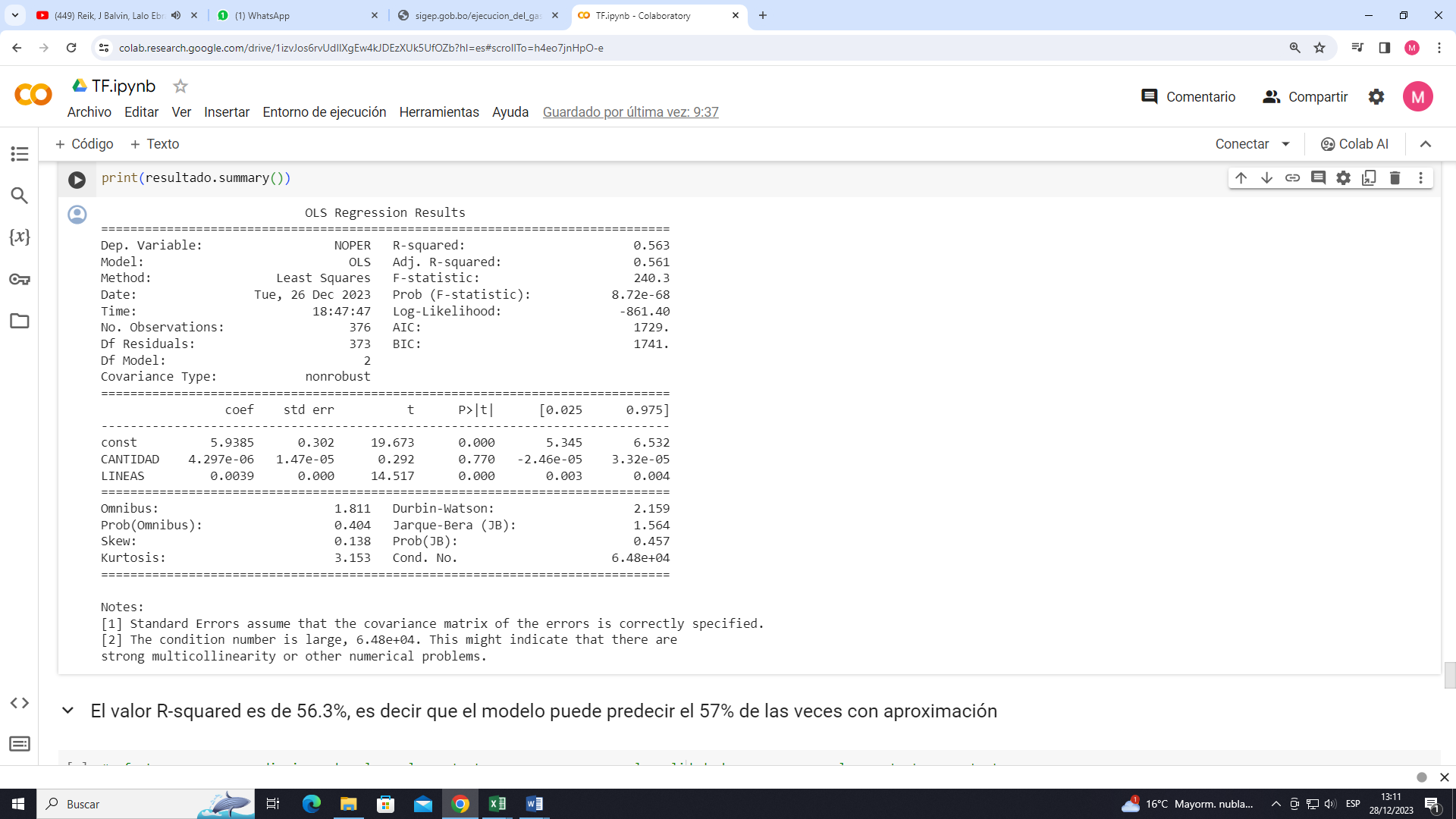
**Selección Justificada:** Se optó por evaluar el rendimiento del modelo utilizando métricas como el R² y el error cuadrático medio (MSE), por otro lado, el método grafico de análisis de correlaciones es una buena alternativa a aplicar en el conjunto de datos de prueba.

**Justificación de la Selección:** Estas estadísticas seleccionadas son las más utilizadas al momento de evaluar una regresión lineal, estas proporcionan una comprensión cuantitativa del rendimiento del modelo al comparar las predicciones con los valores reales.

**5. Predicciones Utilizando el Modelo**

Una vez entrenado y validado el modelo de regresión lineal utilizando el conjunto de datos de entrenamiento y prueba, se procedió a realizar predicciones utilizando el modelo para estimar la cantidad de operarios requeridos en situaciones futuras basadas en la cantidad de líneas y artículos a pickear.

**Interpretación de Resultados**

Se crea el modelo y se ajusta utilizando OLS del cual presenta el siguiente resultado:

Las predicciones generadas por el modelo representan la cantidad estimada de operarios necesarios para el pickeo de materiales en base a las características proporcionadas (cantidad de líneas y artículos a pickear). Estos resultados ofrecen una guía para la asignación óptima de recursos humanos, permitiendo a Logistic Group tomar decisiones más informadas y eficientes en su proceso de operaciones diarias.

Como primer paso de la aplicación de la regresión lineal se divide las variables en dos grupos de dependientes e independientes. La variable Y que es la dependiente esta conformada por la variable NOPER, mientras que las variables independientes X están conformadas por las variables CANTIDAD y LINEAS. En segundo lugar, se divide la data en entrenamiento que contiene el 70% (376 datos) de la data y test que contiene el 30% (162 datos) restante.

Posteriormente se realiza un ajuste al modelo agregando un “intercept” con valor constante “1” que es aplicado a las variables independientes.

Con este ajuste del modelo obtenemos un R-cuadrado igual a 0.563, es decir que el modelo propuesto puede predecir aproximadamente el 56% de las veces con un MSE de 0.302.

Para finalizar se realiza la predicción del NOPER, y se observa que los datos predichos son cercanos a los valores reales, se tiene un error de 2.57 lo que nos indica que la predicción se aleja en 2.57 unidades del valor real. Por lo que los valores predecidos no son exactos, por tanto, se debe realizar ajustes sobre el modelo para mejorar el rendimiento y su confiabilidad, entre las opciones a tomar esta la transformación de variables ya que la escala o valores altos de algunas variables puede tener un efecto negativo en el modelo, por otro lado, se considera que los valores outliers no han tenido un buen tratamiento

**6. Conclusiones del Proyecto**

Durante el desarrollo de este proyecto de predicción de asignación de operarios, se lograron alcanzar varios hitos significativos:

Se identificó y abordó el problema de asignación ineficiente de operarios para el proceso de pickeo en Logistic Group.

Se entrenó y validó un modelo de regresión lineal para predecir la cantidad de operarios requeridos en función de la cantidad de líneas y artículos a pickear.

Se realizaron predicciones exitosas utilizando el modelo entrenado, proporcionando estimaciones confiables de la cantidad de operarios necesarios en diferentes situaciones operativas.

Se evidenció la importancia de la evaluación constante del modelo y la posible iteración hacia modelos más complejos para mejorar la precisión.

**7. ANEXOS**

* LIBRERÍAS UTILIZADAS PARA REALIZAR EL TRABAJO:

import pandas as pd

* LIBRERÍAS GRÁFICAS:

from matplotlib.pyplot import subplots

import seaborn as sns

* OTRAS LIBRERÍAS:

import statsmodels.api as sm

from sklearn.model\_selection import

from sklearn.metrics import mean\_squared\_error

train\_test\_split

* CÓDIGO PARA LA PREPARACION DE LOS DATOS

Primero efectuamos la suma de cantidades y el conteo por fecha, adicionalmente contamos los operarios distintos en esa fecha:

logdf\_aggr = logdf.groupby(by=['FECHAC'],sort=False).agg ({'CANTIDAD':['sum','count'], 'OPER' : ['nunique']})

logdf\_aggr

* CÓDIGO PARA EL ANÁLISIS DE LOS DATOS

PARA CANTIDAD

logdf\_aggr.plot(x='NOPER', y='CANTIDAD', kind='scatter', ax=ax)

PARA LINEAS

logdf\_aggr.plot(x='NOPER', y='LINEAS', kind='scatter', ax=ax)

Efectuamos el análisis de correlación:

corr = logdf\_aggr.corr()

corr

* CÓDIGO PARA CREAR EL MODELO DE REGRESION LINEAL

Creamos el modelo y lo ajustamos utilizando OLS

modelo = sm.OLS(y\_train, x\_train)

resultado = modelo.fit()

resultado.params

Efectuaremos una predicción sobre los valores test y compararemos con la realidad, hay que agregar la constante en x\_test

x\_test = sm.add\_constant(x\_test)

x\_test

Efectuaremos una predicción sobre los datos de test

y\_test\_pred = resultado.predict(x\_test)

y\_test\_pred

**CÓDIGO EN EXTENSO**

import pandas as pd

# importar csv con delimitador pipe y codificacion ansi  
logdf\_orig = pd.read\_csv('LogisticData.txt',delimiter='|',encoding='ansi')

logdf\_orig.shape # --> el df inicialmente tiene 801705 lineas y 40 columnas

(801705, 40)

# vemos las estadísticas de los datos numericos  
logdf\_orig.describe()

GEST NRO LARGO ANCHO ALTO IT \  
count 801705.000000 801705.000000 0.0 0.0 0.0 8.017050e+05   
mean 2022.512303 5504.544041 NaN NaN NaN 1.380488e+06   
std 0.499849 3309.458981 NaN NaN NaN 1.353566e+06   
min 2022.000000 1.000000 NaN NaN NaN 1.000000e+04   
25% 2022.000000 2665.000000 NaN NaN NaN 3.600000e+05   
50% 2023.000000 5378.000000 NaN NaN NaN 9.800000e+05   
75% 2023.000000 8433.000000 NaN NaN NaN 2.000000e+06   
max 2023.000000 11471.000000 NaN NaN NaN 1.333000e+07   
  
 CMAS CINN CANTIDAD BULTOS \  
count 801705.000000 440228.000000 801705.000000 249950.000000   
mean 29.350937 4.928544 16.100984 45.736539   
std 50.995959 6.903372 83.551120 505.610482   
min 0.000000 0.000000 0.000000 0.000000   
25% 10.000000 1.000000 4.000000 6.000000   
50% 12.000000 2.000000 8.000000 11.000000   
75% 24.000000 6.000000 15.000000 20.000000   
max 800.000000 100.000000 48000.000000 23540.000000   
  
 CANMAS CANUNI NUMCAJ CANORIG   
count 243589.000000 244678.000000 197380.000000 615335.000000   
mean 0.328233 8.394862 50.468543 14.088614   
std 0.910521 20.141989 40.426863 45.886587   
min 0.000000 0.000000 1.000000 0.000000   
25% 0.000000 2.000000 19.000000 4.000000   
50% 0.000000 6.000000 43.000000 8.000000   
75% 0.000000 12.000000 73.000000 12.000000   
max 100.000000 2000.000000 346.000000 5000.000000

# vemos las características de cada columna  
logdf\_orig.info()

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 801705 entries, 0 to 801704  
Data columns (total 40 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 REC 801705 non-null object   
 1 INSA 801705 non-null object   
 2 GEST 801705 non-null int64   
 3 NRO 801705 non-null int64   
 4 FECHAC 801705 non-null object   
 5 ESTADO 801705 non-null object   
 6 USUA 801705 non-null object   
 7 FECHAI 796444 non-null object   
 8 OBS 372851 non-null object   
 9 OP 801705 non-null object   
 10 DOCREF1 801491 non-null object   
 11 SOLIC 375768 non-null object   
 12 DOCREF2 250011 non-null object   
 13 TIPSAL 801705 non-null object   
 14 SITIO 801505 non-null object   
 15 APP 615541 non-null object   
 16 FECAPP 801270 non-null object   
 17 LARGO 0 non-null float64  
 18 ANCHO 0 non-null float64  
 19 ALTO 0 non-null float64  
 20 IT 801705 non-null int64   
 21 MUNDO 801705 non-null object   
 22 SKU 801705 non-null object   
 23 BARRA 801705 non-null object   
 24 DESCR 801705 non-null object   
 25 CMAS 801705 non-null int64   
 26 CINN 440228 non-null float64  
 27 CANTIDAD 801705 non-null int64   
 28 ALMA 801705 non-null object   
 29 UBIC 801705 non-null object   
 30 BULTOS 249950 non-null float64  
 31 OPER 790677 non-null object   
 32 OBSOPE 179920 non-null object   
 33 ESTOPE 609024 non-null object   
 34 FECHAE 244681 non-null object   
 35 ESTEMB 197378 non-null object   
 36 CANMAS 243589 non-null float64  
 37 CANUNI 244678 non-null float64  
 38 NUMCAJ 197380 non-null float64  
 39 CANORIG 615335 non-null float64  
dtypes: float64(9), int64(5), object(26)  
memory usage: 244.7+ MB

## LIMPIEZA DE DATA

# analizamos que columnas tiene datos únicos  
for elem in logdf\_orig.columns:  
 print('Columna ', elem,': ', len(logdf\_orig[elem].unique()))

Columna REC : 1  
Columna INSA : 1  
Columna GEST : 2  
Columna NRO : 10043  
Columna FECHAC : 554  
Columna ESTADO : 3  
Columna USUA : 2  
Columna FECHAI : 584  
Columna OBS : 214  
Columna OP : 5  
Columna DOCREF1 : 14578  
Columna SOLIC : 70  
Columna DOCREF2 : 1543  
Columna TIPSAL : 3  
Columna SITIO : 30  
Columna APP : 2  
Columna FECAPP : 560  
Columna LARGO : 1  
Columna ANCHO : 1  
Columna ALTO : 1  
Columna IT : 1331  
Columna MUNDO : 44  
Columna SKU : 15494  
Columna BARRA : 16201  
Columna DESCR : 15358  
Columna CMAS : 95  
Columna CINN : 35  
Columna CANTIDAD : 573  
Columna ALMA : 3  
Columna UBIC : 2267  
Columna BULTOS : 317  
Columna OPER : 37  
Columna OBSOPE : 507  
Columna ESTOPE : 2  
Columna FECHAE : 523  
Columna ESTEMB : 2  
Columna CANMAS : 37  
Columna CANUNI : 164  
Columna NUMCAJ : 347  
Columna CANORIG : 450

# Eliminamos columnas con dato único innecesario para análisis  
logdf = logdf\_orig.drop(columns=['REC','INSA','LARGO','ANCHO','ALTO'])  
# eliminamos columnas de texto que no tienen data relevante para analisis  
logdf.drop(columns=['OBS','OBSOPE','SOLIC','DOCREF1','DOCREF2','NRO','IT'],axis=1, inplace=True)

logdf.head(5).T

0 \  
GEST 2022   
FECHAC 03/01/2022   
ESTADO APR   
USUA HOGARMARKET   
FECHAI 05/01/2022   
OP EC   
TIPSAL BOD   
SITIO LP01   
APP USE   
FECAPP 03/01/2022   
MUNDO CASABAÑO   
SKU 3203836000528   
BARRA 3203836000528   
DESCR TOALLA SABANA CONCEPTO SPA, Marron, 0006   
CMAS 16   
CINN 4.0   
CANTIDAD 4   
ALMA CI   
UBIC FRB24   
BULTOS 13.0   
OPER JTANCARA   
ESTOPE PIC   
FECHAE 03/01/2022   
ESTEMB EMB   
CANMAS 0.0   
CANUNI 4.0   
NUMCAJ 8.0   
CANORIG 4.0   
  
 1 \  
GEST 2022   
FECHAC 03/01/2022   
ESTADO APR   
USUA HOGARMARKET   
FECHAI 05/01/2022   
OP EC   
TIPSAL BOD   
SITIO LP01   
APP USE   
FECAPP 03/01/2022   
MUNDO CASABAÑO   
SKU 3203836000542   
BARRA 3203836000542   
DESCR TOALLA SABANA CONCEPTO S, Petroleo, 0006   
CMAS 16   
CINN 4.0   
CANTIDAD 8   
ALMA CI   
UBIC FRB20   
BULTOS 19.0   
OPER JTANCARA   
ESTOPE PIC   
FECHAE 03/01/2022   
ESTEMB EMB   
CANMAS 0.0   
CANUNI 8.0   
NUMCAJ 24.0   
CANORIG 8.0   
  
 2 \  
GEST 2022   
FECHAC 03/01/2022   
ESTADO APR   
USUA HOGARMARKET   
FECHAI 05/01/2022   
OP EC   
TIPSAL BOD   
SITIO LP01   
APP USE   
FECAPP 03/01/2022   
MUNDO CASABAÑO   
SKU 3220679000171   
BARRA 3220679000171   
DESCR ALFOMBRA DE BAÑO MF CONCEP, Blanco, 0001   
CMAS 24   
CINN 6.0   
CANTIDAD 8   
ALMA CI   
UBIC ERA26   
BULTOS 20.0   
OPER AVILLCA   
ESTOPE PIC   
FECHAE 03/01/2022   
ESTEMB EMB   
CANMAS 0.0   
CANUNI 8.0   
NUMCAJ 19.0   
CANORIG 8.0   
  
 3 \  
GEST 2022   
FECHAC 03/01/2022   
ESTADO APR   
USUA HOGARMARKET   
FECHAI 05/01/2022   
OP EC   
TIPSAL BOD   
SITIO LP01   
APP USE   
FECAPP 03/01/2022   
MUNDO CASABAÑO   
SKU 3220679000188   
BARRA 3220679000188   
DESCR ALFOMBRA DE BAÑO MF CONC, Palorosa, 0001   
CMAS 24   
CINN 6.0   
CANTIDAD 8   
ALMA CI   
UBIC ERA25   
BULTOS 18.0   
OPER AVILLCA   
ESTOPE PIC   
FECHAE 03/01/2022   
ESTEMB EMB   
CANMAS 0.0   
CANUNI 8.0   
NUMCAJ 19.0   
CANORIG 8.0   
  
 4   
GEST 2022   
FECHAC 03/01/2022   
ESTADO APR   
USUA HOGARMARKET   
FECHAI 05/01/2022   
OP EC   
TIPSAL BOD   
SITIO LP01   
APP USE   
FECAPP 03/01/2022   
MUNDO CASABAÑO   
SKU 3220679000195   
BARRA 3220679000195   
DESCR ALFOMBRA DE BAÑO MF CONCEP, Marron, 0001   
CMAS 24   
CINN 6.0   
CANTIDAD 4   
ALMA CI   
UBIC FRA15   
BULTOS 12.0   
OPER AVILLCA   
ESTOPE PIC   
FECHAE 03/01/2022   
ESTEMB EMB   
CANMAS 0.0   
CANUNI 4.0   
NUMCAJ 39.0   
CANORIG 4.0

logdf['ESTADO'].unique() # --> filtramos por estado 'APR', los otros estados son intermedios o anulados

array(['APR', 'ANU', 'PAP'], dtype=object)

logdf.shape

(801705, 28)

filtroAPR = logdf['ESTADO'] == 'APR'  
  
logdf = logdf[filtroAPR]

# eliminamos columna ESTADO, no nos interesa columnas de fecha adicionales  
logdf.drop(['ESTADO','FECHAI','FECAPP','FECHAE'], axis=1, inplace=True)

logdf.head(5).T

0 \  
GEST 2022   
FECHAC 03/01/2022   
USUA HOGARMARKET   
OP EC   
TIPSAL BOD   
SITIO LP01   
APP USE   
MUNDO CASABAÑO   
SKU 3203836000528   
BARRA 3203836000528   
DESCR TOALLA SABANA CONCEPTO SPA, Marron, 0006   
CMAS 16   
CINN 4.0   
CANTIDAD 4   
ALMA CI   
UBIC FRB24   
BULTOS 13.0   
OPER JTANCARA   
ESTOPE PIC   
ESTEMB EMB   
CANMAS 0.0   
CANUNI 4.0   
NUMCAJ 8.0   
CANORIG 4.0   
  
 1 \  
GEST 2022   
FECHAC 03/01/2022   
USUA HOGARMARKET   
OP EC   
TIPSAL BOD   
SITIO LP01   
APP USE   
MUNDO CASABAÑO   
SKU 3203836000542   
BARRA 3203836000542   
DESCR TOALLA SABANA CONCEPTO S, Petroleo, 0006   
CMAS 16   
CINN 4.0   
CANTIDAD 8   
ALMA CI   
UBIC FRB20   
BULTOS 19.0   
OPER JTANCARA   
ESTOPE PIC   
ESTEMB EMB   
CANMAS 0.0   
CANUNI 8.0   
NUMCAJ 24.0   
CANORIG 8.0   
  
 2 \  
GEST 2022   
FECHAC 03/01/2022   
USUA HOGARMARKET   
OP EC   
TIPSAL BOD   
SITIO LP01   
APP USE   
MUNDO CASABAÑO   
SKU 3220679000171   
BARRA 3220679000171   
DESCR ALFOMBRA DE BAÑO MF CONCEP, Blanco, 0001   
CMAS 24   
CINN 6.0   
CANTIDAD 8   
ALMA CI   
UBIC ERA26   
BULTOS 20.0   
OPER AVILLCA   
ESTOPE PIC   
ESTEMB EMB   
CANMAS 0.0   
CANUNI 8.0   
NUMCAJ 19.0   
CANORIG 8.0   
  
 3 \  
GEST 2022   
FECHAC 03/01/2022   
USUA HOGARMARKET   
OP EC   
TIPSAL BOD   
SITIO LP01   
APP USE   
MUNDO CASABAÑO   
SKU 3220679000188   
BARRA 3220679000188   
DESCR ALFOMBRA DE BAÑO MF CONC, Palorosa, 0001   
CMAS 24   
CINN 6.0   
CANTIDAD 8   
ALMA CI   
UBIC ERA25   
BULTOS 18.0   
OPER AVILLCA   
ESTOPE PIC   
ESTEMB EMB   
CANMAS 0.0   
CANUNI 8.0   
NUMCAJ 19.0   
CANORIG 8.0   
  
 4   
GEST 2022   
FECHAC 03/01/2022   
USUA HOGARMARKET   
OP EC   
TIPSAL BOD   
SITIO LP01   
APP USE   
MUNDO CASABAÑO   
SKU 3220679000195   
BARRA 3220679000195   
DESCR ALFOMBRA DE BAÑO MF CONCEP, Marron, 0001   
CMAS 24   
CINN 6.0   
CANTIDAD 4   
ALMA CI   
UBIC FRA15   
BULTOS 12.0   
OPER AVILLCA   
ESTOPE PIC   
ESTEMB EMB   
CANMAS 0.0   
CANUNI 4.0   
NUMCAJ 39.0   
CANORIG 4.0

# analizamos las coolumnas de operadores, la primera 'OP' no es relevante, la segunda 'OPER' nos da una mejor idea  
logdf['OP'].unique()

array(['EC', 'RR', 'CJ', 'HP', 'FL'], dtype=object)

logdf['OPER'].unique()

array(['JTANCARA', 'AVILLCA', 'RRODRIGUEZ', nan, 'ACACHACA', 'CROJAS',  
 'GCONDORI', 'ROLANDOS', 'RLIMACHI', 'MTICONA', 'FLOPEZ', 'MMARCA',  
 'HPAYE', 'JCORIA', 'ELOY', 'JPEREYRA', 'VARUCUTIPA', 'MMONTERO',  
 'JVALENCIA', 'FVALENCIA', 'FARUCUTIPA', 'IVALENCIA', 'DCACHACA',  
 'ECORIA', 'KGONZALES', 'OVALERIANO', 'RCOLQUEHUANCA', 'JRAMOS',  
 'RRAMOS', 'JCALLE', 'MMIRANDA', 'CALDERON', 'ENZO', 'RCACHACA',  
 'PVASQUEZ'], dtype=object)

# eliminamos la columna 'OP'  
logdf.drop(['OP'], axis=1, inplace=True)

# analizamos tipo de salida, el dato SOB nos interesa, son solo 49 registros, lo filtramos  
logdf['TIPSAL'].unique()

array(['BOD', 'SIT', 'SOB'], dtype=object)

filtroSOB = logdf['TIPSAL'] == 'SOB'  
  
len(logdf[filtroSOB])

49

filtroSOB = logdf['TIPSAL'] != 'SOB'  
  
logdf = logdf[filtroSOB]  
  
logdf.shape

(796137, 23)

# eliminamos columna APP que solo indica si se uso la app movil  
logdf['APP'].unique()

array(['USE', nan], dtype=object)

logdf.drop(['APP'], axis=1, inplace=True)  
logdf.shape

(796137, 22)

# analizamos las cantidades, CMAS, CINN, CANMAS, CANUNI, BULTOS, NUMCAJ, CANORIG, son datos operativos referidos a las cantidades por caja y a lo efectivamente pickeado  
# unicamente nos interesa el dato CANTIDAD que es lo solicitado realmente por cada producto  
logdf.drop(['CMAS','CINN','CANMAS','CANUNI','BULTOS','NUMCAJ','CANORIG'], axis=1, inplace=True)  
logdf.shape

(796137, 15)

logdf.head(5).T

0 \  
GEST 2022   
FECHAC 03/01/2022   
USUA HOGARMARKET   
TIPSAL BOD   
SITIO LP01   
MUNDO CASABAÑO   
SKU 3203836000528   
BARRA 3203836000528   
DESCR TOALLA SABANA CONCEPTO SPA, Marron, 0006   
CANTIDAD 4   
ALMA CI   
UBIC FRB24   
OPER JTANCARA   
ESTOPE PIC   
ESTEMB EMB   
  
 1 \  
GEST 2022   
FECHAC 03/01/2022   
USUA HOGARMARKET   
TIPSAL BOD   
SITIO LP01   
MUNDO CASABAÑO   
SKU 3203836000542   
BARRA 3203836000542   
DESCR TOALLA SABANA CONCEPTO S, Petroleo, 0006   
CANTIDAD 8   
ALMA CI   
UBIC FRB20   
OPER JTANCARA   
ESTOPE PIC   
ESTEMB EMB   
  
 2 \  
GEST 2022   
FECHAC 03/01/2022   
USUA HOGARMARKET   
TIPSAL BOD   
SITIO LP01   
MUNDO CASABAÑO   
SKU 3220679000171   
BARRA 3220679000171   
DESCR ALFOMBRA DE BAÑO MF CONCEP, Blanco, 0001   
CANTIDAD 8   
ALMA CI   
UBIC ERA26   
OPER AVILLCA   
ESTOPE PIC   
ESTEMB EMB   
  
 3 \  
GEST 2022   
FECHAC 03/01/2022   
USUA HOGARMARKET   
TIPSAL BOD   
SITIO LP01   
MUNDO CASABAÑO   
SKU 3220679000188   
BARRA 3220679000188   
DESCR ALFOMBRA DE BAÑO MF CONC, Palorosa, 0001   
CANTIDAD 8   
ALMA CI   
UBIC ERA25   
OPER AVILLCA   
ESTOPE PIC   
ESTEMB EMB   
  
 4   
GEST 2022   
FECHAC 03/01/2022   
USUA HOGARMARKET   
TIPSAL BOD   
SITIO LP01   
MUNDO CASABAÑO   
SKU 3220679000195   
BARRA 3220679000195   
DESCR ALFOMBRA DE BAÑO MF CONCEP, Marron, 0001   
CANTIDAD 4   
ALMA CI   
UBIC FRA15   
OPER AVILLCA   
ESTOPE PIC   
ESTEMB EMB

# tras nueva revisión observamos que TIPSAL no es necesario ya tenemos los sitios tiendas en SIT, eliminamos TIPSAL  
logdf.drop(['TIPSAL'], axis=1, inplace=True)  
logdf.shape

(796137, 14)

# analizamos ALMA y UBIC  
logdf['ALMA'].unique()

array(['CI', 'KE', 'PN'], dtype=object)

len(logdf['UBIC'].unique())

2266

# eliminamos alma y ubic que se refieren a las ubicaciones y almacenes de donde provienen los productos  
logdf.drop(['ALMA','UBIC'], axis=1, inplace=True)  
logdf.shape

(796137, 12)

# analizamos ESTOPE y ESTEMB  
logdf['ESTOPE'].unique()

array(['PIC', nan], dtype=object)

logdf['ESTEMB'].unique()

array(['EMB', nan], dtype=object)

# los eliminamos porque son resultado del trabajo de piqueo y patra nuestro análisis no tiene relevancia  
logdf.drop(['ESTOPE','ESTEMB'], axis=1, inplace=True)  
logdf.shape

(796137, 10)

logdf.head(5).T

0 \  
GEST 2022   
FECHAC 03/01/2022   
USUA HOGARMARKET   
SITIO LP01   
MUNDO CASABAÑO   
SKU 3203836000528   
BARRA 3203836000528   
DESCR TOALLA SABANA CONCEPTO SPA, Marron, 0006   
CANTIDAD 4   
OPER JTANCARA   
  
 1 \  
GEST 2022   
FECHAC 03/01/2022   
USUA HOGARMARKET   
SITIO LP01   
MUNDO CASABAÑO   
SKU 3203836000542   
BARRA 3203836000542   
DESCR TOALLA SABANA CONCEPTO S, Petroleo, 0006   
CANTIDAD 8   
OPER JTANCARA   
  
 2 \  
GEST 2022   
FECHAC 03/01/2022   
USUA HOGARMARKET   
SITIO LP01   
MUNDO CASABAÑO   
SKU 3220679000171   
BARRA 3220679000171   
DESCR ALFOMBRA DE BAÑO MF CONCEP, Blanco, 0001   
CANTIDAD 8   
OPER AVILLCA   
  
 3 \  
GEST 2022   
FECHAC 03/01/2022   
USUA HOGARMARKET   
SITIO LP01   
MUNDO CASABAÑO   
SKU 3220679000188   
BARRA 3220679000188   
DESCR ALFOMBRA DE BAÑO MF CONC, Palorosa, 0001   
CANTIDAD 8   
OPER AVILLCA   
  
 4   
GEST 2022   
FECHAC 03/01/2022   
USUA HOGARMARKET   
SITIO LP01   
MUNDO CASABAÑO   
SKU 3220679000195   
BARRA 3220679000195   
DESCR ALFOMBRA DE BAÑO MF CONCEP, Marron, 0001   
CANTIDAD 4   
OPER AVILLCA

# analizamos SKU y cod\_barra  
print('Nro SKU: ',len(logdf['SKU'].unique()))  
print("Nro barcode: ",len(logdf['BARRA'].unique()))

Nro SKU: 15494  
Nro barcode: 16201

# eliminamos barcode, bajo la premisa que los SKU son únicos  
logdf.drop(['BARRA'], axis=1, inplace=True)  
logdf.shape

(796137, 9)

# hacemos analisis de nulos:  
logdf.info()

<class 'pandas.core.frame.DataFrame'>  
Index: 796137 entries, 0 to 801677  
Data columns (total 9 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 GEST 796137 non-null int64   
 1 FECHAC 796137 non-null object  
 2 USUA 796137 non-null object  
 3 SITIO 795986 non-null object  
 4 MUNDO 796137 non-null object  
 5 SKU 796137 non-null object  
 6 DESCR 796137 non-null object  
 7 CANTIDAD 796137 non-null int64   
 8 OPER 785595 non-null object  
dtypes: int64(2), object(7)  
memory usage: 60.7+ MB

# existen OPER nulos, por lo que los eliminamos  
logdf.dropna(inplace=True)  
logdf.shape

(785589, 9)

# creamos un dataframe que contenga la relación única SKU/DESCR para posteriores cruces  
SKUdf = logdf.drop(['GEST','FECHAC','USUA','SITIO','MUNDO','CANTIDAD','OPER'],axis=1)  
SKUdf

SKU DESCR  
0 3203836000528 TOALLA SABANA CONCEPTO SPA, Marron, 0006  
1 3203836000542 TOALLA SABANA CONCEPTO S, Petroleo, 0006  
2 3220679000171 ALFOMBRA DE BAÑO MF CONCEP, Blanco, 0001  
3 3220679000188 ALFOMBRA DE BAÑO MF CONC, Palorosa, 0001  
4 3220679000195 ALFOMBRA DE BAÑO MF CONCEP, Marron, 0001  
... ... ...  
801673 10020497 HILO DENTAL COLGATE TOTAL 25MT  
801674 10018834 TOALLA HUM PANOLINI ALOE 20UN  
801675 10020319 PASTA DENTAL COLGATE TRIPLE ACCION 90GR  
801676 10020389 PASTA DENTAL COLGATE TRIPLE ACCION 180GR  
801677 10020907 AFEITADORA BIC CONFORT3 PIEL SENSIB 1UN  
  
[785589 rows x 2 columns]

# elimina los duplicados del dataframne SKU dejando un único registro por SKU  
SKUdf.drop\_duplicates(inplace=True,subset=['SKU'])  
len(SKUdf)

15432

len(SKUdf['SKU'].unique())

15432

# haboiendo guardado en otro df las descripciones de los SKU eliminamos la columna DESCR  
logdf.drop(['DESCR'], axis=1, inplace=True)  
logdf.shape

(785589, 8)

logdf.head(5)

GEST FECHAC USUA SITIO MUNDO SKU CANTIDAD \  
0 2022 03/01/2022 HOGARMARKET LP01 CASABAÑO 3203836000528 4   
1 2022 03/01/2022 HOGARMARKET LP01 CASABAÑO 3203836000542 8   
2 2022 03/01/2022 HOGARMARKET LP01 CASABAÑO 3220679000171 8   
3 2022 03/01/2022 HOGARMARKET LP01 CASABAÑO 3220679000188 8   
4 2022 03/01/2022 HOGARMARKET LP01 CASABAÑO 3220679000195 4   
  
 OPER   
0 JTANCARA   
1 JTANCARA   
2 AVILLCA   
3 AVILLCA   
4 AVILLCA

logdf.info()

<class 'pandas.core.frame.DataFrame'>  
Index: 785589 entries, 0 to 801677  
Data columns (total 8 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 GEST 785589 non-null int64   
 1 FECHAC 785589 non-null object  
 2 USUA 785589 non-null object  
 3 SITIO 785589 non-null object  
 4 MUNDO 785589 non-null object  
 5 SKU 785589 non-null object  
 6 CANTIDAD 785589 non-null int64   
 7 OPER 785589 non-null object  
dtypes: int64(2), object(6)  
memory usage: 53.9+ MB

# convertimos GEST en string  
logdf['GEST'] = logdf['GEST'].astype(str)

# este es nuestro dataset definitivo con 785589 líneas y 8 atributos cada uno de ellos con valores necesarios para el análisis   
for elem in logdf.columns:  
 print('Columna ', elem,': ', len(logdf[elem].unique()))

Columna GEST : 2  
Columna FECHAC : 538  
Columna USUA : 2  
Columna SITIO : 26  
Columna MUNDO : 42  
Columna SKU : 15432  
Columna CANTIDAD : 526  
Columna OPER : 34

## PREPARACION DE LOS DATOS

# queremos analizar la cantidad de operarios que son necesarios por fecha enfuncion de las lineas y cantidades que atendieron  
# será necesario hacer dos agregaciones, lineas por día, operarios diferentes por día y canttidades por día  
  
  
# primero efctuamos la suma de cantidades y el conteo por fecha, adicionalmente contamos los operarios distintos en esa fecha  
logdf\_aggr = logdf.groupby(by=['FECHAC'],sort=False).agg({'CANTIDAD' : ['sum','count'], 'OPER' : ['nunique']})  
logdf\_aggr

CANTIDAD OPER  
 sum count nunique  
FECHAC   
03/01/2022 21846 1657 11  
04/01/2022 33897 1819 12  
05/01/2022 36440 1540 11  
06/01/2022 41940 1743 11  
07/01/2022 3320 84 4  
... ... ... ...  
14/12/2023 25483 1467 14  
15/12/2023 6341 472 8  
16/12/2023 4053 132 5  
18/12/2023 35965 2706 14  
19/12/2023 9193 320 8  
  
[538 rows x 3 columns]

logdf\_aggr.info()

<class 'pandas.core.frame.DataFrame'>  
Index: 538 entries, 03/01/2022 to 19/12/2023  
Data columns (total 3 columns):  
 # Column Non-Null Count Dtype  
--- ------ -------------- -----  
 0 (CANTIDAD, sum) 538 non-null int64  
 1 (CANTIDAD, count) 538 non-null int64  
 2 (OPER, nunique) 538 non-null int64  
dtypes: int64(3)  
memory usage: 16.8+ KB

# renombramos atributos para una mejor interpretacion:  
logdf\_aggr.columns = ['CANTIDAD','LINEAS','NOPER']  
logdf\_aggr.info()

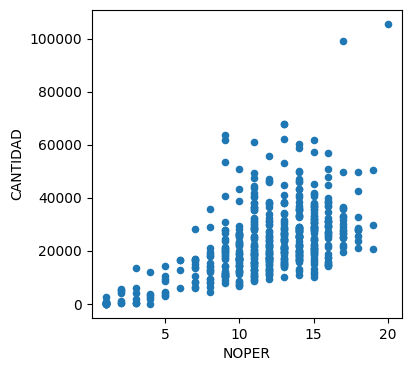
<class 'pandas.core.frame.DataFrame'>  
Index: 538 entries, 03/01/2022 to 19/12/2023  
Data columns (total 3 columns):  
 # Column Non-Null Count Dtype  
--- ------ -------------- -----  
 0 CANTIDAD 538 non-null int64  
 1 LINEAS 538 non-null int64  
 2 NOPER 538 non-null int64  
dtypes: int64(3)  
memory usage: 16.8+ KB

## ANALISIS DE LOS DATOS

# importamos las librerias gráficas  
from matplotlib.pyplot import subplots  
import seaborn as sns

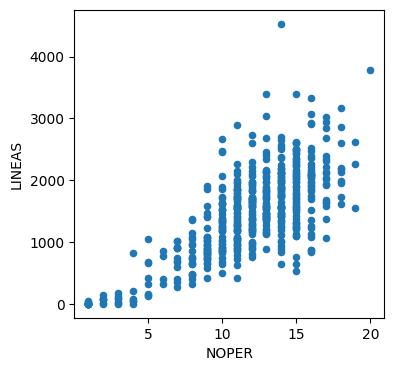
# efectuamos la representacion grafica de los datos respecto a nuestra variable objetivo:  
fig, ax = subplots(figsize=(4,4))  
  
# para CANTIDAD  
logdf\_aggr.plot(x='NOPER', y='CANTIDAD', kind='scatter', ax=ax)

<Axes: xlabel='NOPER', ylabel='CANTIDAD'>



# para LINEAS  
fig, ax = subplots(figsize=(4,4))  
logdf\_aggr.plot(x='NOPER', y='LINEAS', kind='scatter', ax=ax)

<Axes: xlabel='NOPER', ylabel='LINEAS'>

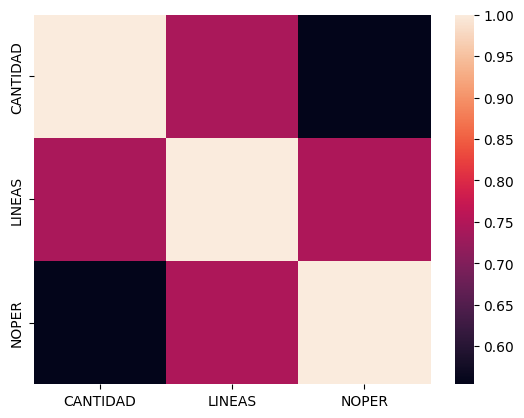


# efectuamos el análisis de correlación:  
corr = logdf\_aggr.corr()  
corr

CANTIDAD LINEAS NOPER  
CANTIDAD 1.000000 0.741878 0.553824  
LINEAS 0.741878 1.000000 0.744760  
NOPER 0.553824 0.744760 1.000000

# graficamos la correlacion mediante heatmap  
sns.heatmap(corr)

<Axes: >



### Observamos para nuestra variable objetivo NOPER que existe correlacion positiva tanto para LINEAS (0.74) y CANTIDAD (0.55)

### Entonces existe una relación lineal

# REGRESIÓN LINEAL

!pip install statsmodels

Requirement already satisfied: statsmodels in c:\users\gusta\appdata\local\packages\pythonsoftwarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (0.14.1)  
Requirement already satisfied: numpy<2,>=1.18 in c:\users\gusta\appdata\local\packages\pythonsoftwarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from statsmodels) (1.26.0)  
Requirement already satisfied: scipy!=1.9.2,>=1.4 in c:\users\gusta\appdata\local\packages\pythonsoftwarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from statsmodels) (1.11.4)  
Requirement already satisfied: pandas!=2.1.0,>=1.0 in c:\users\gusta\appdata\local\packages\pythonsoftwarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from statsmodels) (2.1.1)  
Requirement already satisfied: patsy>=0.5.4 in c:\users\gusta\appdata\local\packages\pythonsoftwarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from statsmodels) (0.5.4)  
Requirement already satisfied: packaging>=21.3 in c:\users\gusta\appdata\local\packages\pythonsoftwarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from statsmodels) (23.2)  
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\gusta\appdata\local\packages\pythonsoftwarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from pandas!=2.1.0,>=1.0->statsmodels) (2.8.2)  
Requirement already satisfied: pytz>=2020.1 in c:\users\gusta\appdata\local\packages\pythonsoftwarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from pandas!=2.1.0,>=1.0->statsmodels) (2023.3.post1)  
Requirement already satisfied: tzdata>=2022.1 in c:\users\gusta\appdata\local\packages\pythonsoftwarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from pandas!=2.1.0,>=1.0->statsmodels) (2023.3)  
Requirement already satisfied: six in c:\users\gusta\appdata\local\packages\pythonsoftwarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from patsy>=0.5.4->statsmodels) (1.16.0)

!pip install scikit-learn

Requirement already satisfied: scikit-learn in c:\users\gusta\appdata\local\packages\pythonsoftwarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (1.3.2)  
Requirement already satisfied: numpy<2.0,>=1.17.3 in c:\users\gusta\appdata\local\packages\pythonsoftwarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from scikit-learn) (1.26.0)  
Requirement already satisfied: scipy>=1.5.0 in c:\users\gusta\appdata\local\packages\pythonsoftwarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from scikit-learn) (1.11.4)  
Requirement already satisfied: joblib>=1.1.1 in c:\users\gusta\appdata\local\packages\pythonsoftwarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from scikit-learn) (1.3.2)  
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\gusta\appdata\local\packages\pythonsoftwarefoundation.python.3.11\_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from scikit-learn) (3.2.0)

# importamos las librerias necesarias  
import statsmodels.api as sm  
from sklearn.model\_selection import train\_test\_split

# dividimos nuestras variables dependientes e independientes  
Y = logdf\_aggr['NOPER']  
X = logdf\_aggr[['CANTIDAD','LINEAS']]

# efectuamos el split con 70% de entrenamiento  
x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, Y, train\_size=0.7)

# veamos cuantos tiene cada uno:  
print("X\_train:", len(x\_train))  
print("X\_test:", len(x\_test))  
print("Y\_train:", len(y\_train))  
print("Y\_test:", len(y\_test))

X\_train: 376  
X\_test: 162  
Y\_train: 376  
Y\_test: 162

x\_train

CANTIDAD LINEAS  
FECHAC   
23/02/2023 37931 3035  
24/02/2022 18527 910  
13/01/2022 17392 1260  
06/12/2022 35724 2646  
26/01/2022 18685 1323  
... ... ...  
30/10/2023 28219 2115  
29/08/2023 25700 1563  
23/12/2022 23187 1348  
18/04/2023 27254 1869  
15/02/2022 26138 1723  
  
[376 rows x 2 columns]

## CON STATSMODEL

# agregamos un intercept con 1 a las variables independientes  
x\_train = sm.add\_constant(x\_train)  
x\_train

const CANTIDAD LINEAS  
FECHAC   
23/02/2023 1.0 37931 3035  
24/02/2022 1.0 18527 910  
13/01/2022 1.0 17392 1260  
06/12/2022 1.0 35724 2646  
26/01/2022 1.0 18685 1323  
... ... ... ...  
30/10/2023 1.0 28219 2115  
29/08/2023 1.0 25700 1563  
23/12/2022 1.0 23187 1348  
18/04/2023 1.0 27254 1869  
15/02/2022 1.0 26138 1723  
  
[376 rows x 3 columns]

# creamos el modelo y lo ajustamos utilizando OLS  
modelo = sm.OLS(y\_train, x\_train)  
resultado = modelo.fit()  
resultado.params

const 5.923622  
CANTIDAD 0.000002  
LINEAS 0.003934  
dtype: float64

print(resultado.summary())

OLS Regression Results   
==============================================================================  
Dep. Variable: NOPER R-squared: 0.560  
Model: OLS Adj. R-squared: 0.558  
Method: Least Squares F-statistic: 237.3  
Date: Tue, 26 Dec 2023 Prob (F-statistic): 3.30e-67  
Time: 19:47:41 Log-Likelihood: -888.32  
No. Observations: 376 AIC: 1783.  
Df Residuals: 373 BIC: 1794.  
Df Model: 2   
Covariance Type: nonrobust   
==============================================================================  
 coef std err t P>|t| [0.025 0.975]  
------------------------------------------------------------------------------  
const 5.9236 0.302 19.629 0.000 5.330 6.517  
CANTIDAD 2.443e-06 1.47e-05 0.166 0.868 -2.64e-05 3.13e-05  
LINEAS 0.0039 0.000 13.951 0.000 0.003 0.004  
==============================================================================  
Omnibus: 3.882 Durbin-Watson: 1.898  
Prob(Omnibus): 0.144 Jarque-Bera (JB): 3.880  
Skew: -0.158 Prob(JB): 0.144  
Kurtosis: 3.384 Cond. No. 6.26e+04  
==============================================================================  
  
Notes:  
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.  
[2] The condition number is large, 6.26e+04. This might indicate that there are  
strong multicollinearity or other numerical problems.

### El valor R-squared es de 56.3%, es decir que el modelo puede predecir el 57% de las veces con aproximación

# efectuaremos una prediccion sobre los valores test y compararemos con al realidad, hayq ue agregar la constante en x\_test  
x\_test = sm.add\_constant(x\_test)  
x\_test

const CANTIDAD LINEAS  
FECHAC   
13/10/2022 1.0 13578 990  
11/02/2023 1.0 2504 44  
02/09/2022 1.0 14072 732  
15/09/2023 1.0 10572 859  
25/05/2022 1.0 20698 1552  
... ... ... ...  
21/01/2022 1.0 36300 1473  
09/07/2022 1.0 16841 351  
08/10/2022 1.0 1660 41  
31/10/2023 1.0 19835 1772  
27/05/2022 1.0 13012 640  
  
[162 rows x 3 columns]

# efectuaremos una prediccion sobre los datos de test  
y\_test\_pred = resultado.predict(x\_test)  
y\_test\_pred

FECHAC  
13/10/2022 9.851656  
11/02/2023 6.102844  
02/09/2022 8.837838  
15/09/2023 9.328932  
25/05/2022 12.080074  
 ...   
21/01/2022 11.807385  
09/07/2022 7.345669  
08/10/2022 6.088979  
31/10/2023 12.943492  
27/05/2022 8.473301  
Length: 162, dtype: float64

# unimos el test con la predicción  
comparacion = x\_test.merge(y\_test, left\_index=True, right\_index=True)  
comparacion = comparacion.merge(pd.DataFrame(y\_test\_pred), left\_index=True, right\_index=True)  
comparacion = comparacion.drop(columns=['const'])  
comparacion.columns = ['CANTIDAD','LINEAS','NOPER\_REAL','NOPER\_PRED']  
comparacion['NOPER\_PRED'] = round(comparacion['NOPER\_PRED'],0)  
comparacion['NOPER\_PRED'] = comparacion['NOPER\_PRED'].astype('int')  
comparacion

CANTIDAD LINEAS NOPER\_REAL NOPER\_PRED  
FECHAC   
13/10/2022 13578 990 12 10  
11/02/2023 2504 44 4 6  
02/09/2022 14072 732 8 9  
15/09/2023 10572 859 9 9  
25/05/2022 20698 1552 19 12  
... ... ... ... ...  
21/01/2022 36300 1473 11 12  
09/07/2022 16841 351 7 7  
08/10/2022 1660 41 4 6  
31/10/2023 19835 1772 14 13  
27/05/2022 13012 640 15 8  
  
[162 rows x 4 columns]

resultado.predict([[1,30938,1969]])

array([13.7456538])

# Observamos que en efecto el modelo si bien se acerca no predice muy bien

# --> entonces se deben efectuar ajustes al modelo, quizas tratamiento de outliers

# encontramos que el error estiamdo por el métodod   
from sklearn.metrics import mean\_squared\_error  
rmse\_rm = mean\_squared\_error(y\_true = y\_test, y\_pred = y\_test\_pred, squared = False)  
print(f"El error (rmse) de test es: {rmse\_rm}")

El error (rmse) de test es: 2.410457520400003

## El error de 2.57 significa que la predicción se aleja en promedio en 2.57 del valor real

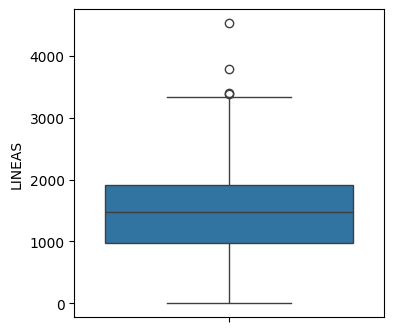
# CORRECION AL MODELO

logdf\_aggr.describe()

CANTIDAD LINEAS NOPER  
count 538.000000 538.000000 538.000000  
mean 23343.416357 1460.202602 11.765799  
std 13396.219207 706.435874 3.782861  
min 2.000000 1.000000 1.000000  
25% 14988.750000 978.750000 10.000000  
50% 21475.000000 1476.500000 12.000000  
75% 29363.500000 1920.500000 14.000000  
max 105588.000000 4533.000000 20.000000

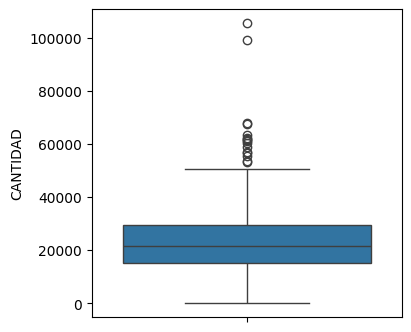
# Efectuaremos la limpieza de outliers para ver si incide en en los resultados  
# efectuando los diagramas de caja de nuestra dos variables independientes  
import seaborn as sns  
  
fig, ax = subplots(figsize=(4,4))  
  
sns.boxplot(logdf\_aggr['LINEAS'])

<Axes: ylabel='LINEAS'>



fig, ax = subplots(figsize=(4,4))  
sns.boxplot(logdf\_aggr['CANTIDAD'])

<Axes: ylabel='CANTIDAD'>



logdf\_aggr.shape # --> existen 538 registros antes de la limpieza de outliers

(538, 3)

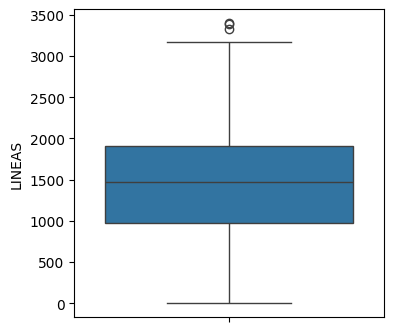
# Efectuamos una reducción de outliers utilizando el método de puntuacoióm Z-score  
import numpy as np  
import scipy.stats as stats  
  
# Encontramos el valor z-score de cada observación  
z = np.abs(stats.zscore(logdf\_aggr))  
# Solo mantenemos las filas con puntuación z menor que el valor absoluto de 3   
logdf\_clean = logdf\_aggr[(z<3).all(axis=1)]  
# Calculamos cuantas filas quedan  
logdf\_clean.shape

(532, 3)

### se eliminaron 6 registros de valore atípicos tras la limpieza

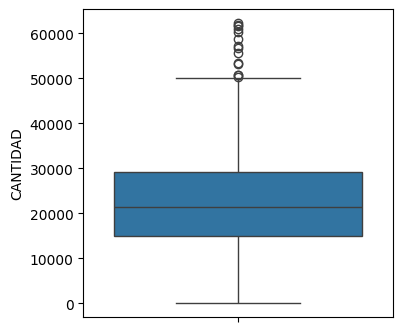
# vemos cómo quedan nuestros boxplots tras la reducción  
  
fig, ax = subplots(figsize=(4,4))  
  
sns.boxplot(logdf\_clean['LINEAS'])

<Axes: ylabel='LINEAS'>



fig, ax = subplots(figsize=(4,4))  
sns.boxplot(logdf\_clean['CANTIDAD'])

<Axes: ylabel='CANTIDAD'>



## REGRESION NUEVA

# dividimos nuestras variables dependientes e independientes en el df limpio  
Y = logdf\_clean['NOPER']  
X = logdf\_clean[['CANTIDAD','LINEAS']]

# efectuamos el split con 70% de entrenamiento  
x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, Y, train\_size=0.7)

# agregamos un intercept con 1 a las variables independientes  
x\_train = sm.add\_constant(x\_train)  
x\_train

const CANTIDAD LINEAS  
FECHAC   
30/05/2022 1.0 34062 2039  
17/09/2022 1.0 20 3  
19/05/2023 1.0 11518 762  
15/07/2023 1.0 3187 136  
23/05/2022 1.0 15116 1350  
... ... ... ...  
06/07/2023 1.0 10408 1028  
02/04/2022 1.0 5130 82  
19/12/2022 1.0 41055 1854  
25/04/2023 1.0 24391 2075  
19/05/2022 1.0 28337 1620  
  
[372 rows x 3 columns]

# creamos el modelo y lo ajustamos utilizando OLS  
modelo = sm.OLS(y\_train, x\_train)  
resultado = modelo.fit()  
resultado.params

const 5.527163  
CANTIDAD 0.000009  
LINEAS 0.004152  
dtype: float64

print(resultado.summary())

OLS Regression Results   
==============================================================================  
Dep. Variable: NOPER R-squared: 0.581  
Model: OLS Adj. R-squared: 0.579  
Method: Least Squares F-statistic: 255.8  
Date: Tue, 26 Dec 2023 Prob (F-statistic): 1.98e-70  
Time: 20:32:03 Log-Likelihood: -869.52  
No. Observations: 372 AIC: 1745.  
Df Residuals: 369 BIC: 1757.  
Df Model: 2   
Covariance Type: nonrobust   
==============================================================================  
 coef std err t P>|t| [0.025 0.975]  
------------------------------------------------------------------------------  
const 5.5272 0.305 18.135 0.000 4.928 6.126  
CANTIDAD 8.521e-06 1.68e-05 0.508 0.612 -2.45e-05 4.15e-05  
LINEAS 0.0042 0.000 14.391 0.000 0.004 0.005  
==============================================================================  
Omnibus: 0.754 Durbin-Watson: 2.112  
Prob(Omnibus): 0.686 Jarque-Bera (JB): 0.533  
Skew: 0.051 Prob(JB): 0.766  
Kurtosis: 3.156 Cond. No. 5.88e+04  
==============================================================================  
  
Notes:  
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.  
[2] The condition number is large, 5.88e+04. This might indicate that there are  
strong multicollinearity or other numerical problems.

## OBSERVAMOS QUE MEJORAMOS UN POCO nuestro valor de R-squared a 58.1%

# efectuaremos una prediccion sobre los valores test y compararemos con al realidad, hayq ue agregar la constante en x\_test  
x\_test = sm.add\_constant(x\_test)  
x\_test

const CANTIDAD LINEAS  
FECHAC   
04/01/2022 1.0 33897 1819  
07/05/2022 1.0 10412 412  
15/11/2023 1.0 17222 1733  
02/02/2023 1.0 21684 1544  
13/09/2022 1.0 35213 2203  
... ... ... ...  
09/05/2023 1.0 27588 2001  
18/05/2023 1.0 20770 1525  
08/12/2023 1.0 19630 873  
27/11/2023 1.0 24145 2258  
13/10/2022 1.0 13578 990  
  
[160 rows x 3 columns]

# efectuaremos una prediccion sobre los datos de test  
y\_test\_pred = resultado.predict(x\_test)  
y\_test\_pred

FECHAC  
04/01/2022 13.368574  
07/05/2022 7.326527  
15/11/2023 12.869410  
02/02/2023 12.122693  
13/09/2022 14.974173  
 ...   
09/05/2023 14.070487  
18/05/2023 12.036016  
08/12/2023 9.319168  
27/11/2023 15.108225  
13/10/2022 9.753388  
Length: 160, dtype: float64

# encontramos que el error estiamdo por el métodod   
from sklearn.metrics import mean\_squared\_error  
rmse\_rm = mean\_squared\_error(y\_true = y\_test, y\_pred = y\_test\_pred, squared = False)  
print(f"El error (rmse) de test es: {rmse\_rm}")

El error (rmse) de test es: 2.4643964391718294

## nuestro valor de predicciones ahora es de 2.46, mejorando un poco

# CONCLUSIONES

* Nuestro modelo mejorado tiene un nivel de exactitud de 58.1%, es decir, en ese porcentaje nuestro modelo de regresión predice la realidad
* Asimismo nuestras predicciones se alejan un 2.46 de los valores reales en promedio
* Si bien nuestro modelo puede predecir con mediana exactitud nos sirve para contratar nuevo personal de acuerdo a las líneas y cantidades solicitadas por día
* Es necesario efectuar nuevos análisis y quizás aplicar otros métodos para mejorar los resultados