predicción de la lista nominal para las elecciones de este año 2021 y analisis de defunciones por covid-19 en San Francisco del Rincón

Universidad de Guanajuato, División de Ciencias e Ingenierías
Herramientas Informaticas y Gestión de la Información
Raygoza Aguirre Ana Teresa

***Resumen*—En esta práctica se estudió la ley de Enfriamiento de Newton a través de la plataforma didáctica física, para observar como se enfría un material en contacto con otro.**

I. Introducción

Los datos abiertos permiten transparencia, monitoreo, innovación e involucramiento ciudadano en la mejora de políticas y servicios públicos. Excel es una herramienta que nos permite hacer varios tipos de cálculos matemáticos, entre ellos estadística para análisis de datos, el siguiente reporte tiene como objetivo utilizar las herramientas aprendidas en clase de Herramientas Informáticas y Gestión de la Información, aplicando lo aprendido para analizar el crecimiento de la lista nominal en el estado de Guanajuato, y hacer una predicción de la lista nominal para las elecciones de este año 2021, y con ello predecir también el número de casillas a instalar en el estado, además de realizar un análisis meteorológico del municipio San Francisco del Rincón para predecir el las posibles lluvias del año 2022. (Electoral, 2020)

II. Objetivos

Objetivos generales: Dado los conceptos aprendidos, aplicarlos al realizar una prediccion del numero de casillas y analisis de datos utilizando las herramientas vistas en clase, especificamente python y latex.

III. Montaje Experimental

III-A. Materiales

- -Computadora
- -Python 3
- - Latex

III-B. Procedimiento Análisis de datos INE

- -Se importaron librerias a python para el analisis de datos, numpy y matplotlib.pyplot
- -Para el análisis de datos se descargaron los archivos de la lista nominal por mes de https://www.ine.mx/transparencia/datos-abiertos//archivo/estadistica-padron-electoral-lista-nominal-electores y se utilizaron los meses de Septiembre 2019 a Diciembre 2020
- -Se importo el achivo a python y mediante el siguiente codigo fue posible que python leyera el archivo:
  import glob files=glob.glob("./data/*.txt")
  data1=pd.read$_csv(files[0], usecols = [0, 1, 2, 3, 13])$
  $data1$
  $-Seseleccionoelestadode Guanajuatomedianteelsiguientecodig$

  $data1_=data1[data1['ENTIDAD'] == 11]$
-con ayuda de la libreria numpy, se creo un arreglo donde aparecieran los municipios del estado de guanajuato:
mpos=np.unique(data1$_GTO['MUNICIPIO'])$
$mpos$
$-seingresanlosmunicipiosaunciclo forconelobjetivodecrearunatabla$
$ln_mpo = []$
$for mpoinmpos[1:]:$
$ln_mpo.append(data1_GTO[data1_GTO['MUNICIPIO'] == mpo]['LISTA_NAL'].sum())$

pd.DataFrame('MPO':mpos[1:],'LISTA$_NAL_MPO_2$0200531' : ln_mpo)
$LISTA_NAL_MPO = data1_GTO[1 :].groupby(['MUNICIPIO']).sum()['LISTA_NAL']$
$LISTA_NAL_MPOs = pd.DataFrame(LISTA_NAL_MPO)$
$LISTA_NAL_MPOs$

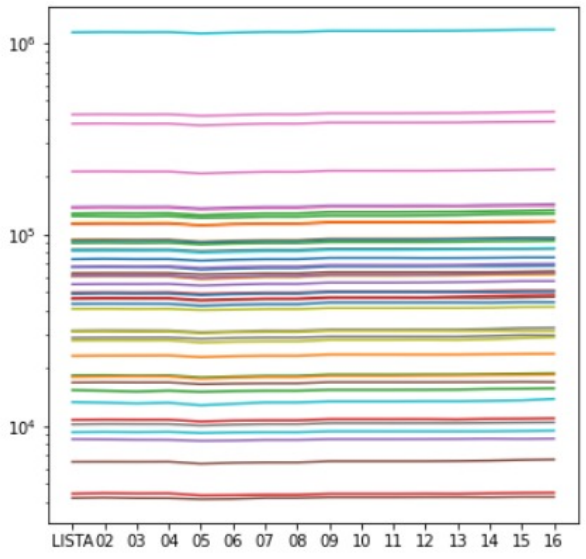
	MPO	LISTA_NAL_MPO_20200531
0	1	68057
1	2	94967
2	3	130720
3	4	50283
4	5	69125
5	6	4244
6	7	383007
7	8	31800
8	9	60931
9	10	9376
10	11	75392
11	12	23598
12	13	18553
13	14	115367
14	15	141334
15	16	16919
16	17	428437
17	18	29248
18	19	41248

- Para la regresion lineal se multiplico por dos los datos de la lista nominal:
  LISTA$_{N}AL_{M}POs['LISTA_{N}AL2'] = LISTA_{N}AL_{M}POs['LISTA_{N}AL'] * 2$
  $LISTA_{N}AL_{M}POs$
  $conelsiguientecodigoseacomodanlosarchivosporfecha$:
  $files$
  $date = []$
  $date_{=}[]$
  $files_{=}[]$

  for i,file in enumerate(files):
  date.append(re.findall(r'+',file)[0])

  temp=sorted(range(len(date)), key=date.$_{getitem_{)}}$
  for i in temp:
  date.$_{append(date[i])print(date[i],files[i])files._{append(files[i])}}$
- com el siguiente codigo se grafica la linea de tendencia:
  plt.figure(figsize=(6,6))

  for i in range(46):
  plt.plot(df$_{m}po.iloc[i])$
  $plt.yscale('log')$



$pararealizarlapr$
$fits = []$
$prediction_{l}nal = []$

for i in range(len(municipios)):
xx=np.arange(len(municipios[i]))
ma, ba = np. polyfit(xx, municipios[i],1,w=municipios[i])
fits.append([ma,ba])
pred=ma*(xx[-12]+12)+ba
if pred <municipios[i][-1]:
pred=municipios[i][-1]

prediction$_{l}nal.append(pred)$
$df_{m}po['PredictionLNAL'] = prediction_{l}nal$

- podemos ver algunas de las predicciones en municipios

MUNICIPIO	LISTA	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	Prediction LNAL
1	67427	67541	67450	67463	65505	66245	66750	66736.0	68057	68054	68052	68093	68244	68241	68422	68611	68526.354095
2	94040	94139	93825	93970	91712	92759	93343	93340.0	94967	94967	94952	95009	95209	95595	96005	95934	95916.339636
3	128446	128937	128731	128978	125884	127461	128544	128543.0	130720	130717	130708	130992	131172	131903	132669	133489	132884.440484
4	49656	49787	49699	49818	48907	49311	49596	49596.0	50283	50281	50273	50287	50324	50647	50867	50877	50824.359919
5	67868	68120	67992	68218	66789	67553	68063	68058.0	69125	69124	69114	69132	69287	69745	70072	70194	70131.713743

- ya que tenemos una prediccion de lista nominal de cada seccion, se calcula cuantas casillas se requieren, dado que debe haber una casilla por cada 750 actas para obtener esta informacion se dividio entre 750:
  prediccion=df$_{m}po['PredictionLNAL']$
  $prediccion.sum()$
  $prediccion.sum()/750$
  $conestoconcluimosqueelnumerodecasillasserian$:
  6037,966979796251

IV. Procedimiento Análisis de datos de Covid-19

- Se importaron librerias a python para el analisis dedatos, numpy y matplotlib.pyplot

-Para el análisis de datos se descargaron los archivos de la lista nominal por mes de https://coronavirus.guanajuato.gob.mx/

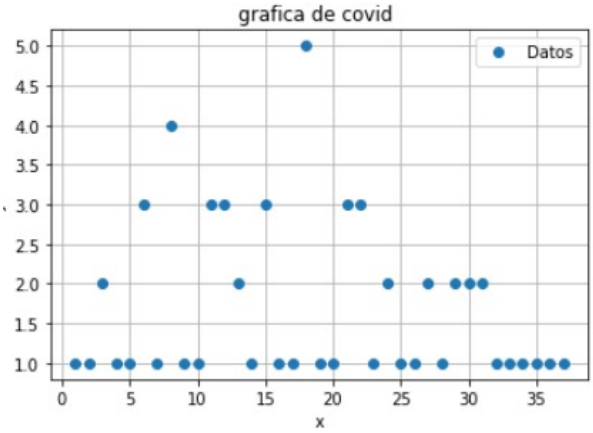
Se importo el achivo a python y mediante el siguientecodigo fue posible que python leyera el archivo:
covid$_{S}FR = pd.read_{c}sv('/content/drive/MyDrive/HIGI/proyecto.$
$covid_{S}FR$
$serealizoelsiguientecodigoconlafinalidaddehacerunaregresionlinea$
$fecha = pd.read_{c}sv('/content/drive/MyDrive/HIGI/fecha.csv')de$

$pd.read_csv('/content/drive/MyDrive/HIGI/defunciones.csv')$
$dias = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37]$
$x = np.array(dias)$
$y = np.array(defunciones)$

n = len(x)
sumx = sum(x)
sumy = sum(y)
sumx2 = sum(x*x)
sumy2 = sum(y*y)
sumxy = sum (y*x)
promx = sumx/n
promy = sumy/n
m = (sumx*sumy - n*sumxy)/(sumx**2 - n*sumx2)
b = (promy - m*promx)

con el siguiente codigo se graficaron los fatos obtenidos:

plt.plot (x,y , 'o', label= 'Datos')
plt.plot(x, m*x + b, label = 'ajuste')
plt.xlabel('x')
plt.ylabel('y')
plt.title('grafica de covid ')
plt.grid()
plt.legend()
plt.show



se utilizo otro tipo de metodo para graficar los datos, llamado interpolacion grafica, el cual consiste en encapsular la serie de datos y crear funciones cubicas en matrices, la interpolacion junta todas las funciones.

def Matrices(x,y):
"
Devuelve las matrices A y b de los coeficientes que necesitaremos para la interpolacion

PARAMETROS
x: arreglo con los valores de la variable independiente
y: arreglo con los valores de la variable dependiente"
n = len(x)
A = np.zeros((4*(n-1), 4*(n-1)))
b = np.array([])

Condicion 1: $S_i(x_i) = y_i$
for i in range(n-1):
for j in range(4):
for $_i nrange(n-1)$:
$A[i, i*4+j] = x[i]**(3-j)$
$b = np.append(b, y[i])$

Condicion 2: $S_i(x_i+1) = y_i + 1$
$foriinrange(n-1)$:
$forjinrange(4)$:
$for_i nrange(n-1)$:
$A[i+(n-1), i*4+j] = x[i+1]**(3-j)$
$b = np.append(b, y[i+1])$

Condicion 3: [iI([+1])=[+1]([+1])
for i in range(n-2):
for j in range(4):
for $_i nrange(n-1)$:
$A[i+(n-1)*2, i*4+j] = (3-j)*x[i+1]**(3-j))*(-1)$
$A[i+(n-1)*2, (i+1)*4+j] = ((3-j)*x[i+1]**(3-j))*(-1)$
$b = np.append(b, 0)$

Condicion 4: [ ]([+1])= [+1]([+1])
6[2][1]+2[2]6[1][1]2[1]=0
for i in range(n-2):
for j in range(2):
A[i+(n-1)*3-1, (i+j)*4] = 6*x[i+1]*(-1)**j
A[i+(n-1)*3-1, (i+j)*4+1] = 2*(-1)**j
b = np.append(b,0)

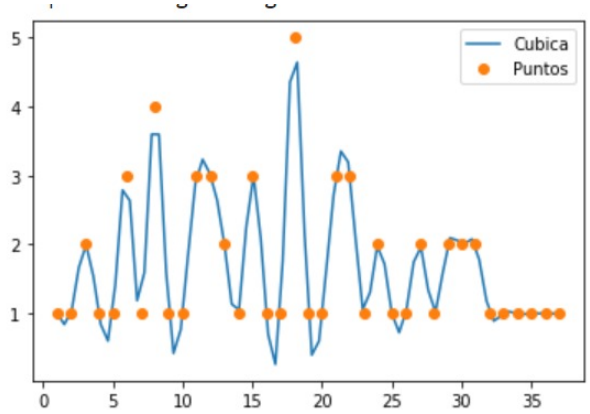
Condicion 5 y 6: [1]([1])=0 y [1]([])=0,
6[1][1]+2[1]=0 , 6[1][]+2[1]=0
A[-2][0] = 6*x[0]
A[-2][1] = 2
b = np.append(b,0)

A[-1][-4] = 6*x[-1]
A[-1][-3] = 2
b = np.append(b,0)
return A,b

def interpol$_c ubic(x, y)$:
$"""PARAMETROS$
$x : arregloconlosvaloresdelavariableindependiente$
$y : arregloconlosvaloresdelavariabledependiente$
"""
$A, b = Matrices(x, y)$
$b = b[:, np.newaxis]transpuesta$
$coef = np.dot(np.linalg.inv(A), b)$

def f(x1):
for i in range(len(x)):
if x[i] <= x1 <= x[i+1]:
salida = coef[4*i]*x1**3+coef[4*i+1]*x1**2+coef[4*i+2]*x1+coef[4*i+3
return salida
return f

gra1= interpol$_cubic(x, y)$
$x_new = np.linspace(1, 37, 70)$
$y_new = []$
$for i in x_new:$
$y_new.append(gra1(i))$
$plt.plot(x_new, y_new, label = "Cubica")$
$plt.plot(x, y, "o", label = "Puntos")$
$plt.legend()$



V. BIBLIOGRAFICA

https://coronavirus.guanajuato.gob.mx/

Electoral, I. N. (31 de diciembre de 2020). Instituto Nacional Electoral. Obtenido de Instituto Nacional Electoral: https://www.ine.mx/transparencia/datosabiertos//archivo/estadistica-padron-electoral-lista-nominal-electores