División de Ciencias e Ingenierías

Proyecto final: Reporte

Herramientas Informáticas y Gestión de la Información

Análisis de datos en Python

Alumno:
Azalia Orozco Salgado

NUA:
427308

18 de junio de 2021

Índice

1. Predicción de casillas del INE

1.1. Introducción

Para este análisis se retomaron los datos de las secciones del INE desde el mes de septiembre de 2019 hasta dicienbre de 2021 cuyas bases de datos se encuentran en Con el objetivo de predecir el número de casillas que se instalarían en Geanajuato para el mes de febrero de 2021

1.2. Desarrollo

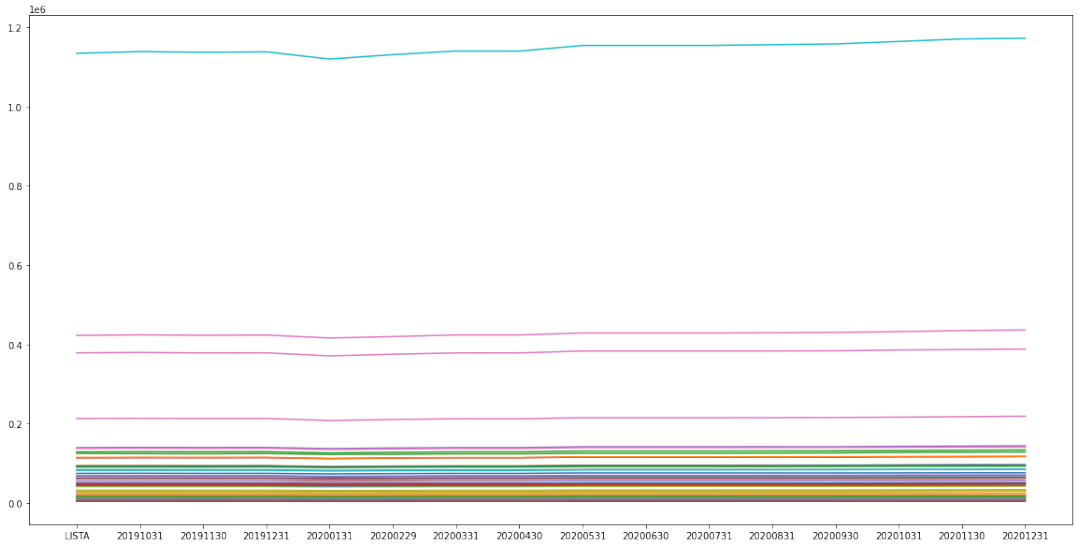
Se utilizó jupyter notebook para la realización del análisis utilizando principalmente las librerías de pandas, numpy u scikit-learn. Primero se importaron todos los archivos con la librería glob y se leyeron con pandas. Para organizar los datos se realizó el código mostrado a continuación, la organización se hizo por municipio y por cada uno se hizo la suma de lista nominal

```
    for i,file in enumerate(files_):
  data = pd.read_csv(file)
  data = data[1:]
  data = data[data['ENTIDAD']==11][1:]
  sec = data.groupby(['SECCION']).sum()
  if i==0:
    if 'LISTA_NAL' in sec.columns:
      df_sec = pd.DataFrame(sec['LISTA_NAL'])
    if 'LISTA_NACIONAL' in sec.columns:
      df_sec = pd.DataFrame(sec['LISTA_NACIONAL'])
    if 'LISTA' in sec.columns:
      df_sec = pd.DataFrame(sec['LISTA'])
  else:
    if 'LISTA_NAL' in sec.columns:
      df_sec[date_[i]]=sec['LISTA_NAL']
    if 'LISTA_NACIONAL' in sec.columns:
      df_sec[date_[i]]=sec['LISTA_NACIONAL']
    if 'LISTA' in sec.columns:
        df_sec[date_[i]]=sec['LISTA']
```

Ua vez organizados se graficaron con matplotlib con el siguiente código

```
     plt.figure(figsize=(20,10))
for i in range(len(df_sec)):
  plt.plot(df_sec.iloc[i])
```

Y dando los siguientes resultados

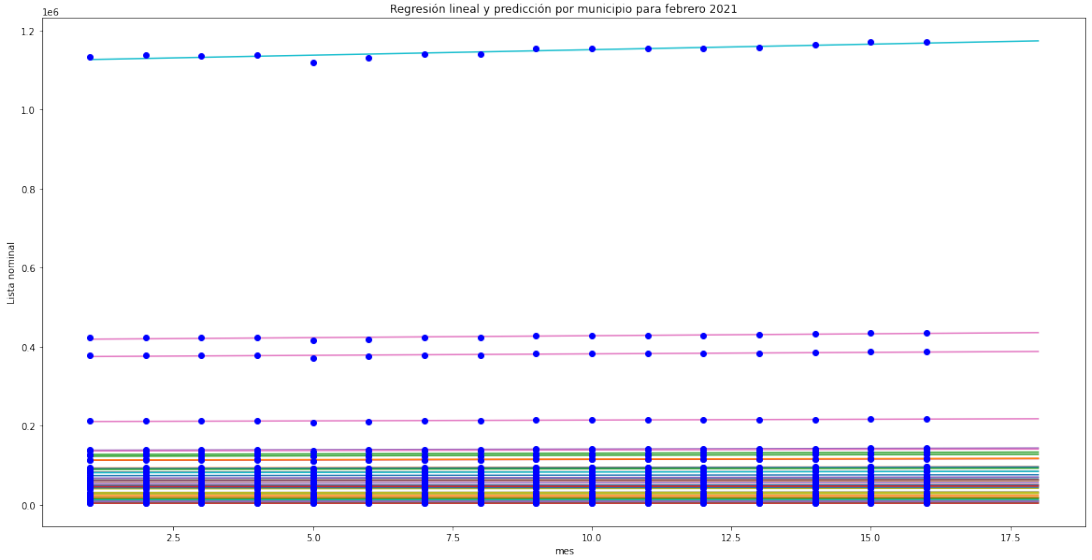


Para la la regresión lineal y la predicción para el me meás de febrero se utilizó la librería de scikit-learn como se muetra a continuación.

```
    model = LinearRegression()
x = np.linspace(1,16,16).reshape(-1,1)
y = np.asarray(df_mpo)
m = []
b = []
R_sq = []
y_predict = []
y_feb = []
y_sum_feb = 0
plt.figure(figsize=(20,10))
plt.title('Regresión lineal y predicción por municipio para febrero 2021')
for i in range (len(df_mpo)):
  model.fit(x,y[i])
  R_sq = model.score(x, y[i])
  y_predict = model.predict(x)
  b = model.intercept_
  m = model.coef_
  x_feb = np.asarray([18]).reshape(-1,1)
  y_feb = model.predict(x_feb)
  x_new = np.array(np.concatenate([x, x_feb]))
```

```
y_new = np.array(np.concatenate([y_predict, y_feb]))
plt.plot(x_new, y_new,)
plt.plot(x, y[i], 'bo')
plt.xlabel('mes')
plt.ylabel('Lista nominal')
y_sum_feb += y_feb
```

1.3. Resultados y conclusiones



Con la predicción de lista nominal para el mes de febrero, lo unico que se hizo fue hacer la división entre 750, lo que dió como resultado que para este mes se tendrían que instaar un aproximado de 6050 casillas en el estado de Guanajuato

2. Índice de Marginación en el Estado de Guanajuato

2.1. Introducción

Para la realización de este análisis se tomó una base de datos del gobierno de méxico, la cual contiene datos desde 1990 hasta 2015, y puede ser encontada en el siguiente enlace: https: //datos.gob.mx/busca/dataset/indice- de-marginacion-carencias-poblacionales-por-localidad-municipio-y-entidad

Se relizó el análisis utilizando las librerías de pandas, numpy y scikit-learn

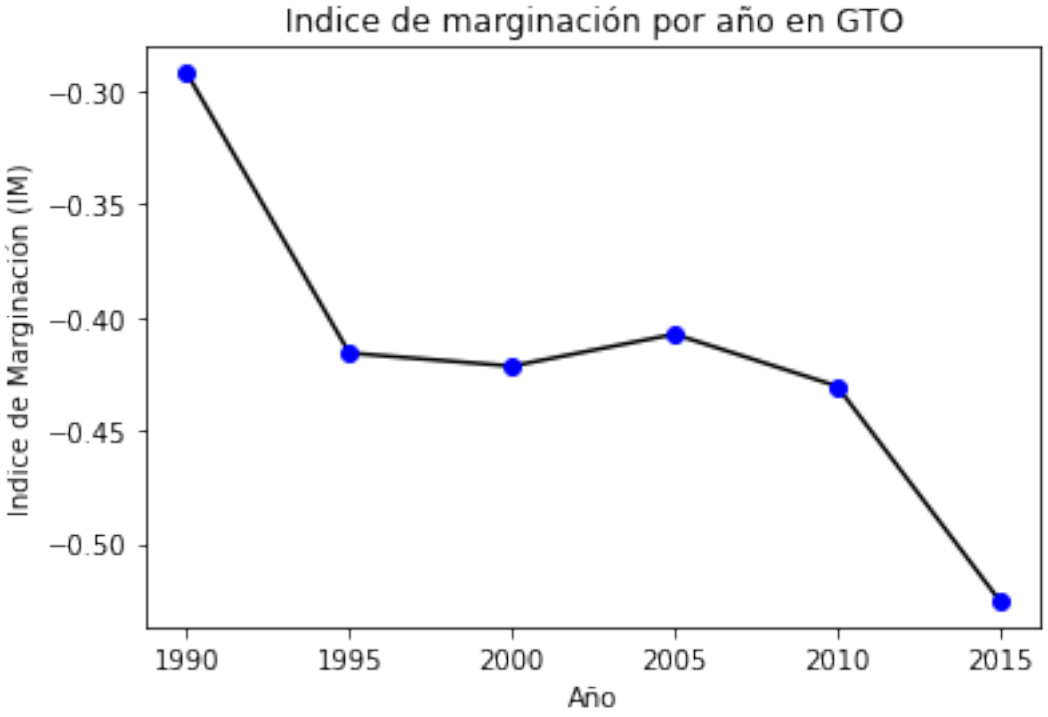
2.2. Desarrollo

Lo primero que se hizo fue filtrar y organizar los datos con pandas y se obtuvo el promedio por municipio de cada año, para lo que se utilizó el siguente código

```
    def convert_array_float(array):
  return [float(x) for x in array]

def clean_data_for_value_state(state):
  clean_up = drop_colums['ENT'] == state
  drop_ent = drop_colums[clean_up]
  return drop_ent.drop('ENT', axis=1)

def get_mean_im(data_csv):
  mean = {}

  for i in range(years[0], years[1]):
    data = data_csv[data_csv['AÑO'] == i]
    if len(data) >= 1:
      mean[i] = np.mean(convert_array_float(data['IM'].values))

  return mean
  new_csv = clean_data_for_value_state('Guanajuato')
  for i in range(years[0], years[1]):
  data = new_csv[new_csv['AÑO'] == i]
  if len(data) >= 1:
    print(data)
  data_mean = get_mean_im(new_csv)
  data_mean
```

Una vez organizados los datos se realizó la gráfica y se observó un comportamiento medianamente lineal.

```
    x = [key for key in data_mean.keys()]
    y  = [value for value in data_mean.values()]
    plt.plot(x, y, color = 'black')
    plt.plot(x, y, 'bo')
    plt.title('Indice de marginación por año en GTO')
    plt.xlabel('Año')
    plt.ylabel('Indice de Marginación (IM)')
    plt.show()
```

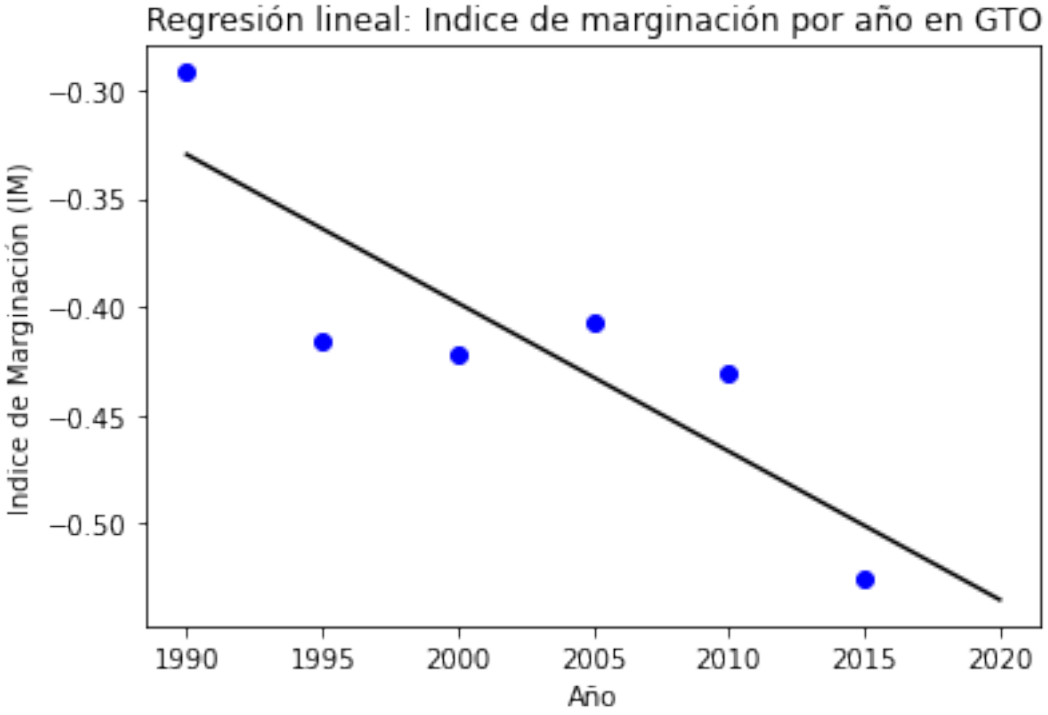


Se procedió a realizar una regresión lineal untilizando LinearRegression de la de sklearn como se muetras a conticuación

```
x_1 = np.array(x).reshape(-1, 1)
model = LinearRegression()
model.fit(x_1, y)
R_sq = model.score(x_1, y)
print("R^2 = ", R_sq)
print("b = ", model.intercept_)
print("m = ", model.coef_)
# Nuevos valores y de la regresión
y_predict = model.predict(x_1)
print("Valores de y en para la regresión lineal: ", y_predict)
# Extrapolar para el año 2020
x_2020 = np.array([2020]).reshape(-1,1)
y_2020 = model.predict(x_2020)
# Nuevas coordenadas para la gráfica
x_new = np.array(np.concatenate([x_1, x_2020]))
y_new = np.array(np.concatenate([y_predict, y_2020]))
# Graficar la regresión lineal
plt.plot(x_new, y_new, color = 'black')
```

```
plt.plot(x, y, 'bo')
plt.title('Regresión lineal: Indice de marginación por año en GTO')
plt.xlabel('Año')
plt.ylabel('Indice de Marginación (IM)')
plt.show()
```

2.3. Resultados y conclusiones



Con la regresión linal se extrapolaron los datos para el año 2020, Podemos darnos cuenta que el índice de marginación a lo largo del tiempo ha hido en decremento, lo que sugiere que cada vez más, en el estado, se vamejorando las condiciones de vida de la población, esto puede ser debido a quela educación es cada vez es más accesible.