Análisis sobre salarios en la Industria Manufacturera

Proyecto Final

Profesor: Dr. Cesar Isaza Bohorquez

Curso: Tecnología de Ciencia de Datos

Alumno: Angel Mario Alemán T. Matrícula: 220037640

Alumno: Marco Medina Matricula: 220037533

Universidad Politécnica de Querétaro
Dirección: Carretera Estatal 420 S/N, El Marqués, Querétaro
C.P. 76240 Teléfono: 52 (442) - 101 - 9000

DESCRIPCIÓN DE LA ACTIVIDAD

Realizar la estimación de un modelo matemático para realizar predecir el comportamiento de un sistema a partir de información de base de datos, utilizando los 6 pasos del proceso de ciencia de datos.

Problema

1. Predecir el comportamiento de los salarios en la industria manufactura México y su predicción de crecimiento en contraste con la estimación del comportamiento de la inflación en México.

Actividades

1. Definir la investigación – que problemas queremos resolver
2. Recuperación de los Datos
3. Preparación de los Datos – Hay datos basura, necesitamos limpiarlos.
4. Exploración de los Datos – Como procesamos la información, qué características tiene. Encontrar o generar modelos de referencia
5. Modelado de los Datos – Comparar la información con un modelo de referencia
6. Presentación y automatización – Resultados

MARCO TEÓRICO

Regresión Lineal

La regresión lineal es un método estadístico que trata de modelar la relación entre una variable continua y una o más variables independientes mediante el ajuste de una ecuación lineal. Se llama regresión lineal simple cuando solo hay una variable independiente y regresión lineal múltiple cuando hay más de una. A la variable modelada se le conoce como variable dependiente o variable respuesta, y a las variables independientes como regresores, predictores o *features*.

Modelos de regresión lineal en Python

Dos de las implementaciones de modelos de regresión lineal más utilizadas en Python son: **scikit-learn** y **statsmodels**. Aunque ambas están muy optimizadas, **Scikit-learn** está orientada principalmente a la predicción, por lo que no dispone de apenas funcionalidades que muestren las muchas características del modelo que se deben analizar para hacer inferencia. **Statsmodels** es mucho más completo en este sentido.

Definición matemática

El modelo de regresión lineal (Legendre, Gauss, Galton y Pearson) considera que, dado un conjunto de observaciones $\{y_i, x_{i1}, \ldots, x_{np}\}_{ni=1}$, la media μ de la variable respuesta y se relaciona de forma lineal con la o las variables regresoras $x_1 \ldots x_p$ acorde a la ecuación:

$$\mu_y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

El resultado de esta ecuación se conoce como la línea de regresión poblacional, y recoge la relación entre los predictores y la **media** de la variable respuesta.

Otra definición que se encuentra con frecuencia en los libros de estadística es:

$$y_i=\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+...+\beta_p x_{ip}+\epsilon_i$$

En este caso, se está haciendo referencia al valor de y

para una observación i concreta. El valor de una observación puntual nunca va a ser exactamente igual al promedio, de ahí que se añada el término de error ϵ

.

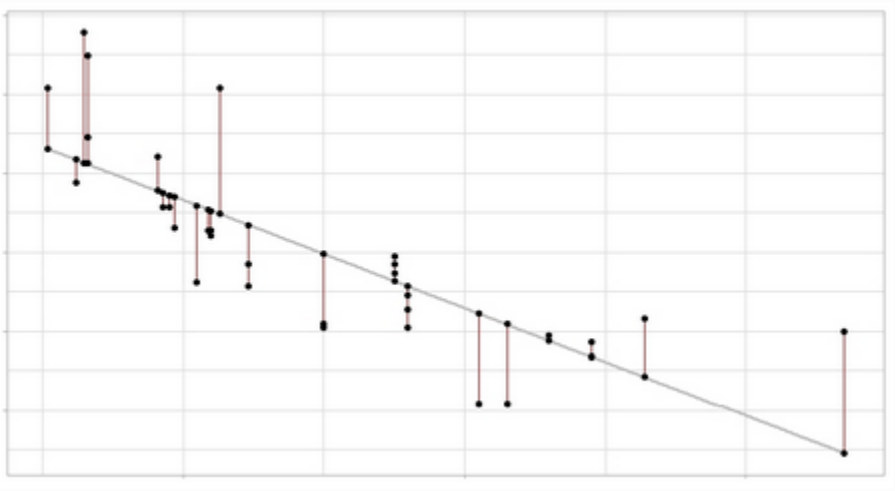
En ambos casos, la interpretación de los elementos del modelo es la misma:

- β_0 : es la ordenada en el origen, se corresponde con el valor promedio de la variable respuesta y cuando todos los predictores son cero.
- β_j : es el efecto promedio que tiene sobre la variable respuesta el incremento en una unidad de la variable predictora x_j , manteniéndose constantes el resto de variables. Se conocen como coeficientes parciales de regresión.
- e : es el residuo o error, la diferencia entre el valor observado y el estimado por el modelo. Recoge el efecto de todas aquellas variables que influyen en y pero que no se incluyen en el modelo como predictores.

En la gran mayoría de casos, los valores β_0 y β_j poblacionales se desconocen, por lo que, a partir de una muestra, se obtienen sus estimaciones $\hat{\beta}_0$ y $\hat{\beta}_j$. Ajustar el modelo consiste en estimar, a partir de los datos disponibles, los valores de los coeficientes de regresión que maximizan la verosimilitud (*likelihood*), es decir, los que dan lugar al modelo que con mayor probabilidad puede haber generado los datos observados.

El método empleado con más frecuencia es el ajuste por mínimos cuadrados ordinarios (*OLS*), que identifica como mejor modelo la recta (o plano si es

regresión múltiple) que minimiza la suma de las desviaciones verticales entre cada dato de entrenamiento y la recta, elevadas al cuadrado.



Modelo de regresión lineal y sus errores: la línea gris representa la recta de regresión (el modelo) y los segmentos rojos el error entre esta y cada observación.

Interpretación del modelo

Los principales elementos que hay que interpretar en un modelo de regresión lineal son los coeficientes de los predictores:

- β_0 es la ordenada en el origen o *intercept*, se corresponde con el valor esperado de la variable respuesta y cuando todos los predictores son cero.
- β_j los coeficientes de regresión parcial de cada predictor indican el cambio promedio esperado de la variable respuesta y al incrementar en una unidad de la variable predictora x_j, manteniéndose constantes el resto de variables.

La magnitud de cada coeficiente parcial de regresión depende de las unidades en las que se mida la variable predictora a la que corresponde, por lo que su magnitud no está asociada con la importancia de cada predictor.

Para poder determinar qué impacto tienen en el modelo cada una de las variables, se emplean los coeficientes parciales estandarizados, que se obtienen al estandarizar (sustraer la media y dividir entre la desviación estándar) las variables predictoras previo ajuste del modelo. En este caso, β_0 se corresponde con el valor esperado de la variable respuesta cuando todos los predictores se encuentran en su valor promedio, y β_j el cambio promedio esperado de la variable respuesta al incrementar en una desviación estándar la variable predictora x_j, manteniéndose constantes el resto de variables.

Si bien los coeficientes de regresión suelen ser el primer objetivo de la interpretación de un modelo lineal, existen muchos otros aspectos (significancia del modelo en su conjunto, significancia de los predictores, condición de normalidad). Estos últimos suelen ser tratados con poco detalle cuando el único objetivo del modelo es realizar predicciones, sin embargo, son muy relevantes si se quiere realizar inferencia, es decir, explicar las relaciones entre los predictores y la variable respuesta.

1. DEFINIR LA INVESTIGACIÓN – QUE PROBLEMAS QUEREMOS RESOLVER

Como se menciona al principio del documento, se busca predecir el comportamiento de los salarios en la industria manufactura México y su predicción de crecimiento en contraste con la estimación del comportamiento de la inflación en México.

Lo anterior nos permitirá plantear los siguientes objetivos;

1. Responder a la pregunta, ¿El incremento de salario, es superior a la inflación en México, manteniendo el poder adquisitivo de la población que trabaja en la industria manufacturera?
2. Estimar el incremento de sueldo a solicitar para evitar el efecto de la inflación en nuestro poder adquisitivo. Es decir, que sueldo solicitar en enero del 2023 para mantenerse sobre el incremento de la inflación para los próximos 2 años.
3. Estimar el incremento de salarios para la industria manufacturera, necesario para planificar el presupuesto de salarios.
4. Estimar el incremento de la inflación en México

RECUPERACIÓN DE LOS DATOS

Para lograr nuestro objetivo necesitamos recuperar la información que corresponda a los salarios de México e históricos de inflación.

La primera base de datos podemos encontrarla en la página del INEGI, en el apartado de Empleo y Ocupación que nos proporciona una base de datos que muestra el salario para la industria Manufacturera en dólares por hora (así porque existen estos indicadores en diferentes países)

https://www.inegi.org.mx/app/tabulados/default.html?nc=539&idrt=18&opc=t

Periodo	Total	Fabricacion c	Fabricacion c	excepto pre	Fabricacion c	Industria de	Industria del	Impresion e	Fabricacionn	Industria qui	Industria del	Fabricacion c	Industrias m	Fabricacion c
2019														
Enero R	2.75	2.1	2.4	1.7	2	4.3	3.8	2.6	3	3.7	2.7	3	2.4	2.6
Febrero	2.9	2.2	2.2	1.8	2.1	5.1	4.2	2.7	3.1	3.8	2.7	3.2	2.6	2.7
Marzo	2.87	2.2	2.2	1.8	2.1	4.4	4	2.8	3	4	2.8	3.3	2.5	2.7
Abril	2.87	2.3	2.2	1.8	2.1	4.5	4	2.7	3.1	3.9	2.8	3.3	2.6	2.7
Mayo	3.08	2.3	2.3	1.8	2.2	6.1	4.2	2.8	3.2	4.2	2.9	3.5	2.5	2.9
Junio	2.83	2.1	2.1	1.8	2	4.6	3.8	2.7	3.1	3.9	2.8	3.3	2.5	2.8
Julio	2.74	2	2	1.7	2	4	3.7	2.6	3.1	4	2.7	3.2	2.5	2.6

La segunda base de datos podemos encontrarla en el **Índice Nacional de Precios al Consumidor (INPC)** en la página del INEGI con inflación mensual anualizada. Que nos entrega una base de datos del valor de la inflación por mes a lo largo del tiempo

https://www.inegi.org.mx/temas/inpc/

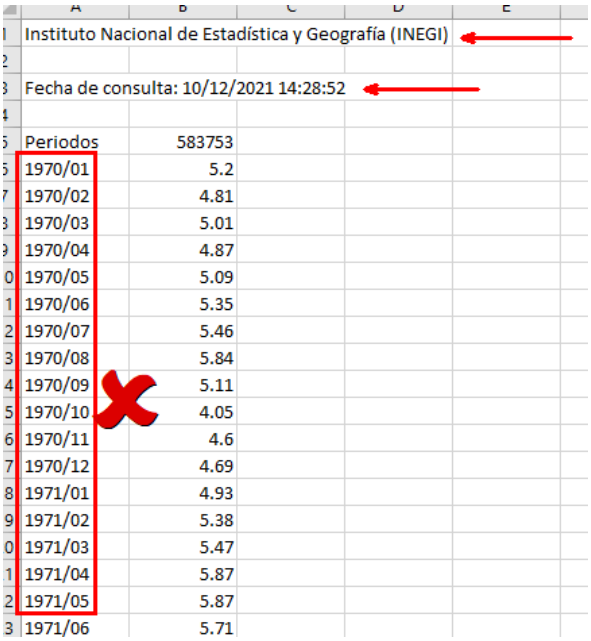
Instituto Nacional de Estadística y Geografía (INEGI)			
Fecha de consulta: 10/12/2021 14:28:52			
Periodos	583753		
1970/01	5.2		
1970/02	4.81		
1970/03	5.01		
1970/04	4.87		
1970/05	5.09		
1970/06	5.35		
1970/07	5.46		
1970/08	5.84		
1970/09	5.11		
1970/10	4.05		
1970/11	4.6		
1970/12	4.69		
1971/01	4.93		

PREPARACIÓN DE LOS DATOS

La siguiente actividad es la limpieza de los datos, esto es eliminar las celdas con valor NULL, celdas con valores en formato STRING (palabras) y la corrección de los encabezados con caracteres no reconocidos

Periodo	Total	de insumos textiles y acabado de textiles	Fabricacion d e productos textiles	excepto prendas de vestir	Fabricacion de prendas de vestir	Industria de la madera	Industria del papel	Impresion e industrias conexas	de productos derivados del petroleo y del carbon	Industria quimica	Industria del plastico y del hule	de productos a base de minerales no metÃ¡licos	Industrias metalicas basicas
2019													
Enero R	2.75	2.1	2.4	1.7	2	4.3	3.8	2.6	3	3.7	2.7	3	2.4
Febrero	2.9	2.2	2.2	1.8	2.1	5.1	4.2	2.7	3.1	3.8	2.7	3.2	2.6
Marzo	2.87	2.2	2.2	1.8	2.1	4.4	4	2.8	3	4	2.8	3.3	2.5
Abril	2.87	2.3	2.2	1.8	2.1	4.5	4	2.7	3.1	3.9	2.8	3.3	2.6
Mayo	3.08	2.3	2.3	1.8	2.2	6.1	4.2	2.8	3.2	4.2	2.9	3.5	2.5
Junio	2.83	2.1	2.1	1.8	2	4.6	3.8	2.7	3.1	3.9	2.8	3.3	2.5
Julio	2.74	2	2	1.7	2	4	3.7	2.6	3.1	4	2.7	3.2	2.5
Agosto P	2.8	2.1	2.2	1.7	2.2	4.6	3.8	2.6	3	4	2.8	3.1	2.4
Septiembre	2.8	2.1	2.1	1.7	2.1	4.1	3.9	2.7	3	4	2.8	3.3	2.5
Octubre	2.81	2.1	2.1	1.7	2	4.8	3.8	2.7	3.1	3.9	2.8	3.2	2.5
Noviembre	2.93	2.2	2.2	1.8	2.2	4.7	4.2	2.7	3.1	4.1	3	3.3	2.6
Diciembre	4.39	4	3.1	2.8	3.4	12.1	6.2	3.8	4	5	3.8	4.6	3.1
2020													
Enero	2.98	2.4	2.4	1.8	2.2	4.3	4.1	2.9	3.2	4.3	3	3.3	2.6
Febrero	3.07	2.3	2.3	1.9	2.2	5.2	4.3	2.9	3.3	4.3	3	3.5	2.7
Marzo	2.6	2	2	1.6	2	3.9	3.4	2.6	2.8	3.7	2.5	3	2.4
Abril	3.54	3.6	2.4	2.6	7.5	3.6	3.5	3.1	3.1	3.9	2.8	3.2	2.4
Mayo	3.3	3.4	2.3	2.3	6.2	4.8	3.6	2.8	3	3.8	2.7	3.3	2.3

También se limpian los datos buscando acotarlos al periodo de análisis de nuestro proyecto, empatándolo con la base de datos de sueldos que inicia en enero del 2019.



Instituto Nacional de Estadística y Geografía (INEGI)	
Fecha de consulta: 10/12/2021 14:28:52	
Periodos	583753
1970/01	5.2
1970/02	4.81
1970/03	5.01
1970/04	4.87
1970/05	5.09
1970/06	5.35
1970/07	5.46
1970/08	5.84
1970/09	5.11
1970/10	4.05
1970/11	4.6
1970/12	4.69
1971/01	4.93
1971/02	5.38
1971/03	5.47
1971/04	5.87
1971/05	5.87
1971/06	5.71

2018/05	4.51
2018/06	4.65
2018/07	4.81
2018/08	4.9
2018/09	5.02
2018/10	4.9
2018/11	4.72
2018/12	4.83
2019/01	4.37
2019/02	3.94
2019/03	4
2019/04	4.41
2019/05	4.28
2019/06	3.95
2019/07	3.78
2019/08	3.16
2019/09	3
2019/10	3.02
2019/11	2.97
2019/12	2.83
2020/01	3.24
2020/02	3.7
2020/03	3.25

Indicadores20211210142852

EXPLORACIÓN DE LOS DATOS

Como procesamos la información, qué características tiene. Encontrar o generar modelos de referencia. La exploración de los datos nos permite adecuar los valores para procesarlos, en el caso de la tabla de salarios, enfocaremos nuestra atención a la celda de Total, el periodo se cambia para hacer coincidir los meses con un valor numérico ID consecutivo para su proceso. Las dos tablas, se unifican en una sola.

```
In [ ]:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import csv
import pylab as pl
#from sklearn.linear_model import BayesianRidge
#def func(x): return np.sin(2*np.pi*x)
```

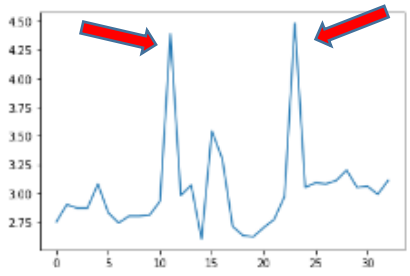
```
In [ ]:
df = pd.read_csv ('Tabuladoprom.csv')
print(df)
```

```
    Total
0    2.75
1    2.90
2    2.87
3    2.87
4    3.08
..    ...
28   3.20
29   3.05
30   3.06
31   2.99
32   3.11
```

```
In [ ]:
```

```
In [ ]:
plt.plot(df)
```

```
Out[ ]:
[<matplotlib.lines.Line2D at 0x7f57bb4016d0>]
```



La exploración de los datos muestra dos picos muy elevados, estos corresponden a los meses de diciembre que incluye salarios integrados con prestaciones, para mejorar la aproximación del modelado, se decide suprimir estos valores, ya que son eventuales, afectan nuestra gráfica, y no son relevantes parta nuestro modelo.

		ID
2019	enero	1
2019	febrero	2
2019	marzo	3
2019	abril	4
2019	mayo	5
2019	junio	6
2019	julio	7
2019	agosto	8
2019	septiembre	9
2019	octubre	10
2019	noviembre	11
2019	diciembre	12
2020	enero	13
2020	febrero	14
2020	marzo	15

MODELADO DE LOS DATOS

Regresión Lineal con Python y SKLearn

Comparar la información con un modelo de referencia, en nuestro caso es un modelo de regresión de lineal Regresión Lineal con Python y SKLearn. Creamos el objeto LinearRegression y lo hacemos “encajar” (entrenar) con el método fit(). Lo que nos va a proporcionar los coeficientes de nuestro modelo.

```
dataX =filtered_data[["ID"]]
X_train = np.array(dataX)
y_train = filtered_data['Total'].values

# Creamos el objeto de Regresión Linear
regr = linear_model.LinearRegression()

# Entrenamos nuestro modelo
regr.fit(X_train, y_train)

# Hacemos las predicciones que en definitiva una línea (en este caso, al ser 2D)
y_pred = regr.predict(X_train)

# Veamos los coeficienetes obtenidos, En nuestro caso, serán la Tangente
print('Coefficients: \n', regr.coef_)
# Este es el valor donde corta el eje Y (en X=0)
print('Independent term: \n', regr.intercept_)
# Error Cuadrado Medio
print("Mean squared error: %.2f" % mean_squared_error(y_train, y_pred))
# Puntaje de Varianza. El mejor puntaje es un 1.0
print('Variance score: %.2f' % r2_score(y_train, y_pred))
```

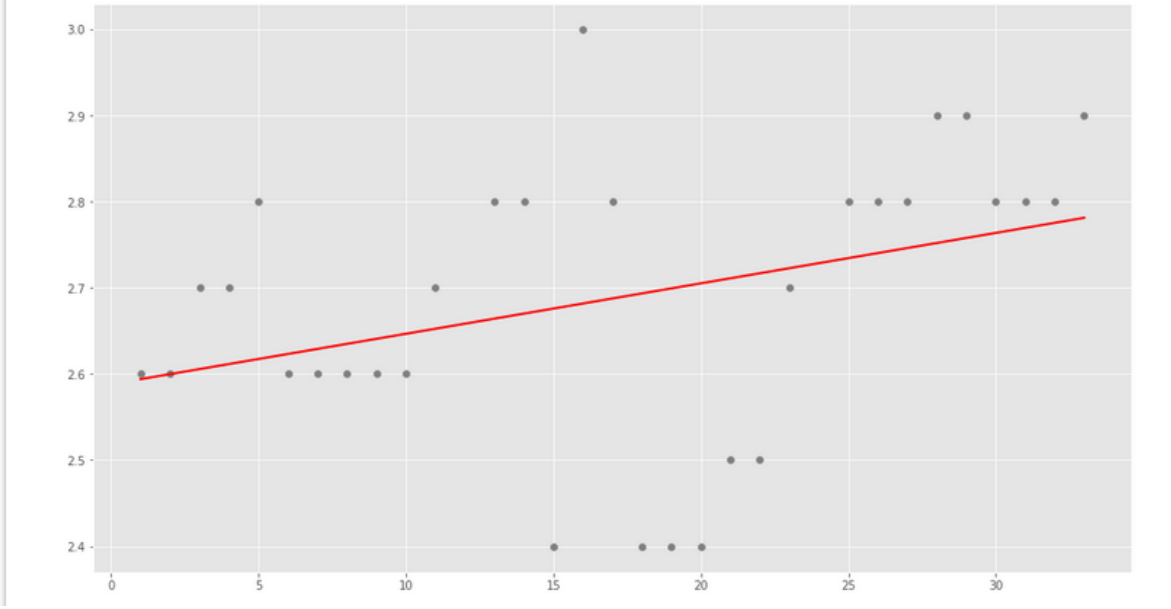
```
Coefficients:
 [0.00585159]
Independent term:
 2.5879972140535528
Mean squared error: 0.02
Variance score: 0.12
```

Ecuación

y = 0.00595159x + 2.58799 R2= 0.02

Grafica

```
plt.scatter(X_train, y_train,  color='gray')
plt.plot(X_train, y_pred, color='red', linewidth=2)
plt.show()
```



Evaluación

Se evalúa la ecuación de tendencia para estimar el valor del salario para enero de 2023

```
[ ]  #Vamos a comprobar:
     # según nuestro modelo, hacemos:
     y_Septiembre2022 = regr.predict([[49]])
     print(float(y_Septiembre2022))
```

```
2.874725274725275
```

Resultando un valor de 2.8747 USD/hora para enero del 2023

Repetimos la operación para la segunda base de datos de la Inflación

```
[ ] dataX =filtered_data[["ID"]]
    X2_train = np.array(dataX)
    y2_train = filtered_data['Inflacion'].values

    # Creamos el objeto de Regresión Linear
    regr2 = linear_model.LinearRegression()

    # Entrenamos nuestro modelo
    regr2.fit(X2_train, y2_train)

    # Hacemos las predicciones que en definitiva una línea (en este caso, al ser 2D)
    y2_pred = regr2.predict(X2_train)

    # Veamos los coeficienetes obtenidos, En nuestro caso, serán la Tangente
    print('Coefficients: \n', regr2.coef_)
    # Este es el valor donde corta el eje Y (en X=0)
    print('Independent term: \n', regr2.intercept_)
    # Error Cuadrado Medio
    print("Mean squared error: %.2f" % mean_squared_error(y2_train, y2_pred))
    # Puntaje de Varianza. El mejor puntaje es un 1.0
    print('Variance score: %.2f' % r2_score(y2_train, y2_pred))
```

```
Coefficients:
 [0.05904316]
Independent term:
 3.055236805448073
Mean squared error: 0.74
Variance score: 0.31
```

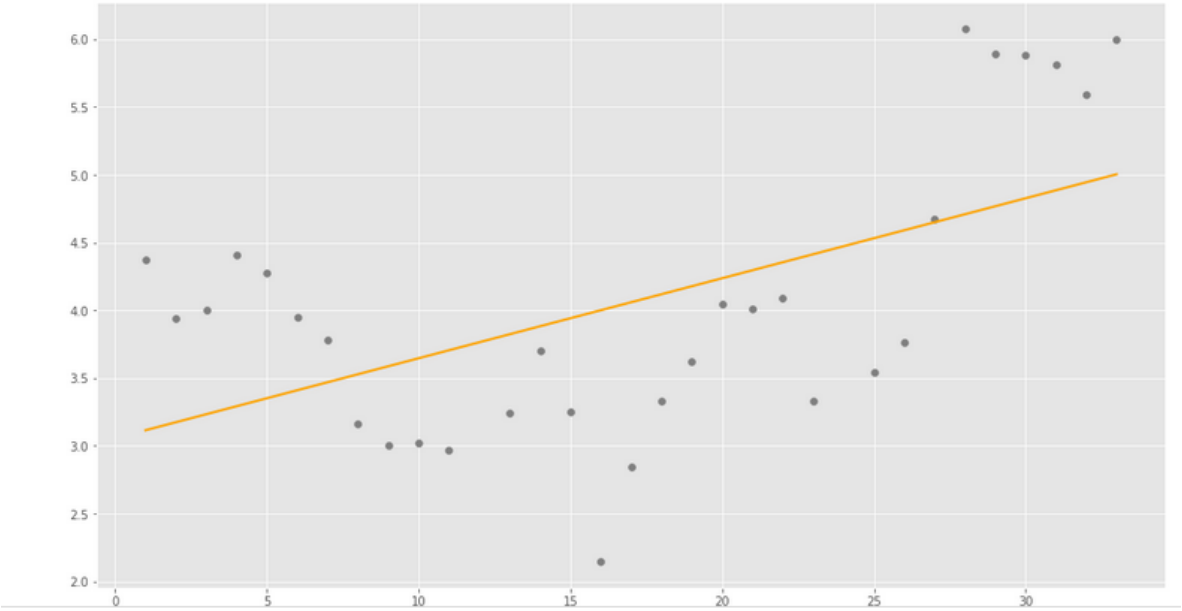
Ecuación

y = 0.0590 + 3.05523 R2= 0.74

Grafica

```
Independent term:
 3.055236805448073
Mean squared error: 0.74
Variance score: 0.31
```

```
[ ] plt.scatter(X2_train, y2_train,  color='gray')
    plt.plot(X2_train, y2_pred, color='orange', linewidth=2)
    plt.show()
```



Evaluación

Se evalúa la ecuación de tendencia para estimar el valor del salario para enero de 2023

```
#Vamos a comprobar:
# según nuestro modelo, hacemos:
p =49
y_Inflacion2022 = regr2.predict([[p]])
print("Inflacion")
print(float(y_Inflacion2022))
y_Septiembre2022 = regr.predict([[p]])
print ("DolaresvsHora")
print(float(y_Septiembre2022))
```

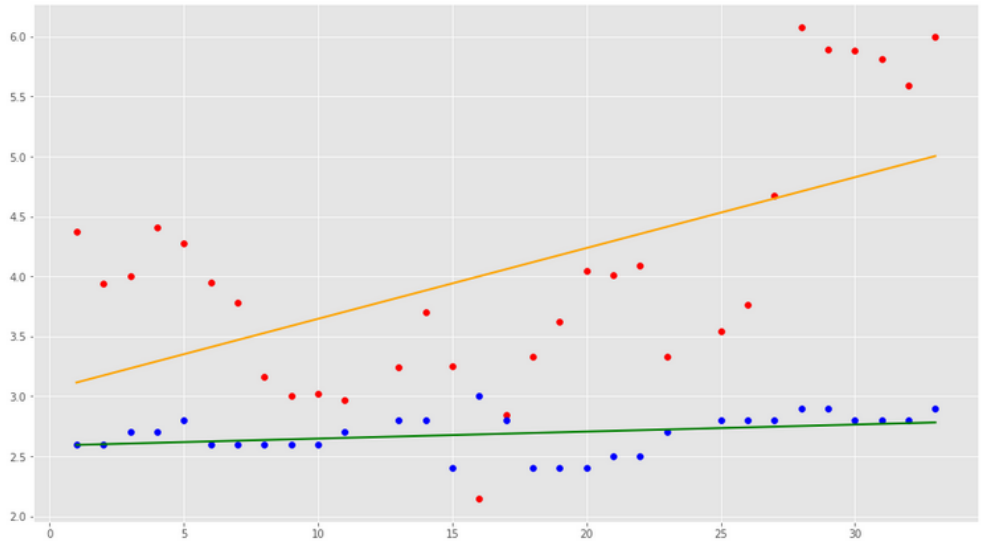
```
Inflacion
5.9483516483516485
DolaresvsHora
2.874725274725275
```

Resultando un valor de 5.9483 % inflación para enero del 2023

PRESENTACIÓN Y AUTOMATIZACIÓN

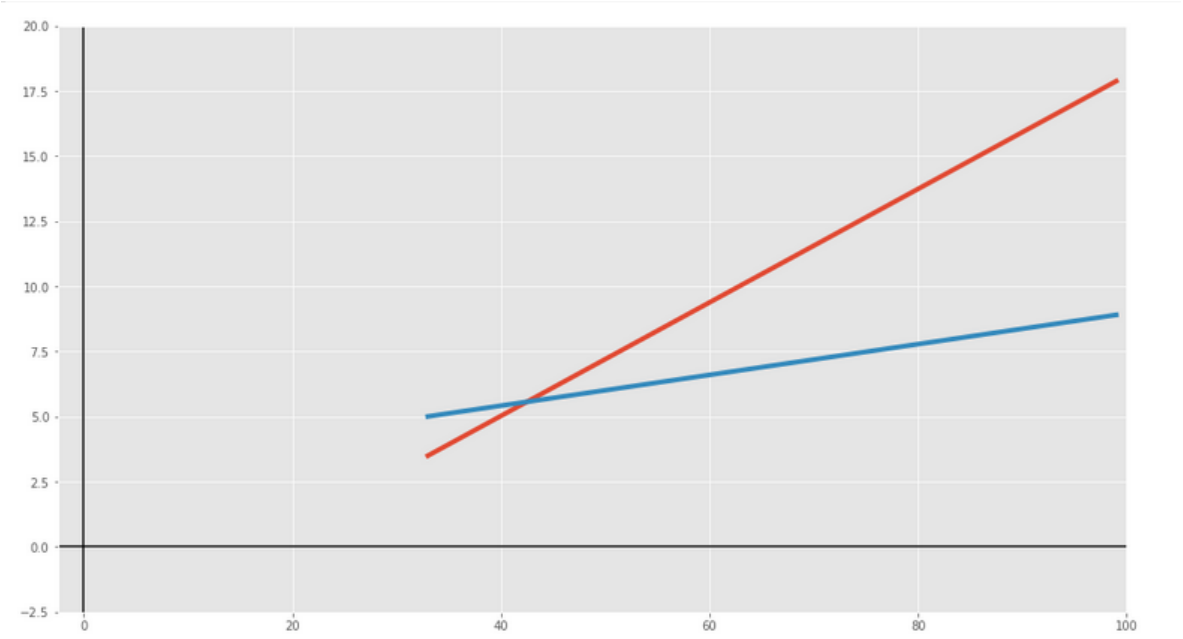
La primer grafica resultante con las dos ecuaciones y valores son como sigue

```
[ ]  plt.scatter(X2_train, y2_train,  color='red')
     plt.plot(X2_train, y2_pred, color='orange', linewidth=2)
     plt.scatter(X_train, y_train,  color='blue')
     plt.plot(X_train, y_pred, color='green', linewidth=2)
     plt.show()
```



Pero estos datos no se encuentran en la misma dimensión por lo que vamos a darle un tratamiento a los datos de salario para hacer su comportamiento en incremento porcentual, en la misma dimensión de la inflación

```
[ ]  def f1(x):
         return (((0.00585159*x + 2.5879972140535528) * 100)/2.6870968) - 100
     # Función lineal.
     def f2(x):
         return 0.05904316*x + 3.055236805448073
     # Valores del eje X que toma el gráfico.
     x = range(33, 100)
     # Graficar ambas funciones.
     pyplot.plot(x, [f1(i) for i in x] , linewidth=4)
     pyplot.plot(x, [f2(i) for i in x], linewidth=4)
     # Establecer el color de los ejes.
     pyplot.axhline(0, color="black")
     pyplot.axvline(0, color="black")
     # Limitar los valores de los ejes.
     pyplot.xlim(-2.5, 100)
     pyplot.ylim(-2.5, 20)
     # Guardar gráfico como imágen PNG.
     pyplot.savefig("output.png")
     # Mostrarlo.
     pyplot.show()
```



Incremento de salario (ROJO)

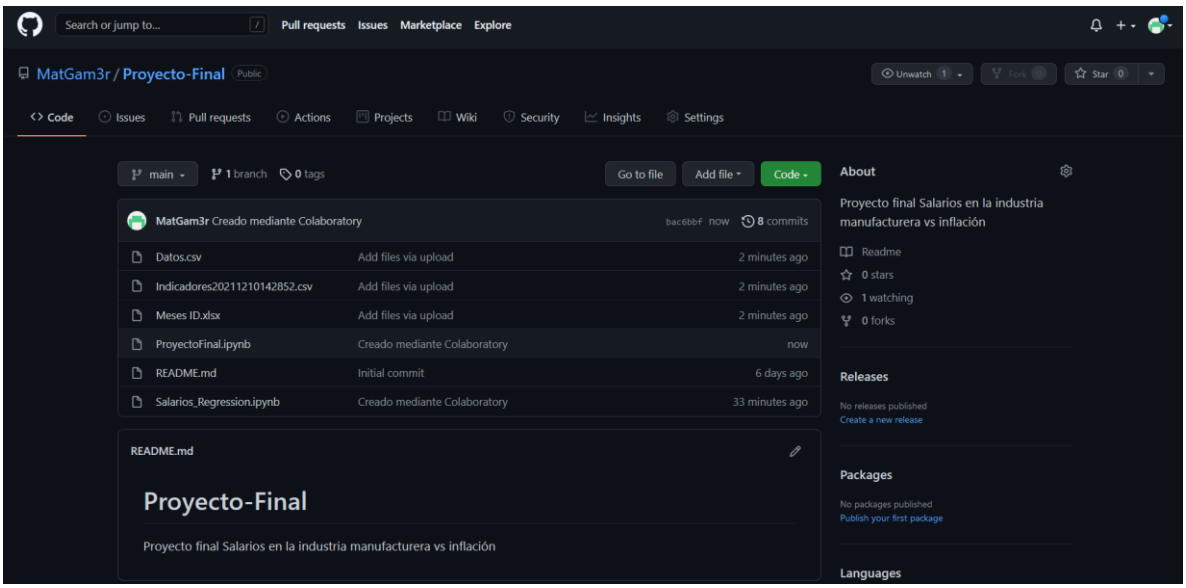
Incremento de la Inflación (AZUL)

Evaluación automatizada cambiando el valor de “p”

```
#Vamos a comprobar:
# según nuestro modelo, hacemos:
p =49
y_Inflacion2022 = regr2.predict([[p]])
print("Inflacion")
print(float(y_Inflacion2022))
y_Septiembre2022 = regr.predict([[p]])
print ("DolaresvsHora")
print(float(y_Septiembre2022))
```

RESULTADOS

Todos los códigos y base de datos se encuentran en un Repositorio de GITHUB para disponibilidad de la comunidad en general.



https://github.com/MatGam3r/Proyecto-Final

En respuesta a las preguntas realizadas, podemos concluir lo siguiente

Las gráficas y comparativas muestran que la tendencia del incremento de salario es superior a la inflación en México, manteniendo el poder adquisitivo de la población que trabaja en la industria manufacturera

REFERENCIAS

https://www.aprendemachinelearning.com/regresion-lineal-en-espanol-con-python/

https://github.com/MatGam3r/Proyecto-Final

https://recursospython.com/codigos-de-fuente/graficar-funciones-matplotlib/

https://yuasaavedraco.github.io/Docs/Regresi%C3%B3n_Lineal_M%C3%BAltiple_con_Python.html

https://www.iartificial.net/regresion-polinomica-en-python-con-scikit-learn/

https://www.inegi.org.mx/app/tabulados/default.html?nc=539&idrt=18&opc=t

https://www.inegi.org.mx/temas/inpc/