Actividad 6
Regresión Lineal y No Lineal

10 de Octubre 2023
ITESM Puebla

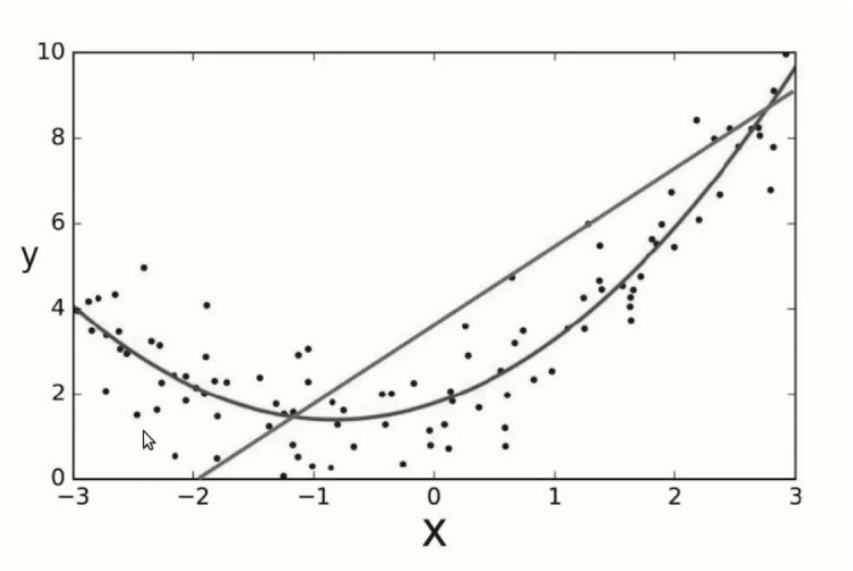
Ángel Rubén Vazquez Rivera (A01735407)
José Israel Pérez Ontiveros (A01423294)
Maximiliano Romero Budib (A01732008)

Índice

1. Introducción

El análisis de datos, involucra varios procesos con el objetivo de extraer información importante de un conjunto de datos. Esté análisis implica no solo ver la información, si no también modificarla (Eliminar datos nulos y datos atípicos) y graficarla. Uno de los elementos más importantes de esta segunda característica es el análisis de regresión lineal y no lineal.

Ambas técnicas de análisis tienen un mismo objetivo: modelar la relación de los datos, pero cada una tiene sus peculiaridades que las vuelve poderosas para diferentes casos. La regresión lineal es, en los términos más simples, una modelación de la relación entre dos variables, esto nos permite visualizar cómo se comportan las variables dependientes contra sus contrapartes independientes, esto nos permite predecir los valores de las variables dependientes, y nos permite dar un análisis más exacto. La desventaja, por otro lado, es que es un análisis muy rígido de las relaciones entre variables, y puede llevar a una gráfica muy imprecisa para una relación muy compleja. Para este tipo de relaciones es cuando utilizamos la regresión no lineal. Este tipo de gráfico nos permite modelar relaciones más complejas como por ejemplo un inicio débil junto a un final débil pero con un punto medio alto, un tipo de relación imposible de graficar con regresión lineal pero posible con regresión no lineal.



Imágen 1. Comparación de regresiones

Incluso viendo esto, aún vale la pena revisar los dos tipos de regresión, esto debido a que, aunque la regresión lineal no sirve tanto para modelar relaciones más complejas, si nos permite ver cómo se desarrolla la información y nos ayuda a predecir su comportamiento.

En este caso, nuestra actividad requirió de un análisis de un dataframe teniendo en cuenta la correlación entre las variables y usando la regresión lineal y no lineal para graficar.

2. Desarrollo

2.1 Extracción de datos:

El primer paso de este análisis es la extracción de los datos, y para esto usamos varias técnicas de extracción de datos como lo es la graficación de la información para así visualizar no solo la cantidad de información a nuestra disposición sino también sus relaciones.

Con la extracción de datos lista, algunos datos interesantes que encontramos fueron los siguientes:

- El 87 % de la población se encuentra fuera de riesgo
- El rango de edades va de 21 a 79 años de edad

```
table = freq_tbl(dataClean['Age'].astype("string"))
Filtro = table[table['frequency']>9]
Filtro_setter = Filtro.set_index("Age")
Filtro_setter
Filtro_setter.plot(kind = "bar" , width=1, figsize=(10,7))
plt.title('Rango de edades')
plt.xlabel('Edades')
plt.ylabel('Personas')
```

```
Text(0, 0.5, 'Personas')
```

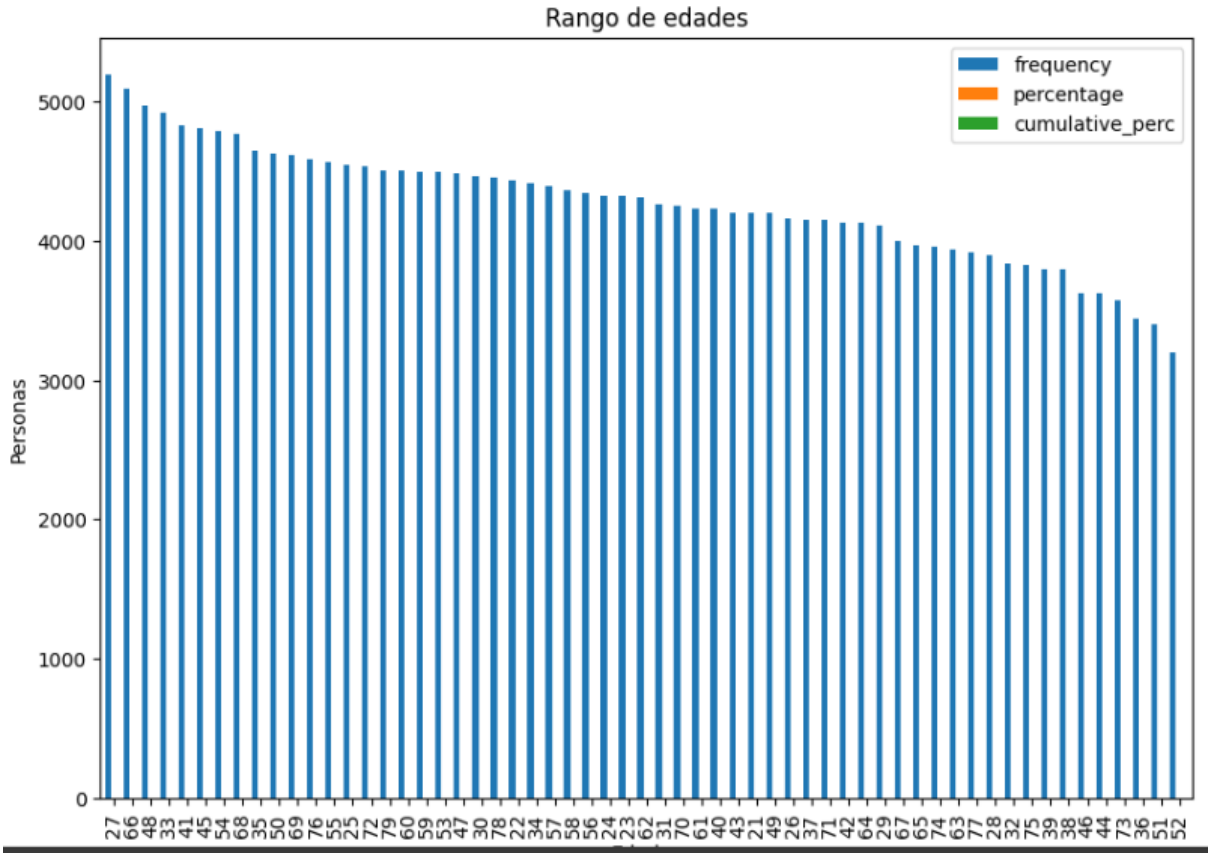


Imagen 2 - Rango de edades

- Experiencia de trabajo va desde 0 a 20 años, con 6 año siendo lo más común
- La mayoría (89.8 %) está soltera
- La mayoría (92%) renta su vivienda, mientras que el 5.1% es dueña y el 2.9% no tiene vivienda propia

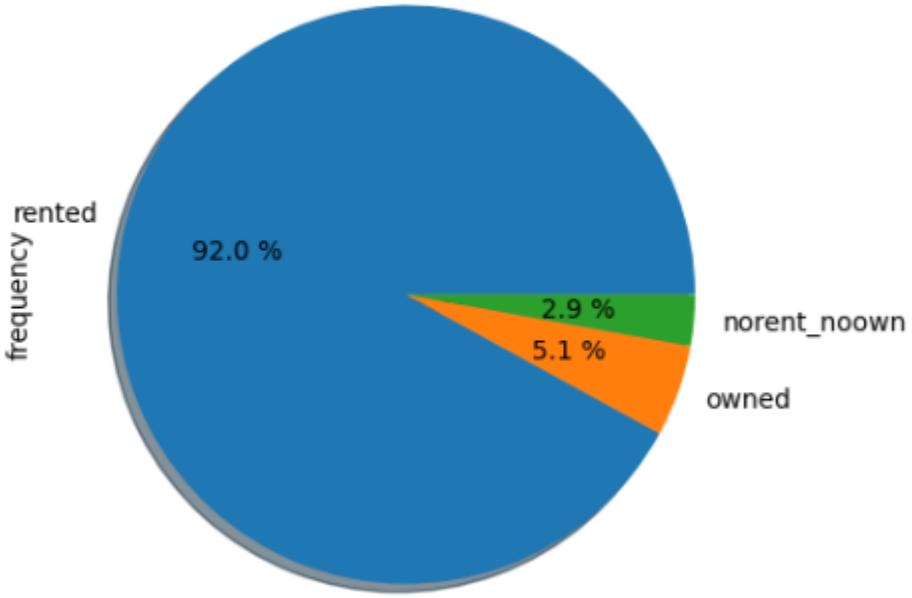


Imagen 3 - Rango de viviendas

- Existen 51 profesiones distintas, Ingeniero siendo la más baja y médico siendo el más común
- Existen 317 ciudades, la más popular siendo Vijayanagaram, y la mayoría de las ciudades siendo de medio oriente y asia (Por la cantidad de ciudades en el dataframe, nos enfocamos en graficar solo las 20 más populares).
- La mayoría de las personas han trabajado poco tiempo en su trabajo actual, 14 años siendo la cantidad más baja de años de trabajo
- Todas las personas tienen un mínimo de 10 años en su vivienda actual

2.2 Los 3 mejores modelos de regresión lineal simple.

El siguiente paso para nuestro análisis es analizar la correlación de los datos que nosotros extraemos, y así encontrar los 3 mejores modelos de regresión lineal simple.

Lo primero que se tuvo que hacer fue separar las variables entre cualitativas y cuantitativas para hacer la graficación correcta sin ningún problema de las variables. Después de esto, graficamos la dispersión de las variables.

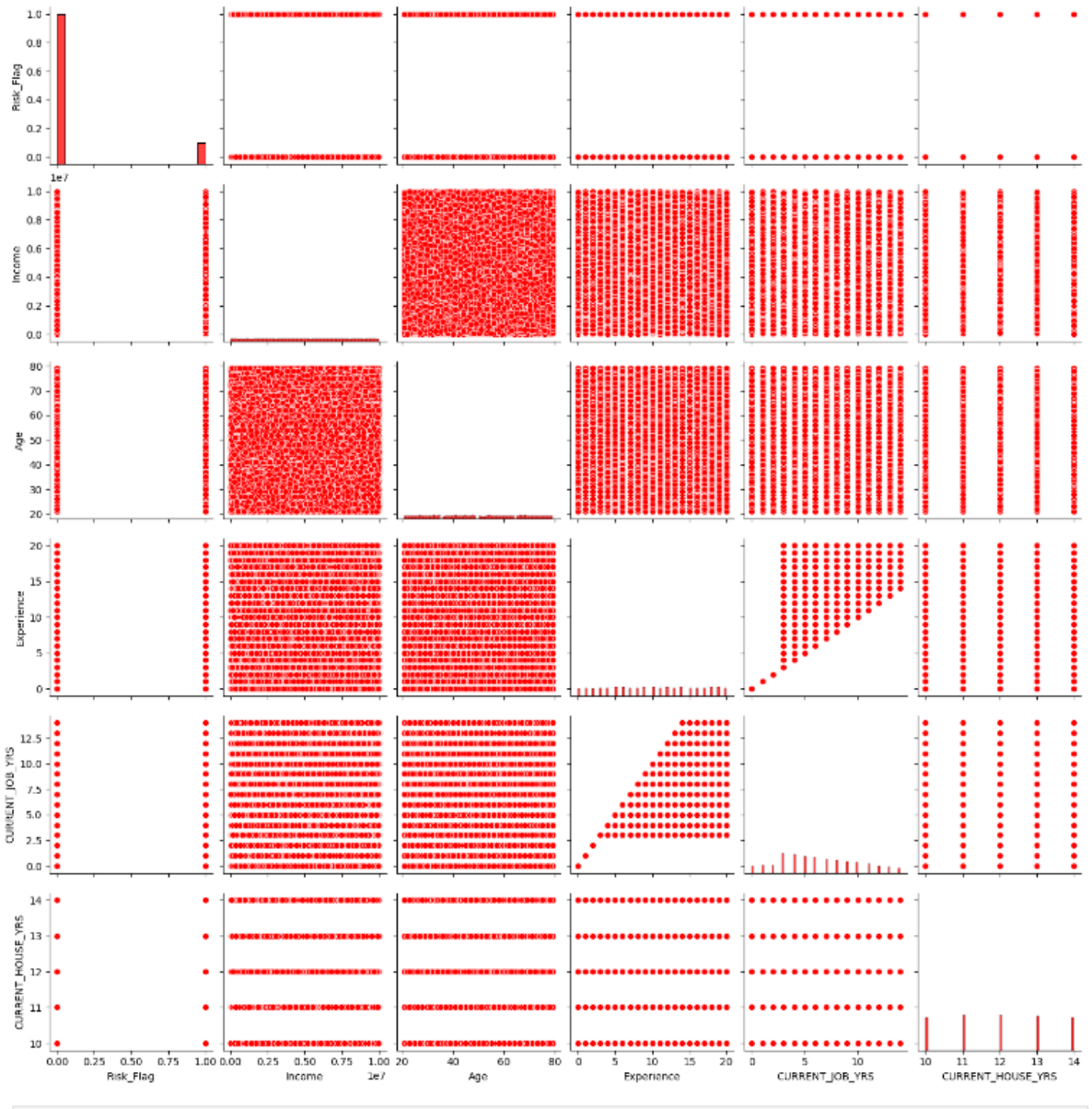


Imagen 4 - Gráficas de dispersión

Y ahora, lo más importante, después de conseguir la correlación, la graficamos con un mapa de calor para encontrar de manera visual las correlaciones más fuertes que podemos usar en nuestro análisis.

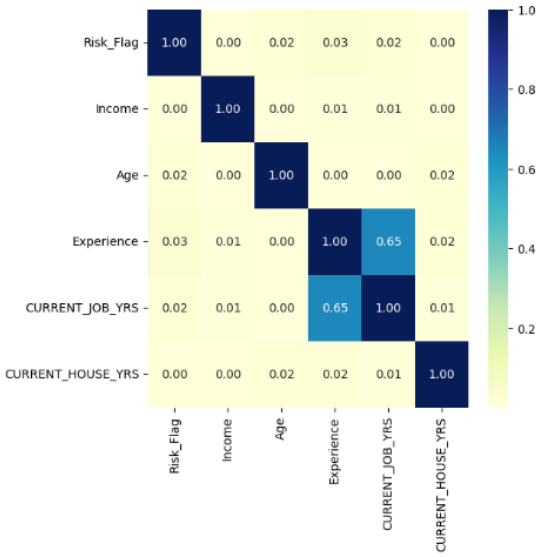


Imagen 5 - Mapa de calor de correlación

Analizando la gráfica, no tenemos mucho con lo cual trabajar, las únicas variables que nos permiten algún tipo de análisis relativamente fructífero son CURRENT_JOB_YRS y Experience.

Lo siguiente que tenemos que hacer con esta información es empezar la regresión lineal simple. Primero empezamos graficando los datos que encontramos que tienen un alto coeficiente de correlación (CURRENT_JOB_YRS y Experience)

```
sns.scatterplot(x='CURRENT_JOB_YRS', y='Experience', color="blue", data=dataCuantitativas)
```

```
<Axes: xlabel='CURRENT_JOB_YRS', ylabel='Experience'>
```

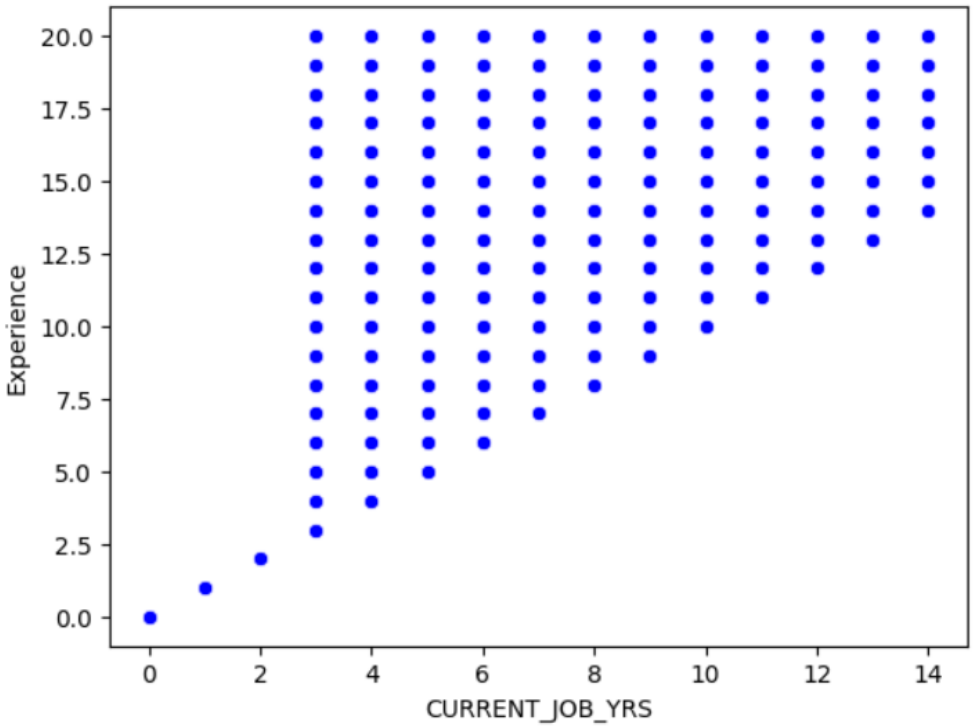


Imagen 6 - Gráfica scatter de Experience y CURRENT_JOB_YRS

Y empezamos con la regresión lineal, que involucra los siguientes pasos:

- Declaramos las variables dependientes e independientes
    - Dependiente = Experience3
    - Independiente = Current Job Years
- Definimos el modelo como la función de regresión lineal
- Ajustamos el modelo con las variables declaradas
- Y procedemos a usar ese modelo para predecir

Lo anterior resulta en la siguiente gráfica:

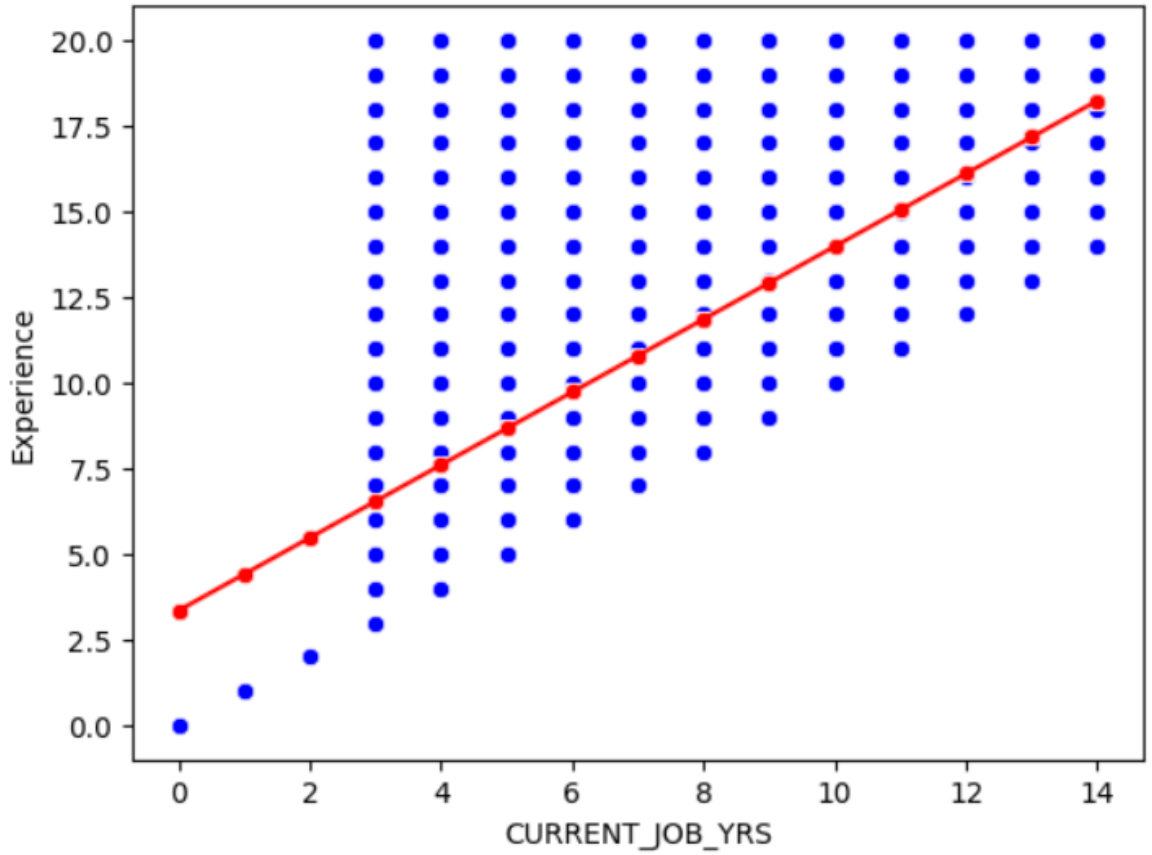


Imagen 7 - Predicción del modelo

Gracias a la regresión lineal, logramos observar que el modelo logró predecir el comportamiento de las variables Experience y CURRENT_JOB_YRS, y muestra un comportamiento ascendente.

Corroborando esta información conseguimos el coeficiente de determinación y correlación:

```
#Corroboramos cual es el coeficiente de Determinación de nuestro modelo
coef_Deter=model.score(X=Vars_Indep, y=Var_Dep)
print(f'Coeficiente de determinacion: {round(coef_Deter,2)}')
```

```
Coeficiente de determinacion: 0.42
```

```
#Corroboramos cual es el coeficiente de Correlación de nuestro modelo
coef_Correl=np.sqrt(coef_Deter)
print(f'Coeficiente de correlacion: {round(coef_Correl,2)}')
```

```
Coeficiente de correlacion: 0.65
```

Imagen 8 - Coeficientes de Experience y Current Job Yrs

Algo que tenemos que aclarar es que la razón por la cual solo tenemos un modelo, es debido a estos datos, ya que en otros modelos probados, el coeficiente era tan bajo que el análisis ya no era viable

2.3 Regresión lineal múltiple

Esta es la parte más complicada de este reporte, por diferentes razones. Debido al carácter de la regresión lineal, este intentará mostrar una predicción de los datos. Pero como ya se había mencionado antes, es algo rígido en cuanto a su modelaje, por lo cual datos más complejos pueden llegar a mostrarse de una manera casi incomprensible.

A continuación explicaremos el proceso para los modelajes y mostraremos los resultados de cada graficación.

El primer paso es Declarar las variables dependientes e independientes (En este caso, utilizaremos Income, junto a Experience, Age, CURRENT_JOB_YRS y CURRENT_HOUSE_YRS)

```
#Declaramos las variables dependientes e independientes para la regresión lineal
Vars_IndepIncome= dataCuantitativas[['Experience', 'Age', 'CURRENT_JOB_YRS', 'CURRENT_HOUSE_YRS']]
Var_DepIncome= dataCuantitativas['Income']
```

Imagen 9 - Declaración de variables independientes y dependientes

Definimos el modelo para luego ajustar el modelo con las variables

```
#Ajustamos el modelo con las variables antes declaradas
modelIncome.fit(X=Vars_IndepIncome, y=Var_DepIncome)
```

Imagen 10 - Ajustamos el modelo

Evaluamos la eficiencia del modelo

```
modelIncome.score(Vars_IndepIncome,Var_DepIncome)
```

```
6.23149509521026e-05
```

Imagen 11 - Evaluación del modelo

Hacemos las predicciones

```
y_predIncome= modelIncome.predict(X=dataCuantitativas[['Experience', 'Age', 'CURRENT_JOB_YRS', 'CURRENT_HOUSE_YRS']])
y_predIncome
```

```
array([4971050.10324344, 5003418.99033241, 4986817.10040466, ...,
       4995332.99506377, 4967441.01602864, 5013313.43842278])
```

Imagen 12 - Predicciónes

Insertamos la columna de predicciones en el DataFrame

```
#Insertamos la columna de predicciones en el DataFrame
dataCuantitativas.insert(0, 'PrediccionesRMIncome', y_predIncome)
dataCuantitativas
```

Id	PrediccionesRMIncome	PrediccionesRL	Risk_Flag	Income	Age	Experience	CURRENT_JOB_YRS	CURRENT_HOUSE_YRS
1	4.971050e+06	6.539208	0	1303834	23	3	3	13
2	5.003419e+06	12.919579	0	7574516	40	10	9	13
3	4.986817e+06	7.602603	0	3991815	66	4	4	10
4	4.968571e+06	5.475813	1	6256451	41	2	2	12
5	4.975598e+06	6.539208	1	5768871	47	11	3	14
...	...	...	...	...	...	...	...	...
251996	5.006359e+06	9.729393	0	8154883	43	13	6	11
251997	5.003681e+06	9.729393	0	2843572	26	10	6	11

Imagen 13 - Inserción de datos

Y ahora visualizamos la gráfica comparativa entre el total real y el predecido (Aquí mostramos Experience contra Income

```
#Visualizamos la gráfica comparativa entre el total real y el total predecido

sns.scatterplot(x='Experience', y='Income', color="blue", data=dataCuantitativas)
sns.scatterplot(x='Experience', y='PrediccionesRMIncome', color="red", data=dataCuantitativas)
```

```
<Axes: xlabel='Experience', ylabel='Income'>
```

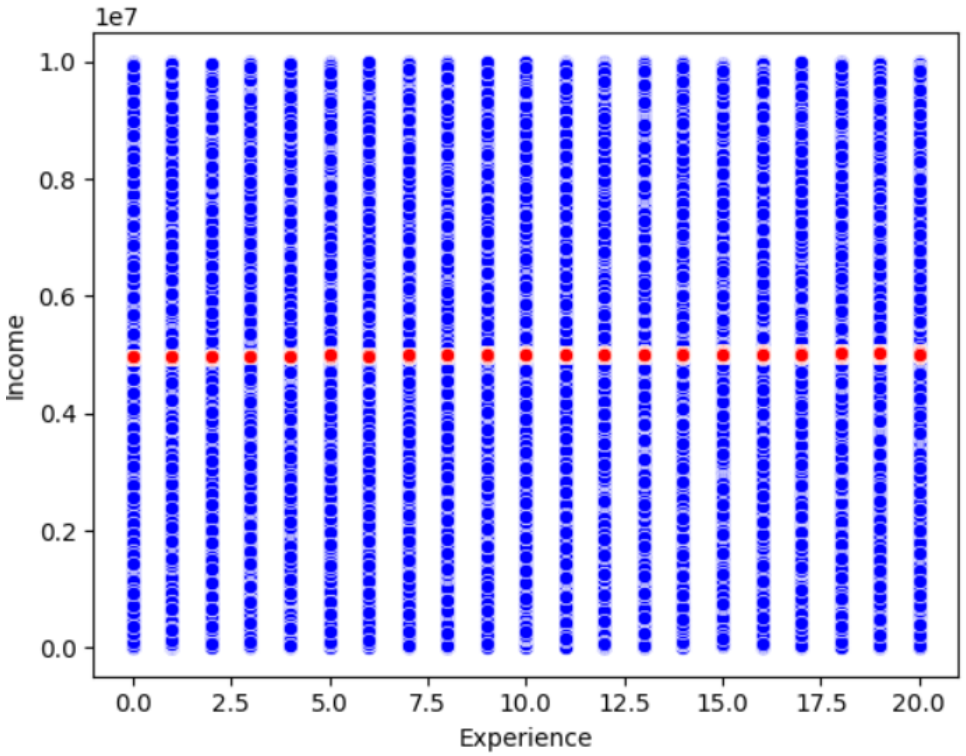


Imagen 14 - Predicción Experience Income

La predicción, aunque sea complicada, se puede explicar: Debido a la enorme cantidad de datos que se encuentran en el data frame, la predicción trata de crear un modelo lo más apegado a los datos, y esto termina en la gráfica que se puede observar anteriormente. La gráfica es correcta, simplemente nos muestra las limitaciones de un modelo de regresión lineal. Este ejemplo se puede ver claramente en la siguiente gráfica:

```
#Visualizamos la gráfica comparativa entre el total real y el total predecido

sns.scatterplot(x='Age', y='Income', color="blue", data=dataCuantitativas)
sns.scatterplot(x='Age', y='PrediccionesRMIncome', color="red", data=dataCuantitativas)
```

```
<Axes: xlabel='Age', ylabel='Income'>
```

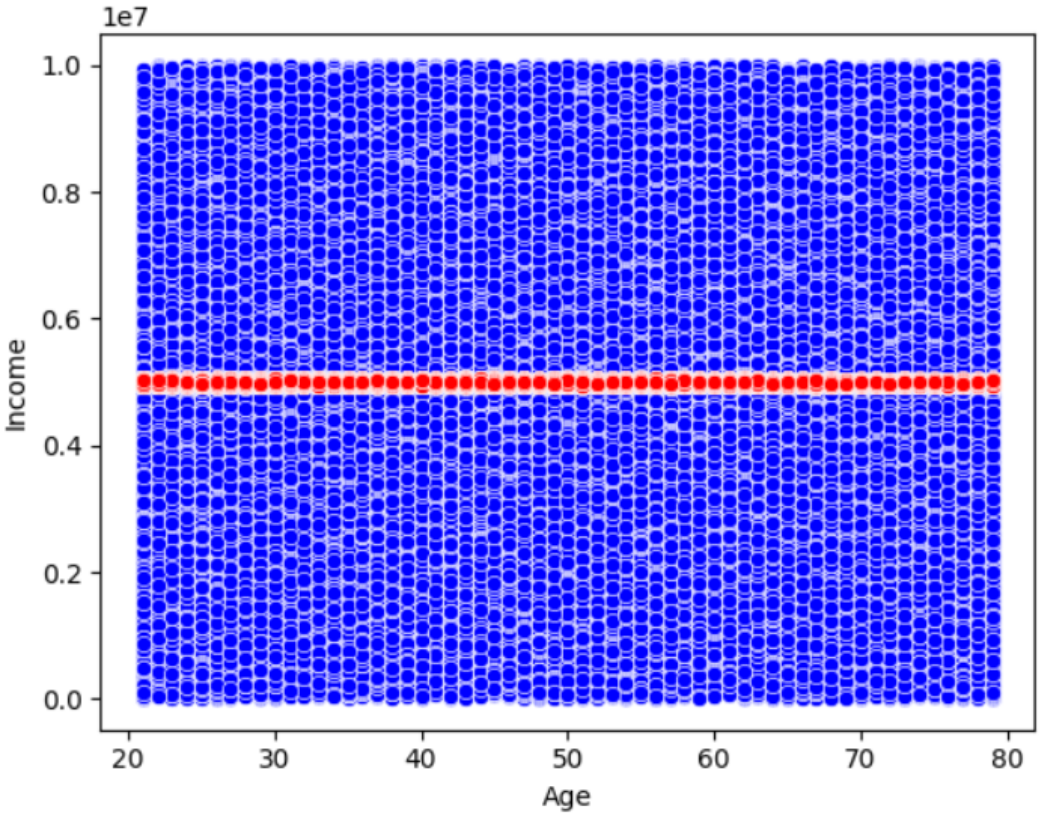


Imagen 15 - Predicción Age Income

Procederemos a mostrar las gráficas de Income y al final explicaremos nuestro análisis.

```
#Visualizamos la gráfica comparativa entre el total real y el total predecido

sns.scatterplot(x='CURRENT_JOB_YRS', y='Income', color="blue", data=dataCuantitativas)
sns.scatterplot(x='CURRENT_JOB_YRS', y='PrediccionesRMIncome', color="red", data=dataCuantitativas)
```

```
<Axes: xlabel='CURRENT_JOB_YRS', ylabel='Income'>
```

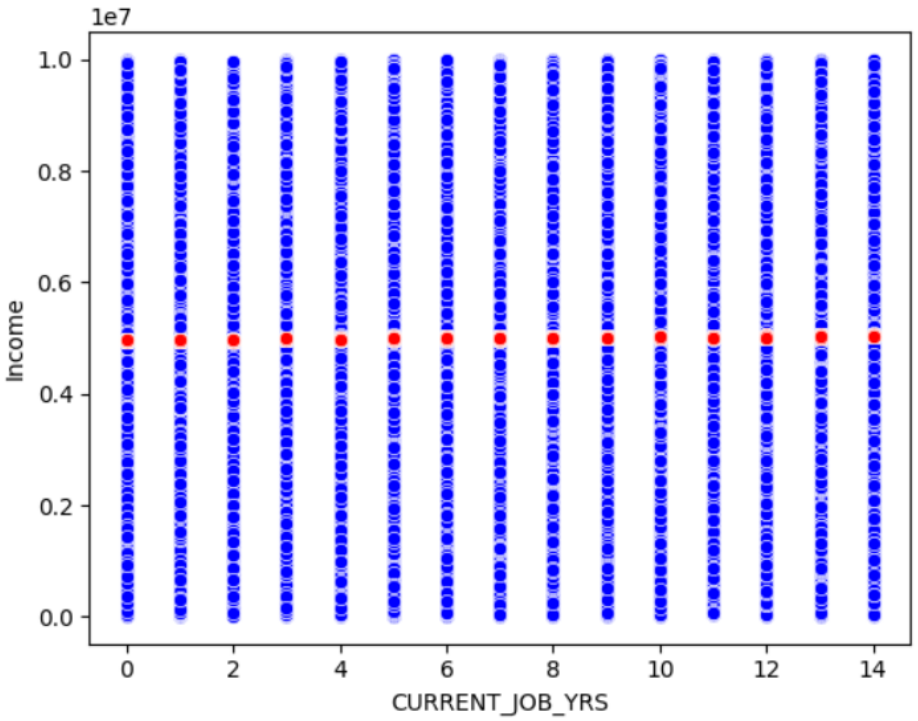


Imagen 16 - Predicción CURRENT_JOB_YRS Income

```
#Visualizamos la gráfica comparativa entre el total real y el total predecido

sns.scatterplot(x='CURRENT_HOUSE_YRS', y='Income', color="blue", data=dataCuantitativas)
sns.scatterplot(x='CURRENT_HOUSE_YRS', y='PrediccionesRMIncome', color="red", data=dataCuantitativas)
```

```
<Axes: xlabel='CURRENT_HOUSE_YRS', ylabel='Income'>
```

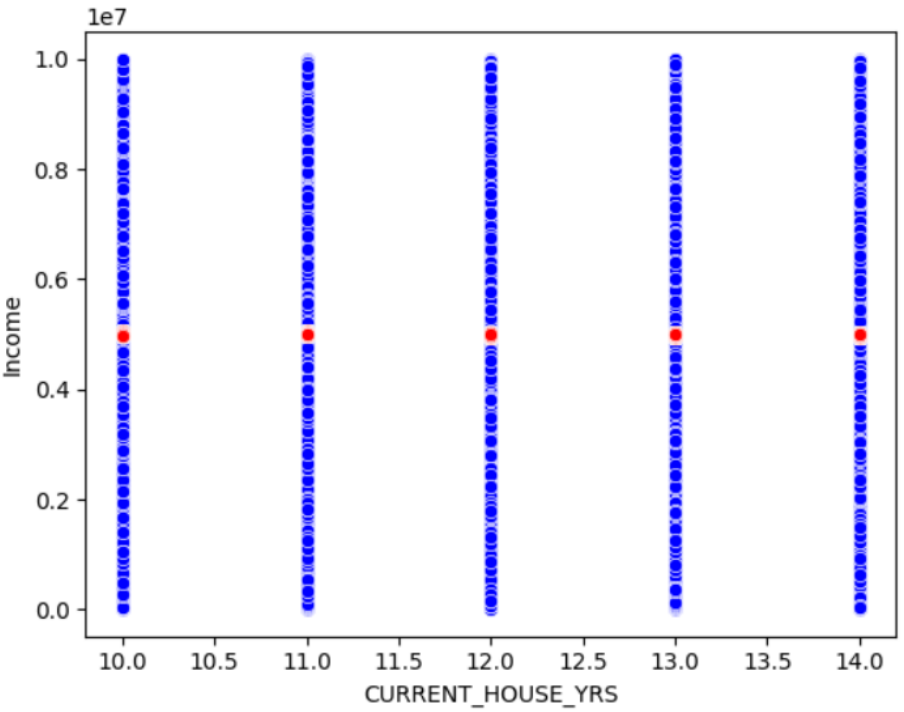


Imagen 17 - Predicción CURRENT_HOUSE_YRS Income

La mayoría de las gráficas en general tuvieron este tipo de comportamiento a excepción de algunos casos específicos, así que un simple análisis visual no es lo suficientemente bueno como para declarar el modelo como bueno, por lo cual utilizamos correlación y determinación para encontrar que tan efectivo fue el modelo.

```
#Corroboramos cual es el coeficiente de Determinación de nuestro modelo
coef_DeterIncome=modelIncome.score(X=Vars_IndepIncome, y=Var_DepIncome)
print(f'Coeficiente de determinacion: {coef_DeterIncome}')
```

```
Coeficiente de determinacion: 6.23149509521026e-05
```

```
#Corroboramos cual es el coeficiente de Correlación de nuestro modelo
coef_CorrelIncome=np.sqrt(coef_DeterIncome)
print(f'Coeficiente de correlacion: {coef_CorrelIncome}')
```

```
Coeficiente de correlacion: 0.007893981945260744
```

Imagen 18 - Coeficientes de variable dependientes

El proceso fue repetido múltiples veces con diferentes variables, estas siendo:

- Income:
  - Previamente mencionado
- Age

```
#Corroboramos cual es el coeficiente de Determinación de nuestro modelo
coef_DeterAge=modelAge.score(X=Vars_IndepAge, y=Var_DepAge)
print(f'Coeficiente de determinacion: {coef_DeterAge}')
```

```
Coeficiente de determinacion: 0.00041925413579979587
```

```
#Corroboramos cual es el coeficiente de Correlación de nuestro modelo
coef_CorrelAge=np.sqrt(coef_DeterAge)
print(f'Coeficiente de correlacion: {coef_CorrelAge}')
```

```
Coeficiente de correlacion: 0.020475696222590233
```

Imagen 19 - Coeficientes Age

- Experience

```
#Corroboramos cual es el coeficiente de Determinación de nuestro modelo
coef_DeterExperience=modelExperience.score(X=Vars_IndepExperience, y=Var_DepExperience)
print(f'Coeficiente de determinacion: {coef_DeterExperience}')
```

```
Coeficiente de determinacion: 0.4177012934333135
```

```
#Corroboramos cual es el coeficiente de Correlación de nuestro modelo
coef_CorrelExperience=np.sqrt(coef_DeterExperience)
print(f'Coeficiente de correlacion: {coef_CorrelExperience}')
```

```
Coeficiente de correlacion: 0.6462981459305864
```

Imagen 20 - Coeficientes Experience

- CURRENT_JOB_YRS

```
#Corroboramos cual es el coeficiente de Determinación de nuestro modelo
coef_DeterJob=modelJob.score(X=Vars_IndepJob, y=Var_DepJob)
print(f'Coeficiente de determinacion: {coef_DeterJob}')
```

```
Coeficiente de determinacion: 0.4175082512923083
```

```
#Corroboramos cual es el coeficiente de Correlación de nuestro modelo
coef_CorrelJob=np.sqrt(coef_DeterJob)
print(f'Coeficiente de correlacion: {coef_CorrelJob}')
```

```
Coeficiente de correlacion: 0.6461487841761434
```

Imagen 21 - Coeficiente CURRENT_JOB_YRS

- CURRENT_HOUSE_YRS

```
#Corroboramos cual es el coeficiente de Determinación de nuestro modelo
coef_DeterHouse=modelHouse.score(X=Vars_IndepHouse, y=Var_DepHouse)
print(f'Coeficiente de determinacion: {coef_DeterHouse}')
```

```
Coeficiente de determinacion: 0.0008688150174576137
```

```
#Corroboramos cual es el coeficiente de Correlación de nuestro modelo

# Notaaaa. La correlacion de CURRENT_HOUSE_YRS si mejora a comparacion al heatmap
# en el mapa de calor el max es de 0.02 y aqui es de 0.03



coef_CorrelHouse=np.sqrt(coef_DeterHouse)
print(f'Coeficiente de correlacion: {coef_CorrelHouse}')
```

```
Coeficiente de correlacion: 0.029475668227499332
```

Imagen 22 - Coeficientes CURRENT_HOUSE YRS

Podemos observar que el mejor modelo fue el utilizado en Current Job Years con un coeficiente de determinación de 0.42 y de correlación de 0.64.

2.4 Regresión no Lineal

Como se había explicado anteriormente, la regresión no lineal se utiliza cuando existe un comportamiento complejo entre las variables, por lo cual en papel, queda perfecto para este análisis debido a las complejas gráficas y datos que nosotros tenemos.

Los pasos son similares a la regresión lineal, primero declaramos las variables dependientes e independientes (empecemos con Income), ajustamos los parámetros de la función curve_fit y ahí podemos empezar a gráficar.

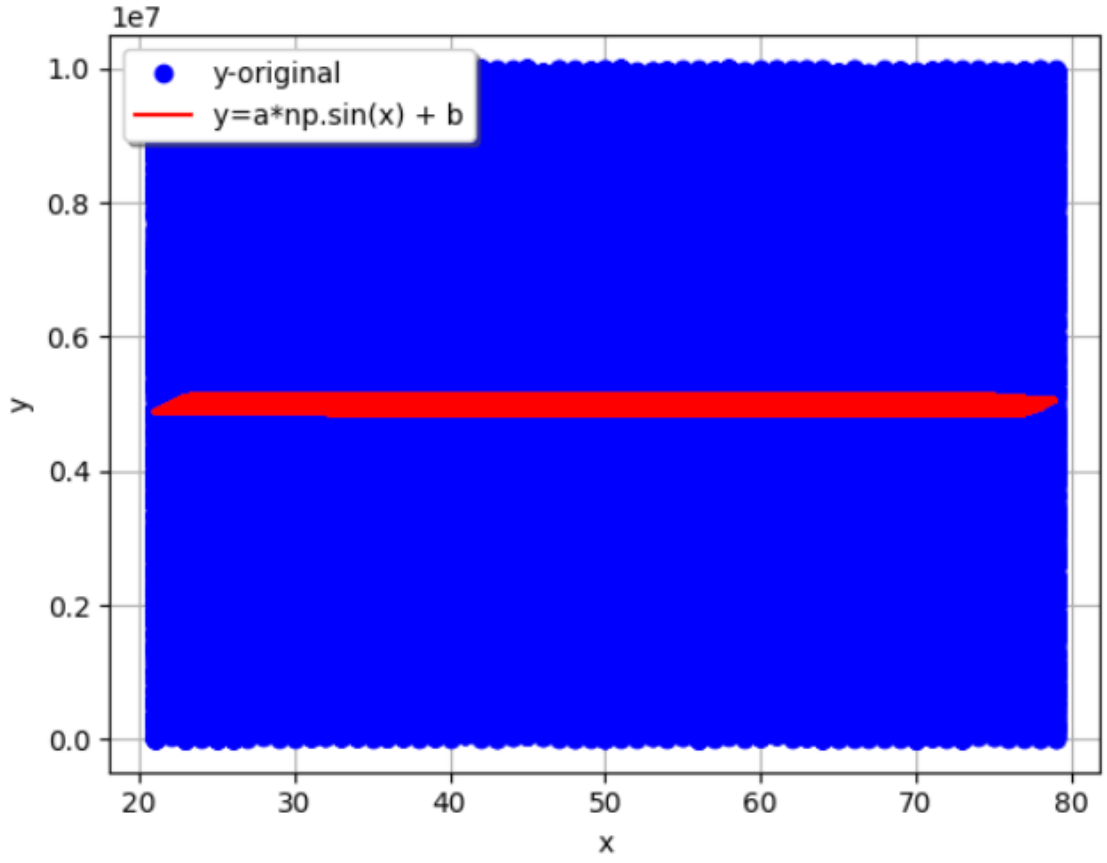
```
#Declaramos las variables dependientes e independientes para la regresión lineal
Vars_IndepIncomeRNL= dataCuantitativas[['Age']]
Var_DepIncomeRNL= dataCuantitativas['Income']
```

```
#Redefinimos las variables
xIncomeRNL= Vars_IndepIncomeRNL
yIncomeRNL= Var_DepIncomeRNL
```

```
#Ajustamos los parámetros de la función curve_fit
parametrosIncomeRNL, covsIncomeRNL= curve_fit(funSin, dataCuantitativas['Age'], dataCuantitativas['Income'])
aIncomeRNL,bIncomeRNL = parametrosIncomeRNL[ 0 ], parametrosIncomeRNL[ 1 ]
yfitIncomeRNL = aIncomeRNL*np.sin(xIncomeRNL) + bIncomeRNL
```

Imagen 23 - Declaración de variables y modelo

Después de graficar, nos quedamos con la siguiente información:




```
# Calculamos el coeficiente de determinación del modelo
r2IncomeRNL = r2_score(yIncomeRNL, yfitIncomeRNL)
print(f'Coeficiente de determinacion: {r2IncomeRNL}')
```

```
Coeficiente de determinacion: 0.0012110706289841788
```

```
# Calculamos el coeficiente de correlacion del modelo

# En este caso mejoró el coef de corr, siendo que en el heatmap es de 0.00 y
# con una funcion senoidal mejora a 0.03

rIncomeRNL=np.sqrt(r2IncomeRNL)
print(f'Coeficiente de correlacion: {rIncomeRNL}')
```

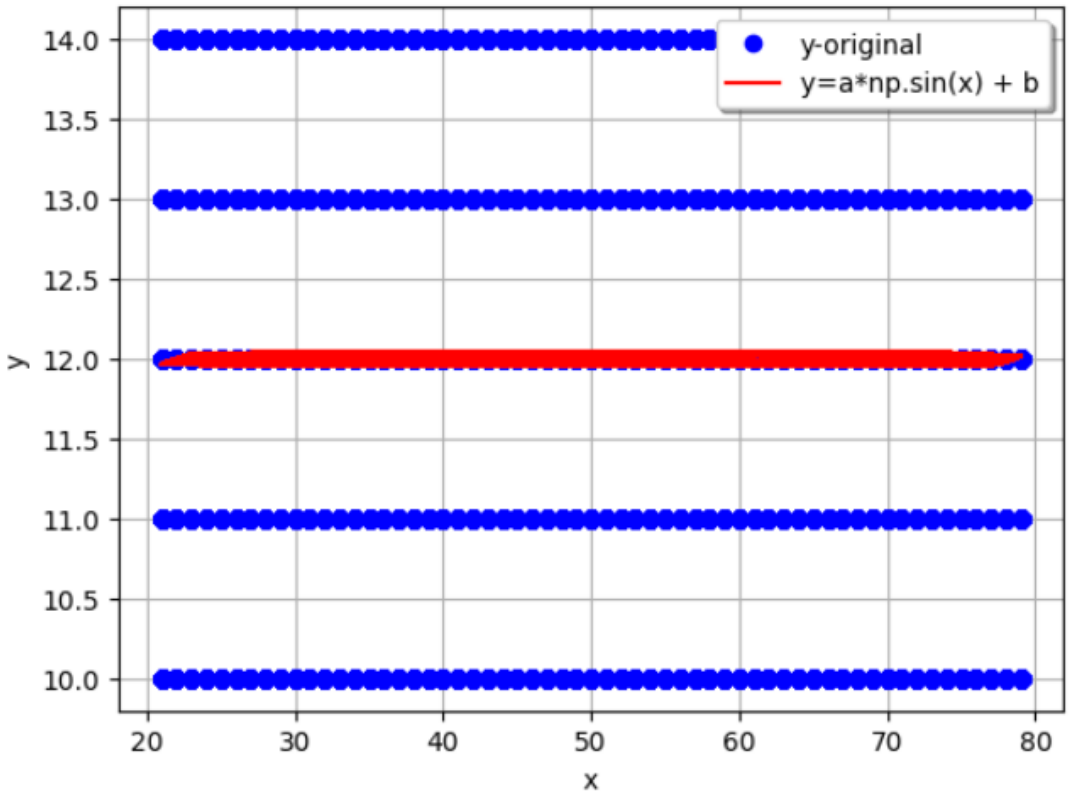
```
Coeficiente de correlacion: 0.034800440068829284
```

Imagen 24 - Gráfica no lineal y coeficientes

Podemos ver que la gráfica puede predecir con mayor área los datos, aunque la cantidad sigue siendo un problema.

Algo muy interesante que podemos ver es que este patrón se repite con las otras variables:

- CURRENT_HOUSE_YRS




```
# Calculamos el coeficiente de determinación del modelo
r2HouseRNL = r2_score(yHouseRNL, yfitHouseRNL)
print(f'Coeficiente de determinacion: {r2HouseRNL}')
```

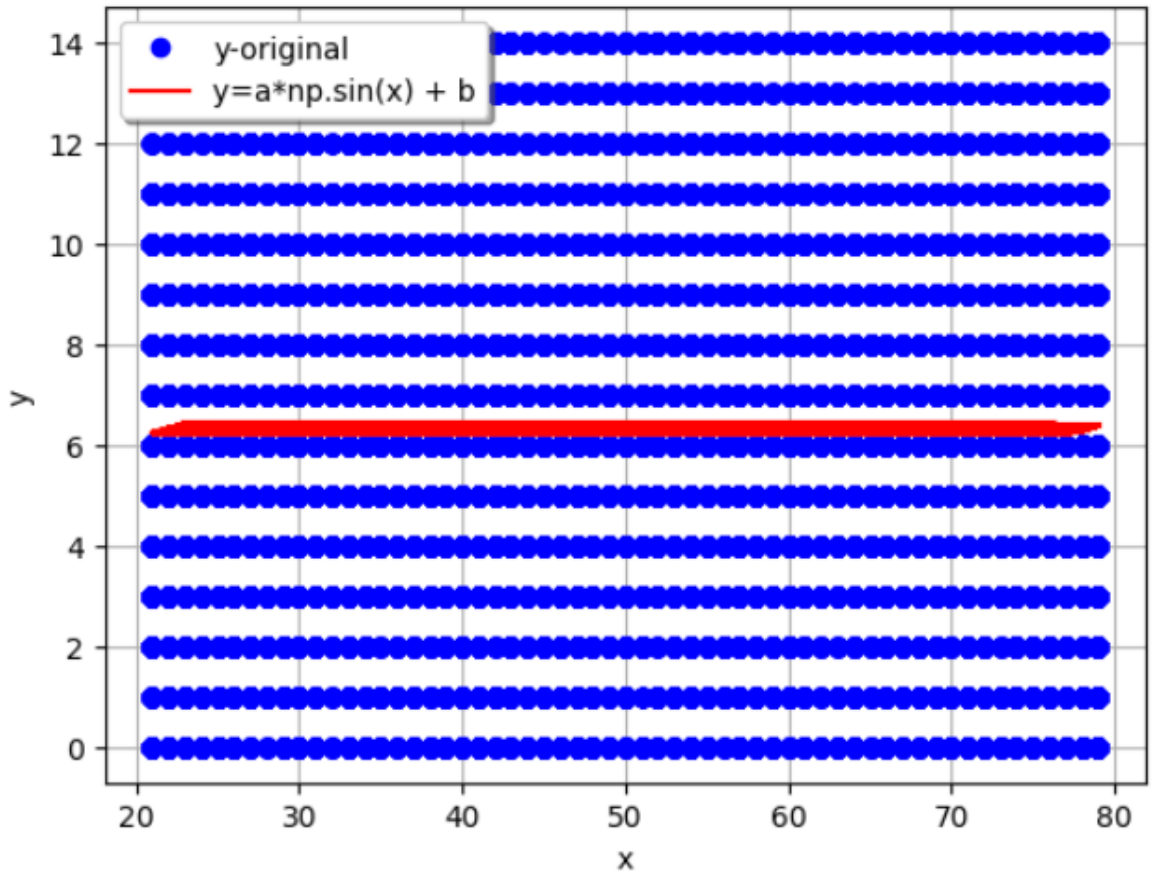
```
Coeficiente de determinacion: 0.0003694726336431753
```

```
# Calculamos el coeficiente de correlacion del modelo
rHouseRNL=np.sqrt(r2HouseRNL)

# En este caso no se encontró modelo que mejorara el factor de correlacion

print(f'Coeficiente de correlacion: {rHouseRNL}')
```

```
Coeficiente de correlacion: 0.01922167093785489
```

Imagen 24 - Gráfica no lineal y coeficientes de CURRENT_HOUSE_YRS

- CURRENT_JOB_YRS




```
# Calculamos el coeficiente de determinación del modelo
r2JobRNL = r2_score(yJobRNL, yfitJobRNL)
print(f'Coeficiente de determinacion: {r2JobRNL}')
```

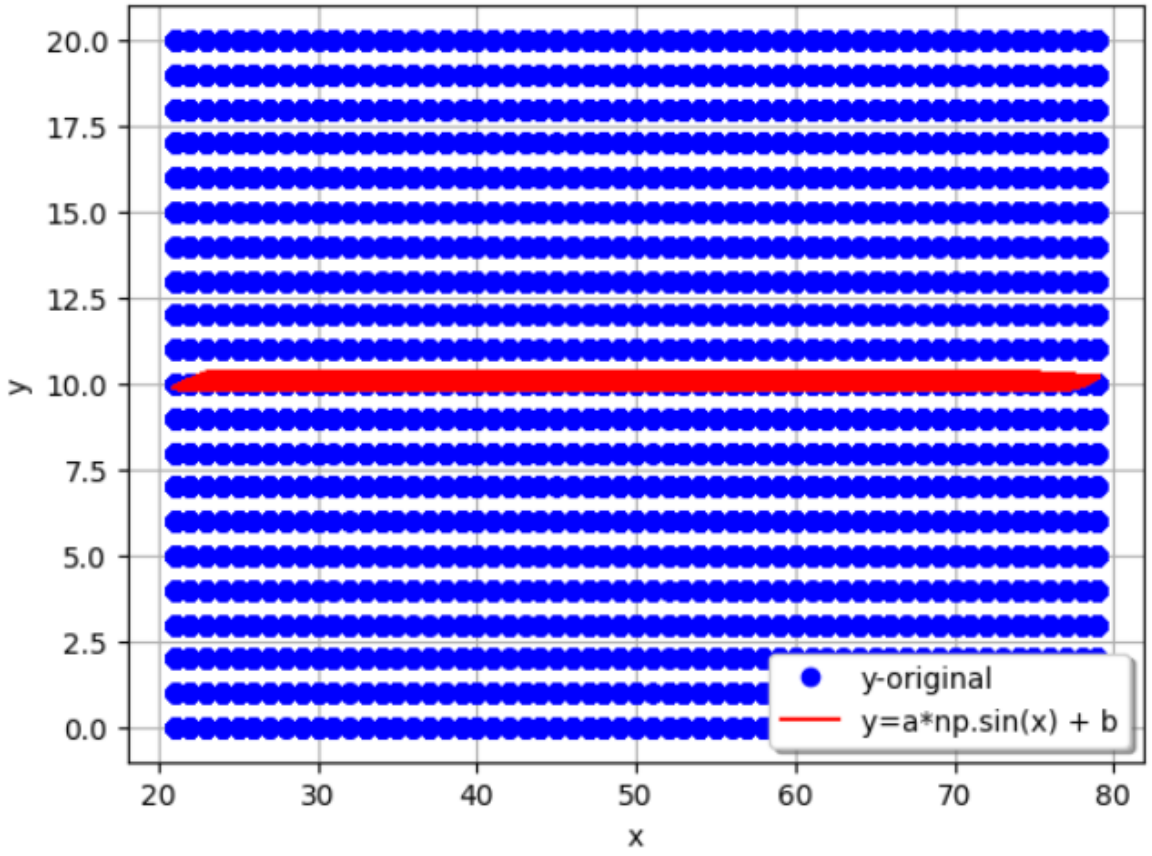
```
Coeficiente de determinacion: 0.00042208725262671276
```

```
# Calculamos el coeficiente de correlacion del modelo

# En este caso mejoró el coef de corr, siendo que en el heatmap es de 0.00 y
# con una funcion senoidal mejora a 0.02

rJobRNL=np.sqrt(r2JobRNL)
print(f'Coeficiente de correlacion: {rJobRNL}')
```

```
Coeficiente de correlacion: 0.020544762170118025
```

Imagen 24 - Gráfica no lineal y coeficientes de CURRENT_HOUSE_YRS

- Experience




```
# Calculamos el coeficiente de determinación del modelo
r2ExperienceRNL = r2_score(yExperienceRNL, yfitExperienceRNL)
print(f'Coeficiente de determinacion: {r2ExperienceRNL}')
```

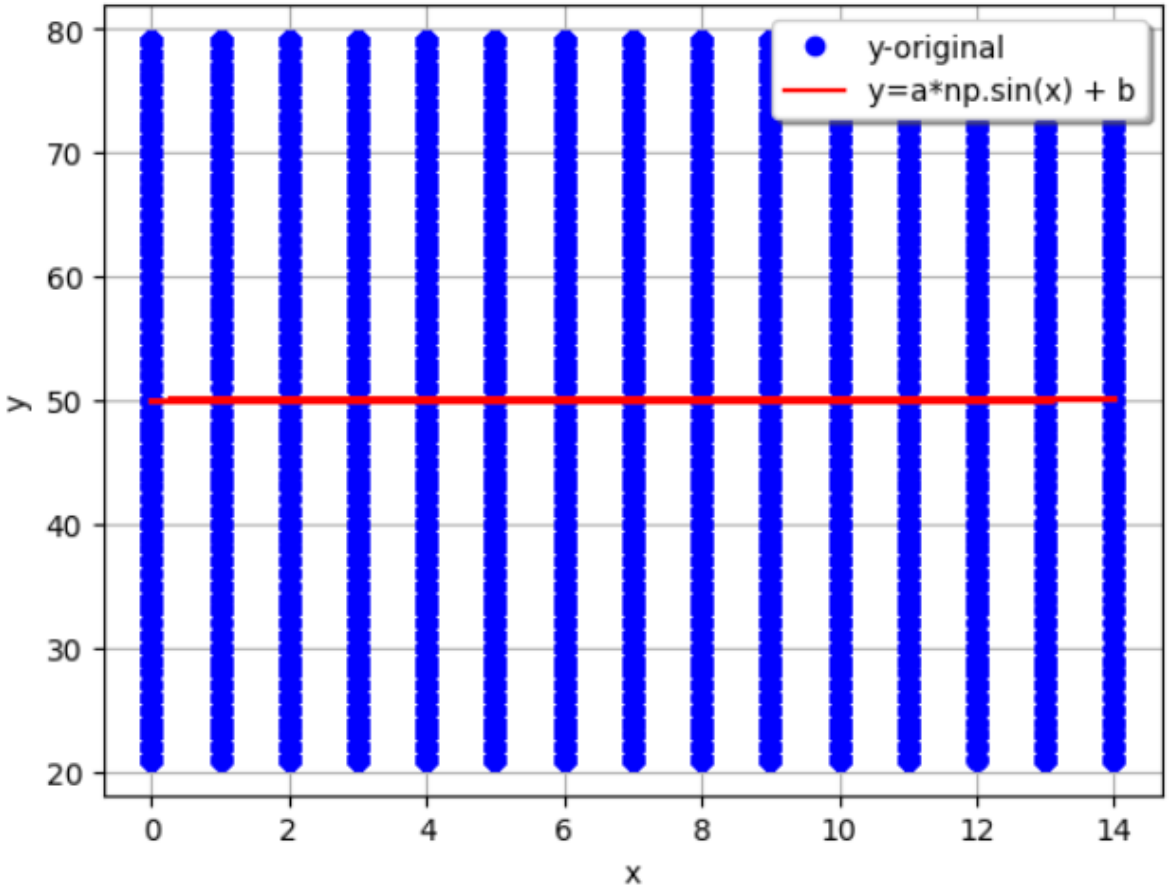
```
Coeficiente de determinacion: 0.0008181516979579584
```

```
# Calculamos el coeficiente de correlacion del modelo
rExperienceRNL=np.sqrt(r2ExperienceRNL)

# En este caso mejoró el coef de corr, siendo que en el heatmap es de 0.00 y
# con una funcion senoidal mejora a 0.02

print(f'Coeficiente de correlacion: {rExperienceRNL}')
```

```
Coeficiente de correlacion: 0.02860335116656715
```

Imagen 25 - Gráfica no lineal y coeficientes Experience

- Age




```
# Calculamos el coeficiente de determinación del modelo
r2AgeRNL = r2_score(yAgeRNL, yfitAgeRNL)
print(f'Coeficiente de determinacion: {r2AgeRNL}')
```

```
Coeficiente de determinacion: 2.03051040829072e-05
```

```
# Calculamos el coeficiente de correlacion del modelo
rAgeRNL=np.sqrt(r2AgeRNL)

# En este caso no se encontró modelo que mejorara el factor de correlacion

print(f'Coeficiente de correlacion: {rAgeRNL}')
```

```
Coeficiente de correlacion: 0.004506118516296171
```

Imagen 26 - Gráfica no lineal y coeficientes de Age161007

Al igual que en el punto anterior, incluso con regresión no lineal, la información es tan grande que los coeficientes son muy bajos para ser considerados óptimos, aunque podemos ver que Income llega a tener el mejor coeficiente de correlación y el de determinación.

2.5 Tabla de comparación

Correlación Regresión Lineal	Correlación Regresión Lineal Multiple	Correlación Regresión Multiple
0.65	0.008	0.03
N/A	0.02	0.02
N/A	0.65	0.02
N/A	0.65	0.03
N/A	0.03	0.005

En esta tabla de comparación, podemos ver los coeficientes de correlación de cada uno de los modelos. Cómo podemos ver: el mejor modelo de regresión fue el de regresión Lineal y lineal múltiple (pero el de múltiple fue de las mismas variables de la regresión lineal). Incluso después de utilizar la regresión no lineal descubrimos que el modelo no mejoró tanto con unas excepciones. Esto nos indica que el modelo probablemente no fue el mejor para el análisis. Definitivamente necesitamos revisar el modelo otra vez.

2.6 Conclusión

La información no siempre se tiene como uno desea, a veces son datos de cantidades tan exorbitantes que los modelos predictivos llegan a tener problemas, aún así, nosotros como equipo consideramos que este es el primer paso para un análisis completo de la información a nuestra disposición, ya que las predicciones de cómo se comportará la información son imperativos en el análisis y nos ayuda a entender la naturaleza de los datos.