Computación Científica
Laboratorio 2

Camilo Andrés Gutierrez Torres

Prof. Hernán Dario Vargas Cardona

Pontificia Universidad Javeriana Cali

Facultad de Ingeniería y Ciencias
Ingeniería de sistemas y Computación

Cali, Colombia
9 de octubre de 2020

1 Resumen

Este informe de laboratorio muestra el proceso realizado para encontrar, dado un conjunto de pares ordenados compuestos por una variable independiente (t) y una dependiente (y), un vector coeficientes con el cual se forma el polinomio que describe la recta que mejor se aproxime a los datos. Se evalúa si se obtuvo el mejor ajuste con el criterio de error mínimo cuadrático. Para hallar el polinomio, se utilizaron dos algoritmos de mínimos cuadrados, el primero es el método de ecucaciones normales y el segundo es trasformaciones Householder en mínimos cuadrados.

2 Abstract

This laboratory report shows the process carried out to find, given a set of ordered pairs composed of an independent variable (t) and a dependent variable (y), a vector of coefficients with which the polynomial that describes the line that best approximates the data is formed. It is evaluated whether the best fit was obtained with the minimum error criterion. To find the polynomial, two least squares algorithms were used, the first is the normal equations method and the second is Householder transformations in least squares.

3 Introducción

El problema de mínimos cuadrados busca hallar un vector x del mejor ajuste, el cuál cada uno de sus elementos representa los coeficientes del polinomio. Para resolver este problema existen métodos por medio de cálculo multivariable, lo que convierte mínimos cuadrados en un problema de optimización. Sin embargo, existen otros métodos hacia los cual está encaminado este laboratorio, que resuelven este problema por medio de álgebra lineal con los métodos de ecuaciones normales y transformaciones de Householder.

4 Materiales

Los materiales utilizados para la realización de este laboratorio son:

- Matlab: El lenguaje de programación en el que se realizaron los algoritmos.
- Laptop Asus con 8gb de memoria RAM y procesador Intel Core i5-5200U
- Python: Lenguaje de programación con el cual se realizó la lectura y posterior conversión de los datos a un formato en el que sea entendible para el lenguaje Matlab.

5 Métodos

Para la realización de los algoritmos, se hizo uso de dos métodos. Ambos métodos reciben una matriz rectangular A de tamaños n*deg, siendo n el tamaño del vector t de entrada y deg el grado del polinomio que queremos formar. Además de recibir un vector b, que contiene el conjunto de datos.

Para realizar las pruebas, se dividieron los datasets en dos subconjuntos, uno de entrenamiento y otro de validación. El subconjunto de entrenamiento sirve para ver gráficamente cómo se comporta el polinomio y qué tan bueno fue el ajuste, mientras que el subconjunto de validación, sirve para validar que gráficamente tiene sentido, porque con estos valores vamos a obtener el error y la desviación estándar del error.

5.1 Ecuaciones normales

El método de ecuaciones normales consiste en obtener una matriz cuadrada a partir de la matriz rectangular A, la manera de conseguir esta matriz es multiplicar $A^T * A$ (para que haya balance en la ecuación, también vamos a multiplicar el vector b por A^T), luego se procede a conseguir una matriz triangular. La matriz triangular se consigue realizando una descomposición de Cholesky de la matriz cuadrada que obtuvimos. Con la descomposición de Cholesky, obtenemos una matriz triangular superior L^T, a la cual al transponerla obtenemos la matriz L. Teniendo estas dos matrices, podemos realizar sustitución sucesiva para obtener los valores de los vectores y y x, a partir de las ecuaciones: $L * y = A^T * b$ y $L^T * x = y$

5.2 Transformaciones de Householder

El método por transformaciones de Householder consiste en obtener una matriz H, la cual se consigue con la siguiente fórmula: $I - 2(v * v^T / v^T * v)$, siendo I la matriz identidad, y v, siendo un vector que se consigue a partir de las columnas de la matriz A, de la siguiente manera: $A[:, i] - norm(A[:, i]) * e$ para i=1,2,3...,deg y siendo e una matriz identidad la cual tiene un 1 en la posición $(i, 1)$. La matriz H obtenida se multiplica por A y por b, hasta llegar haber recorrido todas las columnas. Finalmente, se obtiene una matriz triangular superior, y se puede realizar sustitución sucesiva hacia adelante con las matrices A y b ya transformadas $Ax = b$.

6 Pruebas

Las pruebas de los algoritmos se realizaron teniendo en cuenta dos datasets. El primer dataset, son datos históricos del petróleo Brent, desde el 6 de julio hasta el 6 de octubre, del presente año; el segundo dataset es la esperanza de vida en Colombia desde el año 1964 hasta el año 2018.

6.1 Datos Históricos de Petróleo Brent [1]

En este caso, los datos que representan el eje y, serán el precio en dólares del petróleo Brent, y el eje x representa el tiempo en días, desde el 6 de julio hasta el 6 de octubre, hay algunos días en los que no se tienen datos registrados, por lo tanto, en total se tendrán 67 datos.

Pruebas con transformaciones de Householder:

- **Prueba 1:**

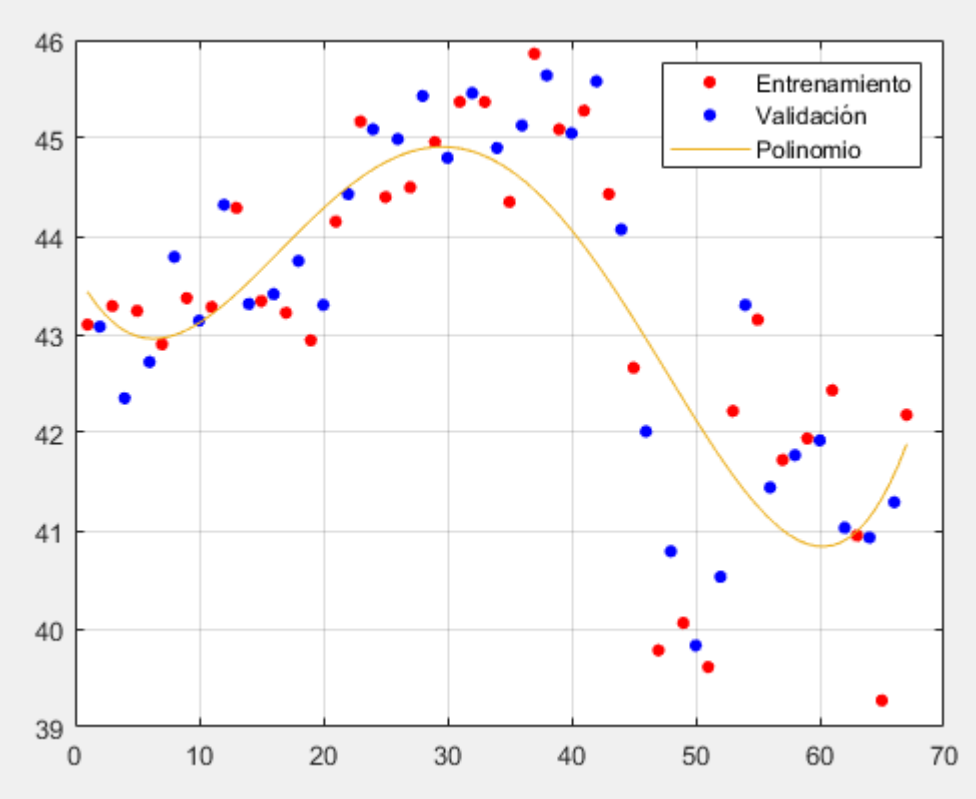

Fig 1. Prueba 1

n=5
Error medio=0.7045
Desviación Estándar del error=0.5958
Tiempo de ejecución: 0.000251 segundos.

- **Prueba 2:**

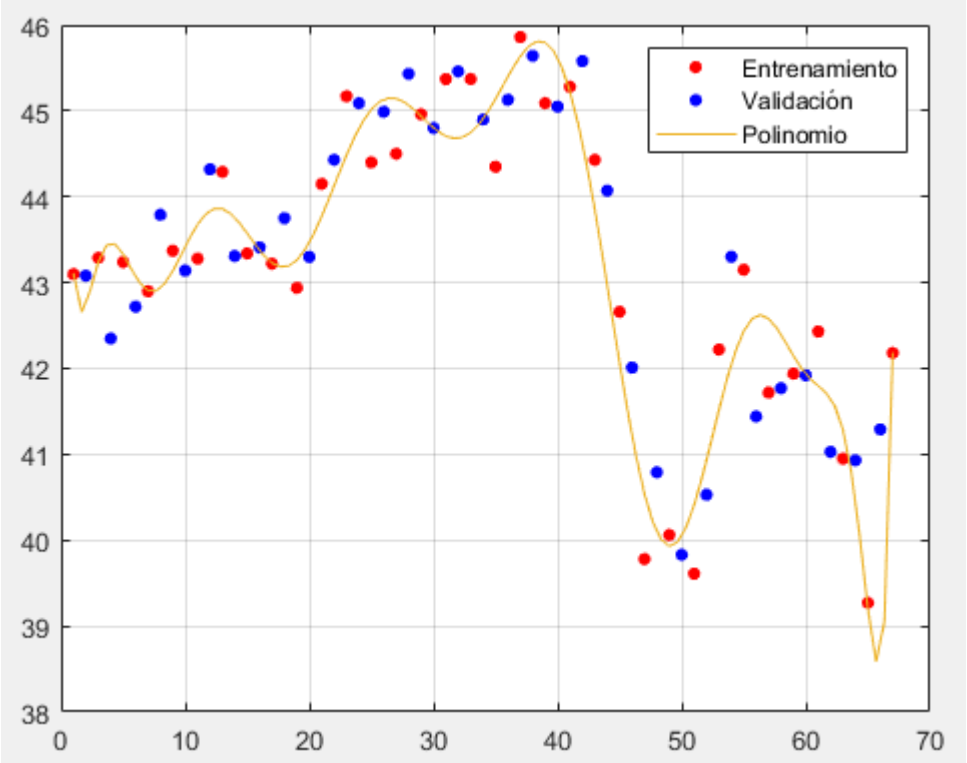

Fig 2. Prueba 2

n=15
Error medio=0.5611
Desviación Estándar del error=0.5189
Tiempo de ejecución: 0.000505 segundos.

Pruebas con ecuaciones normales:

- **Prueba 3:**

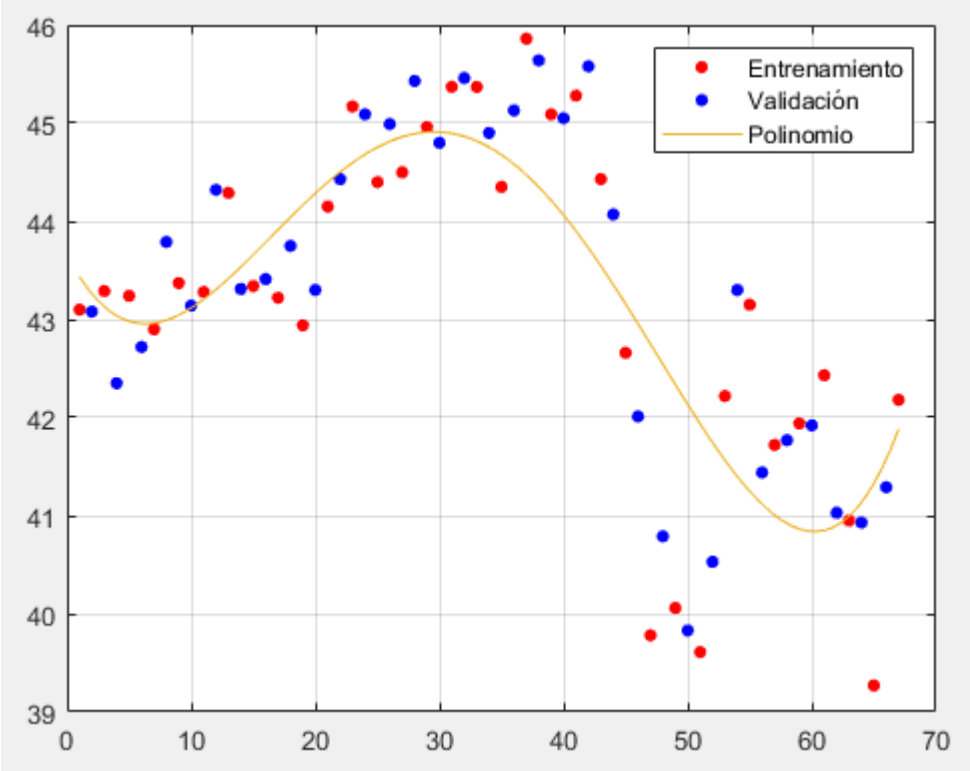

Fig 3. Prueba 3

n=5
Error medio=0.7045
Desviación Estándar del error=0.5958
Tiempo de ejecución: 0.000405 segundos.

- **Prueba 4:**

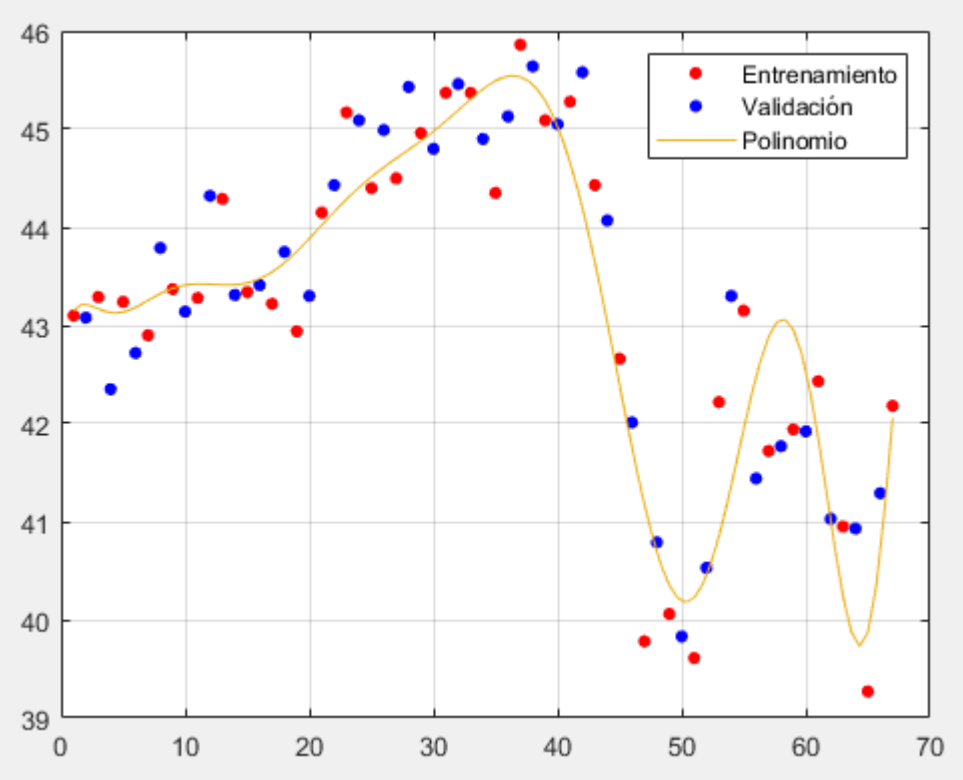

Fig 4. Prueba 4

n=15
Error medio=0.5239
Desviación Estándar del error=0.4565
Tiempo de ejecución: 0.000257 segundos.

6.2 Esperanza de vida en Colombia [2]

En este caso, tendremos en el eje y las edades en años de la esperanza de vida, y en el eje x tendremos los años. Los años están comprendidos desde 1964 hasta el 2018.

Pruebas con transformaciones de Householder:

- **Prueba 5:**

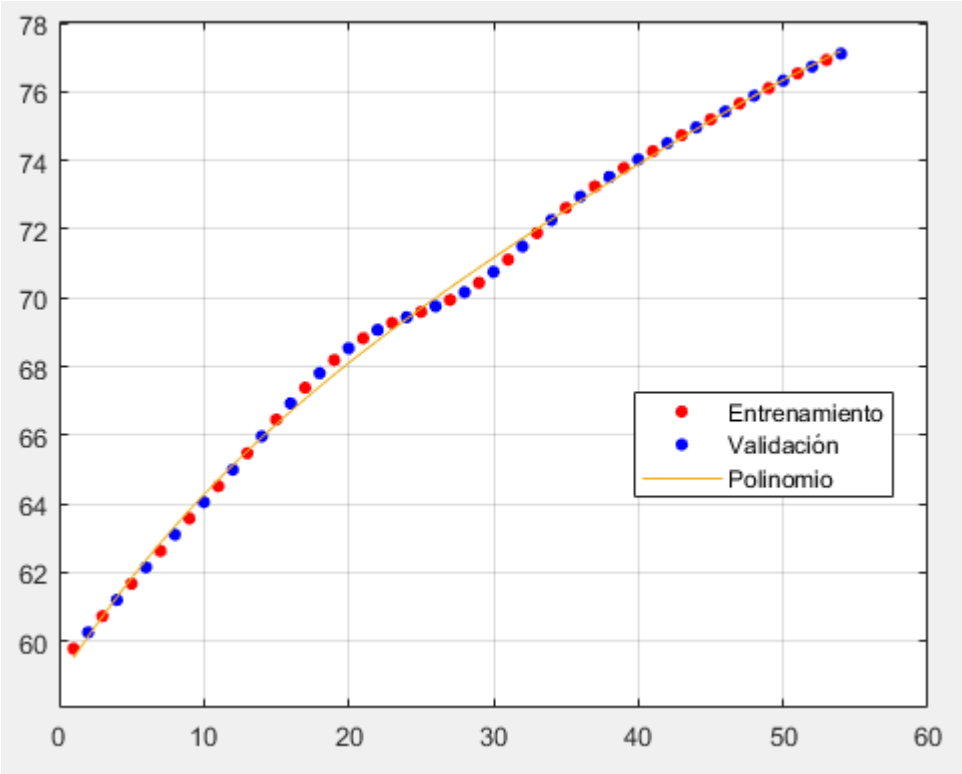

Fig 5. Prueba 5

n=5
Error medio=0.1665
Desviación Estándar del error=0.1346
Tiempo de ejecución: 0.004683 segundos.

- **Prueba 6:**

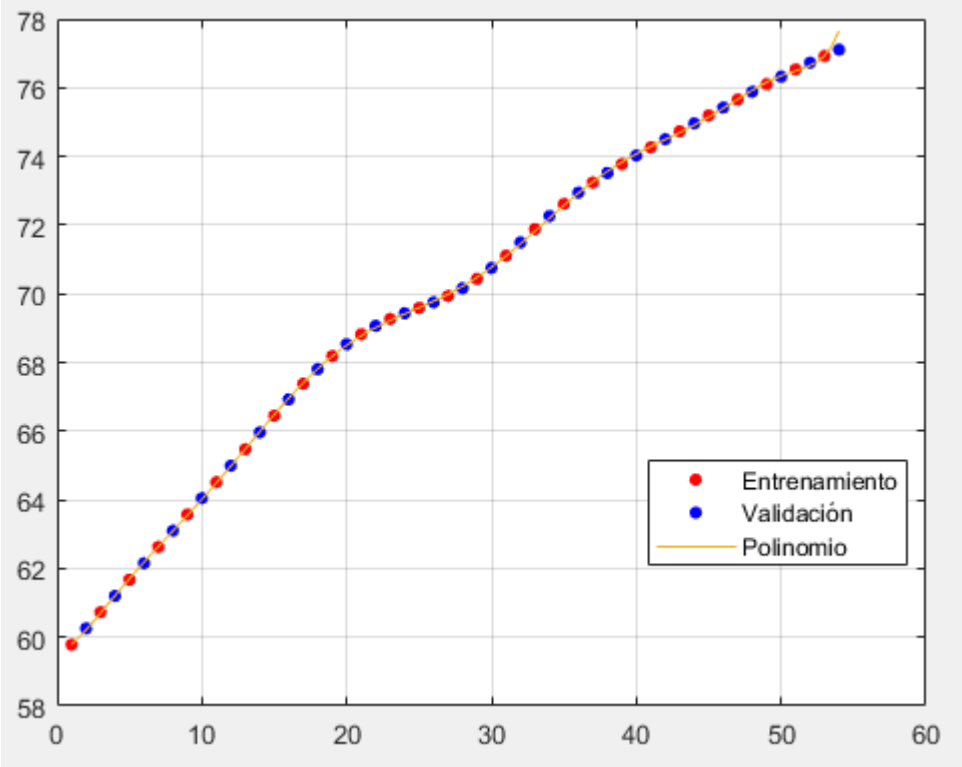

Fig 6. Prueba 6

n=12
Error medio=0.0521
Desviación Estándar del error=0.0990
Tiempo de ejecución: 0.001342 segundos.

Pruebas con ecuaciones normales:

- **Prueba 7:**

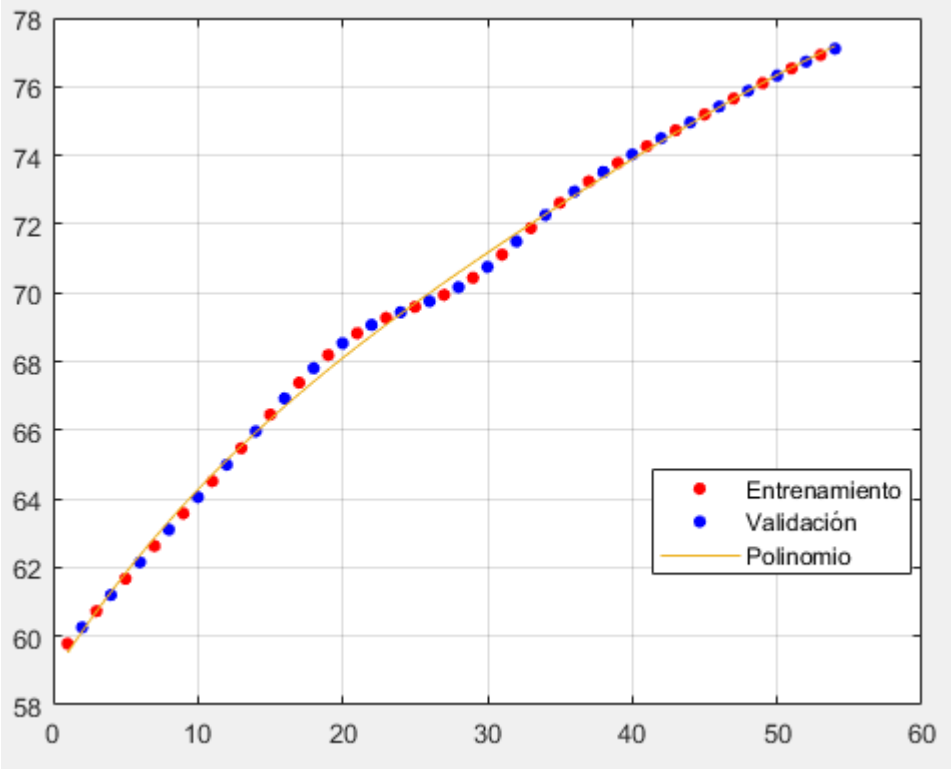

Fig 7. Prueba 7

n=5
Error medio=0.1665
Desviación Estándar del error=0.1346
Tiempo de ejecución: 0.003834 segundos.

- **Prueba 8:**

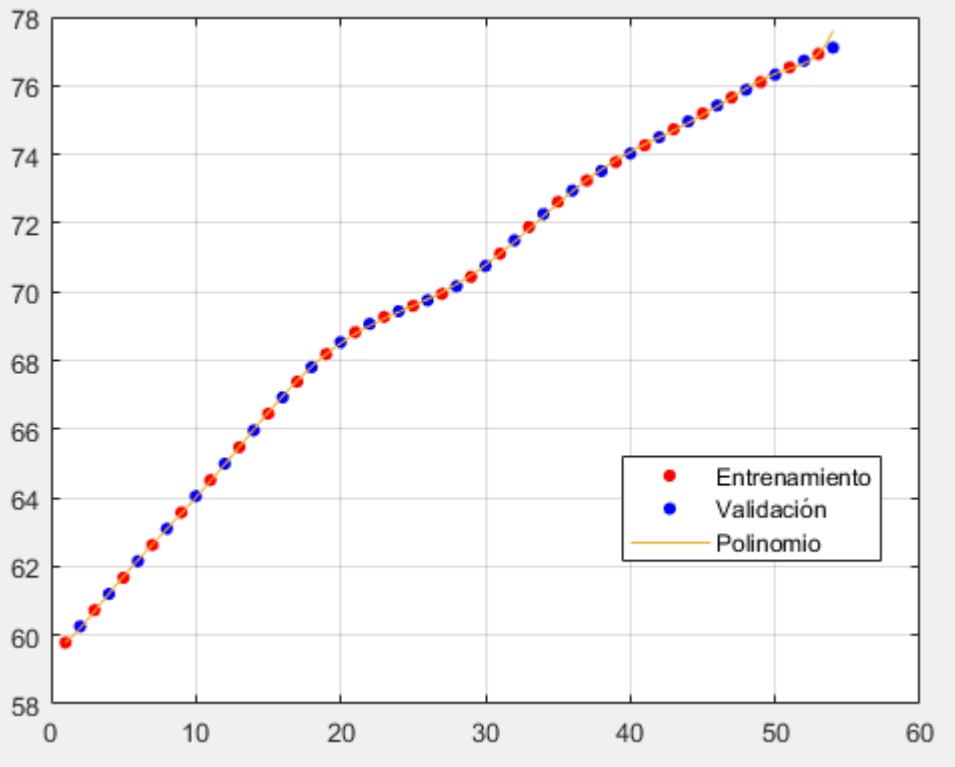

Fig 8. Prueba 8

n=12
Error medio=0.0508
Desviación Estándar del error=0.0890
Tiempo de ejecución: 0.004505 segundos.

7 Análisis de resultados

7.1 Análisis para el dataset "Datos Históricos de Petróleo Brent"

Podemos ver que para este dataset, la variación de los errores no es muy grande cuando se tiene el mismo n. Sin embargo, podemos ver que para un n=15, la gráfica de ambos métodos dan diferentes, sin embargo, esto no se ve muy reflejado en el error ni en la desviación estándar del error. Por lo cual asumimos, por el momento, que ambos métodos a pesar de arrojar diferentes resultados para x, realizan un buen ajuste de datos. La desviación estándar del error para estas dos pruebas, dan similares, pero no iguales, y a partir de ellas podemos interpretar que los errores no están muy dispersos, y por lo tanto, que el polinomio pasa muy cerca por todos los puntos, y finalmente, se concluye que existe buen ajuste de datos en ambos métodos.

Podemos ver también que el tiempo de ejecución de la prueba 4, es menor que el de la prueba 2, pero algunas decimas de segundo.

En cuanto a la prueba 1 y 3, las cuales fueron realizadas con el mismo $n = 5$ pero con diferentes métodos, no existe ninguna diferencia, de hecho, el error da igual, así como la desviación estándar de ese error. El tiempo de ejecución para la prueba 1 con el método de Householder fue menor que el de la prueba 3 con ecuaciones normales, sin embargo, este tiempo no es muy significativo y se puede decir que ambos métodos resuelven el problema de manera eficiente.

7.2 Análisis para el dataset "Esperanza de vida en Colombia"

Para este dataset, se hizo pruebas con un n=5 y n=12, el segundo debido a que la función de cholesky tiene ciertas restricciones en cuanto al tamaño de la matriz la que se le va a realizar la descomposición.

Podemos ver que para las pruebas realizadas con un n=5, ambos algoritmos arrojan el mismo error y, por lo tanto, la misma desviación estándar de ese error. A partir de lo arrojado por estos algoritmos, podemos concluir que existe un buen ajuste de datos, debido a que la desviación estándar del error, nos indica que los datos no están muy alejados de la recta generada por el polinomio. También podemos concluir que ambos algoritmos resolvieron el problema de manera eficiente, ya que el tiempo de ejecución en ambos casos es muy bajo.

Para un n=12, podemos ver que los errores dieron diferentes, pero muy cercanos, esto quiere decir que el polinomio quedó practicamente igual y no se pueden percibir diferencias a simple vista. Vemos que en ambos casos no existe mucha dispersión en cuanto al error, esto quiere decir que los puntos

estaban muy cercanos a la recta generada por el polinomio. Podemos ver que la mayor diferencia que existe, es en el tiempo de ejecución, pues el algoritmo de transformacion de Householder se demoró mucho menos que el de ecuaciones normales. Sin embargo, ambos se realizaron de manera eficiente.

8 Conclusiones

Podemos concluir que va a haber un mejor ajuste de datos mientras más aumentamos el grado del polinomio, sin embargo, puede haber un punto en el que ese ajuste de datos se vea afectado por un grado demasiado alto (30 en adelante aproximadamente). Ambos algoritmos se realizan de manera eficiente pero esto puede variar dependiendo del tamaño del dataset. Para el segundo dataset escogido en este laboratorio, podemos ver que existe una tendencia marcada, por lo tanto, se puede decir que se va a realizar un buen ajuste de datos con ambos métodos. Con el método de Householder, podemos calcular polinomios de mucho mayor grado que con ecuaciones normales, debido a que la descomposición de Cholesky, necesaria para el algoritmo de ecuaciones normales, tiene un límite de grado(aproximadamente grado 15).

9 Referencias bibliográficas

- [1] https://es.investing.com/commodities/brent-oil-historical-data
- [2] https://datos.bancomundial.org/indicator/SP.DYN.LE00.IN