Universitat de les Illes Balears

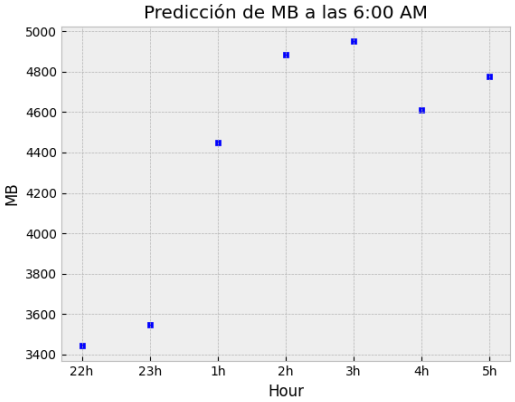
Escola Politécnica Superior

ACSI Cuaderno de Prácticas

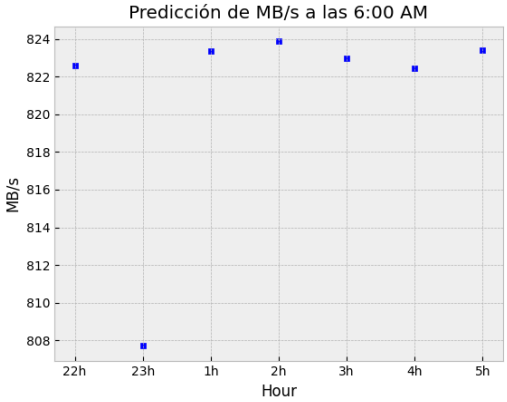
Pablo Núñez Pérez - 43199702K

Práctica 7

1. ¿Qué patrón siguen los datos monitorizados? Proporciona una representación gráfica.



(a) Gráfica tamaño de datos



(b) Gráfica ancho de banda

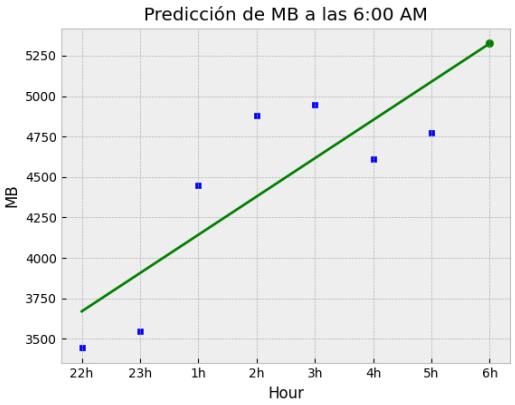
Antes de analizar los gráficos, hay que aclarar un detalle. Como podemos ver a simple vista, en los gráficos no encontramos cientos de miles de puntos. Esto es así porque de cada hora se han hecho las medias respectivas de sus datos (**aritmética** - tamaño, **armónica** - ancho de banda), para cumplir la regla básica de una función: cada elemento de un primer conjunto se le asigna un único elemento del otro conjunto. En estos casos el primer conjunto son las horas (variable independiente) y el segundo el tamaño o el ancho de banda.

En la gráfica del **tamaño de datos**, podemos ver **dos grupos** bien diferenciados: un grupo de ficheros de datos de tamaño pequeño, y otro de tamaño grande. El de tamaño pequeño se comprende entre las 22h y las 23h, y los de mayor tamaño entre las 1AM y 5AM. Esto puede que nos esté indicando que el sistema monitorizado es un servidor, ya que estos suelen llevar a cabo trabajos de mantenimiento, que conllevan archivos más pesados durante las horas nocturnas, cuando no hay muchos usuarios conectados. En cuanto a las tendencia que tiene los datos, podemos decir que ha sido mayoritariamente alcista (entre las 22h y las 3h), con una ruptura de esta tendencia a las 4h, indicando así que posiblemente, el trabajo más pesado ya ha sido realizado.

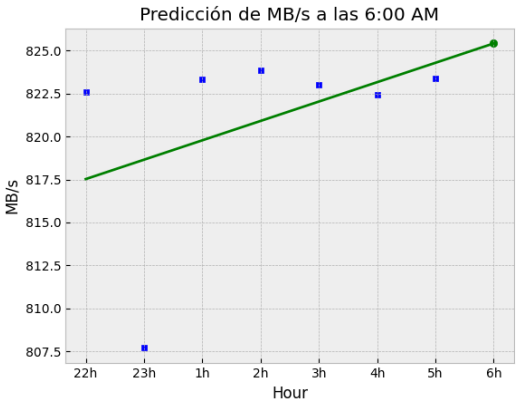
Por otro lado, en la gráfica del **ancho de banda**, obviando las 23h, la distribución ha sido estacionaria. Durante prácticamente todo el tiempo los archivos se han transmitido lo más rápido posible, con muy poca variación de velocidad entre las horas (obviando las 23h). La caída tan brusca en el ancho de banda puede que se deba a un mantenimiento de la empresa proveedora de internet, o bien algún fallo propio del sistema de monitorizado.

2. Calcula los valores solicitados para las 6 a.m. haciendo uso de la regresión lineal, medias móviles (usar los 4 últimos valores) y suavizado exponencial (peso fijo del 60 %).

Regresión lineal



(a) Gráfica tamaño de datos



(b) Gráfica ancho de banda

Para construir las recta y predecir con la función de regresión lineal hemos llevado a cabo estos cálculos:

$$y = a + bx$$

$$b = \frac{\sum_{i=1}^{n} x_i \times y_i - n \times \bar{x} \times \bar{y}}{\sum_{i=1}^{n} x_i^2 - n \times \bar{x}^2}$$

$$a = \bar{y} - b \times \bar{x}$$

Donde x es el parámetro que le pasamos a la función, en este caso la hora a predecir. Las seis en este caso se corresponden con la octava hora a predecir de la lista de horas que tenemos presentes.

- Predicción de tamaño

$$b = 236,69$$

$$a = 3432,55$$

Y la predicción de las 6AM sería por ende:

$$y = 3432,55 + 236,69 * 8$$

$$y = \mathbf{5326{,}07\ MB}$$

- Predicción de ancho de banda

$$b = 1,1264$$

$$a = 816,39$$

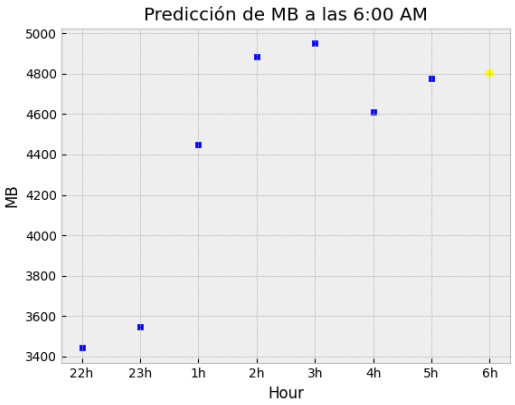
Y la predicción de las 6AM sería por ende:

$$y = 816{,}39 + 1,1535 * 8$$

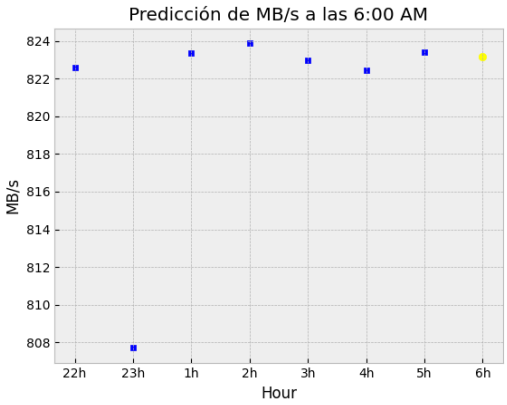
$$y = \mathbf{825{,}40\ MB/s}$$

Como ya podemos ver en las gráficas, la regresión lineal no se ajusta mucho a la distribución de los datos.

Medias móviles



(a) Gráfica tamaño de datos



(b) Gráfica ancho de banda

La función de medias móviles viene dada por la siguiente expresión:

$$f_{t+1} = \frac{y_t + y_{t-1} + ... + y_{t-n+1}}{n}$$

En esta expresión, f_{t+1} indica la predicción, el numerador está compuesto por los valores del eje y que queremos usar para predecir y n indica el número de valores del eje y usados. Como recordatorio, las seis de la mañana corresponden a la octava hora.

- Predicción de tamaño

$$y_t = 4775{,}2293MB$$
$$y_{t-1} = 4611{,}401MB$$
$$y_{t-2} = 4949{,}224MB$$
$$y_{t-3} = 4881{,}890MB$$

Y la predicción de las 6AM sería por ende:

$$f_8 = \frac{4775{,}2293 + 4611{,}401 + 4949{,}224 + 4881{,}89}{4}$$
$$f_8 = \mathbf{4804.43MB}$$

- Predicción de ancho de banda

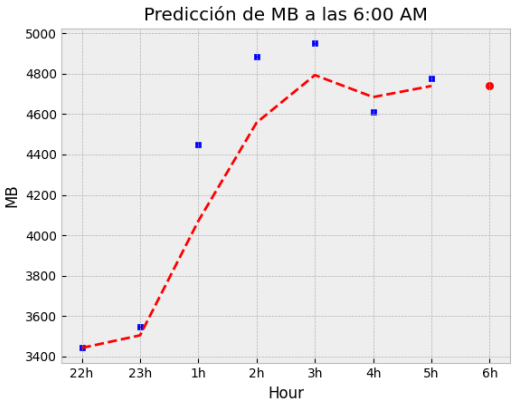
$$y_t = 823{,}3838MB/s$$
$$y_{t-1} = 822{,}451MB/s$$
$$y_{t-2} = 822{,}992MB/s$$
$$y_{t-3} = 823{,}86MB/s$$

Y la predicción de las 6AM sería por ende:

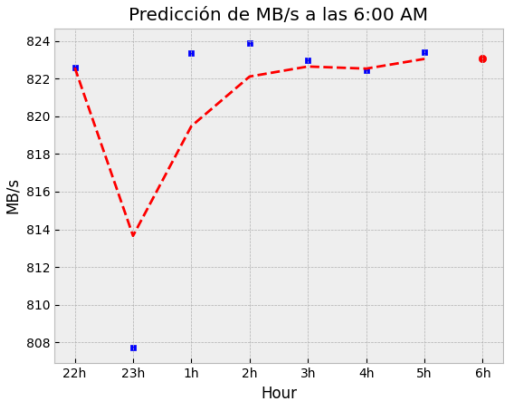
$$f_8 = \frac{823{,}38 + 822{,}45 + 822{,}92 + 823{,}86}{4}$$
$$f_8 = \mathbf{823.15\ MB/s}$$

Las medias móviles funcionan muy bien en ambos casos de predicción. Nos ofrecen unos valores que no distan mucho de los anteriores.

Suavizado exponencial



(a) Gráfica tamaño de datos



(b) Gráfica ancho de banda

La fórmula del suavizado exponencial usada es la siguiente:

$$f_{t+1} = (1 - \alpha)f_t + \alpha(y_{t+1})$$

Los resultados del algoritmo de suavizado exponencial son lo siguientes:

- Predicciones para tamaño de datos

Hora	Media de tamaño MB	Predicción
1	3442.893	3442.893
2	3546.785	3505.22
3	4447.97	4070.875
4	4881.89	4557.48
5	4949.22	4792.52
6	4611.40	4683.852
7	4775.22	4738.67

Por tanto la predicción del tamaño de las 6AM es de **4738,67MB**

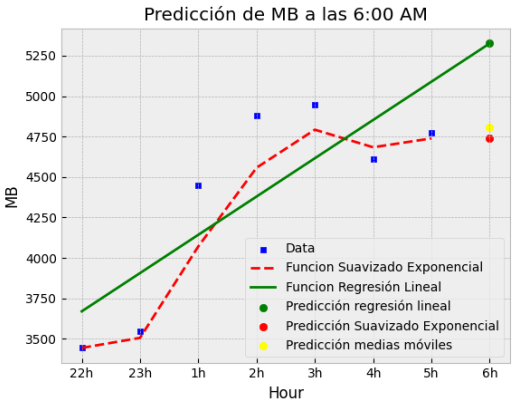
- Predicciones para ancho de banda

Hora	Media de ancho de banda	Predicción
1	822.57	822.57
2	807.70	813.65
3	823.34	819.47
4	823.86	822.106
5	822.99	822.638
6	822.45	822.52
7	823.38	823.040

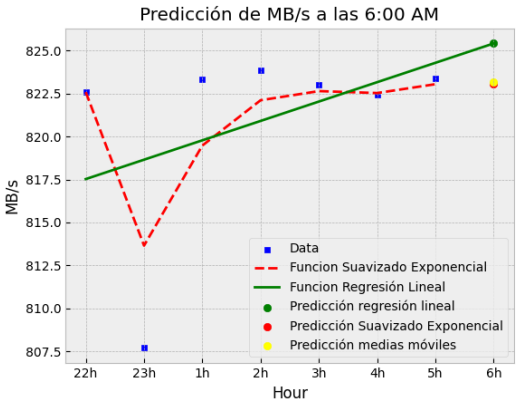
Por tanto la predicción del ancho de banda de las 6AM es de **823,04 MB/s**

3. ¿Qué técnica de predicción funciona mejor? ¿Por qué? ¿Cuál es la más adecuada para los datos con los que contamos?

Resultados finales



(a) Gráfica tamaño de datos



(b) Gráfica ancho de banda

Vamos a analizar los resultados de las funciones de predicción una por una.

En primera instancia, vamos a analizar la **regresión lineal**. Tanto en la gráfica de tamaño, como en la de ancho de banda podemos ver que esta función no se ajusta a la distribución de los datos y que el valor predicho para las 6 AM se aleja demasiado de los datos históricos. El pobre rendimiento de esta función predictiva tiene una razón simple: los datos no siguen una evolución **lineal**, y no tienen una tendencia continua durante todo el intervalo.

En cuanto a las **medias móviles** podemos ver que funcionan realmente bien. Esto es así ya que el intervalo (de los últimos 4 datos históricos) que usamos para calcular el valor de las 6AM es bastante estacionario. Podemos ver de hecho, que cuanto menor es la varianza de los datos usados, mejor es la predicción. Este hecho se puede ver en la gráfica del ancho de banda.

Por último tenemos el **suavizado exponencial**. Como funciona de manera muy similar a las medias móviles, tenemos obviamente valores muy parecidos. Donde encontramos más diferencia es en la predicción del tamaño del fichero a las 6 horas. Esto se debe a la naturaleza del suavizado exponencial, que da más peso a las observaciones más recientes. Teniendo en cuenta que las medias móviles tratan a los valores seleccionados con la misma importancia, y que en ese intervalo (2AM- 5AM) la varianza es apreciable, el resultado es menos preciso y por ende en este caso el suavizado exponencial, podría ofrecer una mejor predicción.

En conclusión, si nos ceñimos simplemente a la teoría, nos decantaríamos por las medias móviles ya que los requisitos idóneos para su uso se dan en ambos casos. Tenemos datos bastante **estacionarios** (más en la gráfica de ancho de banda), y la predicción es a **corto plazo** (1 hora de distancia). Pero, dado lo explicado en el párrafo anterior, la función ideal para la primera gráfica podría ser la de suavizado exponencial. En cuanto a la gráfica del ancho de banda, son casi indistinguibles así que usando la teoría como desempate, usaríamos las medias móviles.

Script

```
import matplotlib.pyplot as plt
import numpy as np
#Colores
color_data = "blue"
color_suaexp  = "red"
color_linealreg  = "green"
color_medias_moviles  = "yellow"
#Datos del eje de abcisas
horas_etiquetas = ["22h", "23h", "1h", "2h", "3h", "4h", "5h", "6h"]

def main():
    #Leemos fichero de datos
    with open("dataProcessed.csv") as file:
        data = file.read().splitlines()
    #Quitamos header
    data.pop(0)

    #Listas de datos
    sizes = np.array([])
    mbss = np.array([])

    data_por_hora = slice_data_time(data)
    #Hacemos media de tamaño y ancho de banda de cada hora
    for data in data_por_hora:
        z_dat = list(zip(*data))
        sizes = np.append(sizes, [arithmetic_mean(np.array(list(z_dat[0])))])
        mbss = np.append(mbss, [harmonic_mean(np.array(list(z_dat[1])))])
    sizes = np.roll(sizes, 2)
    mbss = np.roll(mbss, 2)
    #Dibujamos las listas de datos
    ploting(sizes, "MB")
    ploting(mbss,  "MB/s")

def ploting(data, data_name,):
    x = np.arange(1, len(horas_etiquetas)+1)

    lrf, a, b = regresion_lineal(x[:-1],data)
    print(f"Valor a de {data_name}:{a}, Valor b de {data_name}: {b}")
    plt.figure()
    plt.style.use("bmh")
    plt.xticks(x, horas_etiquetas)
    plt.xlabel("Hour")
    plt.ylabel(data_name)
    plt.title(f"Predicción de {data_name} a las 6:00 AM")
    #Datos
    plt.scatter(x[:-1], data, s=20, marker='s', color=color_data, label="Data")
    #Medias móviles
    plt.scatter(x[-1:], medias_moviles(4, data), marker='o', color=color_medias_moviles, label="Predicc
    #Regresión lineal
    plt.plot([x[0], x[-1]],[lrf(x[0]), lrf(x[-1])] , linestyle='solid', color=color_linealreg, label="F
    plt.scatter(x[-1],lrf(x[-1]) , color=color_linealreg, marker='o', label="Predicción regresión linea
    #Suavizado exponencial
    plt.plot(x[:-1], suavizado_exponencial(0.6, data), linestyle='--', color=color_suaexp, label="Funci
    plt.scatter(x[-1:], suavizado_exponencial(0.6, data)[-1:], marker='o', color=color_suaexp, label="P


    plt.legend(loc="lower right")
    plt.show()
#Troceador de datos
```

```
def slice_data_time(data):
    sliced_data = [[] for _ in range(24)]
    for line in data:
        temp = line.split(",")
        #En la posicion indicada por la hora correspondiente a la fila leída,
        #se añaden los valores de su tamaño y ancho de banda
        sliced_data[int(temp[1])].append([int(temp[0]), float(temp[2])])

    return list(filter(None, sliced_data))

# Medias moviles
def medias_moviles(num_values, data):
    print(f"{data}")
    print(f"MEDIAS MOVILES: {data[-num_values:]}") # De la posicion -4 del Slice hasta el final de la l
    return np.sum(data[-num_values:])/num_values

# Suavizado exponencial
def suavizado_exponencial(alpha, data):
    predicted_data = [data[0]]
    print(f"SUAVIZADO EXPONENCIAL: MEDIA: {data[0]} | PREDICCIÓN {predicted_data}\n")
    for i in range(1, len(data)):
            #ft+1 = (1-alfa)*ft + alfa(data actual)
        predicted_data.append((1-alpha) * predicted_data[i-1] + alpha * data[i])
        print(f"SUAVIZADO EXPONENCIAL: MEDIA {data[i]} | PREDICCIÓN {predicted_data}")

    return predicted_data

# Regresion lineal
def regresion_lineal(x, y, ratio=False):
    # Número de muestras
    n = np.size(x)

    # Medias
    x_hat = np.mean(x)
    y_hat = harmonic_mean(y) if ratio else arithmetic_mean(y)

    b = (np.sum(y*x) - n*x_hat*y_hat)/(np.sum(x*x) - n*x_hat**2)
    a = y_hat - b * x_hat
    return lambda x: a +b*x, a ,b

# Media aritmetica
def arithmetic_mean(data):
    return np.sum(data)/np.size(data)

# Media harmonica
def harmonic_mean(data):
    return np.size(data)/np.sum(1/data)

if __name__ == "__main__":
    main()
```