

# СЕНЗОРСКИ СИСТЕМИ

Лабораториска вежба бр. 2



20 Мај, 2021

Скопје

Марија Грнчаровска

161273

## СОДРЖИНА

<b>1. Имплементација на програмата .....</b>	<b>3</b>
<b>1.1. Користени податоци .....</b>	<b>3</b>
<b>1.2. Дефинирање на функции .....</b>	<b>4</b>
<b>2. Анализа на добиените податоци .....</b>	<b>5</b>
<b>2.1. Графички приказ на <math>MA(1)</math>, <math>MA(2)</math> и <math>MA(3)</math> одделно .....</b>	<b>5</b>
<b>2.2. Анализа на процентот на реализирани трансмисии .....</b>	<b>5</b>
<b>2.3. Анализа на <i>Mean Square Error</i> .....</b>	<b>6</b>

## 1. Имплементација на алгоритмите

### 1.1 Користени податоци

Избраниот dataset се состои од приближно 56 570 сензорски мерења. Сензорите се поставени на 8 локации и вршат мерења на температурата, светлината и влажноста на секои 5 минути.

Преземени се од <https://data.melbourne.vic.gov.au/Environment/Sensor-readings-with-temperature-light-humidity-ev/ez6b-syvw>.

Програмата ја изработив во Jupyter Notebook користејќи различни библиотеки за читање и пресметка на предикциите како numpy, pandas, statsmodels и matplotlib модулот за графичка претстава на резултатите.

На почеток го наоѓаме сензорот со најмногу мерења и вршиме проверка дали постои податок од ненумерички тип.

```
In [23]: # Finding the sensor with most measurements

column_mac = df['mac']

dictionary = dict()
for data in column_mac:
    dictionary.setdefault(data, 0)
    dictionary[data] += 1
dictionary

Out[23]: {'0013a20040b4b755': 2728,
          '0013a20040b516ed': 2915,
          '0013a20040b516f6': 2918,
          '0013a20040b5b318': 2725,
          '0013a20040b315ec': 2903,
          '0013a20040b3155c': 6626,
          '0013a20040b5b337': 4598,
          '0013a20040b31571': 12038,
          '0013a20040b31583': 19119}

In [24]: data_needed = df.loc[df['mac'] == '0013a20040b31583']
data_needed

# 19119 rows x 20 columns
```

Резултатите од тоа се:

Data: 19119  
NaN Data: 0

Threshold-от се движи од 0 – 4, со чекор од 0,1 .

```
# DEFINING THRESHOLD FROM 0 TO 5 WITH STEP 0.1 : 1, 1.1, 1.2, 1.3, 1.4, ..., 4.8, 4.9, 5.0
threshold = [round(x,1) for x in np.arange(0,4.1, 0.1)]
```

## 1.2 Дефинирање на функции

### Алгоритам за МА(1)

```
In [26]: def moving_average_1(serie, threshold):
transmitted = 0;
arr = []
count = 0
mse = 0.0
for data in serie:
    if math.isnan(data):
        continue;
    if count==0:
        arr.append(data)
        transmitted += 1
    else:
        if(abs(data - arr[0]) > threshold):
            arr[0] = data
            transmitted += 1
        else:
            #print(data, arr[0])
            error = round((arr[0] - data)**2,5)
            mse += error
        count +=1
    #print(count, transmitted)
    return round(transmitted*1.0/count*100, 2), mse/count
```

### Дефинирање на генеричка функција за МА

```
In [27]: # DEFINING GENERIC MOVING AVERAGE FUNCTION

def moving_average(n, serie, threshold):
    if n == 1:
        return moving_average_1(serie, threshold) # This generic function is OK even for n = 1 but not enough efficient
        # That's why in tat case we call a special function
    arr = []
    count = 0;
    transmitted = 0;
    mse = 0.0

    for data in serie:
        if math.isnan(data):
            continue
        if count >= n:
            moving_avg = round(np.mean(arr),5)
            for i in range(n - 1):
                arr[i] = arr[i + 1]
            if(abs(moving_avg - data) > threshold):
                arr[n-1] = data
                transmitted = transmitted + 1
            else:
                arr[n-1] = moving_avg
                error = round((arr[n-1] - data)**2,5)
                mse = mse + error
        else:
            arr.append(data)
            transmitted = transmitted + 1
            count = count + 1

    # print(count, transmitted)
    return round(transmitted*1.0/count*100, 2), mse/count
```

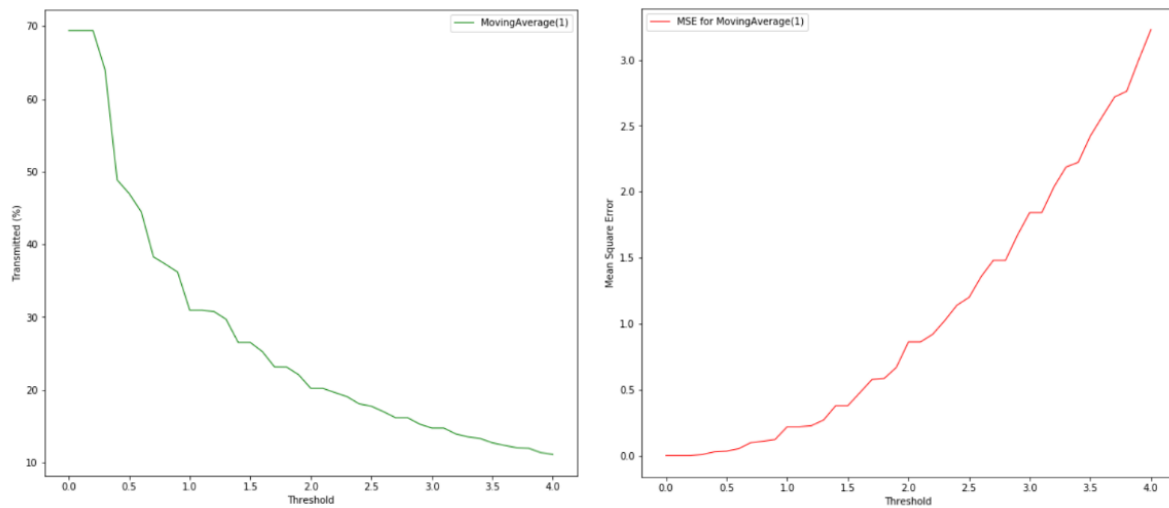
## 2. Анализа на добиените резултати

За графичкиот приказ на податоците користев 2 метрики:

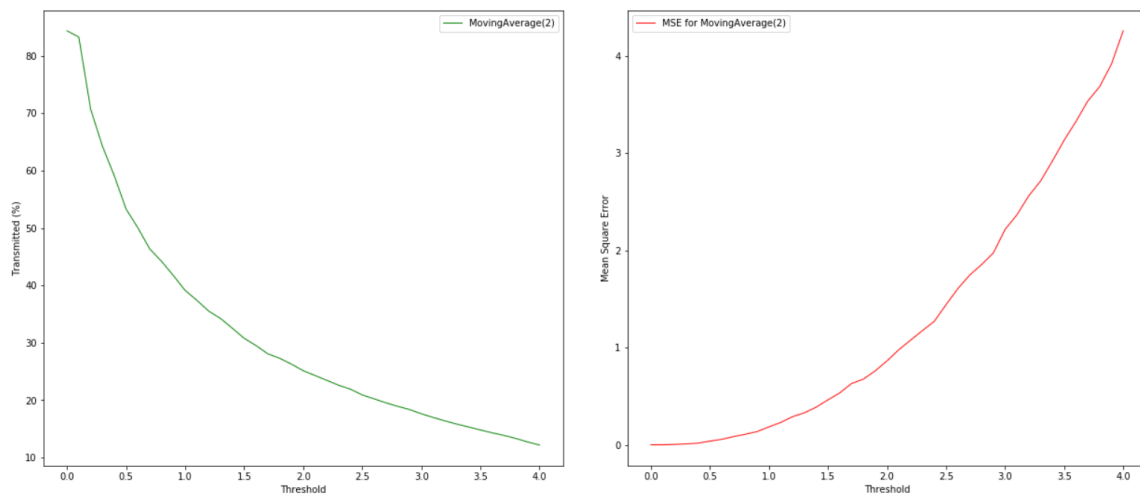
- % на реализирани трансмисии и
- MSE

### 2.1. Графички приказ за MA (1), MA (2) и MA (3)

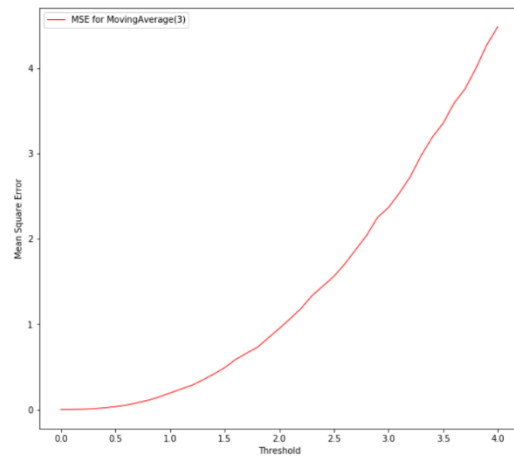
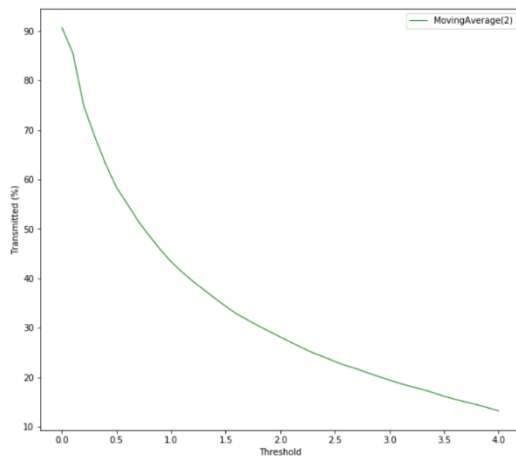
#### ○ Moving Average (1)



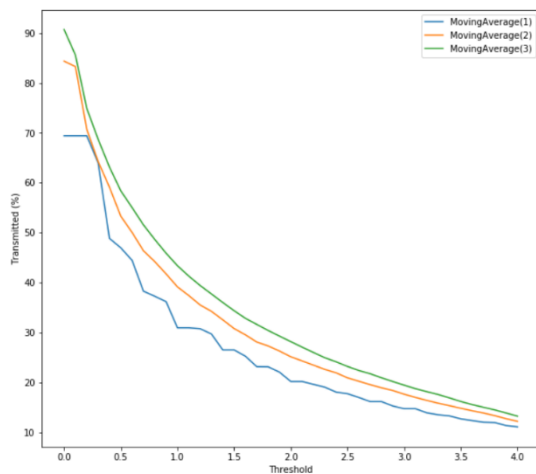
#### ○ Moving Average (2)



### ○ Moving Average (3)



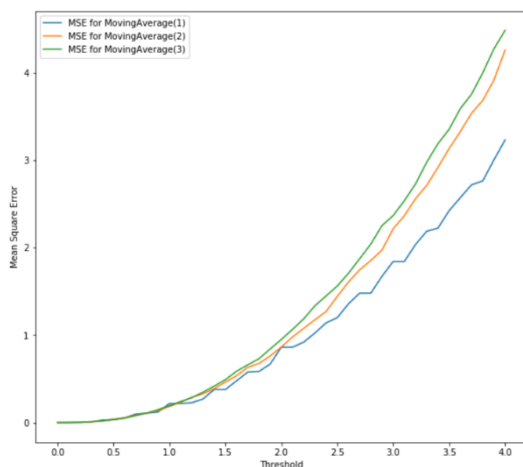
## 2.2. Анализа на процентот на реализирани трансмисии



### TRANSMITTED PERCENTAGE

Од графикот се гледа дека најголем процент на пратени податоци се добиваат со користење на МА(3) алгоритмот, без разлика на вредноста на threshold-от.

## 2.3. Анализа на MSE



### MSE

Во однос на MSE, при помал threshold грешката што се појавува е приближно иста, а за поголемите вредности на threshold-от најмала грешка се појавува со МА(1).