

Міністерство освіти та науки України Львівський
національний університет імені Івана Франка

Звіт

Про виконання лабораторної роботи No1

“ARIMA”

Виконав: студент групи Фес-21 Сало Остап Перевірів: Сінкевич О.О.

Мета Роботи : Ознайомитися з алгоритмами та методами : AR, MA, ARMA та ARIMA. Та використати алгоритм ARIMA на реальних даних температури.

ХІД РОБОТИ :

AR : Перш за все я реалізував алгоритм AR, ось код де реалізований метод найменших квадратів, який реалізований за допомогою бібліотеки **sympy**: Метод найменших квадратів нам потрібний для знаходження двох регресорів(w_0, w_1).

```

def minMNK(a,b):
    function = 0
    w0, w1 = symbols('x y', real=True)

    for i in range(len(a)):
        function += (w1*a[i] + w0 - b[i]) ** 2

    #print(function)

    function1w0 = diff(function,w0)
    function2w1 = diff(function,w1)

    #print(function1w0)
    # print(function2w1)

    function1w0 = str(function1w0)
    function2w1 = str(function2w1)

    result1 = re.findall(r'\d+',function1w0)
    result2 = re.findall(r'\d+',function2w1)

    for i in range(len(result1)):
        result1[i] = int(result1[i])

    for i in range(len(result2)):
        result2[i] = int(result2[i])

    matrix = [result1[0:2],result2[0:2]]
    matrix_result = [result1[2],result2[2]]

    resultAll = np.linalg.solve(matrix,matrix_result)

    new_w0, new_w1 = resultAll

    return new_w0, new_w1

```

Ось тут реалізована Авторегресія :

Авторегресія - це алгоритм який використовує параметр p , Цей параметр означає скільки разів в циклі ми будемо знаходити нові регресори, та додаючи старий регресор до нового, ми будемо знаходити новий вузол.

```
[ ] # Автокорелюція :
def auto_reg(a,b,p=1):
    w0,w1 = minMNK(a,b)
    function = w0
    #print(w0)
    result_array = []
    for i in range(p):
        len_arr = len(a) - i - 1

        w0,w1 = minMNK(a[0:len_arr],b[0:len_arr])
        #print('liner - ',res_liner_regr, 'b - ',b[len_arr])
        function += b[len_arr] * w1

    result_array.append(function)
    return function
auto_reg(x_data,y,p=5)
```

МА: Для Агіма Нам ще потрібен метод МА - скільське середнє. Це алгоритм має параметр q, на якій кількості вузлів потрібно знаходити середню величину. Ось його реалізація :

```
[ ] # Скільське середнє :
def SMA(b,p=3):
    SMA_res = []
    sum_b = 0
    num = 0
    for i in range(len(b) - p):
        for j in range(p):
            sum_b += b[i+j]
        SMA_res.append(sum_b / p)
        sum_b = 0
    return SMA_res
def MA_foresee(b,p):
    SMA_res = 0

    for i in range(p):
        sum_b = len(b) - i - 1
        SMA_res += b[sum_b]
    SMA_res /= p
    return SMA_res
```

ARIMA : Використовує три параметра : p, q, i . p - від AR q - від MA i - Це скільки разів нам потрібно продиференціювати дані, тому що дані можуть мати Тренд, або бути Сезонними, а для сезонних даних реалізовані інші алгоритми. Також хороша практика з логарифмуванням даних перед диференціюванням - це потрібно щоб стабілізувати дисперсію.


Ось код для ARIMA :

```
[ ] def ARIMA_with_Prediction(step, x,y,p,q,d):
    new_y = [0]
    new_y1 = []

    resut_arr = []
    MA_plust_ar = 0

    if d == 0:
        new_x = []
        new_y = []
        new_x = x
        new_y = y
        my_x_x = 199
        for i in range( ):
            MA_res = MA_foresee(new_y,q)
            AR_res = auto_reg(new_x,new_y,p=p)
            MA_plust_ar = MA_res + AR_res
            resut_arr.append(MA_plust_ar)
            new_y.append(MA_plust_ar)
            my_x_x +=1
            new_x.append(my_x_x)

    return resut_arr
print("Predicted = ",ARIMA_with_Prediction(3,x_data,y,p=1,q=7,d=0))
print("True value = ", test_y[201],",", " ",test_y[202],",", " ",test_y[203])
```

 Predicted = [10.243201426497652, 10.239743137128386, 10.273958496080168]
True value = 10.29 , 10.13 , 10.11

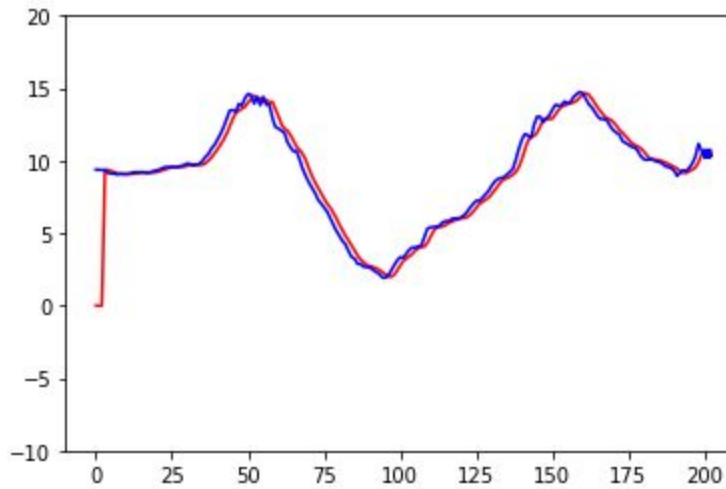
Результати :

```
print("Predicted = ",ARIMA_with_Prediction(3,x_data,y,p=1,q=7,d=0))
print("True value = ", test_y[201],",", " ",test_y[202],",", " ",test_y[203])
```

Predicted = [10.243201426497652, 10.239743137128386, 10.273958496080168]
True value = 10.29 , 10.13 , 10.11



```
result MA = <class 'numpy.float64'>  
result AR = <class 'numpy.float64'>
```



```
predict 10.377690447771062  
True value 10.29
```

Висновок : В цій лабораторній роботі я ознайомився з 4 алгоритмами, які використовуються для роботи з часовими рядами. ARIMA - це один з найпопулярніших алгоритмів, но дуже слабкий, його не можна використати з сезонними даними, та довго працює та невірно працює з великою кількістю даних.