Міністерство освіти та науки України
Львівський національний університет імені Івана Франка

Звіт

Про виконання лабораторної роботи №1

“ARIMA”

Виконав: студент групи ФеС-21 Сало Остап
Перевірив: 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(w0,w1).

```
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
    function1W0 = diff(function,w0)
    function2W1 = diff(function,w1)
    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
minMNK(x,y)
```

Ось тут реалізована Авторегреія :

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.

```
# Автокорегресія :
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])
        function += b[len_arr] * w1
        result_array.append(function)
    return function
auto_reg(x_data,y,p=5)
```

MA:

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

```
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
```

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(x,y,my_x,p,q,d):
    new_y = [0]
    new_y1 = []
    MA_plust_ar = 0
    if d == 0:
        MA_res = MA_foresee(y,q)
        AR_res = auto_reg(x_data,y,p=p)
        print("result MA = ", type(MA_res))
        print("result AR = ", type(AR_res))
        MA_plust_ar = MA_res + AR_res

        arr_MA = [0,0,0]
        temp_MA = MAGrath(y)
        for i in range(len(temp_MA)):
            tr = temp_MA[i]
            arr_MA.append(tr)
        plt.plot(x_data,arr_MA,'r')
        plt.plot(x_data,y,'b')
        plt.axis([-10, 350, -10, 20])
        plt.show()
    elif d == 1:
        for i in range(len(y)):
            num = y[i] - y[i-1]
            new_y.append(int(num))
            num = 0
        MA_res = MA_foresee(new_y,q)
        AR_res = auto_reg(x_data,new_y,p=p)
        print("result MA = ", MA_res)
```

```
        print("result AR = ", AR_res)
        MA_plust_ar = MA_res + AR_res

    elif d == 2:
        for i in range(len(y)):
            num = y[i] - y[i-1]
            new_y.append(int(num))
            num = 0
        for i in range(len(new_y)):
            num = new_y[i] - new_y[i-1]
            new_y1.append(num)
            num = 0
        arr_MA = SMA(new_y1,p)
        res_MA = 0
        for i in range(len(arr_MA)):
            res_MA += arrMA[i]
        arr_AR = auto_reg()

    return "predict " + str(MA_plust_ar)

print(ARIMA(x_data,y,my_x=1,p=1,q=7,d=0))
print("True value",test_y[201])
```

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.