Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №2

з дисципліни «Вступ до Data Science»

СТАТИСТИЧНЕ НАВЧАННЯ З ПОЛІНОМІАЛЬНОЮ РЕГРЕСІЄЮ

Виконав:
Селютін Євген Олександрович
Група ІО-15
Залікова книжка №1519

Перевірив:

Професор кафедри ОТ ФІОТ

Писарчук О. О.

Київ – 2023

Мета: виявити дослідити та узагальнити особливості реалізації процесів статистичного навчання із застосуванням методів обробки Big Data масивів та калмановської рекурентної фільтрації з використанням можливостей мови програмування Python.

Завдання:

1. Отримання вхідних даних із властивостями, заданими в Лр_1;

2. Модель вхідних даних із аномальними вимірами;

3. Очищення вхідних даних від аномальних вимірів. Спосіб виявлення аномалій та очищення обрати самостійно;

4. Визначення показників якості та оптимізація моделі (вибір моделі залежно від значення показника якості). Показник якості та спосіб оптимізації обрати самостійно.

5. Статистичне навчання поліноміальної моделі за методом найменших квадратів (МНК – LSM) – поліноміальна регресія для вхідних даних, отриманих в п.1,2. Спосіб реалізації МНК обрати самостійно;

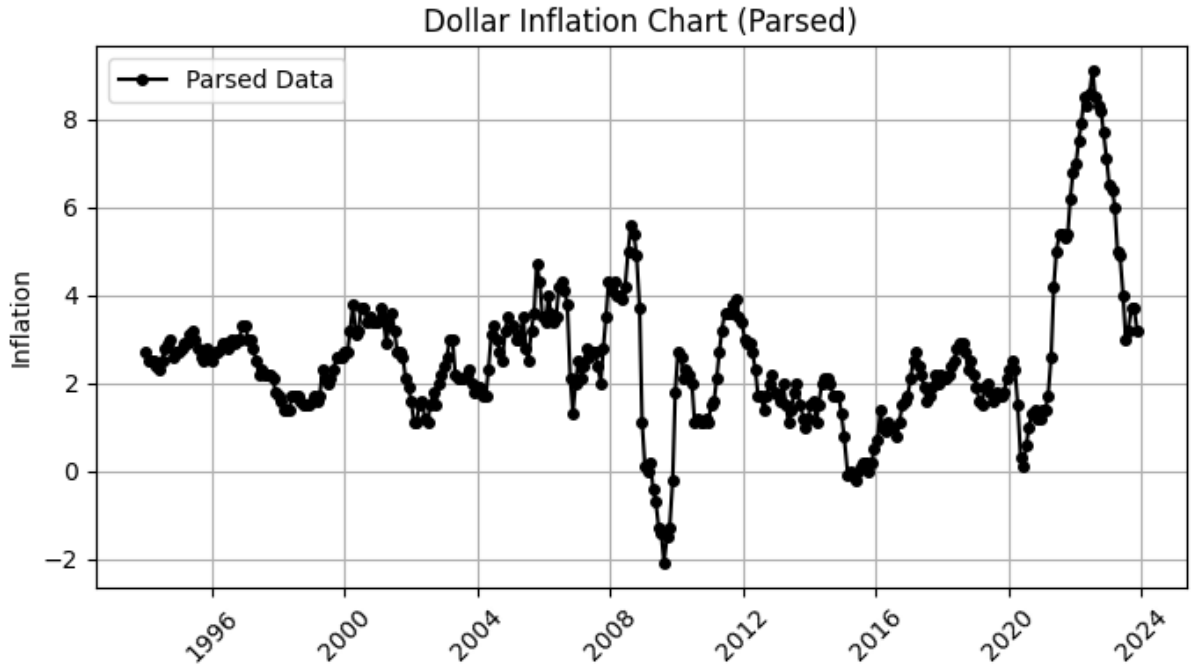
6. Прогнозування (екстраполяцію) параметрів досліджуваного процесу за «навченою» у п.5 моделлю на 0,5 інтервалу спостереження (об’єму вибірки);

7. Провести аналіз отриманих результатів та верифікацію розробленого скрипта.

Виконання:

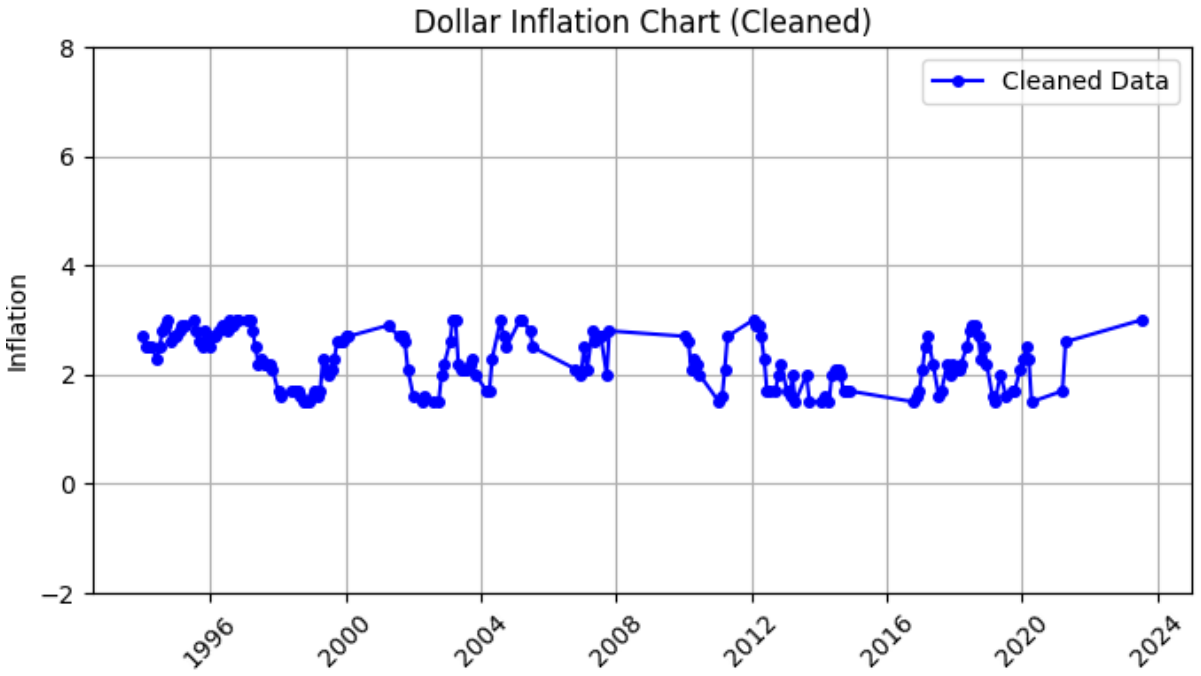
1. Нагадаю, що я працюю з даними про інфляцію долара. Спочатку експортуємо дані з минулої лабі, які ми записали в json файл, розпарсимо у

необхідний формат та побудуємо графік.



2. Як бачимо є явні аномалії, які пов’язані з кризою 2008 року, коронавірусом та війною. Очистимо ці дані. Для цього я буду використовувати бібліотеку sclearn. Також для того щоб зробити очищення коректним я використовую підбір коефіцієнту забруднення. Після цього будуємо графік

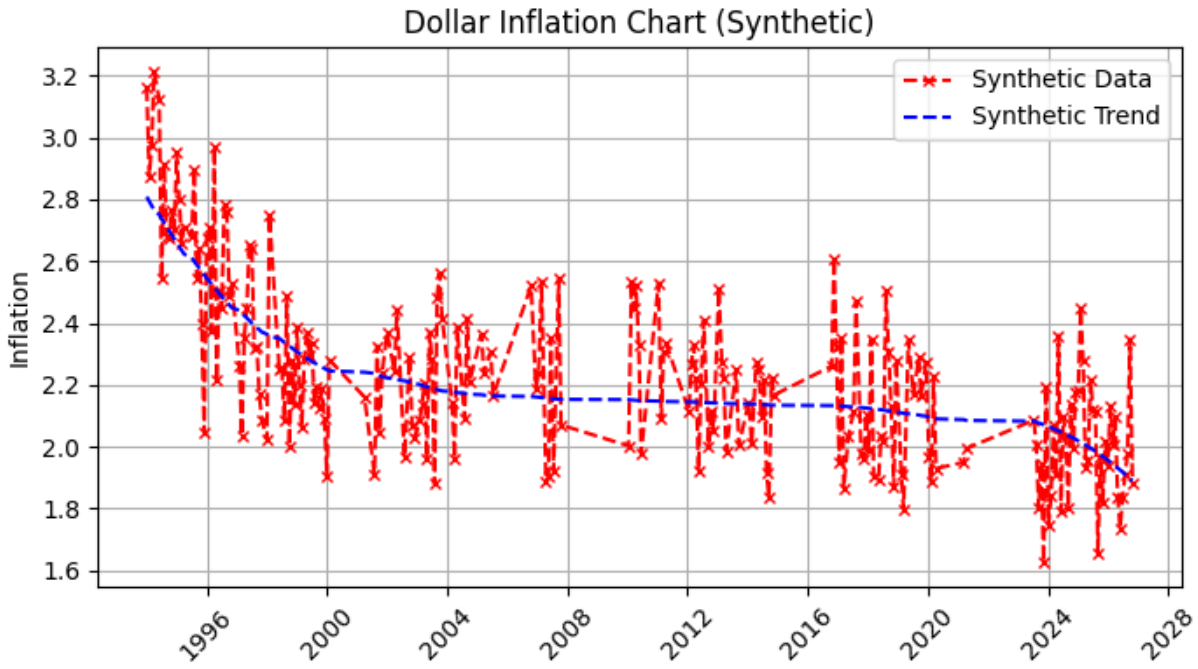
```
53
54      # ________________________________________Cleaned Data_____________________________________
55      X = np.array(parsed_inflation_values).reshape(-1, 1)
56      Y = np.arange(len(parsed_dates))
57
58      # ______________________________Find best values for cleaning_______________________________
59      clf = IsolationForest()
60      contamination_values = np.arange(0.05, 0.3, 0.05)
61      param_grid = {'contamination': contamination_values}
62
63      grid_search = GridSearchCV(clf, param_grid, cv=5, scoring='neg_mean_squared_error')
64
65      grid_search.fit(X, Y)
66
67      best_params = grid_search.best_params_
68      print("Кращі параметри для очистки функції:", best_params)
69
70      best_contamination = best_params['contamination']
71      clf = IsolationForest(contamination=0.5)
72      outliers = clf.fit_predict(X)
73
74      cleaned_dates = [parsed_dates[i] for i in range(len(parsed_dates)) if outliers[i] == 1]
75      cleaned_inflation_values = [parsed_inflation_values[i] for i in range(len(parsed_inflation_values)) if
76                                  outliers[i] == 1]
```



3. Дані очищено, ми отримали графік, який максимально наближено очищено від аномалій (я використовував критерій середнього відхилення). Далі спробуємо побудувати математичну модель та передбачити дані. Для цього я трохи змінив свою функцію з минулої лаби, яка тепер буде передбачати на 40 місяців вперед. Ну і в кінці подивимось на графік та показники.

```python
def generate_and_analyze_synthetic_data(dates: list, real_inflation_values: list, is_printed: bool, predict: bool,
                                        degree=15) -> tuple:
    noise_std = 0.2

    np.random.seed(0)
    coefficients = np.polyfit(np.arange(len(dates)), real_inflation_values, degree)
    if predict:
        last_date = max(dates)
        additional_dates = [last_date + timedelta(days=30 * i) for i in range(1, 41)]
        dates += additional_dates
        synthetic_trend_values = np.polyval(coefficients, np.arange(len(dates)))
    else:
        synthetic_trend_values = np.polyval(coefficients, np.arange(len(dates)))
    print("\nModel:")
    print(np.poly1d(coefficients))
    synthetic_inflation_values = synthetic_trend_values + np.random.normal( loc: 0, noise_std, len(dates))
```



4. І останній етап подивимось статистичні критерії нашої вихідної моделі.

```
/home/selik/Projects/Data-Science-Labs/venv/bin/python /home/selik/Projects/Data-Science-Labs/Lab_2/lab2.py
Кращі параметри для очистки функції: {'contamination': 0.05}

Model:
            3             2
-2.894e-07 x + 0.0001116 x - 0.01476 x + 2.809
Real Data Statistics:
Mean inflation value: 2.53008356545961
Standard deviation of inflation: 1.6321415580274319
Median inflation: 2.3
Minimum inflation value: -2.1
Maximum inflation value: 9.1
Inflation variance: 2.663886065440213
Mean Squared Error (MSE): 1.3070555555555559
Root Mean Squared Error (RMSE): 1.1432653040985525
```

Висновки: була розроблена програма, яка очищає вхідні дані залежності інфляції долара від аномалії, робить передбачення на наступні 4 роки та виводить статистичні показники.