УДК 004.432.2

ИСПОЛЬЗОВАНИЕ БИБЛИОТЕКИ NUMPY И ЕЕ МОДУЛЯ POLYFIT В НАУКЕ О ДАННЫХ

Амельченя М.А.

Белорусский государственный университет информатики и радиоэлектроники, г. Минск, Республика Беларусь

Научный руководитель: Алексеев В.Ф. – канд.техн.наук, доцент, доцент кафедры ПИКС

Аннотация. В работе показано применение библиотеки *NumPy* для исследования массивов данных. Изучен модуль *NumPy.polyfit()* библиотеки *NumPy*. Исследована возможность использования модуля *NumPy.polyfit()* для выполнения полиномиальной регрессии. Дана оценка полезности использования модуля для построения регрессии.

Ключевые слова: *NumPy*, полиномиальная регрессия, *NumPy.polyfit*

Введение. Библиотека *NumPy* является фундаментальной библиотекой *Python* для проведения вычислений в *Data Science*. Ключевыми качествами, которыми обладает данная библиотека, являются:

– скорость: массивы *NumPy* работают в 20 раз быстрее, чем стандартные списки *Python*;

– производительность: *NumPy* сочетает в себе простоту использования *Python* со скоростью *C*;

– вычислительные инструменты: *NumPy* обладает широким спектром математических функций и вычислительных инструментов практически для всех нужд, использование данной библиотеки позволяет с легкостью выполнять такие операции, как подгонка кривой, оптимизация, линейная алгебра, преобразования и т.д.

NumPy является основой, на которой построены другие многочисленные библиотеки для научных вычислений:

– библиотеки статистики и *Machine Learning*: *SciPy, Scikit-Learn, SpaCy*;

– библиотеки визуализации: *Matplotlib, Bokeh, Altair*;

– *ETL*: *Pandas*.

Основная часть. Рассмотрим простой пример. Предположим, нам необходимо добраться из пункта А в пункт Б. Существует несколько способов передвижения. Вы можете воспользоваться велосипедом, попытаться пройтись пешком, или, возможно, захотите взять автомобиль. Появляются дополнительные варианты, среди которых у вас есть возможность взять спортивный автомобиль или внедорожник. Если вы не знаете тип местности, по которой будете ехать, вам, возможно, будет лучше использовать внедорожник – универсальное транспортное средство. Однако, если вы стремитесь завершить путешествие в кратчайшие сроки, и вы знаете, что дороги хорошо асфальтированы, вы выберете спортивный автомобиль, потому что сможете быстро преодолеть расстояние. В этом и заключается разница между стандартным списком *Python* и массивом *NumPy*.

Список *Python* – это внедорожник, в то время как массив *NumPy* – это спортивный автомобиль [1]. Список *Python* предназначен для обработки всех видов данных, которые можно вставить в список. Предположим, что данные представляют собой однородные данные числового типа (целочисленные, с плавающей запятой, логические, дата-время и т.д.) и требуют большого объема вычислений. В этом случае лучше использовать *NumPy*.

Рассмотрим простой пример ипользования. Допустим, у нас есть простой список из пяти элементов. И мы хотим создать еще один список, содержащий результат деления каждого из этих элементов на три. Нам придется перебирать список и выполнять операцию по элементам, потому что выполнение операции деления по всему списку приведет к ошибке.

```
Mylist = [9,18,27,36,45]
Mylist / 3
```

```
TypeError: unsupported operand type(s) for /: 'list' and 'int'
```

Несмотря на то, что каждый из этих пяти элементов является числовым, списки *Python* не позволят выполнить эту операцию деления, поскольку операции *Python* предназначены для работы со всеми возможными случаями. Поскольку список не обязательно должен содержать только числовые типы данных, *Python* этого не допустит.

Используем *NumPy*. С помощью массива *NumPy* мы можем просто разделить весь массив на пять.

```
Myarray = np.array(mylist)
Myarray / 3
Myarray([ 3., 6., 9., 12., 15.])
```

В библиотеке *NumPy* есть опция, которая дает пользователю возможность выполнить линейную регрессию (простую и полиномиальную), используя метод наименьших квадратов в качестве критерия минимизации. Модуль, который выполняет эту регрессию, называется *polyfit*: *NumPy.polyfit(x, y, deg, rcond=None, full=False, w=None, cov=False)*.

Выше был приведен общий синтаксис функции *NumPy.polyfit()*. Данная функция имеет три обязательных параметра и четыре необязательных параметра, которые по-своему влияют на результат. Параметр *x* представляет собой набор точек, которые должны быть представлены вдоль оси *X, Y* – параметр, который представляет все наборы точек, которые должны быть представлены вдоль оси *Y*.

Кроме того, *NumPy.polyfit()* дает возможность указать степень полиномиальной регрессии с помощью параметра *deg*, а также может вычислить ковариационную матрицу *cov*, которая дает важную информацию о коэффициентах полиномиальной регрессии. Модуль *polyfit* подгоняет данные с помощью метода наименьших квадратов и внутренне запоминает коэффициенты линейной регрессии, найденные во время процедуры подгонки.

Параметр *rcond* является необязательным параметром, который отвечает за определение относительного числового условия подгонки. Параметр *full* – необязательный параметр, который определяет характер возвращаемого значения, по умолчанию значение равно *false*, из-за чего возвращаются только коэффициенты. Если значение задано равным *true*, то также возвращается сингулярное значение декомпозиции. Параметр w представляет веса, применяемые к координате y точек выборки [2].

Для построения функции линейной регрессии необходимо преобразовать уже найденные полиномиальные коэффициенты в полиномиальную функцию с помощью функции *NumPy.poly1d()* [3].

В качестве примера, я использую функцию *NumPy.polyfit()* для выполнения полиномиальной регрессии (*n*=1) для массива *x*, хранящего стоимость проданных на аукционе монет, и массива *y*, хранящего даты продажи монеты. При этом используется следующий код на *Python*:

```
def predict_coin_price(x_period, y_period, start_date, days_to_predict):
  mymodel = NumPy.poly1d(NumPy.polyfit(x_period, y_period, 1))
  current_day = (timezone.now() start_date).days
  for i in range(current_day, current_day+days_to_predict):
        price = mymodel(i)
        if price < 0:
              x_period.append(i)
              y_period.append(0)
              break
        x_period.append(i)
        y_period.append(round(price, 2))
```

Запустив приведенный выше код *Python* на своем компьютере, можно показать график полиномиальной регрессии. *NumPy.polyfit()* дает в результате линию полиномиальной регрессии, представленную на рисунке 1.

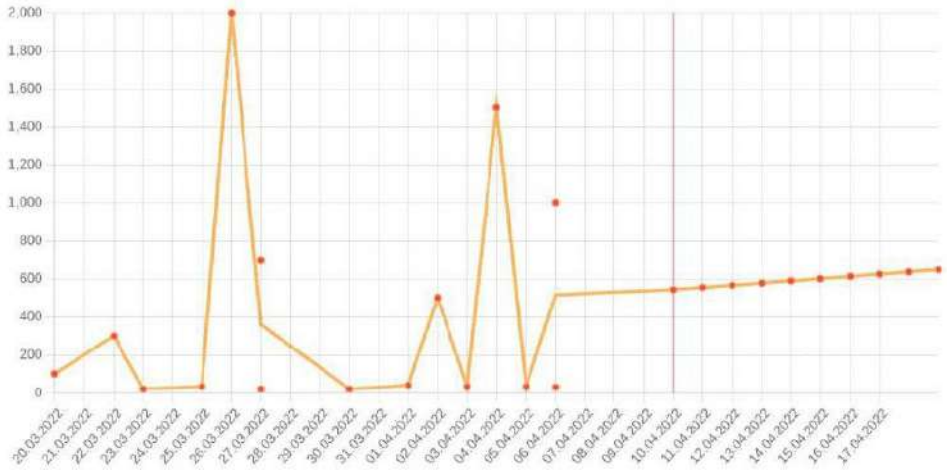

Рисунок 1 – Линия полиномиальной регрессии, выполненная с помощью модуля *polyfit*

Кроме того, *polyfit* дает пользователю возможность узнать коэффициенты линейной или полиномиальной регрессии.

Заключение. Модуль *polyfit* очень полезен для подгонки простой линейной регрессии и полиномиальной регрессии степени *n*. Однако это не дает пользователю возможности использовать линейную регрессию с несколькими предикторными переменными, а именно многомерную регрессию. Таким образом, невозможно использовать *NumPy.polyfit()* для нескольких входных переменных, а только для одной входной переменной. Кроме того, модуль *polyfit* не дает пользователю возможности напрямую вычислять коэффициент детерминации R^2 для оценки степени соответствия, коэффициент корреляции Пирсона *r*, *p*-значение проверки гипотез и ошибки выборки, связанные с коэффициентами регрессии.

Список литературы

1. *NumPy for Data Science Interviews [Электронный ресурс]. – Режим доступа: https://towardsdatascience.com/NumPy-for-data-science-interviews-1f86e7277ddd. – Дата доступа: 12.04.2022.*

2. *NUMPY POLYFIT объяснен примерами [Электронный ресурс]. – Режим доступа: https://pythobyte.com/NumPy-polyfit-22163/. – Дата доступа: 12.04.2022.*

3. *Five Regression Python Modules That Every Data Scientist Must Know [Электронный ресурс]. – Режим доступа: https://towardsdatascience.com/five-regression-Python-modules-that-every-data-scientist-must-know-a4e03a886853. – Дата доступа: 12.04.2022.*

UDC 004.432.2

USING THE *NUMPY* LIBRARY AND ITS POLYFIT MODULE FOR DATA SCIENCE

Amialchenia M.A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Alexeeev V.F. – PhD, assistant professor, associate professor of the department of ICSD

Annotation. The article shows the use of the *NumPy* library for the research of data arrays. The *NumPy*.polyfit() module of the *NumPy* library has been studied. The possibility of using the *NumPy*.polyfit() module to perform polynomial regression is investigated. The usefulness of using the module for regression construction is evaluated.

Keywords: *NumPy*, polynomial regression, *NumPy*.polyfit