**Постановка задачи**: с помощью библиотеки sklearn и matplotlib реализовать 3 метода регрессии на примере двумерных данных на языке программирования *python*

**Этапы выполнения задачи**:

1. **Сгенерировать исходную функцию**

Для генерации исходной функции рекомендуется использовать DeepSeek. Для этого необходимо задать вопрос в новом чате: «***Сгенерируй, пожалуйста, трансцендентно-алгебраическую функцию***». В ответе на данный вопрос DeepSeek предложит случайную функцию (будем называть её )

1. **Определить, является ли сгенерированная функция линейной или нелинейной**
2. **Выбрать 3 метода регрессии для решения задачи**. Ниже приведены 5 методов для линейной и 7 методов для нелинейной регрессии, которыми можно воспользоваться:
   1. Методы линейной регрессии
      * Линейная регрессия (linear regression)
      * Ridge регрессия
      * Lasso регрессия
      * Elastic net
      * Bayessian ridge
   2. Методы нелинейной регрессии
      * Kernel ridge
      * SVR (support vector regression)
      * nuSVR
      * Деревья решений для регрессии (Decision Tree Regressor)
      * Случайный лес для регрессии (Random Forest Regressor)
      * Крутые деревья для регрессии (Extra Trees Regressor)
      * Градиентный бустинг для регрессии (Gradient Boosting Regressor)
3. **Сгенерировать исходные данные на основе полученной функции с заданным шумом**

Всего будет 100 точек для регрессии. Координаты точек по х можно сгенерировать с помощью функции linspace библиотеки numpy:

| import numpy as np x = np.linspace(x\_mix, x\_max, 100) |
| --- |

где ***x\_min*** – значение координаты по х у первой точки

***x\_max*** – значение координаты по х у последней точки

Координаты по y можно сгенерировать следующим способом:

где – некоторый шум, который можно задать, например, следующим образом:

| import random e = [random.uniform(-1, 1) for \_ in range(100)] |
| --- |

5) **Реализовать 3 метода регрессии с помощью библиотеки sklearn**:

Для этого необходимо обратиться к документации sklearn и найти код для создания модели, её обучения и получения прогнозов. В большинстве случаев код будет идентичен и будет похож на следующий кусочек кода:

| model = <название\_класса\_регрессии>() model.fit(x, y) y\_pred = model.predict(x) |
| --- |

6) **Построить графики, исходя из полученных результатов**

По итогам работы регрессий необходимо построить 3 графика для каждого метода. На этих графиках должны присутствовать 3 элемента:

* исходные точки (синий цвет)
* график исходной функции (зелёный цвет)
* график функции, полученной в результате регрессии (красный цвет)

7) **Сделать выводы о том, какая регрессия работает лучше на выбранных данных, а какая хуже**. Сделать этот вывод на основе визуального наблюдения за графиками, а также на основе значения среднеквадратичной ошибки (MSE) по итогам работы методов регрессии