**Отчет к заданию № 1**

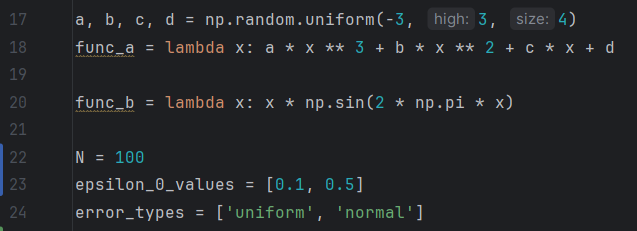
**Первая часть** задания состояла в том, чтобы сгенерировать выборки для заданных функций с моделированием случайной ошибки.

Были даны варианты функций и варианты распределения ошибки на заданном интервале:

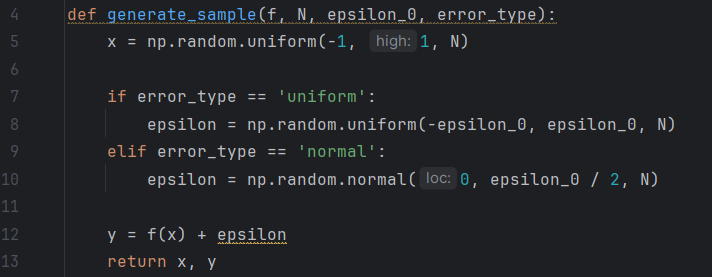
**Варианты распределения ошибки на интервале**   
a) ошибка распределена равномерно;  
б) ошибка распределена нормально.

**Варианты функции f:**  
a) , коэффициенты a,b,c,d сгенерированы случайно из интервала [−3,3];  
б) .

Реализация функций и ошибок:

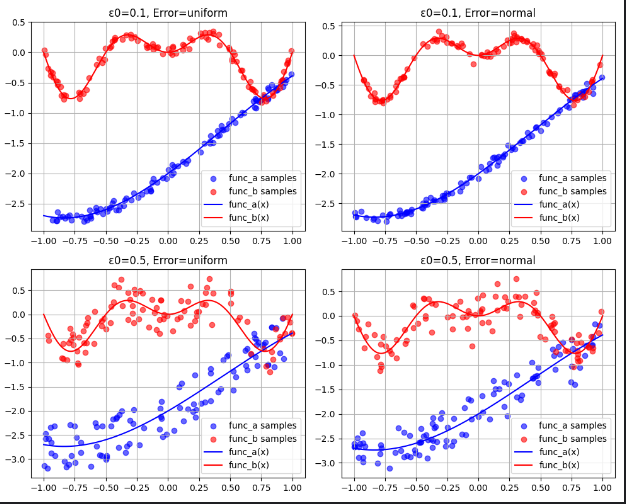


Сами выборки генерировались следующим образом:



\*Саму программу приложил в конце документа

Рассмотрим полученные результаты выборок **(для норм. распред. μ= 0** **и σ =** **)**:

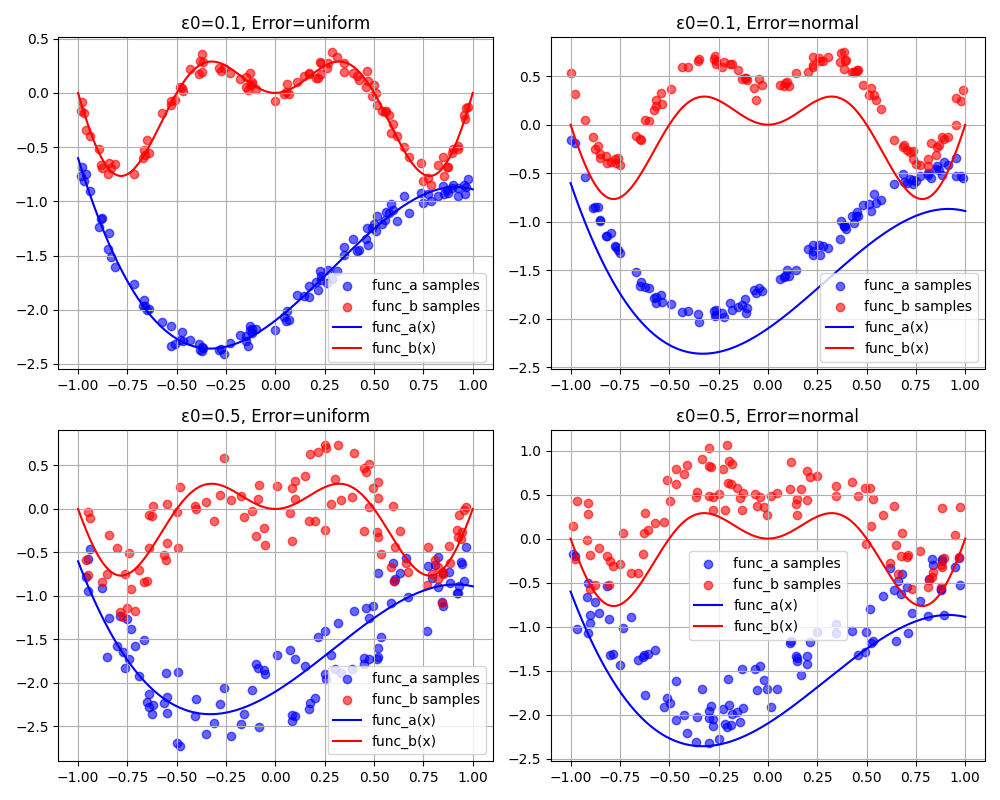
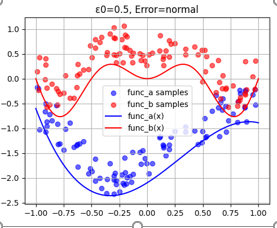


Прямыми обозначены функции, точки обозначают выборку соответственную своей функции и ошибки.

На графиках видно, что чем меньше ошибка , тем меньше расхождения c оригинальной функцией. Кроме этого, можно увидеть, что распределение ошибки совпадает с распределением отделенности точек на графике.

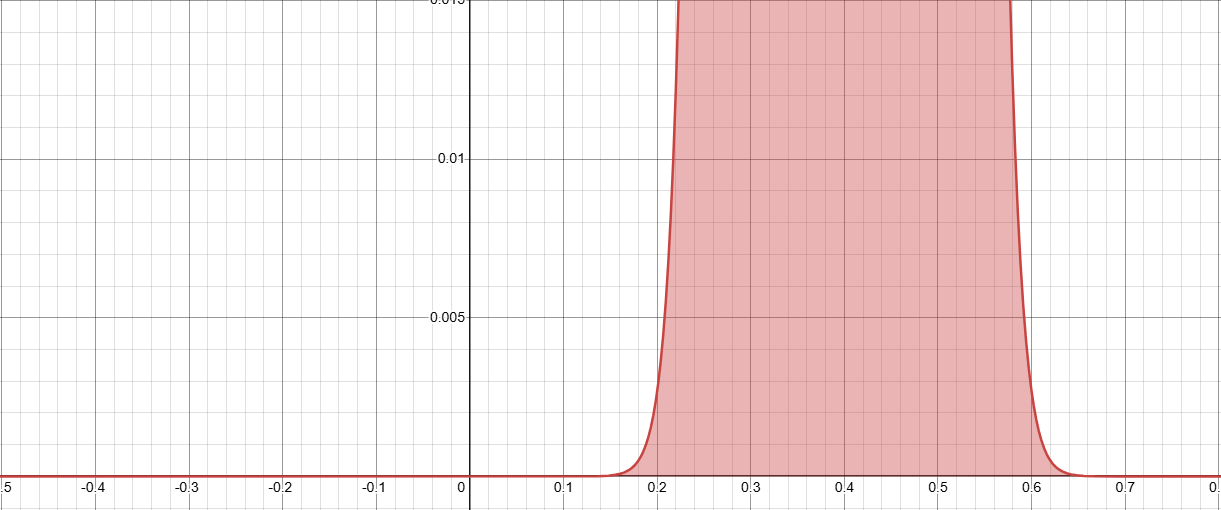
Рассмотрим еще одну выборку, для которой изменили мат. ожидание на

**μ= 0,4, а значение σ = ε0/2 оставим прежним**:

Из-за изменения мат ожидания поменялось и расположение выборки. Картина на графике с выборкой вся сместилась вверх, тк сдвиг сместил центр “горба”, и все значения оказались ближе лежащими к положительной границе промежутка

***;* μ= 0,4; σ = ε0/2**



***;* μ= 0,4; σ = ε0/2**



**Во второй** части задания попросили реализовать с помощью полиномиальной регрессии восстановление функциональной зависимости, по которой была получена выборка и привести примеры выборок и степеней полиномов, при которых:

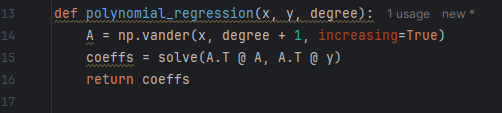
а) происходит недообучение,

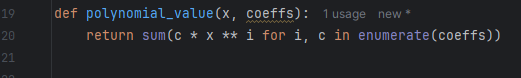
б) происходит переобучение,

в) полученная функциональная зависимость пригодна для прогнозирования значения восстанавливаемой функциональной зависимости в x, не содержащемся в выборке.

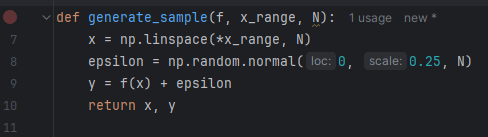
Реализация основывалась на информации с лекции:

Говорилось, что полиномиальную регрессию можно реализовать при помощи матрицы Вандермонда



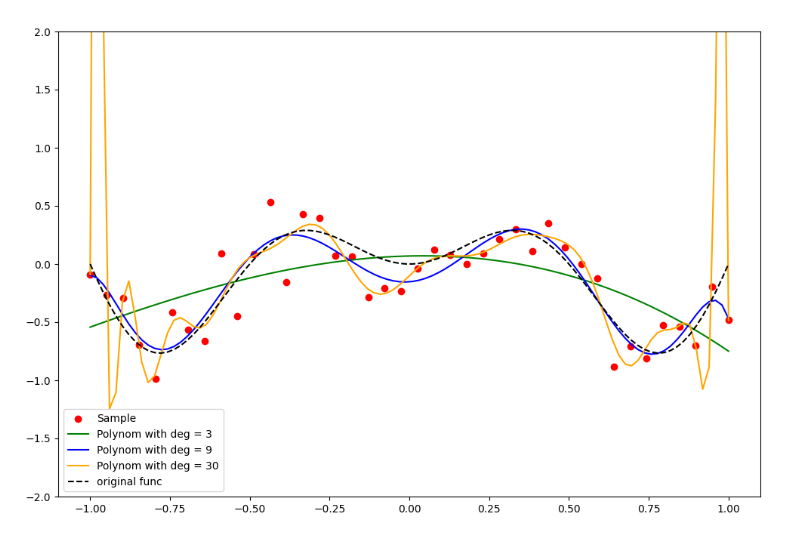


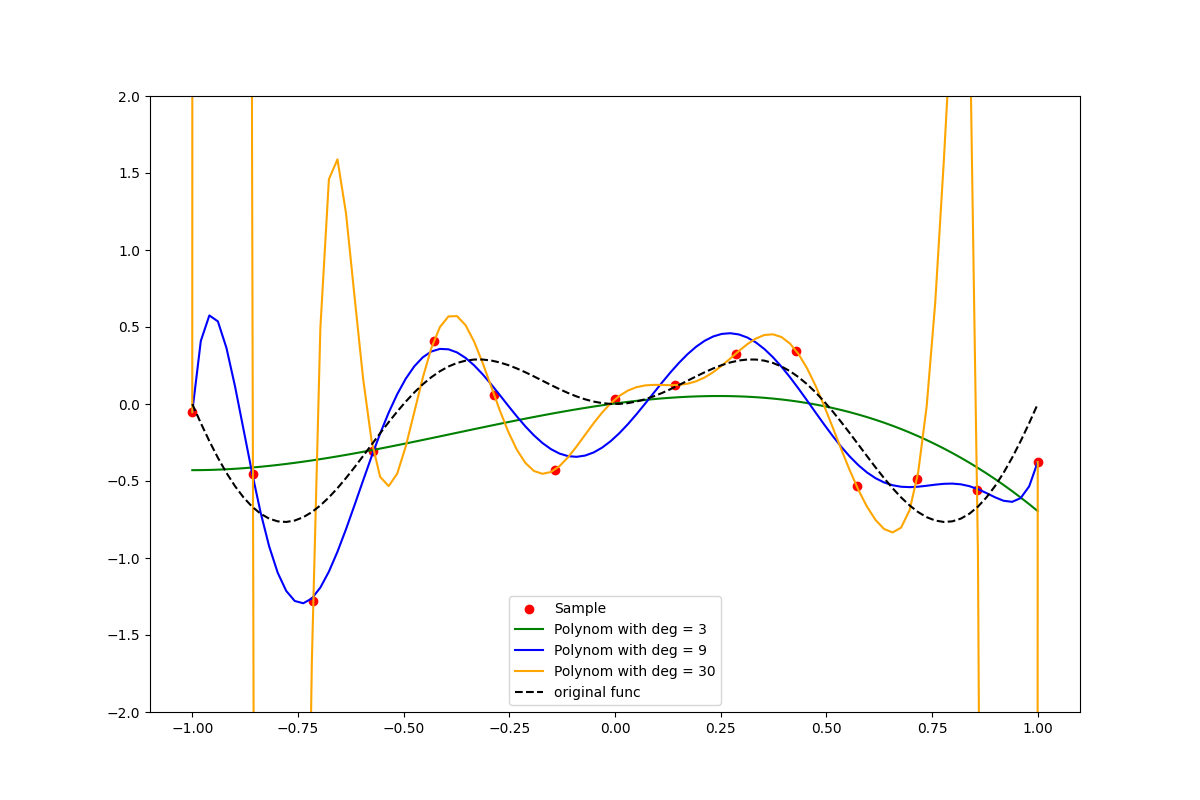
Генерация x происходила по равномерному распределению, а генерация ошибки по нормальному



Приближалась функция: 

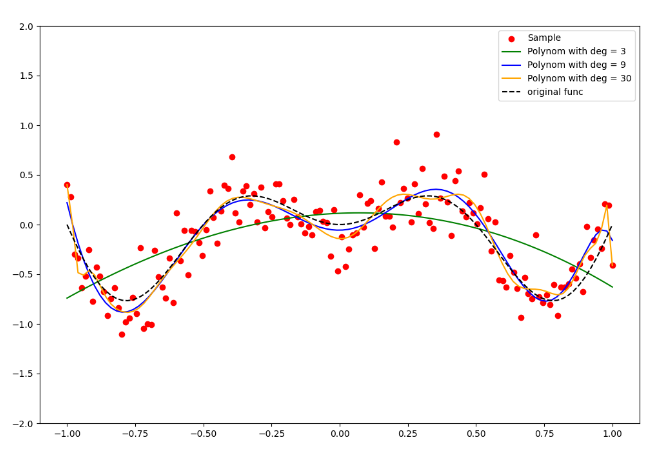
**(на всех граффиках пописанны степени полиномов)**





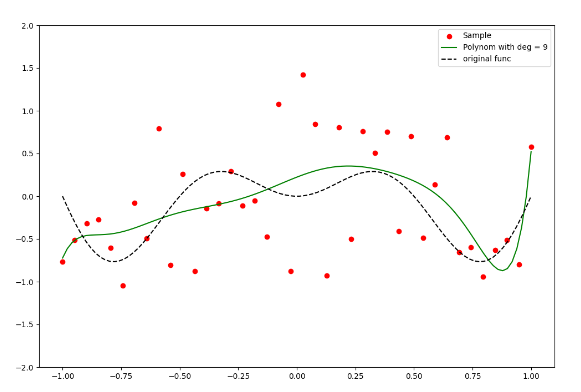
Первый результат от второго отличается количеством точек в выборке (N1 = 40, N2 = 15). Итак, из графиков можно увидеть, что степень полинома может оказаться слишком большой для данной выборки, что вызовет переобучения. А если же степень полинома слишком мала, то полученное приближение будет не совпадать с приближаемой функцией.

Посмотрим на выборку при N = 150



На этом графике уже видно, что полином 30 степени тоже достаточно неплохо приближает функцию.

Давайте заметим, что для решения задачи полиномиальной регрессии методом решения Вандермонда достаточно просто вызвать переобучение, и даже при небольших значениях ошибки появляются расхождения с оригинальной функцией, а тк при этом подходе полином пройдет точно через точки из обучающей выборки, то при неудачном их выборе полином не будет приближать функцию вовсе.

Пример (до этого на графиках полином 9 степени очень хорошо приближал функцию):  
  


Данный случай я бы классифицировал как недообучение, тк тут не достигается достаточная точность приближения. Возможные решения именно этого случая – использовать другой метод, например, MSE (mean squared error) или же LSM (least squares method).

Вывод:

Я сгенерировал различные выборки с разными распределениями. Кроме этого, смог добиться переобучения, обучения и недообучения в методе полиномиальной регрессии.