

**Федеральное государственное образовательное
бюджетное учреждение
высшего образования**

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ
ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ
ФЕДЕРАЦИИ»**

(Финансовый университет)

**Факультет
информационных технологий и анализа больших данных
Кафедра «Прикладная математика и информатика»**

Домашнее задание № 4

«Регрессия»

Студенты группы ПМ19-3:

Филимонова Ю.М.

Корнева Т.А.

Косовская Т.П.

Дубровская А.А.

Кривоносова Д.В.

Руководитель:

Аксенов Дмитрий Андреевич

Москва 2022

Содержание

1	Постановка задачи	3
2	Алгоритмы и математические модели	4
2.1	Алгоритм и математическая модель функции, реализующей модель линейной регрессии с 2 видами регуляторов: L1 и L2	5
2.1.1	Описание входных данных	5
2.1.2	Описание алгоритма решения и математической модели . .	5
2.1.3	Описание выходных данных	6
2.2	Алгоритм и математическая модель функции, реализующей модель полиномиальной регрессии с 2 видами регуляторов: L1 и L2	6
2.2.1	Описание входных данных	6
2.2.2	Описание алгоритма решения и математической модели . .	6
2.2.3	Описание выходных данных	6
2.3	Алгоритм и математическая модель функции, реализующей модель экспоненциальной регрессии с 2 видами регуляторов: L1 и L2	7
2.3.1	Описание входных данных	7
2.3.2	Описание алгоритма решения и математической модели . .	7
2.3.3	Описание выходных данных	7
3	Варианты использования системы	8
3.1	Вариант использования системы для построения линейной и экспоненциальной моделей	8
3.2	Вариант использования системы для построения полиномиальной регрессии	9
3.3	Примеры работы	9
4	Архитектура решения	12
4.1	Функции считывания информации	12
4.2	Функции обработки информации	12
4.3	Функции вывода информации	15
4.4	Вспомогательные функции	17
5	Тестирование	18
6	Заключение	20

1. Постановка задачи

“Рассчитать объемы продаж от затрат заказчика на закупку рекламы”

Необходимо в среде программирования Python реализовать функции, которые будут находить примерные объемы продаж при известных затратах на рекламу.

Условия заказчика:

Заказчик занимается продажей одежды в интернете. Он решил увеличить затраты на рекламу для привлечения новых клиентов. Заказчик планирует в следующем месяце вложить в рекламу 50 тыс рублей, через 2 месяца - 60 тыс рублей, через 3 месяца - 70 тыс рублей.

Ему необходимо узнать, какой объем продаж ожидается при известных затратах на рекламу для того, чтобы заранее договориться с заводами о пошиве новых изделий и чтобы пополнить запасы на складе для перебойной доставки товаров клиентам.

Также он хочет составить таблицу, где будут указаны выделенные денежные средства на рекламу и планируемые объемы продаж, чтобы занести эти данные в квартальный отчет компании. Таблица выглядит следующим образом:

Таблица 1. План заказчика

	1-й месяц	2-й месяц	3-й месяц
Затраты на рекламу	50 тыс руб.	60 тыс руб.	70 тыс руб.
Объем продаж	?	?	?

Постановка задачи:

Необходимо по имеющимся данным (см. файл “Данные заказчика” по ссылке: https://drive.google.com/drive/folders/17AKMyTM1z3H467T1anIBn_2-C3gHgTr-?usp=sharing) построить регрессию и вывести уравнение регрессии, по которому будут рассчитываться ожидаемые объемы продаж на каждый месяц.

2 Алгоритмы и математические модели

В разделе описываются различные алгоритмы и математические модели, выбранные для создания конечного продукта.

Для решения выше поставленной задачи, необходимо построить регрессию и вывести уравнение регрессии, по которому будут рассчитываться примерные затраты на рекламу каждый месяц. В данной работе реализовано 3 метода построения регрессии: линейная, полиномиальная и экспоненциальная регрессии с 2 видами регуляторов: L1, L2.

В L1-регуляризации, которая также называется регуляризацией лассо-регрессией, мы используем специальное L1-нормирование:

$$J_1 = \sum_{i=1}^N (y_n - \hat{y}_n)^2 + \lambda ||w||_1 \quad (1)$$

Запишем J_1 :

$$J_1 = (Y - Xw)^T (Y - Xw) + \lambda |w| \quad (2)$$

Возьмем производную, приравняем к 0 и решим относительно w

$$\frac{\partial J_1}{\partial w} = -2X^T Y + 2X^T Xw + \lambda \text{sign}(w) = 0 \quad (3)$$

Для решения задачи необходимо воспользоваться градиентным спуском. В L2-регуляризации, которая называется регрессией Риджа, мы используем L2-нормирование для поправки весовых коэффициентов:

$$J_2 = \sum_{i=1}^N (y_n - \hat{y}_n)^2 + \lambda ||w||_2^2 \quad (4)$$

где

$$||w||_2^2 = w^T w = w_1^2 + w_2^2 + \dots + w_D^2 \quad (5)$$

Запишем J_2 :

$$J_2 = (Y - Xw)^T (Y - Xw) + \lambda w^T w \quad (6)$$

Возьмем производную, приравняем к 0 и решим относительно w

$$\frac{\partial J_2}{\partial w} = -2X^T Y + 2X^T Xw + 2w\lambda = 0 \quad (7)$$

Из этого получаем

$$w = (\lambda + X^T X)^{-1} X^T Y \quad (8)$$

Приведем математические модели 3-х методов в общем виде и их алгоритмы.

2.1 Алгоритм и математическая модель функции, реализующей модель линейной регрессии с 2 видами регуляторов: L1 и L2

2.1.1 Описание входных данных

На вход подаются массив предикторов (переменные X), массив предсказываемой переменной (переменная Y) и дополнительные параметры: вид регуляции и флаг для возможно построения графика.

2.1.2 Описание алгоритма решения и математической модели

В данном методе связь переменной Y с переменными X_1, \dots, X_n :

$$Y = \beta_0 + \beta_1 X_1 + \dots + \beta_n X_n + \epsilon \quad (9)$$

где $\beta_0, \beta_1, \dots, \beta_n$ - вещественные регрессионные коэффициенты, ϵ - случайная величина, являющаяся ошибкой прогнозирования.

При вычислении оценки вектора параметров $\beta = (\beta_0, \dots, \beta_n)$ в случае многомерной линейной регрессии удобно использовать матрицу плана X размера $m \times (n + 1)$, которая строится по обучающей выборке \tilde{S}_t . j -я строка матрицы плана представляет собой вектор значений переменных X_1, \dots, X_n для объекта s_{jc} , 1.

Пусть $y = (y_1, \dots, y_m) = Y.Y X_1, \dots, X_n$

$$\vec{y} = X\vec{\beta} + \vec{\epsilon} \quad (10)$$

где $\epsilon = (\epsilon_1, \dots, \epsilon_m)$ - вектор ошибок прогнозирования для объектов \tilde{S}_t . Функционал $Q(\tilde{S}_t, \beta_0, \beta_1, \dots, \beta_n)$ может быть записан в виде

$$Q(\tilde{S}_t, \beta_0, \beta_1, \dots, \beta_n) = \sum_{j=1}^m [y_j - \beta_0 - \sum_{i=1}^n \beta_i \dot{x}_{ji}]^2 \quad (11)$$

где \dot{x}_{ji} - элементы матрицы плана X , определяемые равенствами $\dot{x}_{j1} = 1, \dot{x}_{j1} = \dot{x}_{ji-1}$ при $i > 1$

Необходимым условием минимума функционала $Q(\tilde{S}_t, \beta_0, \beta_1, \dots, \beta_n)$ можно записать в матричном виде

$$-2X^t y^t + 2X^t X \beta^t = 0 \quad (12)$$

Из этого следует, что вектор оценок значений регрессионных коэффициентов находится по следующей формуле:

$$\hat{\beta}^t = (X^t X)^{-1} X^t y^t \quad (13)$$

2.1.3 Описание выходных данных

Выходные данные представляют собой получившуюся функцию в аналитическом виде, массив коэффициентов регрессии (веса каждого x), значение свободного члена регрессии и график регрессии, если стоит соответствующий параметр.

2.2 Алгоритм и математическая модель функции, реализующей модель полиномиальной регрессии с 2 видами регуляторов: L1 и L2

2.2.1 Описание входных данных

На вход подаются массив предикторов (переменные X), массив предсказываемой переменной (переменная Y) и дополнительные параметры: вид регуляции и флаг для возможно построения графика.

2.2.2 Описание алгоритма решения и математической модели

Возведем экзогенные переменные в степень от 1 до m , где m - степень полинома. Получим модель множественной регрессии, которая имеет вид:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \dots + \beta_m x_i^m + \epsilon_i \quad (14)$$

В матричном виде модель имеет такой же вид, как и в формуле (10)

$$\vec{y} = X\vec{\beta} + \vec{\epsilon}$$

Отсюда получим вектор оцениваемых коэффициентов полиномиальной регрессии, который находится по формуле (13), т.е.

$$\hat{\beta}^t = (X^t X)^{-1} X^t y^t$$

Таким образом, подставив в формулу $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \dots + \beta_m x_i^m + \epsilon_i$ значения $\hat{\beta}^t$, получим уравнение регрессии с известными коэффициентами независимых переменных.

2.2.3 Описание выходных данных

Выходные данные представляют собой получившуюся функцию в аналитическом виде, массив коэффициентов регрессии (веса каждого x), значение свободного члена регрессии и график регрессии, если стоит соответствующий параметр.

2.3 Алгоритм и математическая модель функции, реализующей модель экспоненциальной регрессии с 2 видами регуляторов: L1 и L2

2.3.1 Описание входных данных

На вход подаются массив предикторов (переменные X), массив предсказываемой переменной (переменная Y) и дополнительные параметры: вид регуляции и флаг для возможно построения графика.

2.3.2 Описание алгоритма решения и математической модели

Модель полиномиальной регрессии имеет вид:

$$\ln(Y) = \beta_0 + \beta_1 X_1 + \dots + \beta_n X_n + \epsilon \quad (15)$$

Заменим $\ln(Y)$ на Y , получим модель линейной регрессии, т.е.

$$Y = \beta_0 + \beta_1 X_1 + \dots + \beta_n X_n + \epsilon$$

Отсюда найдем вектор оцениваемых коэффициентов экспоненциальной регрессии, который находится по формуле (13) $\hat{\beta}^t = (X^t X)^{-1} X^t y^t$

Таким образом, подставив в формулу $\ln(Y) = \beta_0 + \beta_1 X_1 + \dots + \beta_n X_n + \epsilon$ значения $\hat{\beta}^t$, получим уравнение регрессии с известными коэффициентами независимых переменных.

2.3.3 Описание выходных данных

Выходные данные представляют собой получившуюся функцию в аналитическом виде, массив коэффициентов регрессии (веса каждого x), значение свободного члена регрессии и график регрессии, если стоит соответствующий параметр.

3. Варианты использования системы

3.1. Вариант использования системы для построения линейной и экспоненциальной моделей

При работе с генерацией и построением данных для линейной и экспоненциальной моделей имеется следующий вариант использования системы (* - необязательный параметр):

1. Пользователь вводит количество регрессоров (предикатов):
 - a. Пользователь должен ввести целое число без лишних символов.
 - b. Допустимо любое количество предикатов, но пользователю стоит учитывать, что точность модели может уменьшаться с увеличением независимых переменных.
2. Пользователь вводит количество наблюдений:
 - a. Пользователь должен ввести целое число без лишних символов.
 - b. Допустимо любое количество наблюдений, но пользователю стоит учитывать, что слишком маленькое или наоборот большое количество наблюдений может уменьшить точность модели.
3. Пользователь должен ввести параметр, отвечающий за дальнейший ввод дополнительных параметров:
 - a. При вводе 1(Да) пользователь продолжит ввод данных.
 - b. При вводе 0(Нет) оставшиеся параметры будут взяты по умолчанию, и ввод остановится.
 - c. Допустимы лишь значения 0 и 1.
4. *Пользователь вводит вид регуляризации:
 - a. Допустимы значения 0, 1 и 2.
 - b. При вводе 0, регрессия будет построена без регулятора.
 - c. При вводе 1, регрессия будет построена с L1-регуляризацией.
 - d. При вводе 2, регрессия будет построена с L2-регуляризацией.
5. *Пользователь вводит параметр для построения графика:
 - a. Допустимы значения 0 и 1.
 - b. При вводе 0(Нет) график разброса данных и оцененной регрессии не будет построен.
 - c. При вводе 1(Да) график разброса данных и оцененной регрессии будет построен.
 - d. Построение графика доступно для одного регрессора(предиката).

3.2. Вариант использования системы для построения полиномиальной регрессии

При работе с генерацией и построением данных для полиномиальной модели имеется следующий вариант использования системы (* - необязательный параметр):

1. Пользователь вводит степень полинома:
 - a. Пользователь должен ввести целое число без лишних символов.
 - b. Допустим ввод любой степени полинома, но пользователю стоит учитывать, что точность модели может уменьшаться с увеличением степени.
2. Пользователь вводит количество наблюдений:
 - a. Пользователь должен ввести целое число без лишних символов.
 - b. Допустимо любое количество наблюдений, но пользователю стоит учитывать, что слишком маленькое или наоборот большое количество наблюдений может уменьшить точность модели.
3. Пользователь должен ввести параметр, отвечающий за дальнейший ввод дополнительных параметров:
 - a. При вводе 1(Да) пользователь продолжит ввод данных.
 - b. При вводе 0(Нет) оставшиеся параметры будут взяты по умолчанию, и ввод остановится.
 - c. Допустимы лишь значения 0 и 1.
4. *Пользователь вводит вид регуляризации:
 - a. Допустимы значения 0, 1 и 2.
 - b. При вводе 0, регрессия будет построена без регулятора.
 - c. При вводе 1, регрессия будет построена с L1-регуляризацией.
 - d. При вводе 2, регрессия будет построена с L2-регуляризацией.
5. *Пользователь вводит параметр для построения графика:
 - a. Допустимы значения 0 и 1.
 - b. При вводе 0(Нет) график разброса данных и оцененной регрессии не будет построен.
 - c. При вводе 1(Да) график разброса данных и оцененной регрессии будет построен.

3.3. Примеры работы

3.3.1. Линейная регрессия

Введите количество регрессоров. Пример: 2. Ввод: 3
Введите количество наблюдений. Пример: 100. Ввод: 100
Хотите ввести дополнительные параметры? 1-Да/0-Нет. Если нет, будут взяты параметры по умолчанию. Ввод:0
Коэффициенты независимых переменных {x0: 21.2069, x1: -10.19024, x2: -27.00405}
Свободный член регрессии: 15.53587
Уравнение регрессии: $21.2069 \cdot x_0 - 10.19024 \cdot x_1 - 27.00405 \cdot x_2 + 15.53587$
Стандартная квадратичная ошибка: 81435.77819993299

рис.1 Пример работы системы с генерацией линейной регрессии - 1

Введите количество регрессоров. Пример: 2. Ввод: 1
Введите количество наблюдений. Пример: 100. Ввод: 100
Хотите ввести дополнительные параметры? 1-Да/0-Нет. Если нет, будут взяты параметры по умолчанию. Ввод:1
Введите вид регуляризации
0 - без регуляризации, 1 - L1-регуляризация, 2 - L2-регуляризация
Пример: 2. Ввод:1
Хотите ли выполнить построение графика (доступно только для парной линейной регрессии)? 1-Да/0-Нет. Ввод:1
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:20: RuntimeWarning: invalid value encountered in subtract
Коэффициенты независимых переменных {x0: 31.57161}
Свободный член регрессии: 2.08632
Уравнение регрессии: $31.57161 \cdot x_0 + 2.08632$
Стандартная квадратичная ошибка: 19942.912228243124

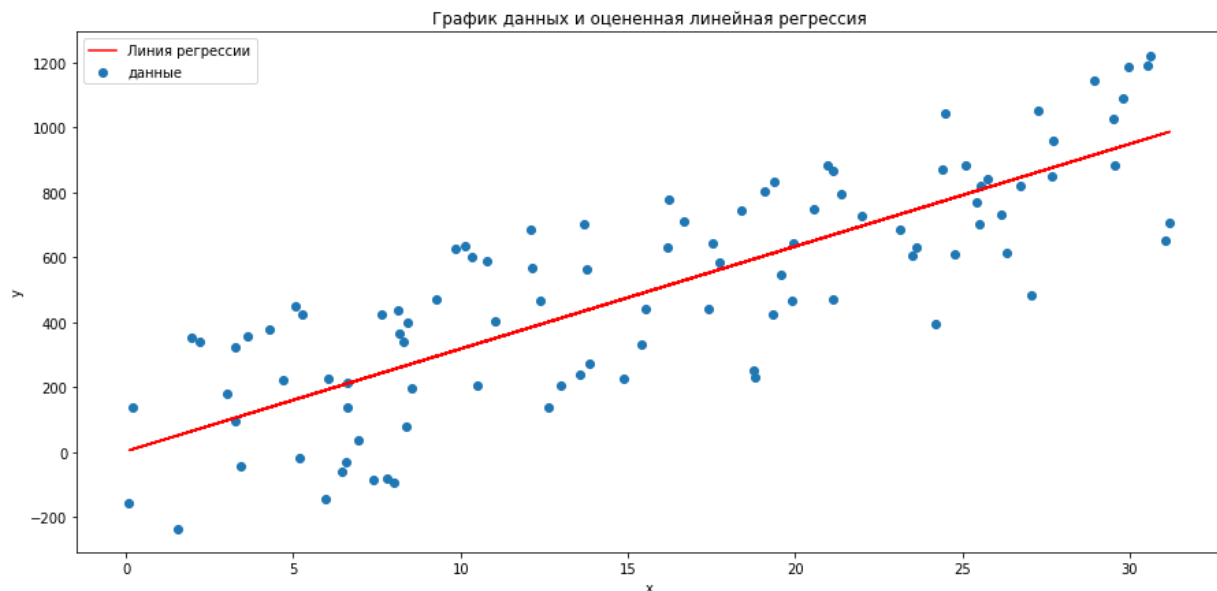


рис.2 Пример работы системы с генерацией линейной регрессии - 2

3.3.2. Полиномиальная регрессия

Введите количество наблюдений. Пример: 100. Ввод: 100
Введите степень полинома. Пример: 2. Ввод: 4
Хотите ввести дополнительные параметры? 1-Да/0-Нет. Если нет, будут взяты параметры по умолчанию. Ввод:0
Коэффициенты независимых переменных {x**1: 300.46523, x**2: -64.49432, x**3: 7.59172, x**4: -0.39331}
Свободный член регрессии: -464.28425
Уравнение регрессии: $300.46523 \cdot x^{**1} - 64.49432 \cdot x^{**2} + 7.59172 \cdot x^{**3} - 0.39331 \cdot x^{**4} - 464.28425$
Стандартная квадратичная ошибка: 872118.6181941745

рис.3 Пример работы системы с генерацией полиномиальной регрессии - 1

Введите количество наблюдений. Пример: 100. Ввод: 100
Введите степень полинома. Пример: 2. Ввод: 2
Хотите ввести дополнительные параметры? 1-Да/0-Нет. Если нет, будут взяты параметры по умолчанию. Ввод:1
Введите вид регуляризации
0 - без регуляризации, 1 - L1-регуляризация, 2 - L2-регуляризация
Пример: 2. Ввод:0
Хотите ли выполнить построение графика? 1-Да/0-Нет. Ввод:1
Коэффициенты независимых переменных {x**1: 13.71142, x**2: -6.32603}
Свободный член регрессии: -2.90366
Уравнение регрессии: $13.71142 \cdot x - 6.32603 \cdot x^2 - 2.90366$
Стандартная квадратичная ошибка: 1603.8787141627442

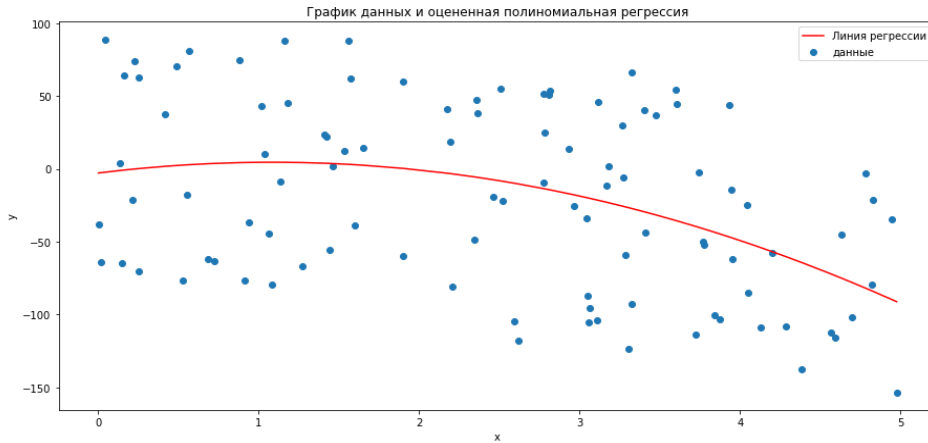


рис.4 Пример работы системы с генерацией полиномиальной регрессии - 2

3.3.3. Экспоненциальная регрессия

Введите количество регрессоров. Пример: 2. Ввод:3
Введите количество наблюдений. Пример: 100. Ввод: 100
Хотите ввести дополнительные параметры? 1-Да/0-Нет. Если нет, будут взяты параметры по умолчанию. Ввод:0
Коэффициенты независимых переменных {x0: 0.03224, x1: -0.52058, x2: 0.00703}
Свободный член регрессии: 0.80581
Уравнение регрессии: $2.23850895693047 \cdot \exp(0.03224 \cdot x_0 - 0.52058 \cdot x_1 + 0.00703 \cdot x_2)$
Стандартная квадратичная ошибка: 0.019618040160026835

рис.5 Пример работы системы с генерацией экспоненциальной регрессии - 1

Введите количество регрессоров. Пример: 2. Ввод:1
Введите количество наблюдений. Пример: 100. Ввод: 100
Хотите ввести дополнительные параметры? 1-Да/0-Нет. Если нет, будут взяты параметры по умолчанию. Ввод:1
Введите вид регуляризации
0 - без регуляризации, 1 - L1-регуляризация, 2 - L2-регуляризация
Пример: 2. Ввод:2
Хотите ли выполнить построение графика(доступно только для одного регрессора)? 1-Да/0-Нет. Ввод:1
Коэффициенты независимых переменных {x0: -0.6885}
Свободный член регрессии: 0.92503
Уравнение регрессии: $2.52194391754081 \cdot \exp(-0.6885 \cdot x_0)$
Стандартная квадратичная ошибка: 0.06970211715027219

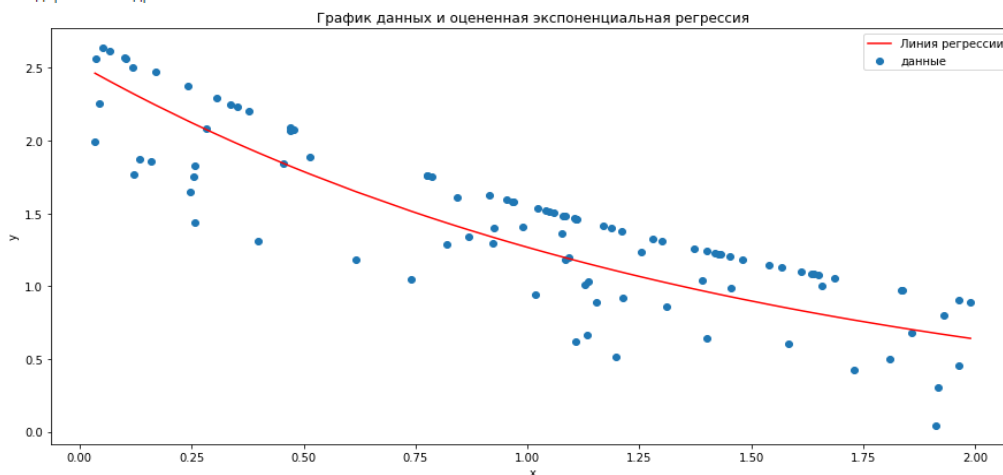


рис.6 Пример работы системы с генерацией экспоненциальной регрессии - 2

4. Архитектура решения

В разделе описываются создаваемые для решения задачи методы (функции), разделенные по 4-м принципиальным блокам.

4.1. Функции считывания информации

```
# def input_lin_reg():
```

```
'''
```

Функция-метод для ввода параметров генерации данных для линейной регрессии

Выходные данные: список введенных параметров, list

```
'''
```

```
# def input_pol_reg():
```

```
'''
```

Функция-метод для ввода параметров генерации данных для полиномиальной регрессии

Выходные данные: список введенных параметров, list

```
'''
```

```
# def input_exp_reg():
```

```
'''
```

Функция-метод для ввода параметров генерации данных для экспоненциальной регрессии

Выходные данные: список введенных параметров, list

```
'''
```

4.2. Функции обработки информации

```
# def predict(X,a):
```

'''

Функция-метод для расчета значений Y по найденным коэффициентам для линейной и полиномиальной регрессий

Входные данные:

X - массив предикатов, array

a - оцененные коэффициенты, array

Выходные данные: предсказанные значения Y , array

'''

def predict_exp(X,a):

'''

Функция-метод для расчета значений Y по найденным коэффициентам для экспоненциальной регрессии

Входные данные:

X - массив предикатов, array

a - оцененные коэффициенты, array

Выходные данные: предсказанные значения Y , array

'''

def error(X,Y,a):

'''

Функция-метод для расчета стандартной квадратичной ошибки регрессии

Входные данные:

X - массив предикатов, array

Y - массив реальных значений зависимой переменной, array

a - оцененные коэффициенты, array

Выходные данные: стандартная квадратичная ошибка, float

'''

def coef_lin_reg(X, Y, reg=0):

'''

Функция-метод для оценки коэффициентов регрессии

Входные данные:

X - массив предикатов, array

Y - массив реальных значений зависимой переменной, array

reg - вид регуляризации, int

Выходные данные: оцененные коэффициенты регрессии, array

'''

def generate_lin_reg(x,row):

'''

Функция-метод для генерации значений X и Y линейной модели

Входные данные:

x - количество регрессоров, int

row - количество наблюдений, int

Выходные данные: сгенерированный массив X и Y, array

'''

def generate_pol_reg(x,row):

'''

Функция-метод для генерации значений X и Y полиномиальной модели

Входные данные:

row - количество наблюдений, int

degree - степень полинома, int

Выходные данные: сгенерированный массив X и Y, array

'''

def generate_exp_reg(x,row):

'''

Функция-метод для генерации значений X и Y экспоненциальной модели

Входные данные:

row - количество наблюдений, int

degree - степень полинома, int

Выходные данные: сгенерированный массив X и Y, array

'''

4.3. Функции вывода информации

def output_lin_reg(X,X_,Y,reg=0,graph=0):

'''

Функция-метод для оценки коэффициентов, вывода итогов регрессии и построения графика, если был указан соответствующий параметр

Входные данные:

X - массив предикатов, array

X_ - массив предикатов с добавленной 1 для свободного члена, array

Y - массив реальных значений зависимой переменной, array

reg - вид регуляризации, int

graph - построение графика, int

Выходные данные: уравнение регрессии и сведения о коэффициентах, график, если был указан соответствующий параметр

'''

def output_pol_reg(X,X_,Y,reg=0,graph=0):

'''

Функция-метод для оценки коэффициентов, вывода итогов регрессии и построения графика, если был указан соответствующий параметр

Входные данные:

X - массив предикатов, array

X_ - массив предикатов с добавленной 1 для свободного члена, array

Y - массив реальных значений зависимой переменной, array

reg - вид регуляризации, int

graph - построение графика, int

Выходные данные: уравнение регрессии и сведения о коэффициентах, график, если был указан соответствующий параметр

'''

def output_exp_reg(X,X_,Y,reg=0,graph=0):

'''

Функция-метод для оценки коэффициентов, вывода итогов регрессии и построения графика, если был указан соответствующий параметр

Входные данные:

X - массив предикатов, array

X_ - массив предикатов с добавленной 1 для свободного члена, array

Y - массив реальных значений зависимой переменной, array

reg - вид регуляризации, int

graph - построение графика, int

Выходные данные: уравнение регрессии и сведения о коэффициентах, график, если был указан соответствующий параметр

'''

4.4. Вспомогательные функции

def all_f_lin_reg():

'''

Функция-метод для объединения работы всех функций

'''

def all_f_pol_reg():

'''

Функция-метод для объединения работы всех функций

'''

def all_f_exp_reg():

'''

Функция-метод для объединения работы всех функций

'''

5. Тестирование

В разделе приводится тестирование работы программы.

Оптимальный способ представления результатов тестирования – это вывод графиков всех видов регрессий (линейной, полиномиальной и экспоненциальной) с использованием регуляризации (L1 и L2) или без ее использования:

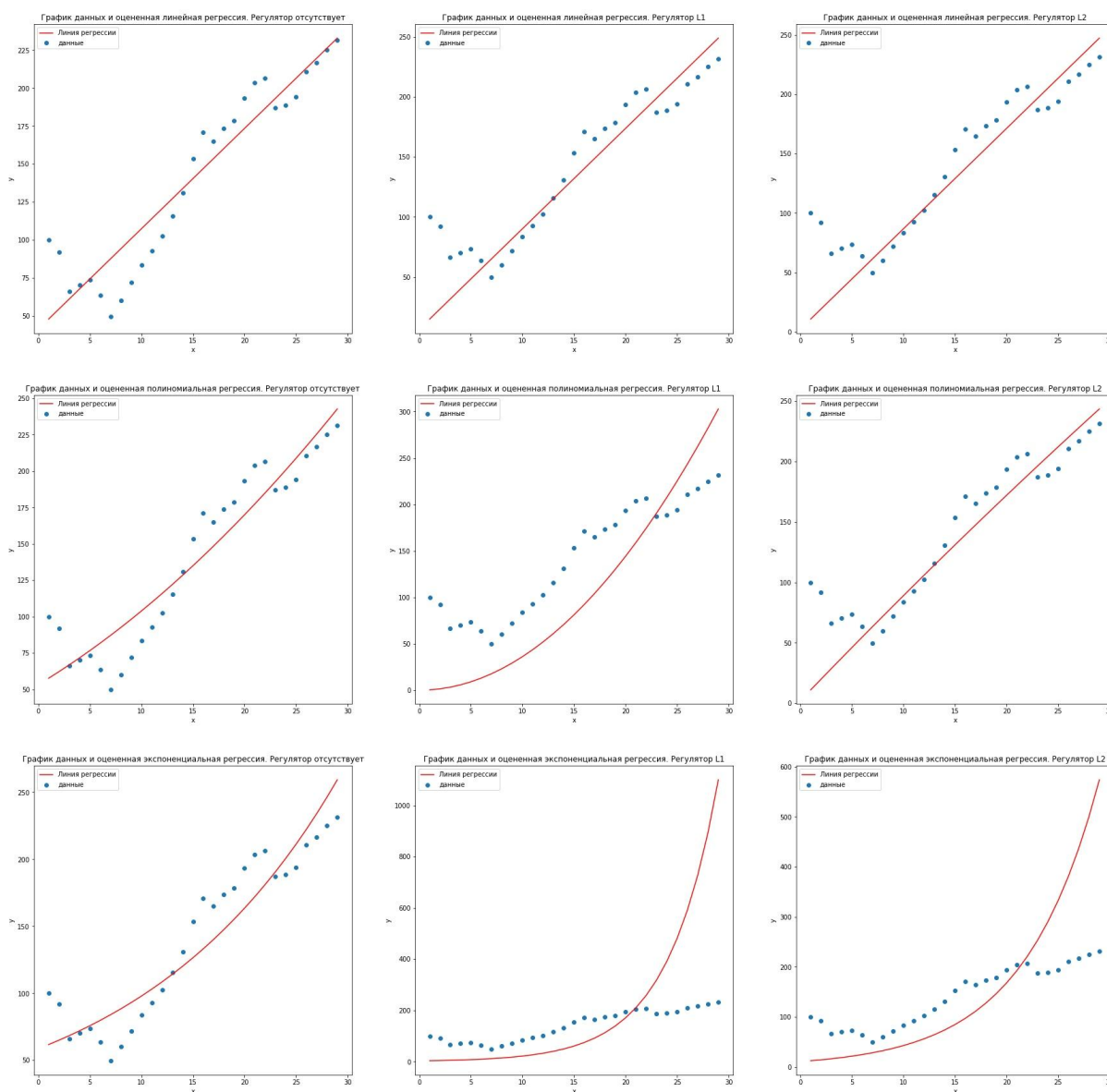


рис.7 Графики данных и оцененные регрессии

А также расчет стандартной ошибки по всем оцененным регрессиям с выводом результатов в виде таблицы:

	Регрессия	Регулятор	Стандартная ошибка
0	Линейная	Без регулятора	210.985285
1	Линейная	L1	353.134074
2	Линейная	L2	393.867997
3	Полиномиальная	Без регулятора	199.234906
4	Полиномиальная	L1	1550.569062
5	Полиномиальная	L2	375.308839
6	Экспоненциальная	Без регулятора	10905.988679
7	Экспоненциальная	L1	10935.412805
8	Экспоненциальная	L2	10921.340901

рис.8 Стандартные ошибки по данным

Как можно заметить, самую наименьшую стандартную ошибку имеет полиномиальная регрессия без регуляторов. Также невысокую стандартную ошибку имеет линейная регрессия без регуляторов. Самые худшие результаты у экспоненциальной регрессии, где практически не имеет значение присутствует ли регулятор.

Таким образом, мы приходим к выводу, что для данных лучше всего подходит полиномиальная регрессия без регуляторов.

6. Заключение

Подводя итоги, можно констатировать следующее: согласно требованиям заказчика в среде программирования Python были реализованы функции, которые находят примерные объемы продаж при известных затратах на рекламу с помощью линейной, полиномиальной и экспоненциальной регрессий.

Произведем сравнение реализованных функций по данным заказчика, чтобы выбрать наилучшее уравнение регрессии, которое дает наименьшую стандартную ошибку, т.е. дает наиболее точный прогноз.

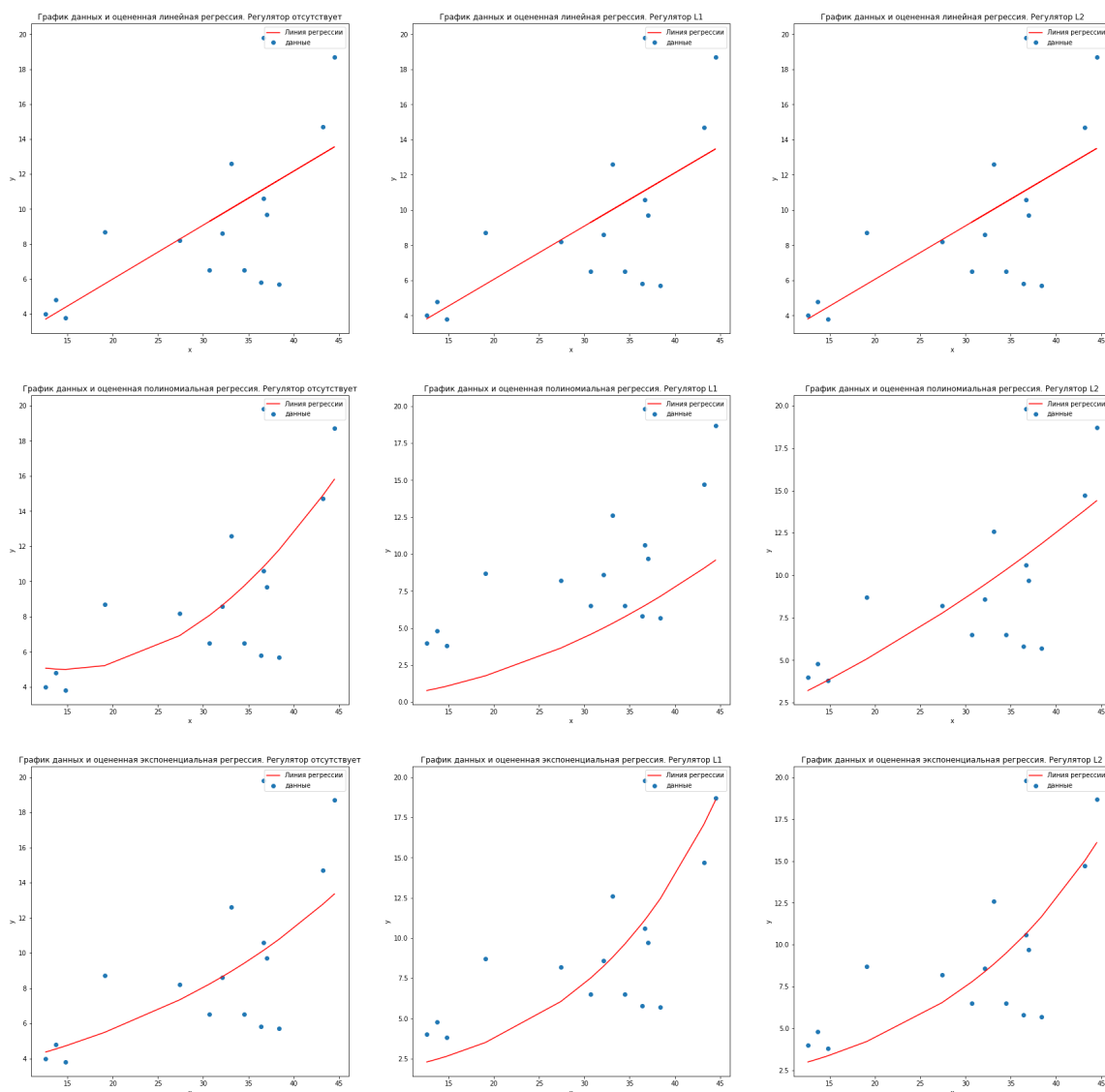


рис.9 Данные заказчика и оцененные регрессии

	Регрессия	Регулятор	Стандартная ошибка
0	Линейная	Без регулятора	6.593705
1	Линейная	L1	6.595355
2	Линейная	L2	6.594862
3	Полиномиальная	Без регулятора	6.037400
4	Полиномиальная	L1	15.357507
5	Полиномиальная	L2	6.462850
6	Экспоненциальная	Без регулятора	36.123360
7	Экспоненциальная	L1	36.006096
8	Экспоненциальная	L2	36.073482

рис.10 Стандартные ошибки по данным заказчика

По рисункам можно сделать вывод о том, что для данных заказчика лучше всего подходит полиномиальная регрессия без регуляторов.

Тогда уравнение регрессии имеет следующий вид:

$y = 0.01236881x^2 - 0.36949017 + 7.75994933$. Из этого уравнения находится значение y (предположительный объем продаж за месяц) по уже известным значениям x (денежные затраты на рекламу) .

Полученные результаты округлим до целой части и занесем их в Таблицу 1. Получается следующее:

Таблица 2. Итоговый план заказчика

	1-й месяц	2-й месяц	3-й месяц
Затраты на рекламу	50 тыс руб.	60 тыс руб.	70 тыс руб.
Объем продаж	20 млн руб.	30 млн руб.	43 млн руб.

Получается, что при известных затратах на рекламу заказчик может ожидать в первый месяц объем продаж примерно равный 20 млн рублей, во второй - 30 млн рублей, в третий - 43 млн рублей.

Уважаемый заказчик, наша команда создала модель, которая позволяет решить поставленную задачу и удовлетворяет всем условиям. Мы рассчитываем на дальнейшее сотрудничество. В случае заключения договора, на долгосрочной основе, мы готовы производить новые расчеты и прогнозы для Вашей компании.