

[Дикая и живописная река Дешут](https://commons.wikimedia.org/wiki/File:Deschutes_Wild_and_Scenic_River_(28260152076).jpg) ( [CC BY 2.0](https://creativecommons.org/licenses/by/2.0/deed.en) )

**Иллюстрированное руководство по нуль-инфляционной модели[[1]](#footnote-1) регрессии Пуассона (модели регрессии Пуассона с избыточным количеством нулей)**

An Illustrated Guide to the Zero Inflated Poisson Regression Model

<https://towardsdatascience.com/an-illustrated-guide-to-the-zero-inflated-poisson-model-b22833343057>

[Sachin Date](https://sachin-date.medium.com/?source=post_page-----b22833343057--------------------------------)

[2 мая 2020 г.](https://towardsdatascience.com/an-illustrated-guide-to-the-zero-inflated-poisson-model-b22833343057?source=post_page-----b22833343057--------------------------------)

[Нуль-инфляционная регрессионная модель Пуассона 3](#_Toc73571046)

[Структура модели ZIP 3](#_Toc73571047)

[Знакомство с φ 6](#_Toc73571048)

[Как оценить φ? 7](#_Toc73571049)

[Обучение модели ZIP с помощью Python 9](#_Toc73571050)

[Интерпретация результатов обучения 14](#_Toc73571051)

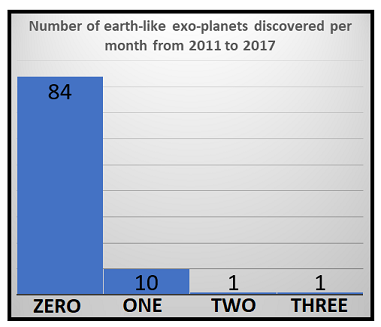
[Интересное упражнение 17](#_Toc73571052)

[Предлагаемые статьи для дальнейшего чтения 17](#_Toc73571053)

В статье узнаем, как построить регрессионную модель для **счетного** **набора данных** (count based dataset)**,** в котором зависимая переменная содержит **большое количество данных с нулевым значением**.

В счетном наборезависимой переменной является событие, например:

* количество транспортных средств, пересекающих перекресток в час;
* количество поступивших больных в приемный покой каждый месяц;
* количество заявлений на выплату автомобильной страховки в год;
* количество дефектов в печатной плате при серийном производстве.



Набор данных, содержащий множество нулевых отсчетов (Изображение [автора](https://sachin-date.medium.com/))

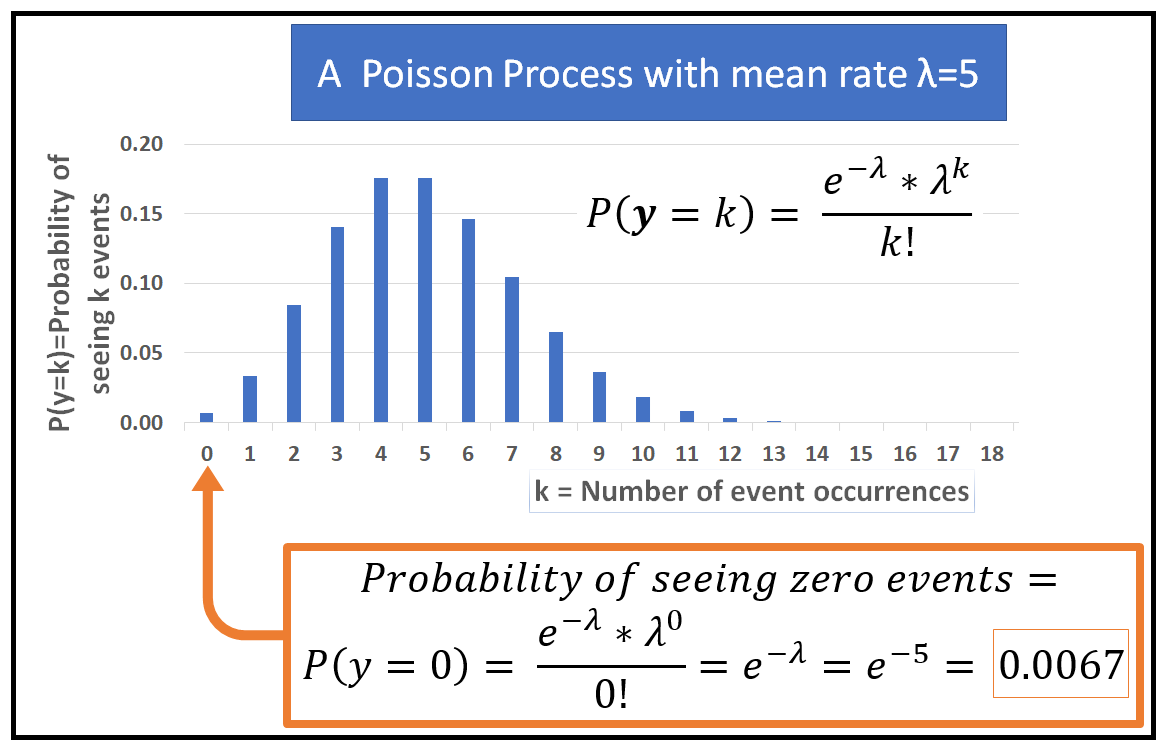
Многие явления в реальном мире часто приводят к нулевым показателям (отсутствиям событий). Например:

* количество отказов машины в месяц;
* количество экзопланет, обнаруживаемых за год;
* количество миллиардеров, проживающих в произвольном городе мира.

С такими данными трудно работать, применяя традиционные дискретные модели, такие как модель [**Пуассона**](https://towardsdatascience.com/an-illustrated-guide-to-the-poisson-regression-model-50cccba15958), модели [**биномиальной**](https://towardsdatascience.com/the-binomial-regression-model-everything-you-need-to-know-5216f1a483d3) или [**отрицательной биномиальной**](https://towardsdatascience.com/negative-binomial-regression-f99031bb25b4) (NB) регрессии.

Это происходит потому, что подобные наборы **содержат значительно большее количество нулевых значений, чем обычно присутствует при использовании традиционных распределений вероятностей**.

Например, если предположить, что явление подчиняется процессу Пуассона с λ=5 (англ. Poisson(5)), то количество ожидаемых нулевых значений не превысит 0,67% случаев.



Процесс Пуассона (λ=5) будет генерировать нули примерно в 0,67% наблюдений (Изображение [автора](https://sachin-date.medium.com/))

Если нулевые результаты встречаются гораздо чаще, то говорят, что набор данных содержит **избыток нулей.**

Если для таких наборов данных используется стандартная модель регрессии Пуассона, биномиальная или NB модель, то она может плохо соответствовать реалии и давать прогнозы низкого качества, независимо от того, насколько качественно вы настраиваете ее параметры.

Что делать аналитику, столкнувшись с данными, содержащими очень большое количество нулей?

# Нуль-инфляционная регрессионная модель Пуассона

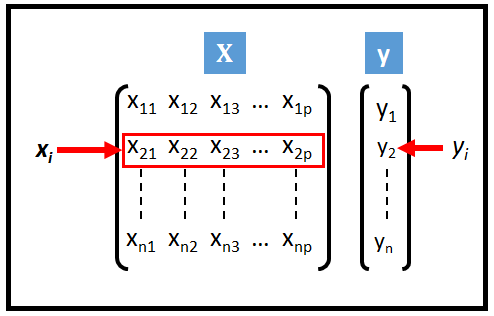
К счастью, есть способ изменить стандартную модель, такую ​​как Пуассон или NB, чтобы учесть наличие большого множества нулей. На самом деле, есть как минимум два способа. Один известен как **барьерная** **модель (hurdle model),** а второй – как **нуль-инфляционная модель (zero-inflated model).**

В статье рассмотрим нуль-инфляционную модель регрессии. В частности, сосредоточимся на **нуль-инфляционной** **модели регрессии Пуассона**, часто называемой **моделью ZIP (Zero Inflated Poisson Regression Model)**.

## Структура модели ZIP

Сначала рассмотрим структуру обычной модели Пуассона, прежде чем перейдем к ее модификации для обработки большого количества нулевых событий.

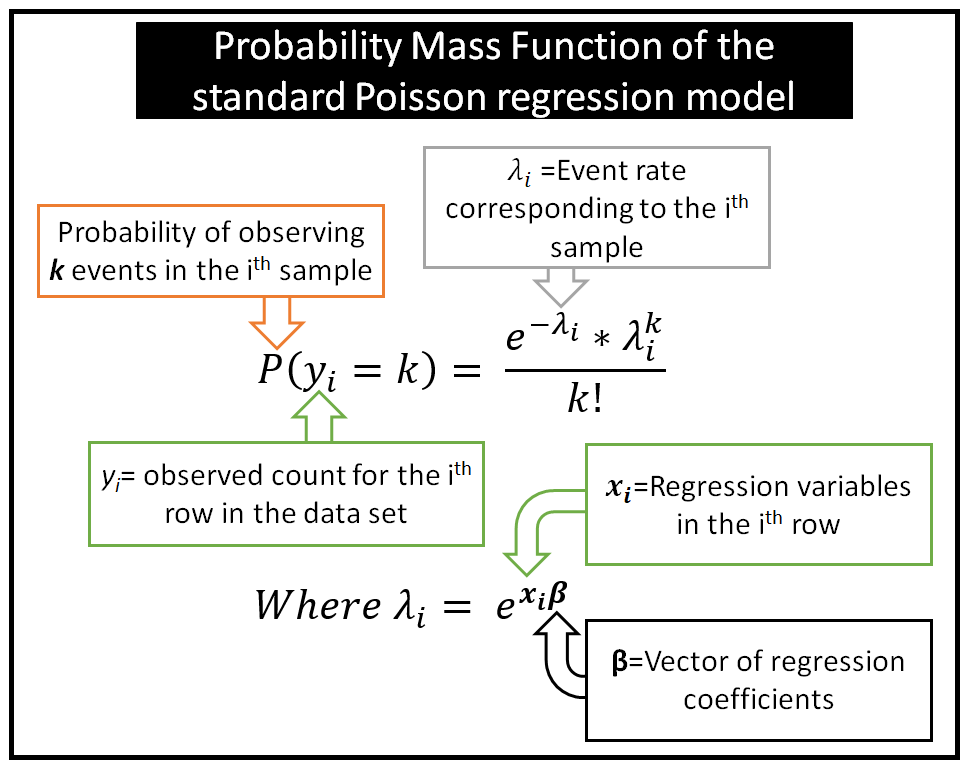
Представим себе набор данных, содержащий *n* событий и *p* регрессионных переменных (признаков, предикторов) на событие. Следовательно, переменные регрессии ***X*** могут быть представлены матрицей размера *(n × p),* где каждая строка ***xi*** в матрице ***X*** представляет собой вектор размера *(1 × p),* соответствующий значению зависимой (целевой) переменной *yi*.



Набор данных (*X, y* ) в матричной записи (Изображение [автора](https://sachin-date.medium.com/))

Если предположить, что ***y*** является случайной величиной, соответствующей распределению Пуассона, то для такого набора данных можно построить модель регрессии Пуассона. Модель Пуассона состоит из двух частей:

1. Функция вероятности (Probability Mass Function, PMF). Обозначается как *P (yi = k),* используется для вычисления вероятности обнаружения *k* событий за фиксированный промежуток времени с заданным математическим ожиданием случайной величины λ (средним количеством событий за фиксированный промежуток времени).
2. Функция связи, выражающая λ через регрессионные переменные ***X***.



Функция вероятности стандартной модели регрессии Пуассона (Изображение [автора](https://sachin-date.medium.com/))

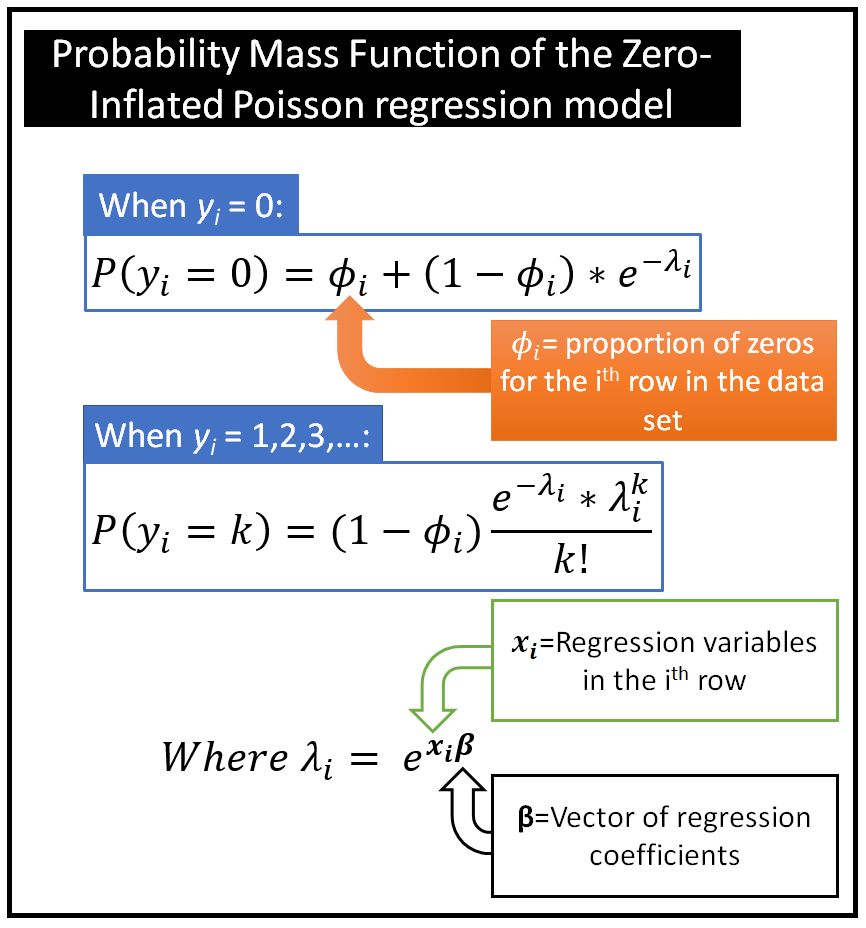
Обычно мы предполагаем, что существует некоторый базовый процесс, производящий наблюдаемые значения в соответствии с *PMF* Пуассона *P (yi = k)*.

Идея, лежащая в основе ZIP-модели, заключается в том, что ***существует второй базовый процесс, который определяет, является ли наблюдаемое значение нулевым или ненулевым***. Если значение определено как ненулевое, то работает обычный процесс Пуассона для определения фактического ненулевого значения на основе PMF Пуассона.

Таким образом, регрессионная модель ZIP состоит из трех частей:

1. PMF *P (yi = 0),* которая используется для вычисления вероятности наблюдения с нулевым значением;
2. вторая PMF *P (yi = k),* которая используется для вычисления вероятности наблюдения *k* событий при условии, что k> 0;
3. функция связи, выражающая λ через регрессионные переменные

***X.***



Функция вероятности модели ZIP (Изображение [автора](https://sachin-date.medium.com/) )

Как и раньше, *yi* ­– это случайная величина, которая обозначает наблюдаемое количество, соответствующее строке переменных регрессии ***xi*** *= [xi1, xi2, xi3, …, xip].*

*φi* – доля избыточных нулей, соответствующая i-й строке *(yi****, xi****)* в наборе данных.

## Знакомство с *φ*

Простой способ понять *φi.*

Представим, что берутся 1000 наблюдений *yi*, каждое с ***одинаковой*** комбинацией значений переменных регрессии ***xi*** *= [xi1, xi2, xi3, …, xip].* Поскольку *yi –* это случайная величина, которая соответствует распределению Пуассона, увидим разные значения *yi* в каждом из 1000 наблюдений.

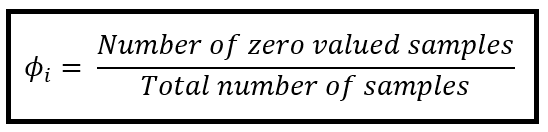
Предположим, что из 1000 наблюдаемых значений *yi* получили 874 нулевых значения. Определяем, сколько из этих 874 нулевых значений сможет объяснить обычное распределение Пуассона, которое приняли в данном случае (λ=5) для *yi.* Получаем не более 7 нулевых значений. Таким образом, оставшиеся 867 нулевых значений являются избыточными. Итак, для *i-й* строки в нашем наборе данных *φi* = *867/1000* = 0,867.

Когда в наборе данных нет лишних нулей в зависимой переменной, значение ***φ*** оказывается равным нулю, и PMF модели ZIP уменьшается до PMF стандартной модели Пуассона (легко проверяется, установив ***φ*** равным 0 в PMF модели ZIP).

## Как оценить *φ?*

Как оценить значение *φi*?

Простой и грубый способ оценки *φi* ­– установить для каждого *φi* следующее соотношение:



Простой, но неточный способ оценки *φi* (Изображение [автора](https://sachin-date.medium.com/) )

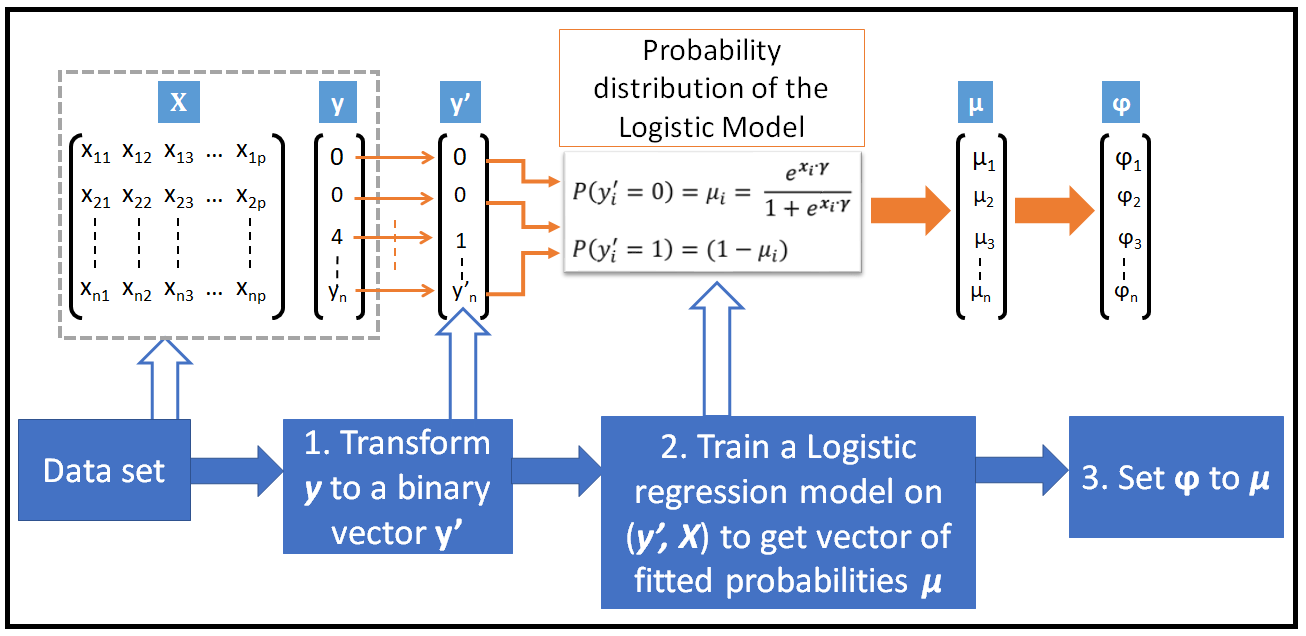
Количество наблюдений с нулевыми значениями / Общее количество наблюдений

Возможно, более эффективным способом расчета *φi* является оценка ее как функции регрессии переменных ***X***. Обычно это делается путем преобразования переменной ***y*** в бинарную случайную величину ***y'***, которая принимает значение 0, если базовый ***y*** равен 0, и 1 во всех остальных случаях. Затем **настраиваем модель логистической регрессии** на преобразованной ***y'***. Обучаем модель логистической регрессии на наборе данных [***X, y'***], и она дает вектор ***настроенных*** вероятностей ***µ\_fitted*** *= [µ1, µ2, µ3,…, µn]* (поскольку именно это делает модель логистической регрессии) *.*

Как только мы получили *вектор* ***µ\_fitted, мы просто устанавливаем его равным вектору φ.***

*Таким образом, [φ1 = µ1, φ2 = µ2, φ3 = µ3, …, φn = µn]*.

Данный процесс оценки ***ϕ*** проиллюстрирован на рисунке.

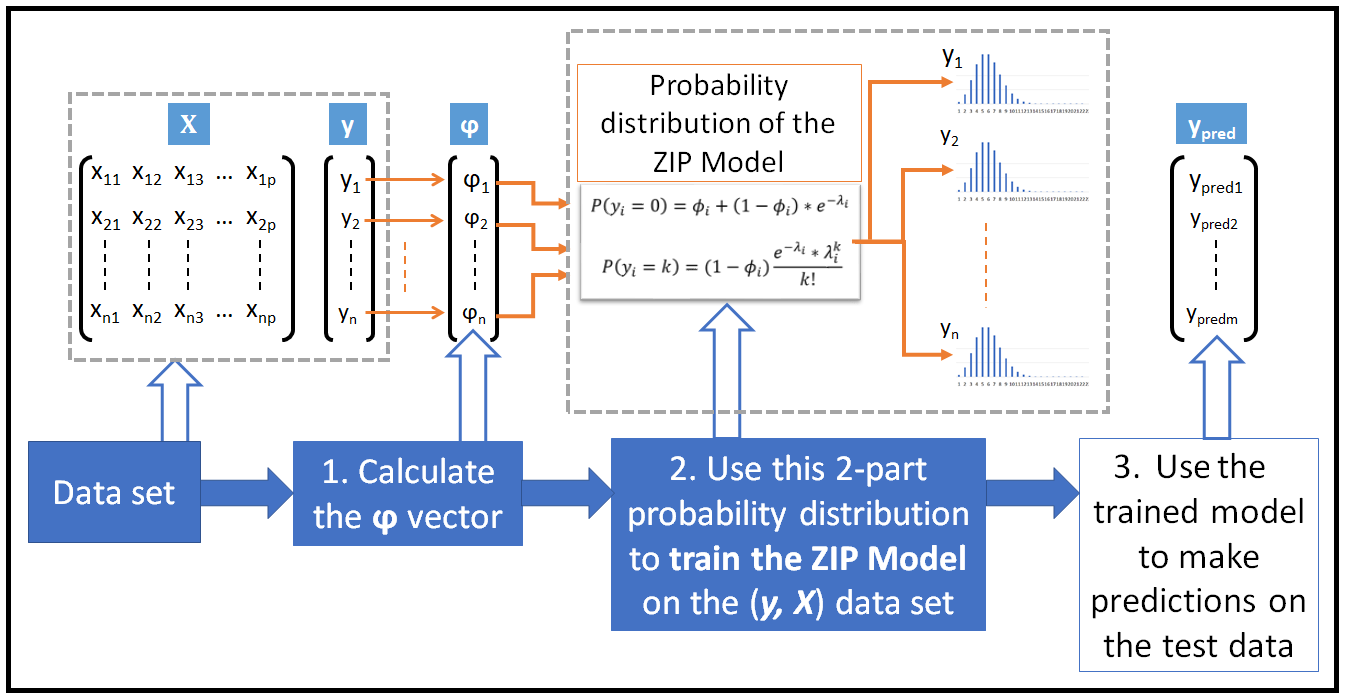


Обучающая последовательность для оценки параметра лишних нулей ϕ (изображение [автора](https://sachin-date.medium.com/))

После того, как оценили вектор ***φ***, встраиваем его в функции вероятности ZIP- модели и применяем то, что известно как **метод максимального правдоподобия** ( Maximum Likelihood Estimation, MLE ) для обучения ZIP-модели на наборе данных с избытком нулевых событий.

Пожалуйста, прочтите мою статью о [**Пуассоновской модели регрессии**](https://towardsdatascience.com/an-illustrated-guide-to-the-poisson-regression-model-50cccba15958) для понимания того, как работает MLE.

На следующем рисунке показана последовательность обучения модели **ZIP.**



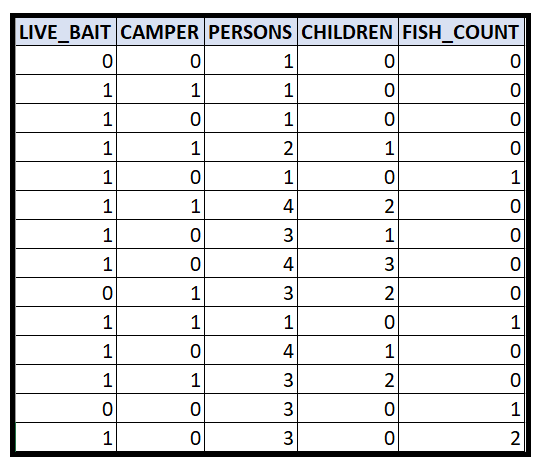
Последовательность обучения ZIP-модели (Изображение [автора](https://sachin-date.medium.com/))

Существует множество статистических пакетов, которые автоматизируют процедуру оценки ***φ*** и использования оцененного ***φ*** для обучения модели ZIP с применением метода MLE.

В оставшейся части статьи будем использовать библиотеку Python *statsmodels* для создания и обучения модели ZIP.

# Обучение модели ZIP с помощью Python

Будем использовать набор данных о походах, совершенных 250 группами людей.



Набор данных о походах (изображение [автора](https://sachin-date.medium.com/))

Набор данных доступен [здесь](https://gist.github.com/sachinsdate/09cfd42b7701c48ec68b04c786786434). Вот несколько важных особенностей этого набора:

* группа могла рыбачить или не рыбачить во время поездки;
* если группа рыбачила, она поймала от нуля рыб и более;
* мы хотим оценить не только, сколько рыбы было поймано (если рыбалка была у группы), но и вероятность того, что группа вообще что-либо поймала.

Задействуем два различных процесса генерации данных.

1. Процесс, который определяет, была ли успешной рыбалка в группе. ZIP-модель будет использовать модель логистической регрессии для моделирования этого бинарного процесса.
2. Второй процесс, который определяет, сколько рыбы было поймано группой, предполагая, что группа поймала хотя бы одну рыбу. ZIP-модель будет использовать обычную модель Пуассона для моделирования этого процесса.

Набор данных о походах содержит следующие переменные:

**FISH\_COUNT ­–** количество пойманной рыбы. Это зависимая переменная ***y***;

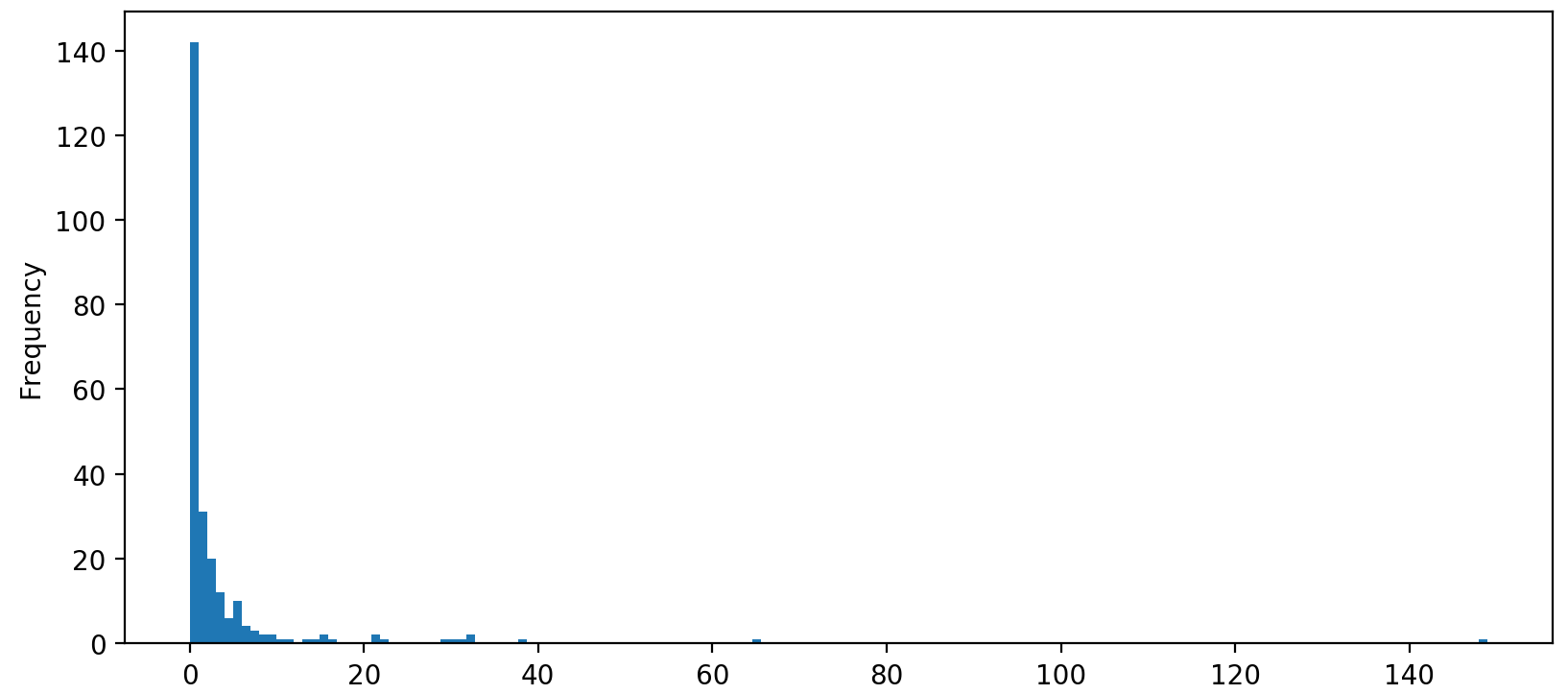
**LIVE\_BAIT –** двоичная переменная, указывающая, использовалась ли наживка;

**CAMPER –** использовала ли группа автофургон;

**PERSONS –** общее количество человек в группе;

**CHILDREN –** количество детей в группе.

Частотное распределение зависимой переменной FISH\_COUNT:



Частотное распределение FISH\_COUNT (изображение [автора](https://sachin-date.medium.com/))

Видим, что в наборе данных могут быть избыточные нули. Обучим модель ZIP на этом наборе, чтобы проверить нашу гипотезу и, возможно, добиться лучшего соответствия, чем обычная модель Пуассона.

**Цель регрессии**

Наша цель для этого набора данных – предсказать количество пойманной группой рыбы (FISH\_COUNT) на основе значений переменных LIVE\_BAIT, CAMPER, PERSONS и CHILDREN.

**Стратегия регрессии**

Наша стратегия:

1. FISH\_COUNT будет зависимой переменной ***у***, а [LIVE\_BAIT, CAMPER, PERSONS, CHILDREN] – независимыми переменными (признаками) ***X***.
2. Будем использовать библиотеку Python statsmodelsдля обучения модели регрессии ZIP на наборе данных (***y, X***)
3. Сделаем некоторые прогнозы, применяя модель ZIP к тестовому набору данных, который модель не видела во время обучения.

Начнем с импорта необходимых пакетов.

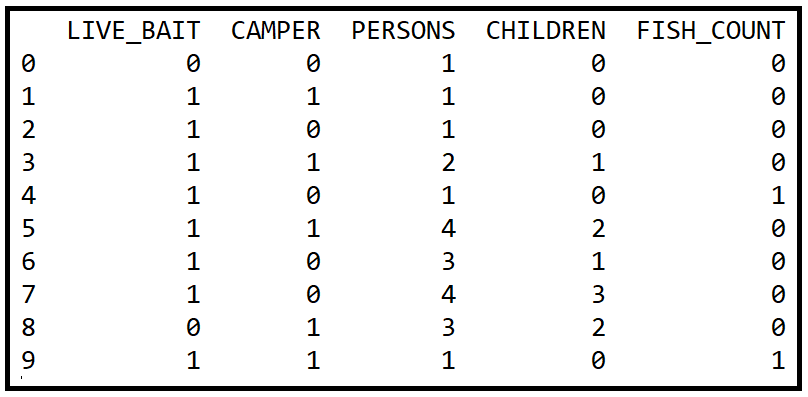
import pandas as pd  
from patsy import dmatrices  
import numpy as np  
import statsmodels.api as sm  
import matplotlib.pyplot as plt

Загрузим набор данных. Вот [ссылка на набор](https://gist.github.com/sachinsdate/09cfd42b7701c48ec68b04c786786434).

df = pd.read\_csv('fish.csv', header=0)

Посмотрим несколько верхних строк набора.

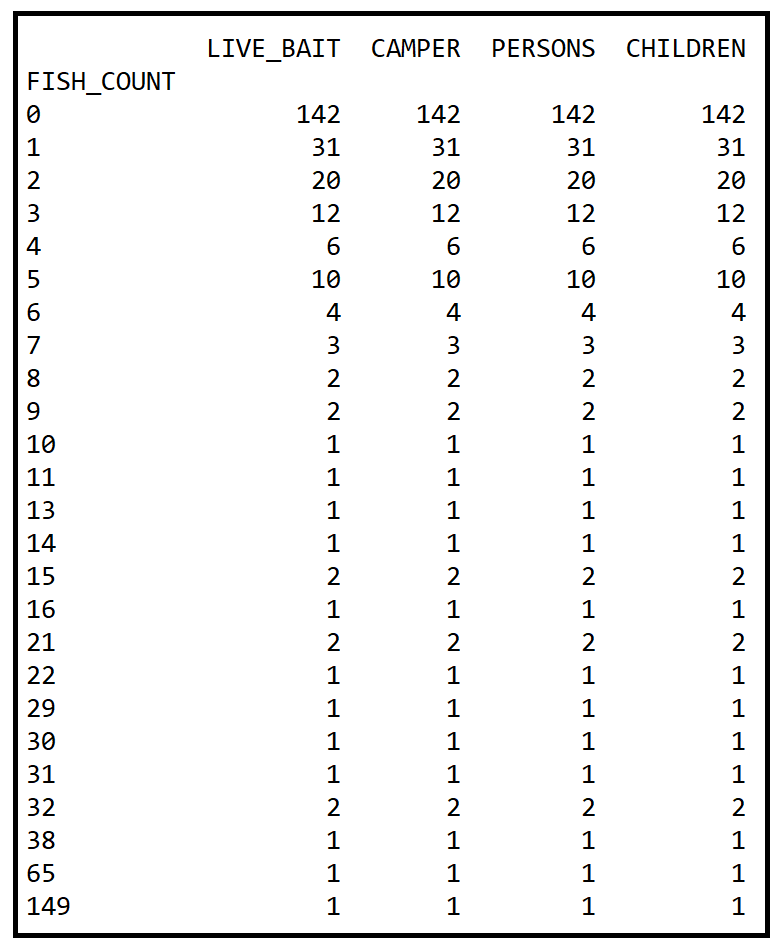
print(df.head(10))



10 первых строк набора данных (изображение [автора](https://sachin-date.medium.com/) )

Выведем частотное распределение значений FISH\_COUNT.

df.groupby('FISH\_COUNT').count()



Частотное распределение количества рыб (изображение [автора](https://sachin-date.medium.com/))

Создаем наборы данных для обучения и тестирования. Обратите внимание, что на данный момент мы не выполняем стратифицированное случайное разбиение.

mask = np.random.rand(len(df)) < 0.8  
df\_train = df[mask]  
df\_test = df[~mask]  
print('Training data set length='+str(len(df\_train)))  
print('Testing data set length='+str(len(df\_test)))

>> Training data set length=196  
>> Testing data set length=54

Установим выражение регрессии в нотации [**Patsy**](https://patsy.readthedocs.io/en/latest/quickstart.html). Мы говорим Patsy, что FISH\_COUNT – наша зависимая переменная ***y,*** и она зависит от переменных LIVE\_BAIT, CAMPER, PERSONS и CHILDREN:

expr = 'FISH\_COUNT ~ LIVE\_BAIT + CAMPER + CHILDREN + PERSONS'

Разделим наборы ***X*** и **y** на обучающие и тестовые данные.

y\_train, X\_train = dmatrices(expr, df\_train, return\_type='dataframe')

y\_test, X\_test = dmatrices(expr, df\_test, return\_type='dataframe')

Используя класс statsmodels [**ZeroInflatedPoisson**](https://www.statsmodels.org/stable/generated/statsmodels.discrete.count_model.ZeroInflatedPoisson.html), создадим и обучим модель регрессии ZIP на обучающем наборе данных.

Но прежде объясню, как использовать два параметра, которые принимает конструктор класса:

* **inflation:** класс ZeroInflatedPoisson будет использовать модель LogisticRegression для оценки параметра **φ**. Следовательно, устанавливаем **inflation** равным «logit». Можно поэкспериментировать с установкой других функций, например, «probit».
* **exog\_infl:** мы хотим, чтобы модель ZIP оценила **φ** как функцию того же набора регрессионных переменных, что и родительская модель, а именно: LIVE\_BAIT, CAMPER, PERSONS и CHILDREN. Следовательно, мы устанавливаем параметр exog\_infl равным X\_train. Можно использовать только подмножество X\_train, установив exog\_infl на совершенно другой набор регрессионных переменных.

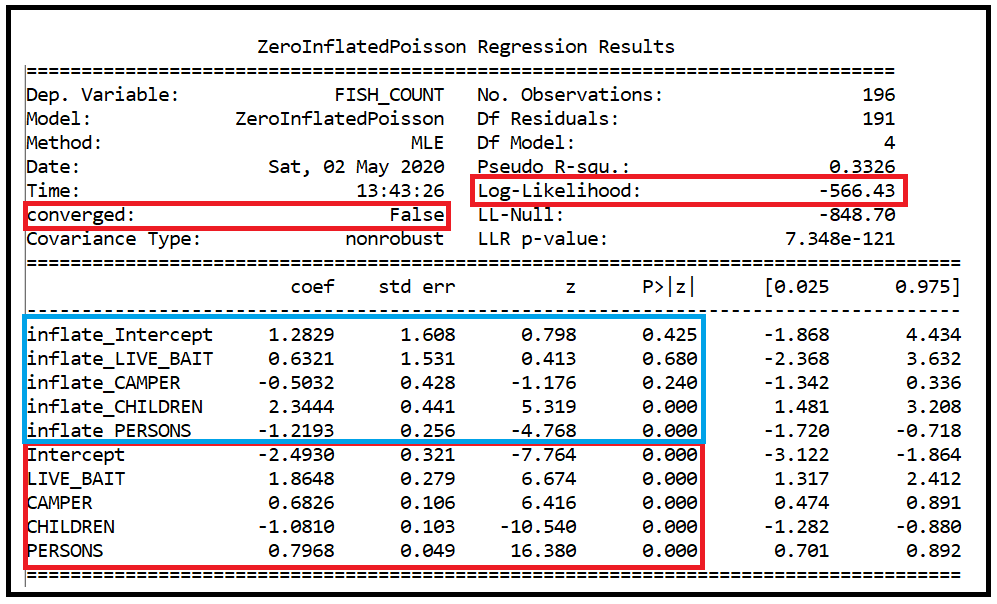
В приведенной ниже строке строится и обучается модель ZIP на обучающем наборе.

zip\_training\_results = sm.ZeroInflatedPoisson(*endog=*y\_train, *exog=*X\_train, *exog\_infl=*X\_train, *inflation=*'logit').fit()

Выведем результаты обучения

print(zip\_training\_results.summary())

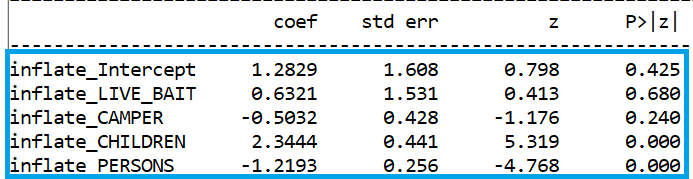
Вот результаты обучения (выделены важные элементы):



Краткое описание модели ZIP (изображение [автора](https://sachin-date.medium.com/))

# ****Интерпретация результатов обучения****

Синее поле содержит информацию о переменных, которые использовала вложенная модель логистической регрессии для оценки вероятности **φ** того, была ли поймана рыба группой**.**



(Изображение [автора](https://sachin-date.medium.com/))

Обратите внимание, что модель логистической регрессии не сочла переменные Intercept, LIVE\_BAIT и CAMPER полезными для оценки **ϕ**. Их коэффициенты регрессии не являются статистически значимыми при уровне достоверности 95%, на что указывают соответствующие значения p :  
inflate\_Intercept = 0,425 ,inflate\_LIVE\_BAIT = 0,680,inflate\_CAMPER = 0,240,которые все больше 0,05 (т.е. порог ошибки равен 5%).

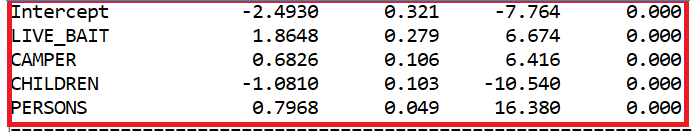
**Наблюдение 1**

Двумя переменными, которые модель логистической регрессии определила как полезные для оценки вероятности того, была ли поймана какая-либо рыба, оказались CHILDREN и PERSONS.

**Наблюдение 2**

Коэффициент регрессии PERSONS отрицательный (inflate\_PERSONS = **-1,2193).** Это означает, что по мере увеличения количества людей в группе вероятность того, что эта группа не поймает рыбу, уменьшается. Это соответствует интуиции.

В красном поле содержится информация о переменных, которые родительская модель Пуассона использовала для оценки FISH\_COUNT при условии, что FISH\_COUNT> 0.



(Изображение [автора](https://sachin-date.medium.com/))

**Наблюдение 3**

Видим, что коэффициенты для всех регрессионных переменных статистически значимы на уровне достоверности 99%, о чем свидетельствует их значение p, которое меньше 0,01. Фактически, значение p меньше 0,001 для всех переменных, потому оно отображается как 0,000.

**Наблюдение 4**

Коэффициент для CHILDREN отрицательный (-1,0810). Это означает, что по мере увеличения количества детей в группе количество рыбы, пойманной группой, уменьшается!

**Наблюдение 5**

Максимальное логарифмическое правдоподобие этой модели составляет 566,43. Это значение полезно для сравнения с другими моделями (см. Интересное упражнение в конце статьи).

**Наблюдение 6**

Наконец, обратите внимание, что алгоритм обучения модели ZIP не смог сойтись на наборе обучающих данных, о чем свидетельствует следующее:



(Изображение [автора](https://sachin-date.medium.com/))

Если бы он сошелся, возможно, получилось бы лучше.

**Прогноз**

Получим прогноз модели ZIP на тестовом наборе и вычислим среднеквадратичную ошибку по фактическим значениям.

zip\_predictions = zip\_training\_results.predict(X\_test,exog\_infl=X\_test)predicted\_counts=np.round(zip\_predictions)

actual\_counts = y\_test[dep\_var][[2]](#footnote-2)  
  
print('ZIP RMSE='+str(np.sqrt(np.sum(np.power(np.subtract(predicted\_counts,actual\_counts),2)))))

>> ZIP RMSE=55.65069631190611

Построим прогнозируемое и фактическое количество рыбы.

fig = plt.figure()

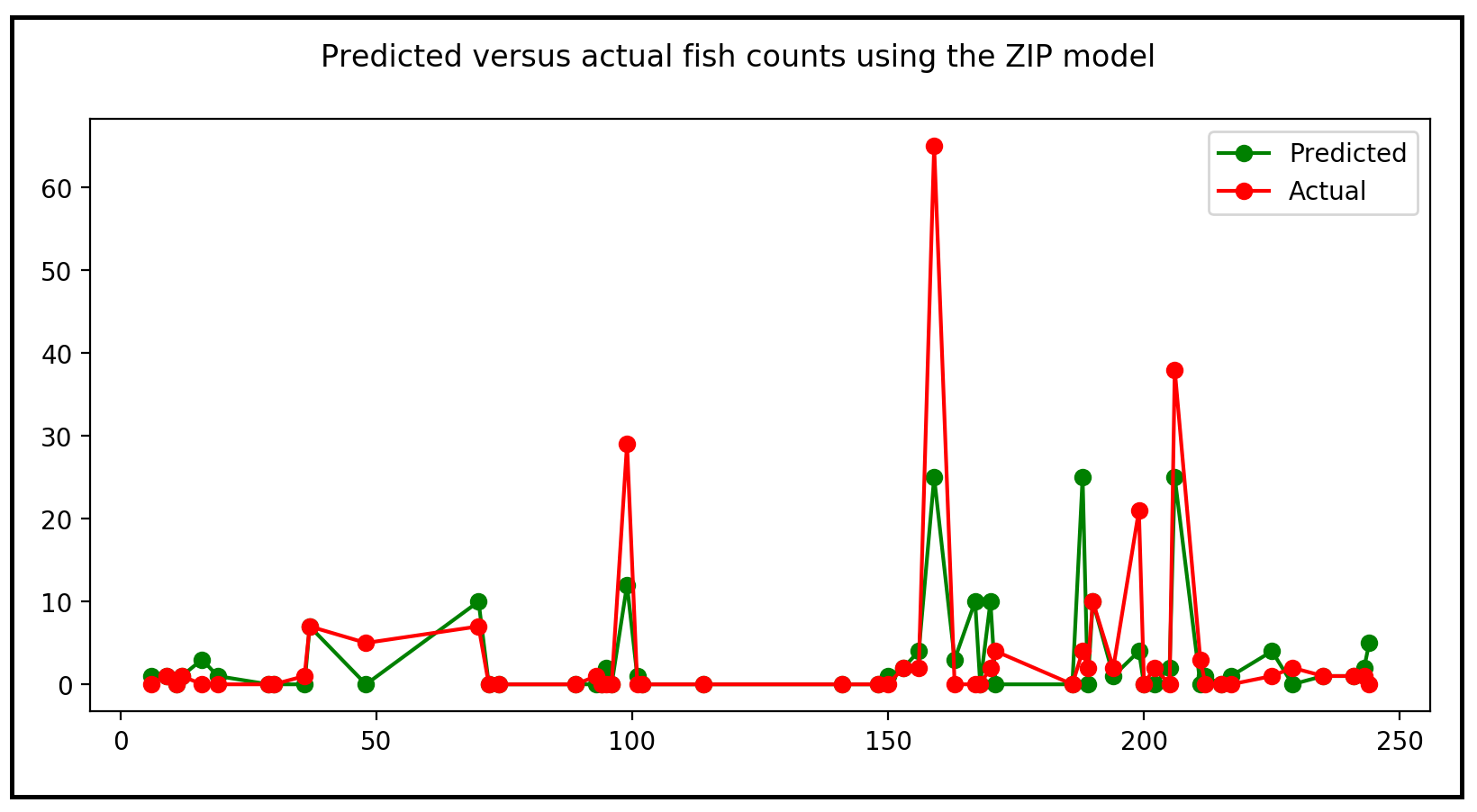
fig.suptitle('Predicted versus actual counts using the ZIP model')

predicted, = plt.plot(X\_test.index, predicted\_counts, 'go-', label='Predicted')

actual, = plt.plot(X\_test.index, actual\_counts, 'ro-', label='Actual')

plt.legend(handles=[predicted, actual])

plt.show()



Прогнозируемая и фактическая пойманная рыба (Изображение [автора](https://sachin-date.medium.com/))

На этом завершаем рассмотрение нуль-инфляционной модели регрессии Пуассона.

# Интересное упражнение

1. Прочтите следующую статью: [Иллюстрированное руководство по модели регрессии Пуассона](https://towardsdatascience.com/an-illustrated-guide-to-the-poisson-regression-model-50cccba15958)
2. Используя Python и statsmodels, обучите стандартную модель Пуассона на наборе данных о походах и сравните производительность модели с производительностью модели ZIP. Вы можете быть удивлены тем, что увидите.
3. Сравните степень согласия двух моделей на наборе обучающих данных, используя значение максимального логарифмического правдоподобия, предоставляемое statsmodels в сводке по обучению обеих моделей. Чем больше это значение, тем лучше степень соответствия.
4. Сравните оценки RMSE (среднее квадратичное отклонение) двух моделей на тестовом наборе.

Удачного моделирования!

# Предлагаемые статьи для дальнейшего чтения

* [The Poisson Process: Everything you need to know](https://towardsdatascience.com/the-poisson-process-everything-you-need-to-know-322aa0ab9e9a)
* [Пуассоновский процесс: все, что вам нужно знать](https://towardsdatascience.com/the-poisson-process-everything-you-need-to-know-322aa0ab9e9a)
* [The Intuition for the Poisson Distribution Formula](https://towardsdatascience.com/the-intuition-for-the-poisson-distribution-formula-34c5e52f4e04)
* [Интуиция в формуле распределения Пуассона](https://towardsdatascience.com/the-intuition-for-the-poisson-distribution-formula-34c5e52f4e04)
* [An illustrated Guide to the Poisson regression model](https://towardsdatascience.com/an-illustrated-guide-to-the-poisson-regression-model-50cccba15958)
* [Иллюстрированное руководство по модели регрессии Пуассона](https://towardsdatascience.com/an-illustrated-guide-to-the-poisson-regression-model-50cccba15958)
* [Generalized Poisson models for Real World Data Sets](https://towardsdatascience.com/generalized-poisson-regression-for-real-world-datasets-d1ff32607d79)
* [Обобщенные модели Пуассона для наборов данных реального мира](https://towardsdatascience.com/generalized-poisson-regression-for-real-world-datasets-d1ff32607d79)
* [The Negative Binomial regression model: A Step by Step Guide](https://towardsdatascience.com/negative-binomial-regression-f99031bb25b4)
* [Модель отрицательной биномиальной регрессии: пошаговое руководство](https://towardsdatascience.com/negative-binomial-regression-f99031bb25b4)
* [A. Colin Cameron and Pravin K. Trivedi, Regression Analysis of Count Data](http://faculty.econ.ucdavis.edu/faculty/cameron/racd2/)
* А. Колин Кэмерон, Правин К. Триведи. Регрессионный анализ счетных данных
* Lambert, D. (1992), “[Zero-Inflated Poisson Regression with an Application to Defects in Manufacturing](https://www.researchgate.net/publication/254333007_Zero-Inflated_Poisson_Regression_With_An_Application_to_Defects_in_Manufacturing),” Technometrics, 34, 1–14.
* Ламберт, Д. (1992), « [Нулевая регрессия Пуассона с приложением к дефектам в производстве](https://www.researchgate.net/publication/254333007_Zero-Inflated_Poisson_Regression_With_An_Application_to_Defects_in_Manufacturing) », Technometrics, 34, 1–14.

Спасибо за прочтение! Я пишу на темы науки о данных, уделяя особое внимание регрессии и анализу временных рядов.

Если вам понравилась эта статья, подпишитесь на меня на [**Sachin Date,**](https://sachin-date.medium.com/) чтобы получать советы, инструкции и советы по программированию по темам, посвященным регрессии и анализу временных рядов.

1. Прим. перев. Не нашел типового названия подобных моделей. [↑](#footnote-ref-1)
2. Прим. перев. Не понял, откуда взялась переменная dep\_var. Заменил на 'FISH\_COUNT' [↑](#footnote-ref-2)