

Отчёт по задаче 4 (Регрессионный анализ)

Владимирова Элина, Евдокимов Данил, Назаров Максим, Стельмах Татьяна

Январь 2021

Содержание

1 Введение	1
2 Постановка задачи	2
3 Описание метода	2
4 Описание данных	2
5 Описание постановки эксперимента	2
6 Результаты	4
7 Краткие выводы	6

1 Введение

В нашем проекте мы будем работать с некоторыми экономическими показателями: **Темп инфляции** (скорость роста цен в стране) и **Уровень безработицы** (отношение численности безработных к численности рабочей силы в процентах), а также спрогнозируем зависимость между уровнем инфляции и уровнем безработицы в долгосрочном периоде в США, Германии, России и Японии.

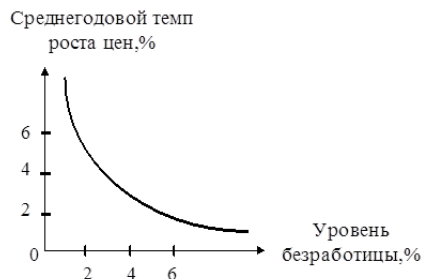


Рисунок 1. Кривая Филлипса

Для краткосрочного периода справедлива "Кривая Филлипса", которая показывает обратную зависимость между уровнями инфляции и безработицы (Рисунок 1)

2 Постановка задачи

Необходимо сделать предположение о регрессионной модели (предварительно построив линейную, экспоненциальную и полиномиальную регрессионные модели) для найденных данных (темпы инфляции и уровень безработицы США, Японии, России и Германии во второй половине 20 и в начале 21 веков), проверить его корректность, построить соответствующие графики, оценить полученные результаты и сделать краткие выводы.

3 Описание метода

В своём проекте мы будем использовать библиотеку **Sklearn** для построения линейной, экспоненциальной и полиномиальной регрессионных моделей. После построения и обучения моделей мы сможем предсказывать поведение темпа инфляции в зависимости от безработицы в долгосрочном периоде.

4 Описание данных

Для получения нужных нам данных в ходе работы мы будем активно пользоваться **Federal Reserve Economic Data** - базой данных, которая ведется исследовательским отделом Федерального резервного банка Сент-Луиса (Рисунок 2). В итоге на вход мы получаем csv-файлы, которые будем считывать непосредственно в самой программе.



Рисунок 2. Federal Reserve Economic Data

5 Описание постановки эксперимента

Кратко опишем код. Снизу приведём используемые библиотеки.

```

from math import sqrt, exp, fabs
import numpy as np
import csv
from sklearn.linear_model import LinearRegression      # модель линейной регрессии
from sklearn.preprocessing import PolynomialFeatures
# генерирует новую матрицу со всеми полиномиальными комбинациями элементов до заданной степени
import matplotlib.pyplot as plt

```

А теперь на примере Японии посмотрим, как мы будем считывать наши данные:

```

with open('FPCPITOTLZGUSA.csv') as csvfile:      # США
    reader = csv.DictReader(csvfile)
    for row in reader:
        y.append(float(row['FPCPITOTLZGUSA'])) # Считываем темпы инфляции (1960-2019)
with open('NROU.csv') as csvfile:
    reader = csv.DictReader(csvfile)
    for row in reader:
        x.append(float(row['NROU']))           # Считываем уровни безработицы

```

1. Опишем построение линейной регрессионной модели:

```

X = np.array(x).reshape(-1,1)      # создаём двумерный массив для обработки его моделью
model1 = LinearRegression().fit(X, Y) # обучаем модель
y_pred1 = model1.predict(X).tolist() # предсказываем Y по X
# изображаем на графике полученную зависимость
scatter1 = plt.plot(x, y_pred1, c = 'red', label = u'Линейная регрессия')

```

2. Опишем построение экспоненциальной модели:

```

i = 0
y_0 = y
x_0 = x
for y0 in y:      # избегаем 0
    if y0 == 0:
        x_0.pop(i)
        y_0.pop(i)
        i-=1
    i+=1
expo = np.polyfit(x_0, np.log(y_0), 1, w=np.sqrt(y_0))
# находит коэффициенты полинома p(x) степени n, который аппроксимирует функцию y(x) в смысле метода наименьших квадратов
y_pred3 = []
x.sort()      # чтобы построить функцию
for x1 in x:
    y_pred3.append(exp(expo[1]) * exp(expo[0] * x1))
scatter1 = plt.plot(x, y_pred3, c = 'yellow', label = u'Экспоненциальная регрессия')

```

Какой же хитростью мы воспользовались? Мы ищем коэффициенты A и B в случае $y = A \cdot e^{B \cdot x}$ - возьмём же натуральный логарифм от обеих частей и получим $\log y = \log A + B \cdot x$. Таким образом, нам достаточно найти коэффициенты полинома первой степени, а затем по ним восстановить y .

3. Опишем построение полиномиальной модели:

```

poly_reg = PolynomialFeatures(degree=4)
x_poly = poly_reg.fit_transform(X)
model2 = LinearRegression()
model2.fit(x_poly, Y)
y_pred2 = model2.predict(x_poly).tolist()
x.sort()
X = np.array(x).reshape(-1,1)
scatter1 = plt.plot(x, model2.predict(poly_reg.fit_transform(X)), c = 'green', label = u'Полиномиальная регрессия')

```

Теперь, когда мы имеем предсказанные значения и модели, мы можем указать наиболее подходящую для нашей ситуации регрессионную модель. Для этого вычислим среднеквадратичное (стандартное) отклонение расстояний от данных точек до предсказанных (точки рассматриваем парами), используя `numpy.std`, - методы этой библиотеки вычисляют его по следующей формуле:

$$S = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^n (a_i - \bar{a})^2}, \text{ где } \bar{a} - \text{среднее арифметическое чисел } a_1, a_2, \dots, a_n.$$

Взглянем на код:

```
dists1 = [] # расстояния до точек, предсказанные линейной регрессией
dists2 = [] # полиномиальной регрессией
dists3 = [] # экспоненциальной регрессией
for j in range(len(x)):
    dists1.append(fabs(y_pred1[j] - y[j])) # вычисляет дробный модуль
    dists2.append(fabs(y_pred2[j] - y[j]))
    dists3.append(fabs(y_pred3[j] - y[j]))
dists = [round(np.std(dists1), 5), round(np.std(dists2), 5), round(np.std(dists3), 5)] # округляет до 5-ти знаков
min_dist = float('inf')
i = 0
indexs = []
for dist in dists:
    if dist < min_dist:
        min_dist = dist # нашли минимальное среднеквадратичное отклонение среди них
for dist in dists:
    if dist > min_dist - 0.001 and dist < min_dist + 0.001:
        indexs.append(i)
    i+=1
```

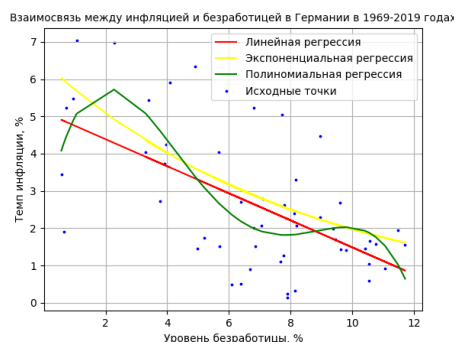
Осталось только построить графики и вывести наиболее подходящую модель регрессии:

```
if indexs.count(0) != 0:
    print('Линейное приближение достаточно хорошо')
if indexs.count(1) != 0:
    print('Полиномиальное приближение достаточно хорошо')
if indexs.count(2) != 0:
    print('Экспоненциальное приближение достаточно хорошо')

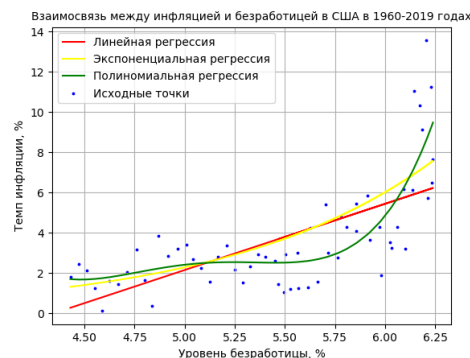
size = 3
trans = 1
scatter1 = plt.scatter(x, y, c='blue', s = size, alpha = trans, label = u'Исходные точки')
plt.grid(True)
plt.xlabel(u'Уровень безработицы, %')
plt.ylabel(u'Темп инфляции, %')
plt.title(u'Взаимосвязь между инфляцией и безработицей в Японии в 1970-2019 годах', fontsize = 10)
plt.legend()
plt.show()
```

6 Результаты

Посмотрим же на долгожданные результаты:



Figure₃.png



Figure₄.png

Во всех случаях нам подошла полиномиальная регрессия. Более того, уже по этим изображениям мы можем сказать о зависимости между темпом инфляции и уровнем безработицы в различных странах в долгосрочном периоде, даже можем предсказать исход снижения или увлечения темпов инфляции, однако наша модель все ещё не совершенна. Так как мы уже уверены в том, что полиномиальная регрессия в этом случае будет наиболее подходящей, то мы можем попробовать увеличить степень полиномиальной регрессии и посмотреть на результат (на примере Японии), но перед этим хотелось бы отметить несколько интересных деталей:

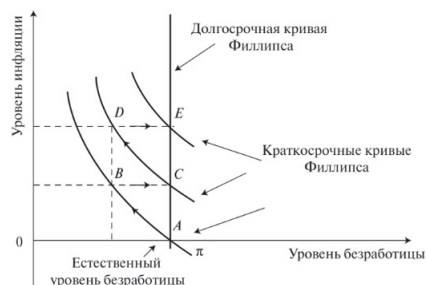


Рисунок 3. Long run

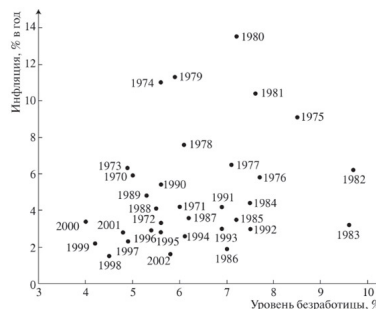
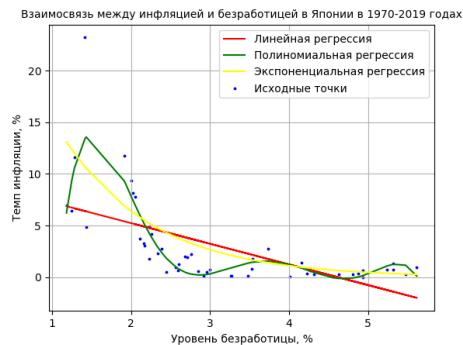
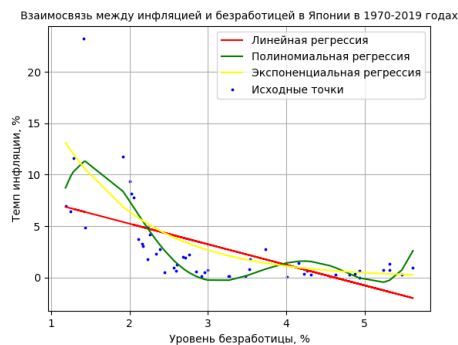


Рисунок 4. New Long run

Если мы посмотрим на получившиеся графики зависимости между темпом инфляции и уровнем безработицы в Германии и Японии, то сможем заметить некоторое их сходство с кривой Филлипса в краткосрочном периоде. Вероятно, в этих странах экономика была более стабильной, чем, например, в России и США (на протяжении 50 лет).

Также получившиеся графики довольно-таки далеки от изображения кривой Филлипса в долгосрочном периоде. Почему? Во-первых, темпы прироста номинальной заработной платы включают показатели ожидаемых темпов инфляции и шоковых изменений предложения, а также учитывают следующие факты: 1). экономика может выходить на уровень полной занятости при

любых темпах инфляции, 2). в длительном периоде инфляция и безработица не корреспондируют друг с другом. Во-вторых, в настоящее время многие страны уже давно пришли к *Современной кривой Филлипса*, которую мы и построили для каждой страны благодаря полиномиальной регрессии. В ходе работы мы столкнулись с несколькими экономическими статьями, оставляющими современную кривую Филлипса лишь в точечном виде (*Рисунок 4*), в нашей же работе была осуществлена попытка наглядно передать эту зависимость для каждой из названных стран. Вернёмся к обещанному анализу эффективности полиномиальной регрессии в зависимости от степени используемого полинома:



Уже на 40-ой степени программа сообщает о наибольшем соответствии действительности экспоненциальной регрессионной модели (хотя ранее программа не считала ее подходящей). Отметим также, что при подобных преобразованиях кривая заметно отделилась от основного скопления точек а, например, на 25-ой степени сильнее приблизилась к нему.

7 Краткие выводы

Мы построили и обучили несколько регрессионных моделей (линейную, полиномиальную и экспоненциальную) и на нашем наборе данных смогли найти наиболее подходящую модель и предположить и установить некоторую зависимость темпов инфляции от уровня безработицы в Японии, Германии, России и США, а также изобразить полученный результат. Мы убедились в том, что полиномиальная регрессионная модель является наиболее подходящей для данной ситуации и может быть улучшена путём нахождения идеальной ее степени для конкретной ситуации (так, опытным путём было установлено, что она лежит в пределах 25-30 для Японии), с сильным же увеличением ее точность может изрядно пострадать. К сожалению, мы не смогли в явном виде изобразить кривую Филлипса в краткосрочном периоде в связи с недостатком данных, однако получили Современную реальную Кривую

Филлипса для нескольких стран и выдвинули свою гипотезу о зависимости этих двух экономических величин.

Идея	Евдокимов Данил, Стельмах Татьяна, Назаров Максим
Тактическая концепция	Евдокимов Данил, Назаров Максим
Код и его проверка	Евдокимов Данил, Назаров Максим, Стельмах Татьяна Владимирова Элина
Отчет	Стельмах Татьяна
Редакция отчета	Владимирова Элина
Подбор данных	Стельмах Татьяна, Евдокимов Данил

Ссылки

<https://www.grandars.ru/student/ekonomicheskaya-teoriya/krivaya-filipsa.html> Упомянутая статья
<https://fred.stlouisfed.org/> Источник данных