

You can also work online [jupyterlite](#)

1. Predicting Apartment Prices Using Linear Regression

Task: Determine the cost of an apartment depending on the area

Source code below in the assignment

Excel files in my GitHub repository

Регрессия – задача предсказать величину конкретного признака объекта в **числовом выражении** используя имеющиеся данные по другим признакам объекта.

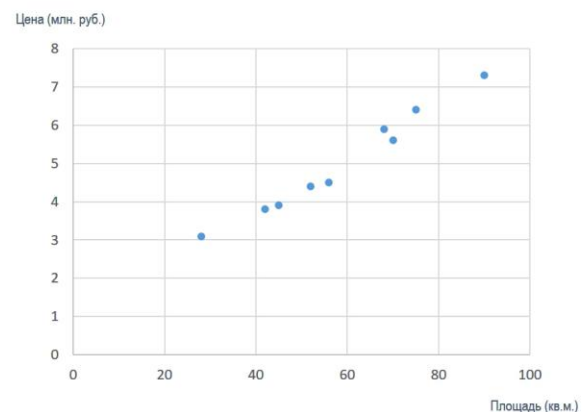
Знаем	
Площадь (кв.м.)	Цена (млн. руб.)
28	2,4
42	3,7
45	3,9
56	4,5
68	5,7
75	6,4
90	7,8
Находим формулу	
$f(x)$	$Y = aX + b$
Предсказываем	
34	?
49	?

Знаем	
Рост (см)	Вес (кг)
158	49
160	53
160	58
173	67
175	77
182	80
184	91
Находим формулу	
$f(x)$	$Y = aX + b$
Предсказываем	
176	?
186	?

Знаем	
Площадь торг.зала (квм)	Продажи (млн)
250	35
160	18
320	38
203	22
545	67
482	60
195	21
Находим формулу	
$f(x)$	$Y = aX + b$
Предсказываем	
230	?
420	?

Задача регрессии:
Определить стоимость квартиры в
зависимости от площади

Знаем	
Площадь (кв.м.)	Цена (млн. руб.)
28	3,1
42	3,8
45	3,9
52	4,4
56	4,5
68	5,9
70	5,6
75	6,4
90	7,3
Находим формулу зависимости	
$f(x)$	
Предсказываем	
34	?
49	?



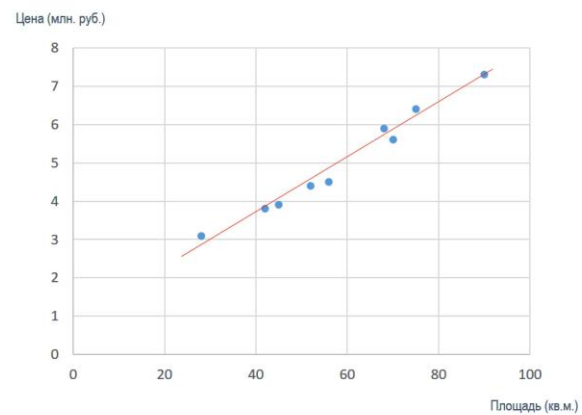
Задача регрессии:
Определить стоимость квартиры в
зависимости от площади

В линейной зависимости используется формула
прямой:

Наша задача – найти коэффициенты a , b

$$Y = aX + b$$

цена площадь



Задача регрессии:
Определить стоимость квартиры в
зависимости от площади

В линейной зависимости используется формула
прямой:

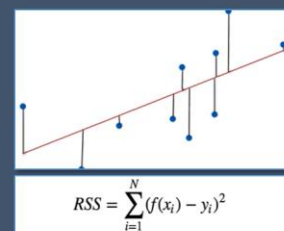
Наша задача – найти коэффициенты a , b

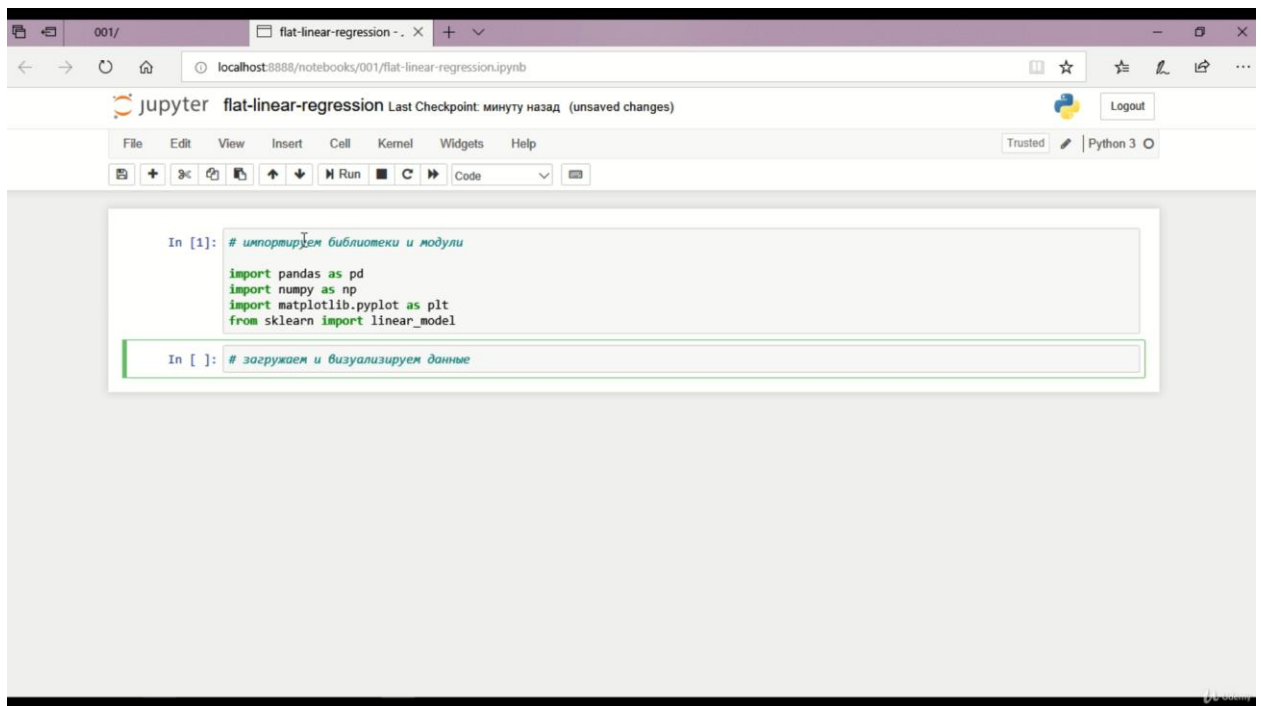
$$Y = aX + b$$

цена площадь

Метод наименьших квадратов

- Найти формулу, при которой сумма квадратных отклонений наименьшая от искомым переменных





```
#!/usr/bin/env python
# coding: utf-8

# In[1]:

# импортируем библиотеки и модули

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import linear_model

# In[2]:

# загружаем и визуализируем данные

# In[3]:

df = pd.read_excel('price1.xlsx')

# In[4]:

df

# In[12]:

get_ipython().run_line_magic('matplotlib', 'inline')
```

```
plt.scatter(df.area, df.price, color='red', marker='^')
plt.xlabel('площадь (кв.м.)')
plt.ylabel('стоимость (млн.руб)')

# In[13]:

# тренируем модель

# In[14]:

reg = linear_model.LinearRegression() #создали модель

# In[15]:

reg.fit(df[['area']],df.price) #обучаем модель на наших данных

# In[16]:

# предсказываем

# In[18]:

reg.predict([[120]])

# In[19]:

reg.predict(df[['area']])

# In[20]:

#  $Y = aX + b$ 

# In[21]:

reg.coef_

# In[22]:

reg.intercept_

# In[23]:

# Стоимость = 0.07148238 * Площадь + 0.8111407046647887
```

```
# In[24]:

0.07148238 * 120 + 0.8111407046647887

# In[26]:

get_ipython().run_line_magic('matplotlib', 'inline')
plt.scatter(df.area, df.price, color='red', marker='^')
plt.xlabel('площадь (кв.м.)')
plt.ylabel('стоимость (млн.руб)')
plt.plot(df.area, reg.predict(df[['area']]))

# In[27]:

pred = pd.read_excel('prediction_price.xlsx')

# In[28]:

pred

# In[30]:

pred.head(3)

# In[32]:

p = reg.predict(pred) # предсказываем цены для новых квартир из нового файла
по нашей модели

# In[33]:

p

# In[34]:

# создаем новую колонку в файле с новыми квартирами и заносим туда
предсказанные цены

# In[35]:

pred['predicted prices'] = p

# In[36]:
```

```
pred

# In[40]:

pred.to_excel('new.xlsx', index=False) # сохраняем файл в Excel без первой
колонки
```

Self task:

- 1) Find the dependence of Russia's GDP on oil prices based on historical data
- 2) Upload the gdprussia.xlsx file to your Jupyter notebook
- 3) Display data as a graph
- 4) Train the model with a linear regression algorithm
- 5) Predict GDP depending on different oil prices

2. Predicting GDP from Oil Prices Using Linear Regression

We use the data from the gdprussia.xlsx file (you can download it in the repository)

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:

# Импортируем модули и библиотеки

# In[3]:

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import linear_model

# In[5]:

df = pd.read_excel('gdprussia.xlsx')

# In[24]:

df

# In[11]:

get_ipython().run_line_magic('matplotlib', 'inline')
plt.scatter(df.oilprice, df.gdp)
plt.xlabel('oil price (US$)')
plt.ylabel('GDP, Russia (bln US$)')
```

```
# In[12]:

# Тренируем модель

# In[13]:

reg = linear_model.LinearRegression()
reg.fit(df[['oilprice']], df.gdp)

# In[14]:

# Предсказываем

# In[15]:

reg.predict(df[['oilprice']])

# In[16]:

get_ipython().run_line_magic('matplotlib', 'inline')
plt.scatter(df.oilprice, df.gdp)
plt.xlabel('oil price (US$)')
plt.ylabel('GDP, Russia (bln US$)')
plt.plot(df.oilprice, reg.predict(df[['oilprice']]))

# In[18]:

reg.predict([[150]])

# In[20]:

# Модель, предсказывающая ВВП в зависимости от 1) года и 2) от цены на нефть

# In[21]:

reg = linear_model.LinearRegression()
reg.fit(df[['year', 'oilprice']], df.gdp)

# In[22]:

reg.predict(df[['year', 'oilprice']])
```

```
# In[23]:  
  
reg.predict([[2025,100]])  
  
# In[ ]:
```