

Работу выполнили: Гарин М.А., Исоков А.А., Квашнин К.А., Манин Д.И

Вариант задания: по данным с сайта объявлений avito (или аналога) построить модель (множественной регрессии), с помощью которой можно предсказать цену на 2х комнатную квартиру площадью 60 кв.м, расположенную на втором этаже кирпичного дома в одном из микрорайонов города Саранска.

Код программы:

```
import pandas as pd
from bs4 import BeautifulSoup
import requests as req
import string
import plotly
import plotly.graph_objs as go

def is_apartment(title):
    f1 = "квартира"
    if (str(title).rfind(f1) != -1): # если найдена подстрока "-к"
        return title # возвращаем всю строку

def is_price(title):
    return title # возвращаем всю строку

def get_room(title):
    a = title.rfind('-к') # позиция вхождения подстроки
    #b = title.rfind('м') - 1 # позиция вхождения подстроки "руб"
    str1 = title[a-1: a] # выделение подстроки с ценой с
    return str1 # возвращаем всю строку

def get_area(title):
    k = title.rfind('квартира,' ) # позиция вхождения подстроки
    b2 = title.rfind('м') - 1 # позиция вхождения подстроки "руб"
    s1 = title[k+9:b2] # выделение подстроки с ценой с
    return s1 # возвращение этой подстроки

df_price = pd.DataFrame({'цена': []}) # создание таблицы pandas
df_area = pd.DataFrame({'площадь': []}) # создание таблицы pandas
df_room = pd.DataFrame({'комнаты': []}) # создание таблицы pandas
pages = 1 # количество анализируемых страниц
base_url = 'https://www.avito.ru/saransk/kvartiry/prodam/2-komnatnye-ASgBAQICAUSSA8YQAUDKCBSCWQ?cd=1&f=ASgBAQECAUSSA8YQAUDKCBSCWQFF4AcXeyJmcm9tIjo1MTIxLCJ0byl6NTEyMX0'
for i in range(pages): # цикл по просматриваемым страницам
    url = base_url.format(str(i)) # форматирование ссылки
    resp = req.get(url) # передача ссылки
```

```

soup = BeautifulSoup(resp.text, "html.parser") # парсинг
links = soup.findAll('a')
prices = soup.findAll(attrs={"class" : "snippet-price"}) #заголовок цены
# print(links) #тест
# print(prices) # тест

for link in links:
    title = link.get('title') # все элементы
    if (title != None):
        # print(title) # вывод элементов
        if is_apartment(title): # если найдена подстрока

            df1 = pd.DataFrame( # добавляем в датафрейм
                {'площадь': [get_area(title)]}) # распределение по столбцам
            df_area = df_area.append(df1, ignore_index=True) # добавить строку df1 в df
        if is_apartment(title): # если найдена подстрока
            # print(content) # вывод элементов
            df3 = pd.DataFrame( # добавляем в датафрейм
                {'комнаты': [get_room(title)]}) # распределение по столбцам
            df_room = df_room.append(df3, ignore_index=True) # добавить строку df1 в df
df = pd.concat([df_area, df_room], axis=1, sort=False)

for price in prices:
    content = price.string # все элементы
    if (content != None):
        # print(content) # вывод элементов
        #print(type(content))
        s = str(content)
        s = s.replace("\n", "")
        s = s.replace(" ", "")
        s = s.replace("₽", "")
        int_price = int(s)
        if is_price(content): # если найдена подстрока
            df2 = pd.DataFrame( # добавляем в датафрейм
                {'цена': [int_price]}) # распределение по столбцам
            df_price = df_price.append(df2, ignore_index=True) # добавить строку df1 в df
df = pd.concat([df, df_price], axis=1, sort=False)

data = pd.read_csv("Apartments.csv")

fig1 = go.Scatter3d(x=data['площадь'], # построение фигуры plotlib
                     y=data['комнаты'],
                     z=data['цена'],
                     marker=dict(opacity=0.9,
                                reversescale=True,

```

```
        colorscale='Blues',
        size=5),
line=dict (width=0.02),
mode='markers')

mylayout = go.Layout(scene=dict(xaxis=dict( title="площадь"), # разметка
                                 yaxis=dict( title="комнаты"),
                                 zaxis=dict(title="цена")))

plotly.offline.plot({"data": [fig1], # построение и вывод в файл
                     "layout": mylayout},
                     auto_open=True,
                     filename=("3DPlot.html"))

df.to_csv("Apartments.csv") # создаем csv из Dataframe
df_room.to_csv("test.csv") # создаем csv из Dataframe

#print(df_price) # вывод df
#print(df_area)
#print(df_room)
#([df_price, df_area, df_room], axis=1, sort=False)
#print(df)
```

Результат выполнения программы:

