1. Загрузка и подготовка данных

In [1]:

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
data_0 = pd.read_csv('/datasets/geo_data_0.csv').drop('id', axis=1)
data_1 = pd.read_csv('/datasets/geo_data_1.csv').drop('id', axis=1)
data_2 = pd.read_csv('/datasets/geo_data_2.csv').drop('id', axis=1)
data_0.head()
```

Out[1]:

	f0	f1	f2	product
0	0.705745	-0.497823	1.221170	105.280062
1	1.334711	-0.340164	4.365080	73.037750
2	1.022732	0.151990	1.419926	85.265647
3	-0.032172	0.139033	2.978566	168.620776
4	1.988431	0.155413	4.751769	154.036647

идентификаторы месторождений роли не играют, поэтому удалю их

2. Обучение и проверка модели

In [3]:

```
predictions=[]
targets = []
for data in [data_0, data_1, data_2]:
    features, target = data.drop('product', axis=1), data.loc[:, 'product']
    features_train, features_valid, target_train, target_valid = train_test_split(features, target, test_size=0.2
5, random_state=12345)
    model = LinearRegression()
    model.fit(features_train, target_train)
    prediction = pd.Series(model.predict(features_valid))
    print(mean_squared_error(target_valid, prediction)**0.5, '-- RMSE\n', prediction.mean(),'-- среднее')
    predictions.append(prediction)
    targets.append(target_valid.reset_index(drop=True))
```

```
37.5794217150813 -- RMSE
 92.59256778438038 -- среднее
0.893099286775616 -- RMSE
 68.728546895446 -- среднее
40.02970873393434 -- RMSE
 94.96504596800489 -- среднее
```

вывод: 1 и 3 регионы примерно аналогичны по показателю среднего предсказанного запаса сырья, и эта величина сильно превышает соответствующий показатель второго региона. вместе с тем для 1 и 3 региона корень из среднеквадратической ошибки составляет весьма заметную часть от среднего показателя добычи, тогда как в регионе 2 предсказания моделей отличаются высокой четкостью. закономерный итог -- регионы 1 и 3 могут быть более прибыльными, но и более рискованными для инвестиций, тогда как вложения в регион 2 должны окупиться в меньшем размере, но с почти единичной вероятностью вложенные средства отобьются

3. Подготовка к расчёту прибыли

In [4]:

```
BUDGET = 10000000000
UNIT_INC = 450000
BEST = 200
THRESHOLD = BUDGET/(BEST*UNIT_INC)
THRESHOLD
i=0
for target in targets:
    i+=1
    if target.mean()>THRESHOLD:
        print('Для региона {}, величина объема нефти среднего месторождения превышает объем, требуемый для безубы
точности месторождения'.format(i))
    else:
        print('Для региона {}, величина объема нефти среднего месторождения не превышает объем, требуемый для без
убыточности месторождения'.format(i))
```

```
Для региона 1, величина объема нефти среднего месторождения не превышает объем, требуемый для безубы
точности месторождения
Для региона 2, величина объема нефти среднего месторождения не превышает объем, требуемый для безубы
точности месторождения
Для региона 3, величина объема нефти среднего месторождения не превышает объем, требуемый для безубы
точности месторождения
```

средняя прибыль 200 лучших месторождений региона должна превышать выведенное выше значение или равняться ему

сравнение объемов со средней величиной:

видно, что средние показатели для каждого из регионов, увы, не позволяют сделать оптимистичных выводов. К счастью, средняя величина вообще позволяет делать какие то выводы лишь для очень небольшого числа ситуаций, поэтому правильным решением будет продолжить анализ

4. Создание функции для оценки прибыли

In [5]:

```
def get_revenue(target, prediction):
    top = target[prediction.sort_values(ascending=False).index][:200]
    total_units = sum(top)
    revenue = total_units*UNIT_INC-BUDGET
    return revenue
get_revenue(targets[0], predictions[0])
```

Out[5]:

```
3320826043.1398544
```

Функция рассчитывает **фактическую** прибыль, полученную в результате эксплуатации 200 вышек, которые, согласно построенной модели, должны показать лучшипе результаты.

5. Вычисление прибыли и рисков для каждого региона

In [9]:

```
import numpy as np

revenues = []
state = np.random.RandomState(12345)
for i in range(3):
    rev = []
    for j in range(1000):
        pred_sample = predictions[i].sample(n=500, replace=True, random_state=state)
        target_sample = targets[i][pred_sample.index]
        rev.append(get_revenue(target_sample, pred_sample))
    rev = pd.Series(rev)
    print(rev.mean(),'среднее для {} региона'.format(i+1))
    print('Доверительный интервал:')
    print(rev.quantile(.025),'-- low', rev.quantile(.975),'-- high')
    print(str(rev[rev<0].count()/len(rev)*100)+'% -- шанс убытков\n')
    revenues.append(rev)
```

```
425938526.910593 среднее для 1 региона
Доверительный интервал:
-102090094.83793645 -- low 947976353.358369 -- high
6.0% -- шанс убытков

518259493.6973477 среднее для 2 региона
Доверительный интервал:
128123231.43310332 -- low 953612982.0669408 -- high
0.3% -- шанс убытков

420194005.3440505 среднее для 3 региона
Доверительный интервал:
-115852609.1600059 -- low 989629939.8445683 -- high
6.2% -- шанс убытков
```

Итог:

Для разработки скважин я бы порекомендовал регион №3, тк средняя прибыль там выше. Месторождения №1 и №3 совсем не перспективны, ведь шанс убытков для каждого региона меньше весьма высоки (допустимый уровень -- 2,5%).