Выбор локации для скважины

Перед нами поставлена задача определить, где бурить новую скважину.

Нам предоставлены пробы нефти в трёх регионах: в каждом 10 000 месторождений, где измерили качество нефти и объём её запасов. Нам надо построить модель машинного обучения, которая поможет определить регион, где добыча принесёт наибольшую прибыль. Проанализируем возможную прибыль и риски техникой *Bootstrap*

Шаги для выбора локации:

- В избранном регионе ищем месторождения, для каждого определяем значения признаков;
- Строим модель и оценивем объём запасов;
- Выбираем месторождения с самым высокими оценками значений. Количество месторождений зависит от бюджета компании и стоимости разработки одной скважины;
- Прибыль равна суммарной прибыли отобранных месторождений.

0.1 Описание данных

Данные геологоразведки трёх регионов:

- id — уникальный идентификатор скважины;
- f0, f1, f2 — три признака точек (нам не раскрыт);
- product — объём запасов в скважине (тыс. баррелей).

0.2 Условия задачи:

- Для обучения модели подходит только линейная регрессия (остальные — недостаточно предсказуемые).
- При разведке региона исследуем 500 точек, из которых с помощью машинного обучения выбираем 200 лучших для разработки.
- Бюджет на разработку скважин в регионе — 10 млрд рублей.
- При нынешних ценах один баррель сырья приносит 450 рублей дохода. Доход с каждой единицы продукта составляет 450 тыс. рублей, поскольку объём указан в тысячах баррелей.
- После оценки рисков оставим те регионы, в которых вероятность убытков меньше 2.5%. Среди них выбирем регион с наибольшей средней прибылью.
- Данные синтетические: детали контрактов и характеристики месторождений не разглашаются.

1 Загрузка и подготовка данных

Ввод [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import StandardScaler
from sklearn.utils import shuffle
from sklearn.metrics import accuracy_score, precision_score, recall_score,
import warnings
warnings.filterwarnings('ignore')
```

1.1 Загрузка данных

Ввод [2]:
```
geo_data_0 = pd.read_csv('./geo_data_0.csv', index_col=0)
```

Ввод [3]:
```
geo_data_1 = pd.read_csv('./geo_data_1.csv', index_col=0)
```

Ввод [4]:
```
geo_data_2 = pd.read_csv('./geo_data_2.csv', index_col=0)
```

1.2 Обзор данных

Ввод [5]:
```
geo_data_0.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 100000 entries, 0 to 99999
Data columns (total 5 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   id       100000 non-null  object 
 1   f0       100000 non-null  float64
 2   f1       100000 non-null  float64
 3   f2       100000 non-null  float64
 4   product  100000 non-null  float64
dtypes: float64(4), object(1)
memory usage: 4.6+ MB
```

Ввод [6]:
```
geo_data_0.describe()
```

Out[6]:

	f0	f1	f2	product
count	100000.000000	100000.000000	100000.000000	100000.000000
mean	0.500419	0.250143	2.502647	92.500000
std	0.871832	0.504433	3.248248	44.288691
min	-1.408605	-0.848218	-12.088328	0.000000
25%	-0.072580	-0.200881	0.287748	56.497507
50%	0.502360	0.250252	2.515969	91.849972
75%	1.073581	0.700646	4.715088	128.564089
max	2.362331	1.343769	16.003790	185.364347

Ввод [7]:
```
geo_data_0.head()
```

Out[7]:

	id	f0	f1	f2	product
0	txEyH	0.705745	-0.497823	1.221170	105.280062
1	2acmU	1.334711	-0.340164	4.365080	73.037750
2	409Wp	1.022732	0.151990	1.419926	85.265647
3	iJLyR	-0.032172	0.139033	2.978566	168.620776
4	Xdl7t	1.988431	0.155413	4.751769	154.036647

Пропущенных данных нет, поле id явно содержит коды скважин и скорее всего не будет нужно

Ввод [8]:
```
geo_data_1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 100000 entries, 0 to 99999
Data columns (total 5 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   id       100000 non-null  object 
 1   f0       100000 non-null  float64
 2   f1       100000 non-null  float64
 3   f2       100000 non-null  float64
 4   product  100000 non-null  float64
dtypes: float64(4), object(1)
memory usage: 4.6+ MB
```

Ввод [9]:
```
geo_data_1.describe()
```

Out[9]:

	f0	f1	f2	product
count	100000.000000	100000.000000	100000.000000	100000.000000
mean	1.141296	-4.796579	2.494541	68.825000
std	8.965932	5.119872	1.703572	45.944423
min	-31.609576	-26.358598	-0.018144	0.000000
25%	-6.298551	-8.267985	1.000021	26.953261
50%	1.153055	-4.813172	2.011479	57.085625
75%	8.621015	-1.332816	3.999904	107.813044
max	29.421755	18.734063	5.019721	137.945408

Ввод [10]:
```
geo_data_1.head()
```

Out[10]:

	id	f0	f1	f2	product
0	kBEdx	-15.001348	-8.276000	-0.005876	3.179103
1	62mP7	14.272088	-3.475083	0.999183	26.953261
2	vyE1P	6.263187	-5.948386	5.001160	134.766305
3	KcrkZ	-13.081196	-11.506057	4.999415	137.945408
4	AHL4O	12.702195	-8.147433	5.004363	134.766305

Пропущенных данных нет

Ввод [11]:
```
geo_data_2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 100000 entries, 0 to 99999
Data columns (total 5 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   id       100000 non-null  object 
 1   f0       100000 non-null  float64
 2   f1       100000 non-null  float64
 3   f2       100000 non-null  float64
 4   product  100000 non-null  float64
dtypes: float64(4), object(1)
memory usage: 4.6+ MB
```

Ввод [12]:
```
geo_data_2.describe()
```

Out[12]:

	f0	f1	f2	product
count	100000.000000	100000.000000	100000.000000	100000.000000
mean	0.002023	-0.002081	2.495128	95.000000
std	1.732045	1.730417	3.473445	44.749921
min	-8.760004	-7.084020	-11.970335	0.000000
25%	-1.162288	-1.174820	0.130359	59.450441
50%	0.009424	-0.009482	2.484236	94.925613
75%	1.158535	1.163678	4.858794	130.595027
max	7.238262	7.844801	16.739402	190.029838

Ввод [13]:
```
geo_data_2.head()
```

Out[13]:

	id	f0	f1	f2	product
0	fwXo0	-1.146987	0.963328	-0.828965	27.758673
1	WJtFt	0.262778	0.269839	-2.530187	56.069697
2	ovLUW	0.194587	0.289035	-5.586433	62.871910
3	q6cA6	2.236060	-0.553760	0.930038	114.572842
4	WPMUX	-0.515993	1.716266	5.899011	149.600746

Пропущенных данных нет.

Поле id не нужно в последующем анализе. Удалим его

Ввод [14]:
```
geo_data_0 = geo_data_0.drop(['id'], axis=1)
geo_data_1 = geo_data_1.drop(['id'], axis=1)
geo_data_2 = geo_data_2.drop(['id'], axis=1)
```

Ввод [15]:
```
geo_data_0.head()
```

Out[15]:

	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

1.3 Вывод

Данные загружены корректно, удалены поля с названиями скважин. Дальнейшая предобработка не требуется

2 Обучение и проверка модели

2.1 Обучаем модель для региона 0

Ввод [16]:
```
features_0 = geo_data_0.drop(['product'],axis=1)
target_0 = geo_data_0['product']
features_train_0, features_valid_0, target_train_0, target_valid_0 = train_
    features_0, target_0, test_size=0.25, random_state=12345)
```

Ввод [17]:
```
model_0 = LinearRegression()
model_0.fit(features_train_0, target_train_0)
prediction_0 = model_0.predict(features_valid_0)
mse_0 = mean_squared_error(target_valid_0, prediction_0)
rmse_0 = mse_0**0.5
mean_pred_0 = prediction_0.mean()
```

Ввод [18]:
```
print('RMSE модели в регионе 0 = {:.3f}'.format(rmse_0))
print('Средний запас предсказанного сырья 0 = {:.3f} т. баррелей'.format(me
```

```
RMSE модели в регионе 0 = 37.579
Средний запас предсказанного сырья 0 = 92.593 т. баррелей
```

2.2 Обучаем модель для региона 1

Ввод [19]:
```
features_1 = geo_data_1.drop(['product'],axis=1)
target_1 = geo_data_1['product']
features_train_1, features_valid_1, target_train_1, target_valid_1 = train_
    features_1, target_1, test_size=0.25, random_state=12345)
```

Ввод [20]:
```
model_1 = LinearRegression()
model_1.fit(features_train_1, target_train_1)
prediction_1 = model_1.predict(features_valid_1)
mse_1 = mean_squared_error(target_valid_1, prediction_1)
rmse_1 = mse_1**0.5
mean_pred_1 = prediction_1.mean()
```

Ввод [21]:
```
print('RMSE модели в регионе 1 = {:.3f}'.format(rmse_1))
print('Средний запас предсказанного сырья 1 = {:.3f} т. баррелей'.format(me
```

```
RMSE модели в регионе 1 = 0.893
Средний запас предсказанного сырья 1 = 68.729 т. баррелей
```

2.3 Обучаем модель для региона 2

Ввод [22]:
```
features_2 = geo_data_2.drop(['product'],axis=1)
target_2 = geo_data_2['product']
features_train_2, features_valid_2, target_train_2, target_valid_2 = train_
    features_2, target_2, test_size=0.25, random_state=12345)
```

Ввод [23]:
```
model_2 = LinearRegression()
model_2.fit(features_train_2, target_train_2)
prediction_2 = model_2.predict(features_valid_2)
mse_2 = mean_squared_error(target_valid_2, prediction_2)
rmse_2 = mse_2**0.5
mean_pred_2 = prediction_2.mean()
```

Ввод [24]:
```
print('RMSE модели в регионе 2 = {:.3f}'.format(rmse_2))
print('Средний запас предсказанного сырья 2 = {:.3f} т. баррелей'.format(me
```

```
RMSE модели в регионе 2 = 40.030
Средний запас предсказанного сырья 2 = 94.965 т. баррелей
```

2.4 Вывод:

Лучшие показатели RMSE модели в Регионе 1 (RMSE = 0.893). То есть для этого региона запас сырья наиболее предсказуем.

Однако средний запас предсказанного сырья в этом регионе (1) = 68.729 т. баррелей, что меньше чем в двух других регионах(92.593 и 94.965).

3 Подготовка к расчёту прибыли

Сохраним ключевые значения для расчетов в отдельных переменных

Ввод [25]:
```
BUDGET = 10000000000
BARREL_P = 450*1000
ALL_T = 500
BEST_T = 200
```

Ввод [26]:
```
cost_t = BUDGET / BEST_T
print('Бюджет бурения одного месторождения, руб:', cost_t)
```

```
Бюджет бурения одного месторождения, руб: 50000000.0
```

Рассчитаем точку безубыточности

Ввод [27]:
```
tochka_b = cost_t / BARREL_P
```

Ввод [28]:
```
print('Объём сырья для безубыточной разработки новой скважины  = {:.3f} т.
```

```
Объём сырья для безубыточной разработки новой скважины  = 111.111 т. барр
елей
```

Средний запас по регионам ниже, чем точка безубыточности в единицах продукции

4 Расчёт прибыли и рисков

Напишем функции, которые будут выводить расчет прибыли и рисков.

Ввод [29]:
```
def revenue(pred, real):
    df = real.to_frame(name='real')
    df['pred'] = pred
    #print(df.info())
    #print(df.head())
    #print(real)
    top = df.sort_values('pred', ascending=False).head(BEST_T)
    volume = top['real'].sum()
    rev = volume * BARREL_P - BUDGET
    return rev
```

Ввод [36]:
```
def revandr(pred, target):
    state = np.random.RandomState(12345)
    values = []
    target_r = target.reset_index(drop=True)
    for i in range(1000):
        target_subsample = target_r.sample(n=500, replace=True, random_stat
        pred_subsample = pred[target_subsample.index]
        values.append(revenue(pred_subsample, target_subsample))

    values_s = pd.Series(values)
    lower_s = values_s.quantile(0.025) / 1000000
    upper_s = values_s.quantile(0.975) / 1000000

    mean_s = values_s.mean() / 1000000
    print("Средняя прибыль: {:.1f} млн. руб.".format(mean_s))
    print("Доверительный интервал: {:.1f} : {:.1f} млн. руб.".format(lower_
    print()
    print('Риск убытков: {:.2%}'.format(len([i for i in values if i < 0])/1
```

4.1 Расчет прибыли для региона 0

Ввод [37]:
```
revandr(prediction_0, target_valid_0)
```

```
Средняя прибыль: 396.2 млн. руб.
Доверительный интервал: -111.2 : 909.8 млн. руб.

Риск убытков: 6.90%
```

4.2 Расчет прибыли для региона 1

Ввод [38]:
```
revandr(prediction_1, target_valid_1)
```

```
Средняя прибыль: 456.0 млн. руб.
Доверительный интервал: 33.8 : 852.3 млн. руб.

Риск убытков: 1.50%
```

4.3 Расчет прибыли для региона 2

Ввод [39]:
```
revandr(prediction_2, target_valid_2)
```

```
Средняя прибыль: 404.4 млн. руб.
Доверительный интервал: -163.4 : 950.4 млн. руб.

Риск убытков: 7.60%
```

4.4 Вывод

Наимболее перспективный регион - 1. На параметрах скважин этого региона модель обучилась наиболее хорошо и дает более корректные результаты. Прибыль с разработки наибольшая и составляет 456.0 млн. руб.. Кроме того, разработка этого региона характеризуется наименьшим риском - всего 1.5%

5 Вывод

По полученным данным было построено три модели - для каждого из трех регионов. Наилучшей пресказательной силой обладает модель, построенная для региона 1. RMSE этой модели = 0.893

Расчитан средний запас предсказанного сырья:

- Регион 0: 92.593 т. баррелей (RMSE = 37.6)
- Регион 1: 68.729 т. баррелей (RMSE = 0.9)
- Регион 2: 94.965 т. баррелей (RMSE = 40.0)

Средний запас предсказанного сырья в 0 и 2 регионах выше, чем в 1. Однако ошибка RMSE перекрывает это преимущество. Для более точного понимания прибыли и рисков была использована технология бутстрап.

С помощью технологии бутстрап произведено моделирование получаемой прибыли при разработке региона. Наиболее перспективным регионом является регион 1. Средняя прогнозируемая прибыль: 456.0, доверительный 95% интервал: 33.8 : 852.3 млн. руб., уровень риска: 1.5%

Остальные регионы имеют меньшую среднюю прогнозируемую прибыль и повышенные уровни риска: 6.9% и 7.6%