Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ имени Н.Э.БАУМАНА
(национальный исследовательский университет)»

Факультет: Информатика и системы управления

Кафедра: Теоретическая информатика и компьютерные технологии

Лабораторная работа № 4

«Анализ результатов проб нефти»

по дисциплине «Моделирование»

Работу выполнил

студент группы ИУ9-82Б

Жук Дмитрий

Москва, 2023

Цель работы

Целью данной работы является построение регрессионной модели для данных о пробах нефти с предварительной очисткой результатов наблюдения с использованием статистических методов для оценки прибыльности разработки месторождений.

Задание

Предоставлены пробы нефти в трёх регионах: в каждом 100 000 месторождений, где измерили качество нефти и объём её запасов. Необходимо построить модель, которая поможет определить регион, где добыча принесёт наибольшую прибыль. Шаги для выбора локации:

1. в избранном регионе ищут месторождения, для каждого определяют значения признаков;
2. строят модель и оценивают объём запасов;
3. выбирают месторождения с самым высокими оценками значений, количество месторождений зависит от бюджета компании и стоимости разработки одной скважины;
4. прибыль равна суммарной прибыли отобранных месторождений.

Предоставлены три набора данных, соответствующие трем разным исследуемым локациям, в них id — уникальный идентификатор скважины; f0, f1, f2 — три признака точек (неважно, что они означают, но сами признаки значимы); product — объём запасов в скважине (тыс. баррелей). Необходимо провести предвварительную обработку данных. Выявить выбросы (если есть), рассчитать квартили, интерквартильный размах, выборочную дисперсию для всех столбцов каждого набора данных. Определить корреляцию целевого признака (product) с зависимыми признаками для каждого набора данных.

Теория

Представленные наборы данных могут быть описаны как функциональные или стохастические зависимости. Функциональная зависимость определяет соответствие между каждым значением из множества X и соответствующим ему значением из множества Y. Стохастическая зависимость, в свою очередь, может иметь несколько значений Y для каждого значения X и характеризуется вероятностной природой. Функциональная зависимость является частным случаем стохастической, который возникает при наиболее тесной связи между переменными. Когда оценивается стохастическая зависимость, применяются методы корреляции, чтобы определить наличие взаимосвязи между переменными, и регрессии, чтобы определить ее характер.

В математической статистике регрессионный анализ – это совокупность методов, используемых для определения связей между независимой переменной Y и одной или несколькими переменными $X_1, X_2, \dots, X_m$. Регрессия представляет собой условное математическое ожидание случайной переменной Y при фиксированном значении другой переменной X. Линейная регрессионная модель является моделью, в которой теоретическое среднее значение зависимой переменной y является линейной комбинацией независимых переменных:

$$\mu = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k$$

Множители $\beta_0, \beta_1, \dots, \beta_k$ представляют собой параметры модели, значения которых должны быть установлены. Они называются коэффициентами регрессии, а β_0 называется свободным или постоянным членом. Модель, более чем с одной переменной x называется моделью множественной регрессии.

Следующие термины используются при анализе данных:

- Квантиль, квартиль и интерквартильный размах.

- α-квантиль ($x\alpha$) для эмпирического распределения можно определить следующим образом: сначала упорядочиваются значения выборки в вариационный ряд $V_0 \le V_1 \le \cdots \le V_{N-1}$, где N – объем выборки, $V_N = V_{N-1}$. Затем вычисляется $\lfloor \alpha(N-1) \rfloor$, и сравнивается с индексом K и значением αN. Если $K + 1 < \alpha N$, то $x_\alpha = V_{K+1}$, если $K + 1 = \alpha N$, то $x_\alpha = \frac{V_K + V_{K+1}}{2}$, а если $K + 1 > \alpha N$, то $x_\alpha = V_K$.

- Первый (нижний) квартиль соответствует 0.25-квантилю, медиана (второй квартиль) соответствует 0.5-квантилю, а третий (верхний) квартиль соответствует 0.75-квантилю.

- Интерквартильный размах определяется как разность между третьим и первым квартилями и используется в качестве характеристики распределения величины, аналогично дисперсии.

- Выброс в статистике — это результат измерения, который выделяется из общей выборки. Для определения выбросов могут использоваться простые методы, основанные на интерквартильном размахе, например, всё, что не попадает в следующий диапазон, считается выбросом:

$$\left[\left(x_{0.25} - 1.5(x_{0.75} - x_{0.25})\right), \left(x_{0.75} + 1.5(x_{0.75} - x_{0.25})\right)\right]$$

Выборочное среднее:

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{N} X_i$$

Выборочное среднее:

$$S_n^2 = \frac{1}{n}\sum_{i=1}^{N} (X_i - \bar{X})^2$$

Коэффициент корреляции Пирсона:

$$\mathrm{r}_{XY} = \frac{\sum_{i=1}^{N}\sum_{i=1}^{N}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{N}(X_i - \bar{X})^2 \sum_{i=1}^{N}(Y_i - \bar{Y})^2}}$$

Реализация

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
import sklearn.pipeline as pipe
first = pd.read_csv('document.txt', sep=r'\s+')
first = first.drop('ind', axis=1)
# first = first[first['product']>0]
second = pd.read_csv('1.txt', sep=r'\s+')
second = second.drop('ind', axis=1)
third = pd.read_csv('2.txt', sep=r'\s+')
third = third.drop('ind', axis=1)
def quant(data: pd.DataFrame, field: str, alpha: float):
d = data.copy()
d = d.sort_values(field)
N = len(d)
K = int(alpha*(N-1))
d = pd.concat([d, d.tail(1)])
# print(float(d.iloc[[K+1]]['product']))
if K + 1 < alpha*N:
return float(d.iloc[[K+1]][field])
elif K+1 == alpha*N:
return (float(d.iloc[[K]][field])+float(d.iloc[[K+1]][field]))/2
else:
return float(d.iloc[[K]][field])

def correl(a: list, b: list) -> float:
mean_a = sum(a)/len(a)
mean_b = sum(b)/len(b)
desp_a = sum([(i - mean_a) ** 2 for i in a])
desp_b = sum([(i - mean_b) ** 2 for i in b])
return sum([(a[i]-mean_a)*(b[i]-mean_b) for i
inrange(len(a))])/(desp_a*desp_b)**0.5
def get_outliers(data):
x25 = quant(data, 'product', 0.25)
x75 = quant(data, 'product', 0.75)
a = x25-1.5*(x75-x25)
b = x75+1.5*(x75-x25)
return data[ (data['product'] < a) | (data['product'] > b)]
def stat(data: pd.DataFrame):
print('квантиль 0.25: ', quant(data, 'product', 0.25))
print('квантиль 0.5: ', quant(data, 'product', 0.5))
print('квантиль 0.75: ', quant(data, 'product', 0.75))
```

```
print('интерквартильный размах: ', quant(data, 'product', 0.75) -quant(data,
'product', 0.25))
product = data['product'].to_list()
f0 = data['f0'].to_list()
f1 = data['f1'].to_list()
f2 = data['f2'].to_list()
mean_p = sum(product)/len(product)
desp = sum([(i - mean_p) ** 2 for i in product])
print('выборочная дисперсия: ', 1/len(product)*desp)
print('cov product f0: ', correl(product, f0))
print('cov product f1: ', correl(product, f1))
print('cov product f2: ', correl(product, f2))
print('cov f0 f2: ', correl(f0, f2))
print('cov f0 f1: ', correl(f0, f1))
print('cov f1 f2: ', correl(f1, f2))
#Небольшая процедура для предварительного анализа данных
def define_dataset(df):
print(df.shape)
print(df.info())
print(df.head(40))
print(df.describe())
return df['id'].value_counts().head(20) #определениеиндексов-дубликатов
```

Функция get_dummies

```
#Формируем наборы признаков и вектор целевого признака для всех трехлокаций,
одинакого исключая из списка признаков
#идентификатор (индекс) месторождения - он никак не может влиять на
объемдобытой нефти

features_1 = first.drop(['id','product'], axis=1)
features_ohe_1 = pd.get_dummies(features_1, drop_first=True)
target_1 = first['product']

features_2 = second.drop(['id','product'], axis=1)
features_ohe_2 = pd.get_dummies(features_2, drop_first=True)
target_2 = second['product']

features_3 = third.drop(['id','product'], axis=1)
features_ohe_3 = pd.get_dummies(features_3, drop_first=True)
target_3 = third['product']

#Разбиваем данные на обучающую и валидационную выборки в соотношении 75:25.
features_train_1, features_valid_1, target_train_1, target_valid_1
=train_test_split(features_ohe_1, target_1,
test_size=0.25,random_state=12345)
features_train_2, features_valid_2, target_train_2, target_valid_2
=train_test_split(features_ohe_2, target_2,
test_size=0.25,random_state=12345)
features_train_3, features_valid_3, target_train_3, target_valid_3
=train_test_split(features_ohe_3, target_3,
test_size=0.25,random_state=12345)
model_1 = LinearRegression() #Применяем модель линейной регрессии
model_2 = LinearRegression()
model_3 = LinearRegression()

#Признаки кодируем во избежание доминирования одного из них
numeric = ['f0','f1','f2']
def scale(features_train, features_valid = None, numeric=['f0','f1','f2']):
```

```
scaler = StandardScaler()
scaler.fit(features_train_1[numeric])
features_train[numeric] = scaler.transform(features_train[numeric])
if features_valid is not None:
features_valid[numeric] =scaler.transform(features_valid[numeric])
return

#Обучаем модель и проводим предсказания на первой валидационной выборке.
def study(model: LinearRegression, features_train,
features_valid,target_train, target_valid, number_location):
model.fit(features_train,target_train) # обучите модель на
первойтренировочной выборке
predictions_valid = model.predict(features_valid) # получитепредсказания
модели на первой валидационной выборке
#Выводим на печать средний запас предсказанного сырья и RMSE моделидля первой
локации.
mse = mean_squared_error(target_valid, predictions_valid)

# < извлекаем корень из MSE >
result = mse ** 0.5
print("Средний запас предсказанного на валидационной
выборке",number_location, "сырья:", predictions_valid.mean(),
'(тыс.баррелей)')
print("RMSE модели линейной регрессии на валидационной
выборке",number_location, ":", result)
return predictions_valid
model_1 = pipe.Pipeline([
('scaler', StandardScaler()),
('model', LinearRegression())
])

# study(model, features_train_1, features_valid_1,
target_train_1,target_valid_1, 1)
study(model_1, features_train_1, features_valid_1,
target_train_1,target_valid_1, 1)
model_2 = pipe.Pipeline([
('scaler', StandardScaler()),
('model', LinearRegression())
])
study(model_2, features_train_2, features_valid_2,
target_train_2,target_valid_2, 2)
model_3 = pipe.Pipeline([
('scaler', StandardScaler()),
('model', LinearRegression())
])
study(model_3, features_train_3, features_valid_3,
target_train_3,target_valid_3, 3)
r1 = pd.read_csv('place1.csv', sep=',')
r2 = pd.read_csv('place2.csv', sep=',')
r3 = pd.read_csv('place3.csv', sep=',')
COSTS = 500_000 #бюджет на разработку
INCOME = 450 #доход с одного бареля нефти
COUNT_REGION = 30 #количество исследуемых точек в одном регионе
BOREHOLES = 16 #количество выбранных скважин для разработки месторождения
loss_threshold = COSTS/(BOREHOLES*INCOME) #Минимальная средняяпродуктивность
скважины для достижения порога окупаемости
```

```
region_threshold = round(BOREHOLES*loss_threshold,1)
#Минимальнаяпродуктивность 200 скважин региона для достижения порога
окупаемости
print('Минимальная средняя продуктивность скважины для достижения
порогаокупаемости:', round(loss_threshold,1), '(тыс. баррелей)')
def calc_profit(data: pd.DataFrame, model: LinearRegression):
d = data.copy()
# d.sample()
d_product = model.predict(d[numeric])
# print(d_product)
d['product'] = d_product
d.sort_values(by='product', inplace=True)
top_d = d.tail(BOREHOLES)
# print(top_d)
return top_d['product'].sum() * INCOME - COSTS

print("Прибыль в первой локации:", calc_profit(r1,
model_1).round(),'тысрублей')
print("Прибыль в первой локации:", calc_profit(r2,
model_2).round(),'тысрублей')
print("Прибыль в первой локации:", calc_profit(r3,
model_3).round(),'тысрублей')
import numpy as np
state = np.random.RandomState(12345) #обеспечим случайность
формируемыхвыборок
def bootstrapped(data: pd.DataFrame, model: LinearRegression):
values = []
d = data.copy()
for _ in range(1000):
profit = calc_profit(d.sample(COUNT_REGION,
replace=False,random_state=state), model)
values.append(profit.round())

values = pd.Series(values)
mean = values.mean() #расчет средней прибыли
print('Средняя прибыль, тыс руб.: {:,.2f}'.format(mean))

lower = values.quantile(.025) #строим доверительный интервал
upper = values.quantile(.975)

print('95% доверительный интервал:', '{:,.2f}'.format(lower),
':','{:,.2f}'.format(upper))
bootstrapped(r1, model_1)
bootstrapped(r2, model_2)
bootstrapped(r3, model_3)
```

Результат

Для первого региона была обнаружена высокая (99%) корреляция между признаком $f2$ и целевым признаком $product$ с помощью статистического анализа. Это позволяет исключить признаки $f0$ и $f1$ при построении линейной регрессии в данном регионе.

Статистический анализ показал наличие одного выброса во втором наборе данных, однако удаление этой записи не представляется целесообразным из-за малого размера выборки.

Для выполнения задачи по выбору 200 точек из 500 с бюджетом 10 млрд. рублей на 200 точек (50 млн. рублей на точку) выбиралось 16 точек, так как в наборе данных было всего 40 записей. Общий бюджет составил 800 млн. рублей (50 млн. рублей на точку), а доход с одного барреля не изменился и составил 450 рублей.

Из-за малого объема набора данных выбор разбиения на обучающую и валидационную выборки может оказывать значительное влияние на результат. Для уменьшения этого фактора использовалась технология bootstrap: из 40 месторождений выбиралось 30 и проводилось предсказание для уменьшенной выборки. В качестве конечного результата было взято среднее значение результатов на 1000 итерациях.

В таблице 1 приведены результаты работы модели.

	Регион 1	Регион 2	Регион 3
Средний запас нефти, тыс баррелей	101	79	99
RMSE на валидационной выборке	1,14	39,4	45,5
Средняя прибыль, тыс руб. (bootstrap)	213163	-36786	105355
95% доверительный интервал (bootstrap)	(78394; 326257)	(-88196; 7432)	(67463; 135708)

Таблица 1 — Результат анализа

Вывод

В процессе выполнения этой лабораторной работы была построена регрессионная модель для данных о пробах нефти, а также был проведен статистический анализ, включающий анализ выбросов и корреляций признаков. Первый регион был выбран как наиболее перспективный, так как оценка прибыли и доверительный интервал показали лучший результат. Кроме того, доверительный интервал для третьего региона показал, что этот регион является безубыточным, в то время как нижняя граница интервала для второго региона находится в отрицательной зоне.