Тренировочный вариант 1

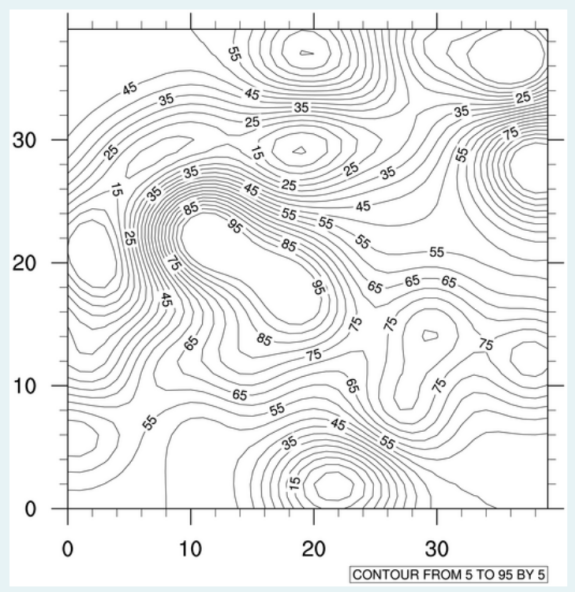
Задания

1 +

Иногда приходится иметь дело с трехмерными поверхностями, т.е. результатом табулирования функции от двух переменных z = f(x, y). Сейчас, как правило, с восприятием таких графиков нет никаких проблем - есть много способов создать интерактивный трехмерный график, который можно вертеть и масштабировать как угодно. Но если мы говорим о статичной картинке, то с пониманием графика возникают некоторые трудности. Выход нашли ещё до появления компьютеров, при издании карт местности, который заключается в том, что строят не трехмерный график, а его отображение на плоскости. Это отображение получают по следующим правилам: берут заданное количество плоскостей, параллельных плоскости xy, и выделяют место пересечения поверхности с каждой из этих плоскостей - контуры, затем проецируют данные контуры на плоскость xy и всё — график готов.

Каждому контуру соответствует определенное значение величины z, его подписывают рядом с контуром.

Перед нами стоит задача минимизировать некоторую функцию потерь, contour plot которой изображен ниже.



Выберите все верные утверждения:

Решение:

- По графику можно точно определить минимальное значение функции, оно равно 15

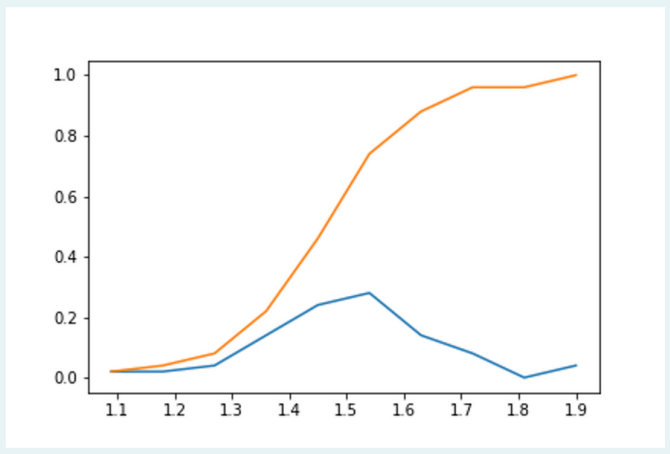
+ На картинке есть как области локальных минимумов, так и области локальных максимумов

- В точке (x,y)=(16,20) значение функции близко к своему минимальному значению.

+ Функция имеет несколько локальных экстремумов

2 +

На картинке нарисованы графики плотности распределения (pdf) и функции распределения (cdf) некоторой случайной величины.



Выберите верный вариант ответа:

Решение:

+ Оранжевый график - это функция распределения, синий - плотность

- Оранжевый график - это плотность, синий - функция распределения

- Мода случайной величины равна примерно 1.8

- Оранжевый график не может быть плотностью

3 +

Аналитик Вася с утра просыпается с головной болью и попадает по нужной клавише на клавиатуре с вероятностью 0.7. Найдите математическое ожидание количества правильно нажатых клавиш в предложении из 40 символов.

Решение:

28

4 +

Инженер данных Сергей построил пайплайн для сбора и хранения данных. Сергей утверждает, что в его алгоритме данные теряются с вероятностью 2%. Затем пришел инженер данных Виталий и, проверяя гипотезу Сергея, заявил, что такая маленькая потеря данных невозможна. Сергей обиделся и протестировал свой алгоритм многократными проверками, при этом каждый раз только 2% данных терялось.

Выберите верное утверждение:

Решение:

+ Виталий совершил ошибку первого рода

- Виталий не совершил ошибку, ошибку совершил Сергей - но какого рода, неизвестно

- Виталий совершил ошибку второго рода

5 +

Предположим, вы проводите исследование благосостояния граждан по странам. Имеющиеся у вас данные включают в себя уровень дохода, среднегодовую температуру в месте проживания респондента, степень удовлетворенности жизнью, ВВП на душу населения, возраст респондента, его рост и другие характеристики для 100 тысяч граждан 100 государств. Хотим построить модель, которая будет по данным для человека предсказывать уровень благосостояния - любое число на отрезке от 1 до 100 (где 1 - низкий уровень, а 100 - высокий уровень). Ответ может быть любым числом из отрезка [1; 100], в том числе и нецелым.

Выберите все верные утверждения (их два):

Решение:

+ Объектом является гражданин страны

- Имеем дело с задачей классификации

- Объектом в этой задаче является страна

+ Будем работать над решением задачи регрессии

6 +

Цель банка - найти мошенников при совершении банковских операций (мошеннических транзакций на несколько порядков меньше, чем не мошеннических). Пусть при обучении классификации для выявления мошеннических транзакций в банке accuracy на тренировочных данных получилось равным 0.9, а на тестовых - 0.85. О чем может говорить эта ситуация?

Для решения задачи использовали логистическую регрессию с Lasso-регуляризацией.

Выберите наиболее подходящий вариант ответа.

Решение:

- Модель переобучилась, так как качество на тесте ниже, чем на трейне.

- Модель недообучилась, так как качество на трейне низкое

+ Неизвестно, переобучилась или недообучилась модель, так как выбрана плохая метрика для данной задачи

- Модель обучилась как надо (нет ни недообучения, ни переобучения)

7 +

Дан текст: “Но не каждый хочет что-то исправлять :(”

После некоторой обработки получилось:

['но', ' ', 'не', ' ', 'каждый', ' ', 'хотеть', ' ', 'что-то', ' ', 'исправлять', ':(\\n']
Выберите все шаги, которые были сделаны с исходным текстом:

Решение:

- Стемминг (stemming)

+ Токенизация (tokenization)

- Векторизация (vectorization)

+ Лемматизация (lemmatization)

8 ?

Мы решаем задачу классификации - определения по МРТ-снимкам, болен пациент некоторой болезнью или нет. Мы не хотим часто ошибаться, то есть называть больных здоровыми, но и здоровых называть больными мы также не хотим. В обучающих данных 8543 пациента, из которых 4100 больных, а остальные здоровые.

Какие метрики можно использовать для измерения качества модели в этой задаче?

Решение:

- MAPE

+ F1-score

- MSLE (mean squared logarithmic error)

+ AUC-PR (площадь под Precision-Recall кривой)

-Accuracy

9 +

Выберите два корректных утверждения про градиентный спуск:

Решение:

- Градиентный спуск применяется для нахождения максимума функции потерь

+ Правильный подбор шага градиентного спуска позволяет уменьшить количество шагов, необходимых для поиска минимума

+ Если сделать длину шага градиентного спуска недостаточно маленькой, то алгоритм может разойтись.

- На каждом шаге алгоритма считается градиент от одного, случайно выбранного элемента.

10 +

Напомним, что существует формула для разложения ошибки модели на шум, смещение и разброс:

$Bias^2$ - смещение модели, Var - разброс, δ^2 - шум в данных.

На некоторых данных обучили линейную регрессию. Как изменятся компоненты $Bias^2$ и Var ошибки линейной регрессии, если добавить в данные для обучения модели полиномиальные признаки степени 2?

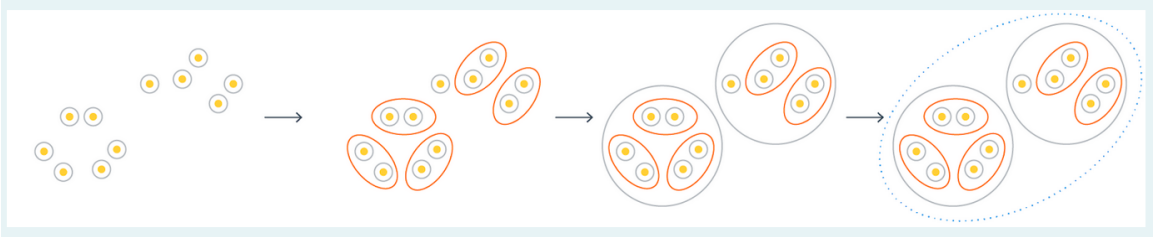
Среди перечисленных ниже два ответа верные.

Решение:

+ Var увеличится
+
$Bias^2$ уменьшится

11 +

На рисунке ниже изображен некоторый алгоритм кластеризации. А какой?



Решение:

- Спектральная кластеризация

+ Аггломеративная (иерархическая) кластеризация

- DBSCAN

- K-means

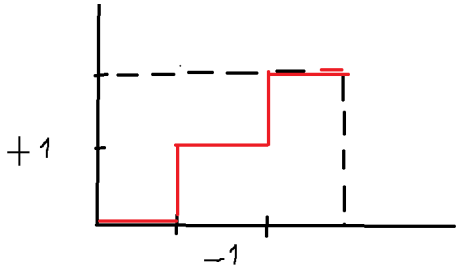
12 +

Алгоритм бинарной классификации для каждого объекта x_i выдает оценку b_i его принадлежности к положительному классу. Ниже в таблице даны предсказания b_i и правильные ответы y_i.

x_i	y_i
0.7	+1
0.6	-1
0.3	-1
0.45	+1
0.92	-1

Вычислите ROC-AUC. Ответ округлите до сотых.

Решение:



$(1/2)(1/3) + 1(1/3) = 0.5$

13 +

В вершине дерева, решающего задачу бинарной классификации, находилось 40 объектов класса 1 и 60 объектов класса 0. После разбиения вершины на две группы по некоторому условию:

- в левой вершине оказалось 20 объектов класса 1 и 50 объектов класса 0
- в правую вершину попали все остальные объекты.

Вычислите Information Gain:

$Q = H(R) - \frac{|R_l|}{|R|}H(R_l) - \frac{|R_r|}{|R|}H(R_r),$

где $|A|$ - количество объектов в вершине A,

$H(R) = \sum_{k=1}^{2} p_k(1 - p_k)$- значение критерия Джини в вершине R.

Ответ округлите до сотых.

Решение:

$((40/100)(1 - 40/100) + (60/100)(1 - 60/100)) - (70/100)((20/70)(1 - 20/70) + (50/70)(1 - 50/70)) - (30/100)((20/30)(1 - 20/30) + (10/30)(1 - 10/30)) = 0.06$

14 +

В машинном обучении есть подход, позволяющий при помощи линейных моделей решать линейно неразделимые задачи классификации: в этом подходе мы переходим в новое пространство признаков и в этом

пространстве решаем задачу при помощи линейной модели. Скалярное произведение векторов в новом пространстве задается функцией, называющейся ядром.

Дано ядро $K(a,b) = exp(-||a-b||^2)$, где $||a-b||$ - евклидова норма (длина) вектора

$a-b$.

Вычислите косинус угла между векторами $a = (1,1,1)$ и $b = (1,2,0)$ в новом признаковом пространстве, в котором скалярное произведение задается функцией $K(a,b)$.

Ответ округлите до сотых.

Решение:

1. Вычислим скалярное произведение в новом пространстве:

   Сначала найдем разность векторов: a - b = (1-1, 1-2, 1-0) = (0, -1, 1)

   Вычислим квадрат евклидовой нормы этой разности: ||a - b||² = 0² + (-1)² + 1² = 2

   Подставим значение в формулу ядра: K(a, b) = exp(-||a - b||²) = exp(-2)

2. Вычислим нормы векторов в новом пространстве:

   ||a|| = sqrt(K(a, a)) = sqrt(exp(-||a - a||²)) = sqrt(exp(0)) = 1
   ||b|| = sqrt(K(b, b)) = sqrt(exp(-||b - b||²)) = sqrt(exp(0)) = 1

3. Вычислим косинус угла:

   Напомним формулу косинуса угла между векторами: cos(θ) = (a, b) / (||a|| ||b||)
   Подставим значения, полученные в новом пространстве: cos(θ) = exp(-2) / (1 1) = exp(-2)

   Округлим ответ:

   cos(θ) ≈ 0.14

15 +

За круглый стол на 201 стул в случайном порядке рассаживаются 199 разработчиков и 2 аналитика. Найдите вероятность того, что между

аналитиками будет сидеть один разработчик.

Решение:

$1 * (2/200) = 0.01$

Инфо

В файлах TrainData.csv и TestData.csv находятся данные о клиентах некоторой компании.

В этом задании предлагается изучить анонимизированные характеристики клиентов и на их основе решить задачу оттока: понять, покинет клиент компанию или нет.

Описание столбцов:

- столбцы 0-13 - анонимизированная информация о клиентах
- столбец target - целевая переменная: 1 - клиент покинет компанию, 0 - не покинет

Считайте данные в два pandas dataframe: df_train и df_test.

```
import numpy as np
import pandas as pd
train = pd.read_csv('1_TrainData.csv')
test = pd.read_csv('1_TestData.csv')
```

16 +

Проверьте, есть ли в тренировочных и в тестовых данных пропуски? Укажите количество столбцов тренировочной выборки, имеющих пропуски

Решение:

```
train.isna().sum()
```

Ответ:

2

17 +

В столбце с наибольшим количеством пропусков заполните пропуски средним значением по столбцу. В ответ запишите значение вычисленного среднего.

Ответ округлите до десятых.

Решение:

```
train['1'].mean()
train['1'].fillna(train['1'].mean(), inplace=True)
```

Ответ:

31.3

18 +

Найдите строки в тренировочных данных, где пропуски стоят в столбце с наименьшим количеством пропусков. Удалите эти строки. Сколько строк вы удалили?

Решение:

```
train.dropna(inplace=True)
```

Ответ:

3

19 +

Переведите столбец с целевой переменной в бинарные значения по правилу: Churn - 1, Not churn - 0. Исправьте опечатки в названиях категорий целевой переменной (до того, как переводить их в бинарные значения).
Сколько опечаток вы исправили?

Решение:

```
train[(train['target']!='Churn') & (train['target']!='Not chu
rn')]

train['target'] = train['target'].replace('Chorn', 'Churn')
train['target'] = train['target'].replace('Not chorn', 'Not c
hurn')

train['target'] = train['target'].replace('Churn', '1')
train['target'] = train['target'].replace('Not churn', '0')
train['target'] = train['target'].astype(int)
```

Ответ:

2

Demo day 23 -

Найдите числовой признак, имеющий наибольшую по модулю корреляцию Пирсона с колонкой target. В ответ напишите модуль корреляции Пирсона с целевой переменной для этого признака. Ответ округлите до сотых.

Решение:

```
train.corr(numeric_only=True)
```

Ответ:

0.44

20 +

В этом задании все пункты выполняйте только по таблице df_train.

Сколько столбцов в таблице (не считая target) содержат меньше 5 различных значений?

Решение:

```
train.nunique()
```

Ответ:

6

21 +

Верно ли, что если значение единственного категориального признака в таблице равно A, то клиент уйдет из компании? (target = 1) В ответ запишите “да” или “нет” без кавычек.

Решение:

```
len(train[train['7'] == 'A']), len(train[(train['7'] == 'A')
& (train['target'] == 'Churn')])
```

Ответ:

нет

22 +

Вычислите долю ушедших из компании клиентов, для которых значение признака 2 больше среднего значения по столбцу, а значение признака 13 меньше медианы по столбцу. Ответ округлите до сотых.

Решение:

```
mean2 = train['2'].mean()
median13 = train['13'].median()
rows1 = train[(train['2'] > mean2) & (train['13'] < median1
3)]
rows2 = rows1['target'] == 1]
len(rows2)/len(rows1)
```

Ответ:

0.51

23+

Закодируйте категориальные столбцы в тренировочных и тестовых данных при помощи label encoding (категории кодируйте подряд идущими числами, начинающимися с 0).

Сколько столбцов после кодировки стало в тестовых данных?

Решение:

```
from sklearn.preprocessing import LabelEncoder
enc = LabelEncoder()
train['7'] = enc.fit_transform(train['7'])
test['7'] = enc.fit_transform(test['7'])
```

Ответ:

14

24 -

Разбейте тренировочные данные на целевой вектор y, содержащий значения из столбца target, и матрицу объект-признак X, содержащую остальные признаки. Обучите на этих данных логистическую регрессию из sklearn (LogisticRegression) с параметрами по умолчанию. Выведите среднее значение метрики f1-score алгоритма на кросс-валидации с тремя фолдами. Ответ округлите до сотых.

При объявлении модели фиксируйте random_state = 42.

Комментарий: параметры по умолчанию предполагаются следующими

penalty='l2', dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None

Решение:

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_score
```

```
# Создаем матрицу объект-признак X и целевой вектор y
X = train.drop('target', axis=1)
y = train['target']

# Создаем модель логистической регрессии
model = LogisticRegression(random_state=42)

# Вычисляем f1-score с помощью кросс-валидации
scores = cross_val_score(model, X, y, cv=3, scoring='f1')

# Выводим среднее значение f1-score
print("Среднее значение f1-score:", round(scores.mean(), 2))
```

Ответ:

0.82

25 -

Подберите значение константы регуляризации C в логистической регрессии, перебирая гиперпараметр от 0.001 до 100 включительно, проходя по степеням 10. Для выбора C примените перебор по сетке по тренировочной выборке (GridSearchCV из библиотеки sklearn.model_selection) с тремя фолдами и метрикой качества - f1-score. Остальные параметры оставьте по умолчанию. В ответ запишите наилучшее среди искомых значение C.

При объявлении модели фиксируйте random_state = 42.

Комментарий: параметры по умолчанию предполагаются следующими

penalty='l2', dual=False, tol=0.0001, fit_intercept=True, intercept_scaling=1, class_weight=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None

Решение:

```
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV
```

```
# Создаем матрицу объект-признак X и целевой вектор y
X = data.drop('target', axis=1)
y = data['target']

# Создаем модель логистической регрессии
model = LogisticRegression(random_state=42)

# Задаем сетку параметров для C
param_grid = {'C': [0.001, 0.01, 0.1, 1, 10, 100]}

# Создаем объект GridSearchCV
grid_search = GridSearchCV(estimator=model, param_grid=param_
grid, cv=3, scoring='f1')

# Обучаем GridSearchCV на тренировочных данных
grid_search.fit(X, y)

# Выводим наилучшее значение C
print("Наилучшее значение C:", grid_search.best_params_['C'])
```

Ответ:

1

26 +

Добавьте в тренировочные и тестовые данные новый признак ‘NEW’, равный произведению признаков ‘7’ и ‘11’.

На тренировочных данных с новым признаком заново с помощью GridSearchCV (с тремя фолдами и метрикой качества - f1-score) подберите оптимальное значение C (перебирайте те же значения C, что и в предыдущих заданиях), в ответ напишите наилучшее качество алгоритма (по метрике f1-score), ответ округлите до сотых.

При объявлении модели фиксируйте random_state = 42.

Комментарий: параметры по умолчанию предполагаются следующими

penalty='l2', dual=False, tol=0.0001, fit_intercept=True, intercept_scaling=1, class_weight=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None

Решение:

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV

# Создаем новый признак 'NEW'
train['NEW'] = train['7'] * train['11']
test['NEW'] = test['7'] * test['11']

# Создаем матрицу объект-признак X и целевой вектор y
X = train.drop('target', axis=1)
y = train['target']

# Создаем модель логистической регрессии
model = LogisticRegression(random_state=42)

# Задаем сетку параметров для C
param_grid = {'C': [0.001, 0.01, 0.1, 1, 10, 100]}

# Создаем объект GridSearchCV
grid_search = GridSearchCV(estimator=model, param_grid=param_
grid, cv=3, scoring='f1')

# Обучаем GridSearchCV на тренировочных данных
grid_search.fit(X, y)

# Выводим наилучшее качество (f1-score)
print("Наилучшее значение f1-score:", round(grid_search.best_
score_, 2))
```

Ответ:

0.85

27+

Теперь вы можете использовать любую модель машинного обучения для решения задачи. Также можете делать любую другую обработку признаков. Ваша задача - получить наилучшее качество по метрике accuracy на тестовых данных.

Качество проверяется на представленных тестовых данных.

- accuracy >= 0.88 - 0.25 балла
- accuracy >= 0.9 - 0.75 балла

Сдайте файл result.csv.

Решение:

```
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler

# Масштабирование признаков (важно для логистической регресси
и)
scaler = StandardScaler()
X = scaler.fit_transform(X)
test = scaler.transform(test)

# Создание и обучение модели логистической регрессии
model = LogisticRegression(random_state=42, max_iter=500)
model.fit(X, y)

# Предсказания на тестовых данных
predictions = model.predict(test)

# Создание DataFrame для результата
result = pd.DataFrame({'target': predictions})
```

```
# Сохранение результата в файл result.csv
result.to_csv("result.csv", index=False)
```

Ответ: