Тренировочный вариант 2

Задания

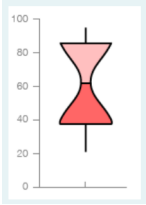
1 +

Vase plot - это тип визуализации данных, который сочетает в себе box plot (ящик с усами) и информацию о плотности распределения данных.

Vase plot содержит в себе:

- Центральную горизонтальную линию - медиану данных
- Верхнюю и нижнюю горизонтали - 75% (Q3) и 25%(Q1)-квантили распределения данных
- Усы заканчиваются в наибольшем значении выборки, не превышающем Q3+1.5IQR, и наименьшем значении выборки, превышающем Q1-1.5IQR, где IQR=Q3-Q1
- Точки выше и ниже усов - выбросы
- Вертикальные стороны фигур, в отличие от классического box plot, это визуализация плотности распределения данных (её части на отрезке от 25% до 75% квантили).

На рисунке изображен vase plot, отображающий распределение веса в некоторой группе людей.



Выберите верные утверждения относительно отображенных данных:

Решение:

+ В данных нет выбросов

+ Медиана набора данных приблизительно равна 60

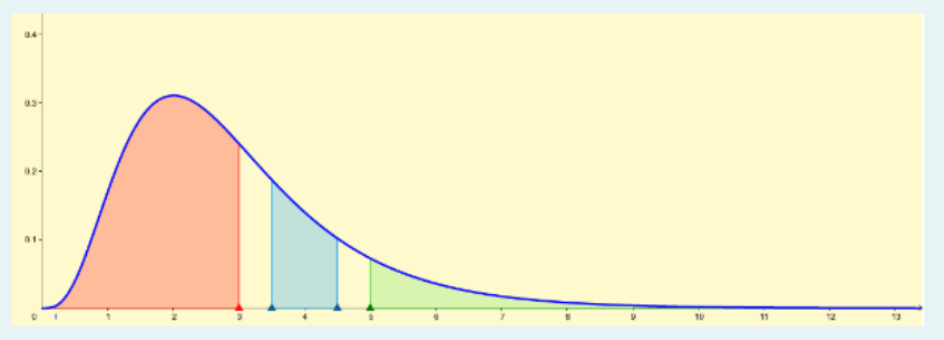
- В данных нет значений, меньших 30

- Интерквартильный размах равен приблизительно 30

- Данные имеют равномерное распределение

2 +

Время между клиентами (в минутах), посещающими магазин, имеет экспоненциальное распределение, показанное на рисунке ниже. По оси x отображено время в минутах.



По этому распределению посчитали три величины:

$P(X \leq 3) = 0.6, P(3.5 \leq X \leq 4.5) = 0.14, P(X \geq 5) = 0.1$

Выберите три верных утверждения.

Решение:

+ Вероятность того, что следующий клиент придет в течение первых трех минут после предыдущего, равна 0.6

- Каждый следующий клиент приходит не позднее, чем через 8 минут после предыдущего

+ Вероятность того, что следующий клиент придет на 7 или больше минут позже предыдущего, меньше 0.1

- Вероятность того, что следующий клиент придет в течение минуты после предыдущего, больше 0.6

+ Вероятность того, что следующий клиент придет на 4 или больше минут позже предыдущего, меньше 0.5

3 +

Виктор пытается дозвониться в телепередачу на радио. Известно, что вероятность дозвониться равна 0.02 и не зависит от предыдущих попыток. Виктор дозвонился с 10й попытки. На сколько попыток раньше он дозвонился, чем в среднем дозваниваются желающие?

Решение:

$1 = n * 0.02$

$n = 50$

$50 - 10 = 40$

Ответ:

40

4 +?

Имеется исследование о влиянии нового учебного метода на успеваемость студентов. Учебный метод внедрялся в группе студентов, исследователи ожидали, что он приведет к улучшению успеваемости.

Нулевая гипотеза: новый учебный метод не влияет на успеваемость студентов.

Пороговое значение статистической значимости (α) установлено на уровне 0.05. После анализа данных исследователи получили p-value = 0.08.

Какую ошибку исследователи могли допустить?

Решение:

Обе ошибки

Ни одной ошибки

Ошибка первого рода

+ Ошибка второго рода

5 +

Аналитики некоторой компании проводят исследование спроса клиентов на шариковые ручки.

Данные представлены в виде таблицы с полями "год покупки ручек", "возраст", "пол", "уровень образования", "средний доход", "город проживания клиентов".

Аналитики пытаются спрогнозировать количество шариковых ручек, которое каждый клиент купил за 2018, 2019, 2020, 2021, 2022 год.

Выберите два верных утверждения.

Решение:

Решается задача обучения без учителя

+ Целевой переменной в данной задаче является клиент

Решается задача классификации

+ Решается задача регрессии

Целевой переменной в данной задаче является количество купленных ручек

6 +

При решении задачи классификации accuracy на тренировочных данных оказалась равна 0.7, а на тестовых - 0.65. Что можно сказать о качестве модели?

Решение:

Модель сильно переобучена

+ Невозможно интерпретировать качество модели, не зная количества классов в задаче и информации о доле объектов каждого класса

Модель имеет низкое качество и на тренировочных, и на тестовых данных

Модель имеет высокое качество - как на тренировочных, так и на тестовых данных

Модель сильно переобучена, поэтому для снижения переобучения в этой задаче рекомендуется использовать регуляризацию

7 +

Выберите три верных утверждения про лемматизацию текстов:

Решение:

- Лемматизация - это обработка слов, в результате которой от каждого слова остается только его основа

+ Лемматизация нужна, чтобы снизить количество различных словоформ в текстах

+ В результате обучения моделей на векторизованных после лемматизации текстах переобучение обычно будет ниже, чем если не делать лемматизацию

- В результате лемматизации количество различных токенов в документе увеличивается

+ Лемматизация - это приведение слова к нормальной (словарной) форме

8 +

Астрономы решают задачу предсказания длительности путешествия от Земли до различных космических объектов в световых годах. Астрономам хочется получить как можно более точный результат, при этом для них гораздо хуже, если алгоритм завысит длительность путешествия, так как тогда астрономы не успеют провести все исследования. Занижение длительности по сравнению с правильным ответом не так страшно.

Какую из метрик астрономам лучше всего использовать для оценки качества модели?

Решение:

f1-score

+ MAE

accuracy

f1-weighted

9 +

Какая из приведенных ниже формул обладает возможностью задавать различные шаги градиентного спуска для разных весов (для различных координат вектора весов)?

Здесь w_k - значения вектора весов на k-й итерации градиентного спуска, $\nabla Q(w)$ - градиент функции потерь, v_k- вспомогательный вектор или скаляр, η, ρ - скаляры, гиперпараметры.

- ☐ $w_{k+1} = w_k - \frac{\eta}{v_k}\nabla Q(w_k)$, где $v_k = k$
- ☐ $w_{k+1} = w_k - \eta v_{k+1}$, где $v_{k+1} = \rho v_k + \nabla Q(w_k)$
- ☐ $w_{k+1} = w_k - \frac{\eta}{\sqrt{v_k + \varepsilon}}\nabla Q(w_k)$, где $v_k = v_{k-1} + (\nabla Q(w_k))^2$
- ☐ $w_{k+1} = w_k - \eta \nabla Q(w_k)$

Ответ:

1 -

2 -

3 +

4 -

10 +

Какая из перечисленных функций используют в качестве критериев информативности для построения решающих деревьев в задаче регрессии?

Решение:

- Критерий Джини

- Энтропия

- Доля ошибок классификации в листе

+ Среднеквадратичная ошибка

11 ?

Как, основываясь на теоретических знаниях о поведении шума, смещения и разброса, должны измениться эти показатели, если к решающему лесу добавить еще одно дерево? Выберите все подходящие варианты ответа.

Решение:

Разброс не изменится

Смещение увеличится

Смещение уменьшится

Разброс увеличится

Разброс уменьшится

Шум не изменится

Шум увеличится

Шум уменьшится

Смещение не изменится

12 +

В выборке 600 объектов положительного класса и 400 объектов отрицательного класса.

Алгоритм бинарной классификации на всех объектах выборки предсказывает одну и ту же вероятность попасть в положительный класс, то есть

$a(x) = \beta$, где β - некоторое число из отрезка $[0; 1]$

Чему равен ROC-AUC алгоритма, посчитанный на этой выборке? Ответ округлите до десятых.

Решение:

Ответ:

0.5

13 ?

Дана выборка для задачи регрессии:

x	y
3	10
2	5
5	15
6	14

Эта задача решается при помощи решающего дерева.

Разбиения в дереве ищутся по условию $x > \theta$, где θ - такой порог, для которого значение Information Gain (IG) максимально.

$IG = H(R) - |Rl||R|H(Rl) - |Rr||R|H(Rr)$, где $H(A)$ - несмещенная дисперсия целевой переменной в вершине A.

Посчитайте Information Gain для разбиения по условию $x > 5$.

Ответ округлите до десятых.

Решение:

Ответ:

14 +

Алгоритм word2vec кодирования текстов закодировал словосочетания “бутерброд с колбасой”, “бутерброд без колбасы”, “бутерброд с сыром” векторами $a = (1, 2, 3, 2, 4), b = (0.5, 2, 2.5, 2, 4)$ и $c = (5, 2, 2, 3, 4)$ соответственно.

Известно, что операции над полученными word2vec-векторами сохраняют смысл текстов, то есть, например, (вектор слова “король”) + (вектор слова “женщина”) = (вектор слова “королева”).

Найдите вектор словосочетания “бутерброд без сыра”. В ответ запишите квадрат его длины (евклидовой нормы).

Решение:

$c - (a - b) = [4.5, 2., 1.5, 3., 4.]$

$4.5^2 + 2^2 + 1.5^2 + 3^2 + 4^2 = 51.5$

Ответ:

51.5

15 +

В проекте по машинному обучению для обработки данных используются три скрипта. Первый скрипт обрабатывает 40% данных, второй - свои 35% данных, третий - оставшиеся 25% данных. Вероятность некорректной работы первого скрипта составляет 0.04, второго - 0.06, а третьего - 0.03.

В результате работы скриптов на некотором объекте в данных обнаружена ошибка.

Какова вероятность того, что ошибка допущена вторым скриптом?

Ответ округлите до сотых.

Решение:

P(1)=0.4

P(2)=0.35

P(3)=0.25

P(f|1)=0.04

P(f|2)=0.06

P(f|3)=0.03

P(2|f)=P(f|2)*P(2)/P(f)=0.06∗0.35/(0.04∗0.4 + 0.06∗0.35 + 0.03∗0.25)=0.47

Ответ:

0.47

Инфо

В файлах train.csv и test.csv находятся данные о мобильных телефонах.

В этом задании предлагается изучить характеристики мобильных телефонов, предоставленные в csv-файлах. Затем на основе характеристик мобильных телефонов построить модель, предсказывающую ценовую категорию устройства.

Описание столбцов:

- Battery_power - мощность батарейки телефона
- Blue - наличие bluetooth
- Clock_speed - скорость микропроцессора
- Dual_sim - есть ли слот для второй сим-карты
- Fc - разрешение фронтальной камеры
- Four_g - есть ли 4G
- Int_memory - размер встроенной памяти
- M_dep - ширина телефона
- Mobile_wt - вес телефона
- N_cores - число ядер процессора
- Pc - разрешение основной камеры
- Px_height - разрешение по высоте в пикселях
- Px_width - разрешение по ширине в пикселях
- Ram - размер памяти RAM
- Sc_h - высота экрана
- Sc_w - ширина экрана
- Talk_time - наибольшее время работы в режиме разговора

- Three_g - есть ли 3G
- Touch_screen - есть ли touch screen
- Wifi - есть ли wifi

Целевая переменная:

- Price_range

16 +

0,25

В скольких столбцах таблицы среднее значение измеряется в сотнях? (не в десятках и не в тысячах).

Решение:

```
import pandas as pd
import numpy as np

train = pd.read_csv('2_train.csv')
test = pd.read_csv('2_test.csv')

train.describe()
```

Ответ:

2

17 +

0,30

Если у телефона нет ни опции 3G, ни опции 4G, ни Wifi - то у пользователей нет доступа к интернету. Создайте колонку is_internet и поставьте туда 0, если у телефона нет ни одной из трех перечисленных опций, и 1 иначе.

Такую же колонку создайте в тестовых данных.

Какая доля телефонов (из train.csv) не имеет доступа к интернету? Ответ округлите до сотых.

Решение:

```
train['is_internet']=((train['three_g'] == 1) | (train['four_g'] == 1) | (train['wifi'] == 1))
train['is_internet']=train['is_internet'].astype(int)

test['is_internet']=((test['three_g'] == 1) | (test['four_g'] == 1) | (test['wifi'] == 1))
test['is_internet']=test['is_internet'].astype(int)

len(train[train['is_internet']==0])/len(train)
```

Ответ:

0.12

18 -

0,42

Какой столбец имеет наибольшую по модулю корреляцию Пирсона с целевой переменной price_range? В ответ запишите коэффициент корреляции (со знаком), округленный до сотых.

Решение:

```
train.corrwith(train['price_range'], numeric_only=True)
```

Ответ:

ram

+0.92

19 +

0,42

Среди телефонов ценовой категории 2 (price_range = 2) вычислите долю тех, которые не имеют bluetooth. Ответ округлите до сотых.

Решение:

```
len(train[(train['price_range']==2) & (train['blue']==0)]) / len(train[train['price_range']==
2])
```

Ответ:

0.52

20 +

0,42

Среди телефонов с talk_time < 10 насколько больше телефонов с опцией dual_sim = 1, чем с dual_sim = 0?

Решение:

```
len(train[(train['talk_time']<10) & (train['dual_sim'] == 1)]) - len(train[(train['talk_time']
<10) & (train['dual_sim'] == 0)])
```

Ответ:

48

21 +

0,30

Разбейте тренировочные данные на целевой вектор y, содержащий значения из столбца price_range, и матрицу объект-признак X, содержащую остальные признаки.

Посмотрите, сколько значений содержит категориальный столбец touch_screen. Значения, встречающиеся меньше, чем в 1% строк, замените на самое частое значение. После этого закодируйте столбец при помощи OneHot-encoding (используйте аргумент drop='first'). Обучайте OHE на тренировочных данных X, но кодируйте при этом и тренировочные, и тестовые данные. Для кросс-валидации лучше использовать функцию cross_val_score из библиотеки sklearn.model_selection.

Сколько изначально различных значений было в столбце touch_screen?

Решение:

```
X_train=train.drop('price_range', axis = 1)
y_train=train['price_range']

X_train['touch_screen'].unique()
len(X_train[X_train['touch_screen']=='-'])/len(X_train)
X_train['touch_screen'].replace('-', 'Yes', inplace=True)
```

```
test['touch_screen'].replace('-', 'Yes', inplace=True)


# encoding
from sklearn.preprocessing import OneHotEncoder
encoder = OneHotEncoder(drop='first', sparse_output=False)

one_hot_encoded = encoder.fit_transform(X_train[['touch_screen']])
one_hot_df = pd.DataFrame(one_hot_encoded, columns=['touch_screen'], dtype=int)
X_train_trans = X_train.copy()
X_train_trans['touch_screen']=one_hot_df

one_hot_encoded = encoder.transform(test[['touch_screen']])
one_hot_df = pd.DataFrame(one_hot_encoded, columns=['touch_screen'], dtype=int)
test_trans = test.copy()
test_trans['touch_screen']=one_hot_df

#или
X_train_trans = pd.get_dummies(X_train, columns = ['touch_screen'], drop_first=True, dtype=in
t)
X_train_trans['touch_screen'] = X_train_trans['touch_screen_Yes']
X_train_trans.drop(columns=['touch_screen_Yes'], inplace=True)
```

Ответ:

3

22 +

0,40

Обучите на этих данных логистическую регрессию из sklearn (LogisticRegression) с параметрами по умолчанию. Выведите среднее значение метрики f1_score с вариантом усреднения ‘weighted’ (или же 'f1_weighted') алгоритма на кросс-валидации с тремя фолдами. Ответ округлите до сотых.

При объявлении модели задайте 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

lr = LogisticRegression(random_state = 42)

lr.fit(X_train_trans, y_train)

scores = cross_val_score(lr, X_train_trans, y_train, cv=3, scoring="f1_weighted")
scores.mean()
```

Ответ:

0.63

23 +

0,25

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_weighted. Остальные параметры оставьте по умолчанию. В ответ запишите наилучшее среди искомых значение C.

При объявлении модели задайте 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=1*

Решение:

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

# 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_weighte
d')

# Обучаем GridSearchCV на тренировочных данных
grid_search.fit(X_train_trans, y_train)

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

Ответ:

1

24 +

0,50

Добавьте в тренировочные и тестовые данные новый признак 'ram_resolution', вычисляемый по формуле ram * (px_height + px_width)

На тренировочных данных с новым признаком заново с помощью GridSearchCV (с тремя фолдами и метрикой качества - f1_weighted) подберите оптимальное значение C (перебирайте те же значения C, что и в предыдущих заданиях), в ответ напишите наилучшее качество алгоритма по метрике f1_weighted (можно посмотреть, обратившись к полю best_score_ обученного GridSearchCV), ответ округлите до сотых.

Комментарий: параметры по умолчанию предполагаются следующими

penalty='l2', *dual=False*, *tol=0.0001*, *fit_intercept=True*, *intercept_scaling=1*, *class_weight=None*, *random_state=None*, *solve*

Решение:

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

```
# 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_weighte
d')

# Обучаем GridSearchCV на тренировочных данных
grid_search.fit(X_train_trans, y_train)

# Выводим наилучшее значение C
print("наилучшее качество алгоритма:", grid_search.best_score_ )
```

Ответ:

0.69

25 +

0,75

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 f1_weighted на тестовых данных.

Качество проверяется на представленных тестовых данных.

- f1_weighted >= 0.88 - 0.25 балла
- f1_weighted >= 0.93 - 0.75 балла

Пример файла для отправки результатов: result.csv

Решение:

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

# Масштабирование признаков (важно для логистической регрессии)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train_trans)
X_test_scaled = scaler.transform(test_trans)

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

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

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

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

Ответ: