

Методические рекомендации для решения конкурсных материалов для проведения практического этапа Московского конкурса межпредметных навыков и знаний «Интеллектуальный мегаполис. Потенциал» в номинации «ИТ-класс в московской школе» по направлению «Большие данные»

1. Назначение конкурсных материалов

Материалы практического этапа Московского конкурса межпредметных навыков и знаний «Интеллектуальный мегаполис. Потенциал» (далее – Конкурс) предназначены для оценки уровня практической подготовки участников Конкурса.

2. Условия проведения

Практический этап Конкурса проводится в форме компьютерного тестирования дистанционно. При выполнении работы обеспечивается строгое соблюдение порядка организации и проведения Конкурса.

3. Продолжительность выполнения

На выполнение заданий практического этапа Конкурса отводится **90 минут**.

4. Содержание и структура

Индивидуальный вариант участника включает 13 заданий, базирующихся на содержании специализированной дисциплины Конкурса. Задания с развернутым ответом передаются на проверку экспертной комиссии.

В каждом варианте содержится 13 заданий. Вариант разработан на базе элективного курса «Большие данные» - https://profil.mos.ru/images/news/14_10_2020_K/rp_spec_big_data.pdf.

5. Система оценивания отдельных частей и работы в целом

Задание считается выполненным, если ответ участника совпал с эталоном. Максимальный балл за выполнение заданий:

- базовый уровень сложности – 3 балла;
- повышенный уровень сложности – 6 балла.

Задание считается выбранным, если на него дан ответ. Участник может изменить свой выбор в процессе выполнения работы путём удаления ответа к одному заданию и сохранения ответа к другому заданию.

Для получения максимального балла на практическом этапе Конкурса предпрофессиональных умений необходимо набрать в сумме за 13 заданий 60 баллов.

Приложение №1. Обобщённый план конкурсных материалов для проведения практического этапа Конкурса

Приложение №2. Демонстрационный вариант конкурсных заданий практического этапа Конкурса с разбором вариантов решения.

**Обобщённый план конкурсных материалов для проведения
практического этапа Конкурса**

№ задания	Уровень сложности	Темы элективных курсов	Контролируемые требования к проверяемым умениям	Балл
1	Базовый	Введение в машинное обучение	Знание и понимание основных понятий; типологий методов машинного обучения; численных методов нахождения лучшего решения.	3
2	Базовый	Введение в Python. Базовые операции.	Знание и понимание типов и структур данных, а также операций над ними.	3
3	Базовый	Введение в SQL. Примеры в PostgreSQL. Текущий контроль.	Умение формировать SQL запросы.	3
4	Базовый	Библиотека pandas. Примеры. Текущий контроль.	Умение построения и исследования моделей из различных предметных областей с использованием выбранных языков программирования и электронных таблиц.	3
5	Базовый	Вектора и матрицы. Текущий контроль.	Знание табличных величин, массивов данных	3
6	Базовый	Вектора и матрицы. Текущий контроль.	Знание табличных величин, массивов данных.	3
7	Повышенный	Библиотека numpy. Примеры.	Умение построения и исследования моделей из различных предметных областей с использованием выбранных языков программирования и электронных таблиц.	6
8	Повышенный	Проектирование баз данных. Цели проектирования. Нормализация данных. Проектирование базы данных в PostgreSQL.	Умение формировать SQL запросы.	6

9	Повышенный	Обучение с подкреплением. Текущий контроль	Знание и изучение методов решения задачи классификации, методов решений задач регрессии.	6
10	Повышенный	Введение в нейронные сети	Знание и понимание основных понятий; типологий методов машинного обучения; численных методов нахождения лучшего решения.	6
11	Повышенный	Введение в Python. Базовые операции.	Знание и изучение синтаксиса и семантики выбранного языка программирования.	6
12	Повышенный	Введение в Python. Базовые операции.	Знание и изучение переменных и их описание.	6
13	Повышенный	Текстовые и бинарные форматы хранения данных json, csv, parquet. Текущий контроль	Знание и понимание основных понятий; текстовых и бинарных форматов хранения данных.	6
ИТОГО				60

Демонстрационный вариант конкурсных заданий практического этапа Конкурса с разбором вариантов решения

1. Тензор с каким рангом создает строка “`t = tf.constant([[1, 2], [3, 4], [5, 6]], dtype=tf.float16)`”?

Ответ: 2

Пояснение: вопрос охватывает тему «Основные понятия. Типология методов машинного обучения. Численные методы нахождения лучшего решения.» Для решения необходимо знать библиотеку глубокого обучения Tensorflow. В примере создается константный (постоянный) объект тензора, с типом данных float(16). Ранг тензора - это количество его измерений. Синонимами ранга являются порядок, степень, размерность. Обратите внимание, что ранг в TensorFlow это не то же самое, что и ранг матрицы в математике. Ранг в Tensorflow соответствует некоторой математической сущности:

Ранг	Математическая сущность
0	Скаляр (только величина)
1	Вектор (величина и направление)
2	Матрица (таблица чисел)
3	3-Тензор (куб чисел)

2. Какой аргумент функции range() в Python необходимо использовать, для того, чтобы она вернула следующую последовательность: 0, 1, 2, 3, 4?

Ответ: 5

Пояснение: range — это встроенная функция Python, которая возвращает итерируемый объект (range object), содержащий целые числа. С помощью функция range() можно сгенерировать последовательность чисел с

определенным. Функция может принимать от одного до трех аргументов. При этом аргументами должны быть целые числа (int).

Синтаксис функции - range(старт,стоп,шаг). Первый аргумент – начальное значение последовательности, второй – конечное значение последовательности (не включая его), третий – шаг с которым выводится последовательность.

В примере мы можем использовать только один аргумент. Вызывая range() с одним аргументом, получаем ряд чисел, начинающихся с 0 и включающих каждое число до, но не включая число, которое обозначили как конечное (стоп). Поэтому в качестве аргумента мы должны ввести число 5.

3. Какой символ в SQL выводит все значения таблицы?

Ответ: *

Пояснение: вопрос относится к теме «Введение в SQL. Примеры в PostgreSQL. Текущий контроль.» Для того, чтобы вывести **все данные** из любой таблицы, можно использовать символ «*», который буквально означает «все столбцы».

4. Заданы следующие списки

a = [1,2,3]

b = a

Что выведет функция print(a is b)?

Ответ: TRUE

Пояснение: Ключевое слово is в Python используется для проверки того, совпадают ли ссылки памяти двух объектов. Оператор is принимает два операнда и возвращает True, если оба объекта имеют одинаковую ссылку на память, и False, если нет. В данном примере оба объекта ссылаются на один участок памяти. Если бы объект b создавался следующим образом –

b=[1,2,3], то оператор is выдал бы ответ FALSE, т.к. объекты ссылаются на разные участки памяти.

5. Задан следующий код на языке Python

```
def add_three_point(x):  
    return x + 3  
li = [1,2,3]  
list(map(add_three_point, li))
```

что будет результатом его работы?

Ответ: [4, 5, 6]

Пояснение: Вопрос направлен на понимание работы встроенной в Python функции map. Функция map() используется для применения функции к каждому элементу итерируемого объекта (например, списка или словаря) и возврата нового итератора для получения результатов. Она возвращает объект (итератор), который перебирает значения, применяя функцию к каждому элементу. В указанном примере функция позволила добавить к каждому элементу в списке 3, за счет обращения к функции add_three_point(x).

6. После выполнения представленного кода на языке Python с использованием библиотеки pandas, какое значение будут содержать все ячейки дата фрейма df1 (введите число)?

```
import pandas as pd  
df1 = pd.DataFrame({'a':[0,0,0], 'b': [1,1,1]})  
df2 = df1  
df2['a'] = df2['a'] + 1  
df1.head()
```

Ответ: 1

Пояснение: pandas - программная библиотека на языке Python для обработки и анализа данных. Работа pandas с данными строится поверх библиотеки NumPy, являющейся инструментом более низкого уровня. Предоставляет

специальные структуры данных и операции для манипулирования числовыми таблицами и временными рядами.

При выполнении представленного кода можно обнаружить, что `df1` изменен. Это потому, что `df2 = df1` не делает копию `df1` и присваивает ее `df2`, а устанавливает указатель, указывающий на `df1`. Таким образом, любые изменения в `df2` приведут к изменениям в `df1`. В программном коде к каждому элементу столбца «а» добавляется единица, что приводит к тому, что все ячейки в `df1` становятся равными единице. Чтобы это исправить, можно сделать либо

```
df2 = df1.copy (),  
либо  
from copy import deepcopy  
df2 = deepcopy(df1)
```

7. Какое значение будет находится в ячейке первой строчки третьего столбца в дата фрейме `df` после выполнения указанного программного кода на языке Python?

```
import pandas as pd  
import numpy as np  
df = pd.DataFrame({'id': [1,2,3], 'c1':[0,0,np.nan], 'c2': [np.nan,1,1]})  
df = df[['id', 'c1', 'c2']]  
df['num_nulls'] = df[['c1', 'c2']].isnull().sum(axis=1)  
df.head()
```

Ответ: 1 (единица)

Пояснение: при построении моделей часто надо исключить строку с большим количеством пропущенных значений или строки со всеми пропущенными значениями. Используя `.isnull()` и `.sum()` для подсчета количества пропущенных значений в указанных столбцах можно решить указанную задачу. Создавая dataframe строчкой `df = pd.DataFrame({'id': [1,2,3], 'c1':[0,0,np.nan], 'c2': [np.nan,1,1]})`, мы можем получить следующий результат

	id	c1	c2
0	1	0.0	NaN
1	2	0.0	1.0
2	3	NaN	1.0

После выполнения строчки `df['num_nulls'] = df[['c1', 'c2']].isnull().sum(axis=1)`, мы подсчитали нулевые элементы каждой строки и добавили их как отдельный столбец. Результат выглядит следующим образом:

	id	c1	c2	num_nulls
0	1	0.0	NaN	1
1	2	0.0	1.0	0
2	3	NaN	1.0	1

Теперь остается посмотреть первую строчку третьего столбца и записать ее значение, это единица.

8. Что будет являться результатом выполнения запроса, представленного в следующем фрагменте кода на языке Python? Ответ запишите в виде слова.

```
import sqlite3
connection = None
connection = sqlite3.connect('./test.db')
cursor = connection.cursor()
cursor.execute("CREATE TABLE if not exists cars (id INT, name TEXT, price INT)")

cursor.execute("INSERT INTO cars VALUES(1,'Audi',52642)")
cursor.execute("INSERT INTO cars VALUES(2,'Mercedes',57127)")
cursor.execute("INSERT INTO cars VALUES(3,'Skoda',9000)")
cursor.execute("INSERT INTO cars VALUES(4,'Volvo',29000)")
cursor.execute("INSERT INTO cars VALUES(5,'Bentley',350000)")
cursor.execute("INSERT INTO cars VALUES(6,'Citroen',21000)")
cursor.execute("INSERT INTO cars VALUES(7,'Hummer',41400)")
cursor.execute("INSERT INTO cars VALUES(8,'Volkswagen',21600)")

cursor.execute("SELECT name FROM cars WHERE price=(SELECT MIN(price)
FROM cars);")
data = cursor.fetchone()
```

Ответ: Skoda

Пояснение: Вопрос оценивает знания в области баз данных и запросов на языке SQL. Первой строчкой импортируется библиотека `sqlite3`, представляющая собой встроенную в Python СУБД. Далее создается объект `connection` и следующей строчкой создается тестовая база `test.db`. Далее создается объект `cursor` для работы с базой данных. Метод `cursor.execute()` выполняет команду SQL. Команды SQL могут быть параметризованными, то есть передаются заполнители вместо литералов SQL.

Далее создается таблица со следующими столбцами: Идентификатор, Название, Цена и происходит заполнение таблицы марками автомобилей и их ценой. В результате получаем таблицу следующего вида:

	id	name	price
0	1	Audi	52642
1	2	Mercedes	57127
2	3	Skoda	9000
3	4	Volvo	29000
4	5	Bentley	350000
5	6	Citroen	21000
6	7	Hummer	41400
7	8	Volkswagen	21600

Далее выполняется запрос на выбор данных, в результате которого необходимо выбрать поле столбца «name», таблицы «cars», в которой стоимость («price») минимальная из всех. Таким образом мы должны выбрать автомобиль с минимальной ценой и вывести его название – Skoda.

```
cursor.execute("SELECT name FROM cars WHERE price=(SELECT MIN(price) FROM cars);")
```

9. Какое значение необходимо вставить вместо символов «__» в код на Python, представленный ниже для того, чтобы размер обучающей выборки был равен 80% от исходных данных?

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=__ , random_state=0)
```

Ответ: 0.2

Пояснение: Вопрос направлен на понимание сути машинного обучения и использование библиотеки Scikit-learn. В примере использован модуль `train_test_split` библиотеки Scikit-learn, который очень полезен для разделения датасетов. Конечно, можно выполнить такие разбиения каким-либо другим способом (возможно, используя только Numpy). Библиотека Scikit-learn включает полезные функции, позволяющие сделать это немного проще.

Для того, чтобы размер обучающей выборки был равен 80%, необходимо, чтобы тестовая выборка была 20%. В связи с этим мы должны указать в качестве параметра размерности тестовой выборки **0.2**.

10. Чему будет равен результат выполнения следующего кода на языке Python?

```
import tensorflow as tf
sess = tf.InteractiveSession()
a = tf.placeholder("float")
b = tf.placeholder("float")
y = tf.multiply(a, b)
print(sess.run(y, feed_dict={a:100, b:500}))
```

Ответ: 50000.0

Пояснение: Вопрос направлен на понимание принципов работы с библиотекой глубокого обучения Tensorflow.

Первой строчкой производится импорт библиотеки Tensorflow

```
import tensorflow as tf # подключаем TF
```

Далее создается сессия

```
sess = tf.InteractiveSession()
```

Далее создаются плейсхолдеры — объекты, которые параметризуют граф и отмечают места для подстановки внешних значений. Как написано в официальной документации, плейсхолдер — это «обещание подставить значение в будущем». Создадим плейсхолдеры и назначаем им типы данных:

```
b = tf.placeholder("float")
y = tf.multiply(a, b)
```

Далее используем эти плейсхолдеры в качестве входных узлов для сумматора и запустим сессию. В результате получим значение **50000**.

11. Какую последовательность данных выдаст следующий код на Python?

```
filtered = [i for i in range(10) if i%2==0]
print(filtered)
```

Ответ: [0, 2, 4, 6, 8]

12. Что выведет в качестве результата следующий программный код на языке Python?

```
def print_argument(func):
    def wrapper(the_number):
        print("Аргументом для ",
              func.__name__,
              "является", the_number)
        return func(the_number)
    return wrapper
@print_argument
def add_one(x):
    return x + 1
print(add_one(1))
```

Ответ:

Аргументом для функции add_one является 1

2

Пояснение: декораторы — один из самых полезных инструментов в Python, однако новичкам они могут показаться непонятными, потому что они выходят за рамки «обычного» процедурного программирования как в Си, где вы объявляете функции, содержащие блоки кода, и вызываете их. То же

касается и объектно-ориентированного программирования, где вы определяете классы и создаёте на их основе объекты. Декораторы не принадлежат ни одной из этих парадигм и исходят из области функционального программирования.

Декоратор — это функция, которая позволяет обернуть другую функцию для расширения её функциональности без непосредственного изменения её кода.

Здесь `print_argument()` является функцией-декоратором. Она является функцией высшего порядка, так как принимает функцию в качестве аргумента, а также возвращает функцию. Внутри `print_argument()` мы определили другую функцию, обёртку, которая обёртывает функцию-аргумент и затем изменяет её поведение. Декоратор возвращает эту обёртку. Добавив `@print_argument` перед определением функции `add_one(x)` мы изменили её поведение.

13. Какой результат выведет следующий программный код на языке Python?

```
import jmespath

persons = {
    "persons": [
        { "name": "erik", "age": 38 },
        { "name": "john", "age": 45 },
        { "name": "rob", "age": 14 }
    ]
}
jmespath.search('persons[*].age', persons)
```

Ответ: [38, 45, 14]

Пояснение: JMESPath — это язык запросов JSON. JMESPath в Python позволяет легко получать необходимые данные из документа или словаря JSON. Эта библиотека доступна для Python. В первой части кода импортирована библиотека JMESPath и создан объект, который содержит данные о людях и их возрасте. Следующим запросом мы извлекаем возраст всех людей ('persons[*].age') из объекта persons.

Можно было бы написать Python-цикл `for` и перебрать всех людей, но циклы работают медленно и усложняют код. В это случае целесообразнее воспользоваться библиотекой **JMESPath**.