СПИСОК ЗАДАЧ

Приведенные ниже вопросы являются примерными.

Тестовые задания

1. Машинное обучение и анализ данных

1. Вопрос. Какие задачи решают деревья решений?

а) Только классификацию.

б) Только регрессию.

в) Классификацию и регрессию. \* правильный ответ

г) Только анализ временных рядов.

* Дерево решений в машинном обучении — **это модель, которая представляет собой дерево с узлами и листьями**. Узлы дерева представляют собой решения, которые необходимо принимать, а листья — конечные результаты.

2. Вопрос. Что такое дерево решений в контексте машинного обучения?

а) Алгоритм для классификации и регрессии, который представляет

собой структуру в виде дерева, где каждый узел представляет атрибут, а листья

— возможные исходы.

б) Метод анализа данных, основанный на машинном обучении, который

используется для прогнозирования будущих событий.

в) Инструмент визуализации данных, позволяющий наглядно

представить структуру и связи между различными элементами.

г) Статистический метод для анализа влияния различных факторов на

результат.

3. Вопрос. Какой тип визуализации данных используется для

демонстрации корреляции между двумя переменными?

а) Диаграмма рассеяния.

б) Гистограмма.

в) Линейный график.

г) Круговая диаграмма

4. Вопрос. Какой тип визуализации данных лучше всего подходит для

отображения распределения числовых данных?

а) Столбчатая диаграмма.

б) Гистограмма.

в) Линейный график.

г) Круговая диаграмма.

5. Вопрос. Какой метод анализа данных применяется для разбиения

данных на группы на основе сходства?

а) Регрессионный анализ.

б) Кластеризация.

в) Дисперсионный анализ.

г) Корреляционный анализ.

6. Вопрос. Какой метод анализа данных используется для определения

взаимосвязи между двумя переменными?

а) Регрессионный анализ.

б) Кластеризация.

в) Дисперсионный анализ.

г) Корреляционный анализ.

7. Вопрос. Какое ключевое слово используется в Python для создания

конструктора?

а) \_\_init\_\_.

б) \_\_class\_\_.

в) \_\_main\_\_.

г) \_\_name\_\_.

8. Вопрос. Какой метод в Python отвечает за удаление элемента списка с

указанной позицией?

а) pop().

б) remove().

в) delete().

г) clear().

9. Вопрос. Какие типы JOIN существуют в SQL? – в каком из sql?

а) INNER JOIN, LEFT JOIN, RIGHT JOIN.

б) CROSS JOIN, FULL JOIN, SEMI JOIN.

в) OUTER JOIN, INNER JOIN, CROSS JOIN.

г) Все вышеперечисленные типы JOIN.

Inner join, left join, right join, cross join, outer join, full join –это точно

Про semi join не уверен

10. Вопрос. Что делает команда SELECT в SQL?

а) Создаёт новую таблицу.

б) Удаляет данные из таблицы.

в) Выбирает данные из таблицы.

г) Изменяет структуру таблицы.

11. Вопрос. Какой из видов машинного обучения основывается на

взаимодействии обучаемой системы со средой? - неуверен

а) Обучение с учителем

б) Глубинное обучения

в) Обучение без учителя

г) Обучение с подкреплением

12. Вопрос. Нейросети хорошо проявляют себя не только в

распознавании, но и в генерации изображений. Но кое с чем у них все-таки

возникают проблемы. С чем именно? – ну тут необьективно

а) Цвет

б) Форма

в) Глубина, количество пикселей

г) Текстуры

источник: <https://vk.com/@-217715624-neironnye-seti-i-mashinnoe-obuchenie>

13. Вопрос. У машинного обучения есть ряд задач. Как называется та,

что направлена на предсказание значения той или иной непрерывной числовой

величины для входных данных?

а) Переобучение

б) Классификация

в) Регрессия

г) Кластеризация

14. Вопрос. Искусственные нейронные сети (ИНС) — модели

машинного обучения, использующие комбинации распределенных простых

операций, зависящих от обучаемых параметров, для обработки входных

данных. Какого вида ИНС не существует?

а) Импульсные

б) Состязательные

в) Наивные

г) Рекуррентные

15. Вопрос. На чем основана модель случайного леса? - неуверен

а) На ансамбле деревьев

б) На ансамбле графов

в) На паре деревьев

г) На матрице смежности для ансамбля графов

16. Вопрос. Что является результатом прогноза логистической

регрессии?

а) Численное значение целевой переменной

б) Вероятность возникновения события

в) Погрешность возникновения некоторого события

г) Коэффициенты линейной функции, описывающей закон

возникновения события

Результатом прогноза логистической регрессии является **вероятность, что величина примет одно из двух бинарных значений**.

17. Вопрос. Что из перечисленного относится к технологиям ИИ?

а) CFD-расчёты

б) СУБД

в) 3D-печать

г) Обработка естественного языка

CFD (Computational Fluid Dynamics) — **это расчётный метод, позволяющий моделировать и изучать потоки жидкостей и газов**. [1](https://strana-rosatom.ru/2017/10/30/zanimatelnaya-gidrodinamika/)

В расчётах учитываются такие параметры, как передача тепла или массы, фазовые изменения веществ, химические реакции, механическое движение, напряжение и деформация твёрдых материалов под воздействием среды. [1](https://strana-rosatom.ru/2017/10/30/zanimatelnaya-gidrodinamika/)

**Результат CFD-расчёта даёт инженерные данные** для концептуальных разработок, детальной проработки изделия, устранения неполадок и внесения изменений в конструкцию. [2](https://kz-engineering.com/wp-content/uploads/2023/12/Vozmozhnosti-ANSYS_CFD.pdf)

**Примеры применения CFD-расчётов**:

* расчёт параметров камер сгорания ракетных двигателей; [1](https://strana-rosatom.ru/2017/10/30/zanimatelnaya-gidrodinamika/)
* физико-химические процессы при обтекании головных частей боеголовок и сверхзвуковых самолётов; [1](https://strana-rosatom.ru/2017/10/30/zanimatelnaya-gidrodinamika/)
* моделирование процессов в реакторных установках; [1](https://strana-rosatom.ru/2017/10/30/zanimatelnaya-gidrodinamika/)
* расчёт поведения среды в насосах, реакторах, парогенераторах, даже аэродинамики автомобиля или самолёта.

18. Вопрос. Что из перечисленного является open-source компонентом?

а) SAS -  **не является открытым исходным кодом**. Это программное обеспечение с закрытой средой, которое не поддерживает открытый исходный код

б) Terradata -  **Teradata является open-source компонентом**. [1](https://www.guru99.com/teradata-tutorial.html)

Это система управления базами данных для разработки крупномасштабных приложений для хранилищ данных.

в) Matplotlib - **Matplotlib является open-source библиотекой** для Python.

г) Power BI – **PowerBI - проприетарный софт от Microsoft, и, не смотря на то, что он умеет работать с самописными open-source модулями, сам по себе он open-source не становится**

19. Вопрос. Как расшифровывается ELT?

а) Extract-List-Transform

б) Extract-Load-Translate

в) Extract-Load-Transform

г) Excel-Load-Trim

20. Вопрос. Какой из перечисленных признаков НЕ является

интервальным?

а) Рост

б) Давление

в) Марка автомобиля

г) Вес

Из перечисленных признаков, марка автомобиля (в) НЕ является интервальным признаком.

Интервальные признаки предполагают наличие числовых значений, которые можно измерять и сравнивать, в то время как марка автомобиля — это категориальная переменная, которая не имеет числового значения и не может быть упорядочена в интервальном формате. Остальные признаки (рост, давление, вес) являются интервальными, так как они количественные и имеют равные интервалы между значениями.

21. Вопрос. из перечисленного НЕ относится к процедурам улучшения

качества данных?

а) "Золотая запись"

б) Идентификационный анализ

в) Бининг

г) Извлечение

Из перечисленного к процедурам улучшения качества данных не относится г) Извлечение.

"Золотая запись", идентификационный анализ и бининг - это подходы и методы, направленные на улучшение качества данных, тогда как извлечение в основном относится к процессу получения данных из различных источников.

2. Алгоритмы и струкутры данных

1. Вопрос: Выберите правильное рекуррентное соотношение для

создания последовательности чисел Фибоначчи:

а) F(0) = 0, F(1) = 1, F(n)=F(n-1)+F(n-2)

б) F(0) = 1, F(1) = 1, F(n)=F(n-1)+F(n-2)

в) F(0) = 1, F(1) = 2, F(n)=F(n-1)+F(n-2)

2. Вопрос: Что такое итерация цикла в программировании?

а) Процесс завершения выполнения программы.

б. Процесс повторного выполнения блока кода до тех пор, пока

выполняется заданное условие.

в) Способ хранения данных в памяти.

г) Метод оптимизации алгоритма для повышения его скорости.

3. Вопрос: Как называется структура, которая позволяет организовать

данные в форме дерева, где каждый узел может иметь несколько дочерних

узлов? - несогласен

а) Линейная структура

б) Структура с иерархией

в) Граф

г) Динамический массив

Правильный ответ: б) Структура с иерархией.

Такая структура данных называется древовидной и позволяет организовать данные в форме дерева, где каждый узел может иметь несколько дочерних узлов.

4. Вопрос: Какое десятичное число соответствует восьмеричному числу

17?

а) 13

б) 14

в) 15

г) 16

17(8)=>1\*(81)+7\*(80)=8+7=15

5. Вопрос: Ниже дан фрагмент кода на языке Python. Что будет выведено

на экран?

a = 17

b = 6

while a > 0:

result = a // b

remainder = a % b

a -= b

print(result, remainder)

а) 2, 5

б) 0, 6

в) 0, 5

г) 3, 4

3. Основы SQL

1. Вопрос. Что такое реляционные базы данных:

а) База данных, в которой информация хранится в виде двумерных

таблиц, связанных между собой

б) База данных, в которой одна ни с чем не связанная таблица

в) Любая база данных - реляционная

г) Совокупность данных, не связанных между собой

**Реляционная база данных** — это **тип базы данных, в которой данные организованы в виде связанных таблиц**. Эти таблицы состоят из строк и столбцов, где каждая строка представляет запись (набор связанных данных), а каждый столбец — атрибут (характеристику) данных.

2. Вопрос. Как выглядит запрос, для вывода ВСЕХ значений из таблицы

Orders:

а) select ALL from Orders;

б) select % from Orders;

в) select \* from Orders;

г) select \*.Orders from Orders;

3. Вопрос. Какие данные мы получим из этого запроса?

select id, date, customer\_name from Orders;

а) Неотсортированные номера и даты всех заказов с именами заказчиков

б) Никакие, запрос составлен неверно

в) Номера и даты всех заказов с именами заказчиков, отсортированные

по первой колонке

г) Номера и даты всех заказов с именами заказчиков, отсортированные

по всем колонкам, содержащим слово Order

4. Вопрос. Есть ли ошибка в запросе?

select id, date, customer\_name from Orders where customer\_name = Mike;

а) Запрос составлен правильно

б) Mike необходимо записать в кавычках 'Mike'

в) Нужно убрать лишние поля из запроса

г) Строчку с where поменять местами с from

В запросе

select id, date, customer\_name from Orders where customer\_name = Mike

**есть ошибка**, так как **значение поля customer\_name необходимо указать в кавычках**, так как это текстовое значение

5. Вопрос. Что покажет следующий запрос:

select \* from Orders where date between '2017-01-01' and '2017-12-31' – вопрос не корректный

а) Все данные по заказам, совершенным за 2017 год, за исключением 01

января 2017 года

б) Все данные по заказам, совершенным за 2017 год, за исключением 31

декабря 2017 года

в) Все данные по заказам, совершенным за 2017 год

г) Ничего, запрос составлен неверно – с одной стороны из-за date

6. Вопрос. С помощью какого запроса можно получить пользователей, в

email которых содержится более двух точек?

a) SELECT \* FROM users WHERE "." IN email MATCHES > 2;

b) SELECT \* FROM users WHERE email LIKE "%.%.%.%";

c) SELECT \* FROM users WHERE email COUNT(".") > 2

**b)** **SELECT \* FROM users WHERE email LIKE '%.%.%.%';**

Этот запрос ищет строки, содержащие по крайней мере три точки, что соответствует требованию "содержится более двух точек".

Запросы (a) и (c) содержат синтаксические ошибки и не будут корректно работать в SQL:

* a) **SELECT \* FROM users WHERE "." IN email MATCHES > 2;** — неверный синтаксис.
* c) **SELECT \* FROM users WHERE email COUNT(".") > 2** — тоже неверный синтаксис.

Таким образом, правильный вариант — это b.

7. Вопрос. Какой механизм в реляционных СУБД позволяет обеспечить

согласованность данных, хранящихся в логически связанных таблицах?

а) Первичный ключ;

б) Внешний ключ;

в) Кластерный индекс;

8. Вопрос. Какой оператор позволяет объединить строки, имеющие

одинаковые значения в заданных полях?

а) ORDER BY;

б) WHERE;

в) GROUP BY;

г) FROM;

9. Вопрос. Какой тип соединения двух таблиц позволит вернуть те

строки, для которых в обеих таблицах выполняется условие соединения?

а) LEFT JOIN;

б) FULL JOIN;

в) RIGHT JOIN;

г) INNER JOIN;

10. Вопрос. Какой инструмент SQL позволяет получить результат

обработки одной или нескольких строк, и подставить этот результат в

текущую строку как новый столбец?

а) Оператор GROUP BY;

б) Команды соединения таблиц;

в) Оконные функции;

Правильный ответ — **в) Оконные функции**.

Оконные функции в SQL позволяют выполнять расчеты над набором строк (окном) и подставлять результаты этих расчетов в текущую строку как новый столбец, что очень удобно для аналитической обработки данных.

11. Вопрос. Какой оператор возвращает уникальные строки, выводимые

левым и правым входными запросами?

а) UNION;

б) EXCEPT;

в) INTERSECT;

Правильный ответ: а) UNION.

Оператор `UNION` объединяет результаты двух или более запросов, возвращая уникальные строки из обоих запросов. Если вам нужно получить строки, которые находятся только в одном из запросов (без дублирования), то используйте `UNION`.

Остальные операторы:

- `EXCEPT` возвращает строки, которые есть в первом запросе, но отсутствуют во втором.

- `INTERSECT` возвращает строки, которые есть в обоих запросах.

4. Анализ больших текстовых данных и текстовый поиск

1. Вопрос. Какой правильное определение модели Sequential:

а) модель разбиения текста на отдельные токены/блоки

б) полносвязная модель нейронной сети прямого распространения

в) рекуррентная нейронная сеть

**Модель Sequential** — это **линейный стек слоёв**

Модель Sequential в Keras (библиотека для построения нейронных сетей на Python) представляет собой линейный стек слоев, где каждый слой имеет единственное предшествующее и последующее состояние, что делает его подходящим для создания полносвязных сетей, а также более сложных сетей, состоящих из последовательных слоев.

2. Вопрос. Что на вход получает encoder модели text2text?

а) Векторное представление вопроса

б) Векторное представление ответа

в) Вектор контекста

г) Векторное представление ответа и контекста

В модели text2text encoder получает на вход текстовые данные, которые могут быть использованы для генерации текстов. Конкретно, encoder обычно принимает строку текста (например, вопрос или запрос), которую нужно преобразовать в векторное представление для дальнейшей обработки.

Таким образом, правильный ответ на ваш вопрос — это аналог такого входа, который наиболее соответствует формату:

а) Векторное представление вопроса

Но стоит учесть, что формулировка вопроса может подразумевать, что encoder получает текстление напрямую, которое затем преобразуется в векторное представление.

3. Вопрос. Какие функции активации обычно есть в слое LSTM?

а) Сигмоид

б) Tanh

в) Relu

г) units

4. Вопрос. Какой результатам соответствует преобразованию текста

методом texts\_to\_matrix&

а) [0, 0, 0, 1, 0, 0, 1, 0, 0, 1]

б) [416, 10, 3]

в) Словарь частот

5. Как задача относится к задачам ML без учителя

а) NER

б) Topic Model

в) Text2Text

г) Суммаризация

Задачи, связанные с машинным обучением без учителя (unsupervised learning), включают в себя такие методы, которые не требуют размеченных данных для обучения. Рассмотрим предложенные варианты:

а) \*\*NER (Named Entity Recognition)\*\* — это задача, как правило, относящаяся к обучению с учителем, поскольку для её успешного выполнения требуется размеченный набор данных с аннотированными именованными сущностями.

б) \*\*Topic Model\*\* — это задача, которая относится к обучению без учителя. Она позволяет выявлять скрытые темы в тексте, используя неразмеченные данные.

в) \*\*Text2Text\*\* — это задача, которая чаще всего требует размеченных данных, так как она подразумевает преобразование одного текста в другой (например, перевод, перефразирование), и, как правило, используется в контексте обучения с учителем.

г) \*\*Суммаризация\*\* — в зависимости от подхода, эта задача может быть как с учителем (где используются размеченные данные), так и без учителя (где алгоритмы обучаются извлекать основные содержания из неразмеченных текстов).

Таким образом, правильный ответ на вопрос о том, какая из задач относится к обучению без учителя — это \*\*б) Topic Model\*\*.

6. Функция активации Relu это

а) Линейный выпрямитель

б) Моделирование векторного пространства

в) Функция, отключающая нейроны по схеме бернулли

5. Основы веб-программирования

1. Вопрос. Как расшифровывается аббревиатуру HTML?

а) HyperText Markup Language

б) HeadText Modulation Language

в) HeadingText Mode Language

2. Вопрос. Где следует располагать тег title?

а) В теге: head

б) В теге: body

в) В теге: footer

3. Вопрос. Существует ли у тега input закрывающий тег в синтаксисе

HTML?

а) Да

б) Нет

в) Такого тега в стандартах HTML не существует

4. Вопрос. Какого вида селектора не существует?

а) Селектор псевдокласса

б) Сестринский селектор

г) Братский селектор

* **а) Селектор псевдокласса:** Это существующий тип селектора CSS. Примеры: :hover, :active, :focus.
* **б) Сестринский селектор:** Это существующий тип селектора CSS, также известный как смежный селектор. Он выбирает элемент, который является непосредственным соседом другого элемента. Обозначается символом +. Например: h1 + p
* **г) Братский селектор:** Такого термина в CSS не существует. Существует смежный сестринский селектор, который описывает только ближайшего соседа, а также общий сестринский селектор (обозначается символом ~), который выбирает все следующие родственные элементы.

5. Вопрос. Какие ключевые слова используются в JavaScript для того,

чтобы объявить переменную? - \*мы не изучали JS

а) var, let, const

б) byte, short, int, long, float

в) int, short,var

6. Вопрос. Что будет, если значению длины непустого массива

присвоить ноль?

(let arr = [‘test’, ’123’]; arr.length = 0;) - если требуется пример

а) Ошибка

б) Массив станет пустым

в) Ничего не произойдет

Когда вы устанавливаете длину массива в 0, все элементы в массиве удаляются. Это стандартное поведение JavaScript.

7. Вопрос. Верно ли, что на самом деле в JS всё является объектом?

а) Да

б) Нет

**Практические задания**

1. Алгоритмы и струкутра данных, python

Устные задания

1. Сколько всего куриц на планете? - Википедия говорит, что около 22 миллиардов

2. Сколько деревьев в Москве? - примерно 8.5 миллионов

3. Сколько теннисных мячиков поместится в автобус? - какого размера мяч? А автобус какой? Что за вопросы от рекрутеров...

2. Основы SQL

Устные задания

1. Чем отличается left join от inner join

LEFT JOIN и INNER JOIN — это два типа соединений (JOIN) в SQL, которые используются для объединения данных из двух и более таблиц на основе общего поля. Основное отличие между ними заключается в том, как обрабатываются строки, которые не соответствуют условиям соединения.

INNER JOIN:

INNER JOIN возвращает только те строки, которые имеют совпадения в обеих таблицах. Если в одной из таблиц нет соответствующей строки, то она не будет включена в результирующий набор данных.

Например, если у вас есть таблицы A и B, и вы выполняете INNER JOIN, вы получите только те строки, где значения в столбцах, по которым происходит соединение, совпадают в обеих таблицах.

LEFT JOIN (или LEFT OUTER JOIN):

LEFT JOIN возвращает все строки из левой таблицы (первой указанной в запросе) и соответствующие строки из правой таблицы. Если в правой таблице нет соответствующих строк, то в результирующем наборе будут присутствовать строки из левой таблицы с значениями NULL для столбцов правой таблицы.

Это позволяет сохранить все данные из левой таблицы, даже если некоторые из них не имеют соответствующих значений в правой таблице.

2.Чем отличаются union от union all

UNION и UNION ALL — это операторы SQL, которые используются для объединения результатов двух или более запросов, но у них есть важные отличия:

Удаление дубликатов:

UNION: Убирает дубликаты из общего результата. Если в разных наборах данных есть одинаковые строки, они будут представлены только один раз.

UNION ALL: Сохраняет все строки, включая дубликаты. Если в наборах данных одинаковые строки, они будут отображены столько раз, сколько раз они встречаются.

Производительность:

UNION: Обычно работает медленнее, чем UNION ALL, потому что он включает в себя этап удаления дубликатов, что требует дополнительной обработки.

UNION ALL: Работает быстрее, так как не требует такой обработки, как удаление дубликатов.

Использование:

Используйте UNION, когда вам нужно объединить результаты и специально удалить дубликаты.

Используйте UNION ALL, когда важно сохранить все строки, включая дубликаты, и когда производительность является приоритетом.

3. Есть две таблицы данных:

table1

Id value

1 A

2 B

2 B

3 C

table2

Id Value2

1 a

2 b

3 c

Сколько строк вернет запрос

Select \* from table1 inner join table2 on table1.Id = table2.ID

4 строки

4. Есть две переменные

A = null

B= null

Что вернет запрос:

IF A=B

NULL

3. Анализ больших текстовых данных и текстовый поиск

Устаные задания:

1. Что такое большая лингвистическая модель?

GPT

Большая лингвистическая модель — это тип искусственного интеллекта, который использует алгоритмы машинного обучения для анализа и генерации текста. Эти модели обучены на огромных объемах текстовых данных и обладают способностью понимать контекст, структуру языка и семантику.

Основные характеристики большой лингвистической модели:

- Обучение на больших датасетах, что позволяет ей осваивать разнообразные темы и стили.

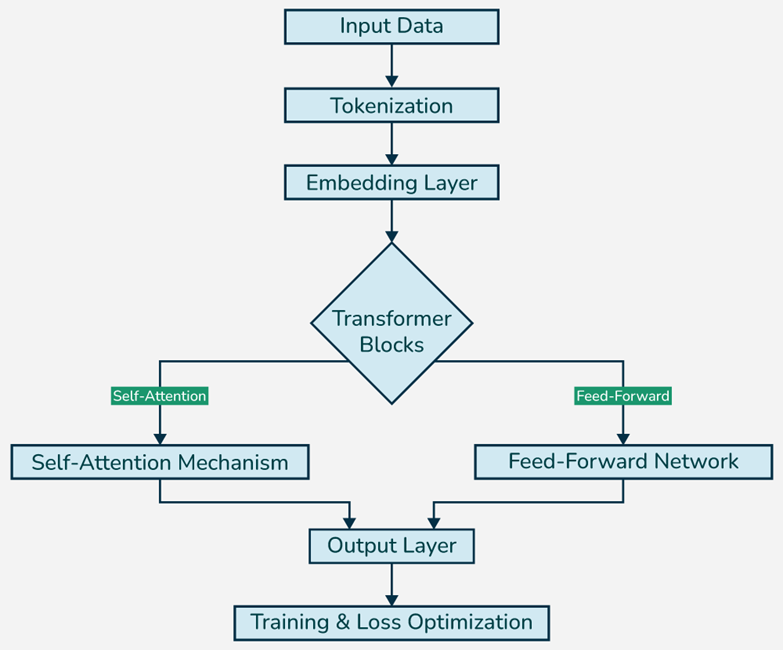
- Способность генерировать осмысленный и связный текст, отвечать на вопросы, переводить и резюмировать информацию.

- Использование архитектур, таких как трансформеры, которые обеспечивают высокую эффективность обработки языковых данных.

Эти модели находят применение в чат-ботах, системах поддержки пользователей, автоматизированном переводе и многих других областях, связанных с обработкой естественного языка.

========================================================================

2. Каковы компоненты архитектуры LLM?



**Входной уровень: Токенизация**

**Токенизация:** входной текст разбивается на более мелкие единицы, называемые токенами, которые могут быть словами, частями слов или символами. Затем эти токены преобразуются в числовые представления (вложения), которые может обрабатывать модель.

Embedding слой

Word Embeddings: каждый токен сопоставляется с плотным вектором в многомерном пространстве, представляющим его семантическое значение. К распространенным методам относятся Word2Vec, GloVe и embedding-ги, полученные в процессе обучения модели.

Positional Embeddings: поскольку трансформеры по своей природе не понимают порядок токенов, к вложениям слов добавляются позиционные вложения, чтобы предоставить модели информацию о расположении токенов в предложении.

Архитектура трансформера

Механизм Self-Attention:

Attention Scores: механизм self-attention вычисляет набор показателей внимания, которые определяют, насколько каждое слово должно быть сосредоточено на других словах в последовательности.

Запрос, ключ и значение (Q, K, V): Это линейные проекции входных векторов, используемых для вычисления внимания. Модель вычисляет релевантность каждого токена по отношению к другим с помощью скалярного произведения векторов запроса и ключа, а затем применяет операцию softmax для получения весовых коэффициентов внимания. Векторы значений затем взвешиваются с помощью этих коэффициентов внимания.

Multi-Head Attention: несколько каналов внимания используются для выявления различных аспектов взаимосвязей между токенами. Каждый канал работает в отдельном подпространстве, а результаты объединяются и проецируются обратно в исходное пространство.

Feedforward Нейронная сеть: После включения механизма внимания выходные данные передаются через нейронную сеть прямой связи (серию плотных слоев с функциями активации), применяемую независимо к каждой позиции.

Нормализация слоёв и остаточные соединения: за каждым подслоем (внимание и прямая связь) следует нормализация слоёв и остаточное соединение, что помогает стабилизировать обучение и позволяет создавать более глубокие сети.

**Укладка слоев**

**Блоки-трансформеры:** Архитектура обычно включает в себя несколько слоёв (или блоков) трансформеров, расположенных друг над другом. Каждый блок состоит из механизма multi-head self-attention и feedforward нейронной сети. Такое расположение позволяет модели изучать сложные иерархические представления данных.

**Выходной уровень: Декодирование**

**Слой Softmax:** конечный слой обычно представляет собой функцию Softmax, которая преобразует выходные данные модели в распределение вероятностей по словарному запасу, позволяя ей выбрать наиболее вероятный следующий токен или заполнить замаскированный токен.

Источник: <https://translated.turbopages.org/proxy_u/en-ru.ru.b3fdb126-67d0c522-07fe50bd-74722d776562/https/www.geeksforgeeks.org/exploring-the-technical-architecture-behind-large-language-models/>

UPD: понял, что из-за переводчика статья не открывается, придётся искать самим по названию( Но статья хорошая, атвечаю.

=================================================================================

3. Какие типы LLM вы знаете?

RNN, LSTM, transformers, CNN, GNN…

Генерация текста (Text-to-Text), Распознавание картинок (Image-to-Text), Генерация звука (Text-to-Audio), Генерация изображений (Text-to-Image).

4. Системы программирования на Python

Устные задания:

1. В чем разница между списком и кортежем?

**Основное различие между списками и кортежами** заключается в том, что **списки являются изменяемыми, а кортежи — нет**.

**Преимущества списков**: списки изменяемы, что делает их гибкими в использовании. **Недостатки**: из-за их изменчивости списки занимают больше памяти, чем кортежи. [1](https://sky.pro/media/razlichiya-mezhdu-spiskami-i-kortezhami-v-python/)

**Преимущества кортежей**: кортежи занимают меньше памяти, чем списки, так как они неизменяемы. Кроме того, они могут использоваться в качестве ключей словаря, в то время как списки — нет.

2. Что будет при делении числа int на 0?

**При делении числа int на 0 в Python произойдёт ошибка**. Интерпретатор выдаст исключение ZeroDivisionError, остановит выполнение кода и выведет в консоль ошибку.

3. Что такое декоратор?

**Декораторы в Python** — это **функции, которые принимают другую функцию в качестве аргумента, добавляют к ней дополнительную функциональность и возвращают функцию с изменённым поведением**. [1](https://ru.hexlet.io/courses/python-functions/lessons/decorators/theory_unit)

Они позволяют изменять поведение функций и классов с помощью добавления или изменения их функциональности без изменения самого кода. [1](https://ru.hexlet.io/courses/python-functions/lessons/decorators/theory_unit)

Декораторы используются для решения множества задач, таких как логирование, кеширование, проверка прав доступа и других.

4. В чем разница между методами экземпляра, класса и статическими

методами в Python?

Класс — это заданный пользователем шаблон или прототип, на основе которого создаются объекты. Классы позволяют объединять данные и функции. Создание нового класса создаёт новый тип объекта, что позволяет создавать новые экземпляры этого типа. К каждому экземпляру класса могут быть привязаны атрибуты для поддержания его состояния. Экземпляры класса также могут иметь методы (определённые в классе) для изменения своего состояния.

Метод экземпляра класса в Python — **это определённая внутри класса функция, которая может вызываться из экземпляра этого класса**. Такие методы **имеют доступ к данным**, связанным с конкретным экземпляром класса, и **могут изменять эти данные**.

Статический метод в Python — **это метод, который связан с классом, а не с его экземплярами**. Чтобы быть вызванным, он не требует создания экземпляра класса и не имеет доступа к экземпляру. [4](https://timeweb.cloud/tutorials/python/staticheskij-metod-python)

Статические методы **используются для выполнения операций**, не требующих доступа к экземпляру класса или его атрибутам. По сути, они являются вспомогательными функциями. [4](https://timeweb.cloud/tutorials/python/staticheskij-metod-python)

Чтобы создать статический метод в Python, необходимо воспользоваться специальным декоратором — **@staticmethod**

5. Машинное обучение и методы обработки данных

Устные задания:

1. Какие метрики качества моделей ML вы знаете?

A screenshot of a black screen

AI-generated content may be incorrect. A screenshot of a computer program

AI-generated content may be incorrect. A screenshot of a black background with white text

AI-generated content may be incorrect.

1. Что такое регуляризация?

Регуляризация в машинном обучении — это метод предотвращения переобучения модели за счёт добавления штрафа за сложность (большие веса) в функцию потерь. Она помогает улучшить обобщающую способность модели на новых данных. A screenshot of a black and white page

AI-generated content may be incorrect.

1. Что такое функция потерь? Как ее можно изменить?

Функция потерь (или функция ошибки, целевая функция) — это метрика, которая оценивает, насколько предсказания модели отличаются от истинных значений. Она определяет, как модель обучается: алгоритм оптимизации (например, градиентный спуск) минимизирует эту функцию. A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

4. Основные этапы разработки ML-моделей

A screen shot of a computer program

AI-generated content may be incorrect.

6. Основы web-программирования

Устные задания:

1. Предназначение HTML, структура гипертекста, правила написания

тегов HTML

<https://practicum.yandex.ru/blog/zachem-nuzhen-html/>

Язык гипертекстовой разметки сайта, или HTML (HyperText Markup Language), — код, помогающий структурировать содержание каждой веб-страницы. С помощью HTML разработчик собирает «скелет». После чего работает с CSS-кодом для стилизации страницы — покрывает «кожей». В финале запускает «кровеносную, пищеварительную и дыхательную системы», программируя на JavaScript.

**Структура HTML-документа** включает в себя следующие основные элементы: [1](https://sky.pro/wiki/html/osnovy-html-struktura-html-dokumenta/)

1. **<!DOCTYPE html>**. Объявление типа документа, указывающее браузеру, что это HTML5-документ. [1](https://sky.pro/wiki/html/osnovy-html-struktura-html-dokumenta/)
2. **<html>**. Корневой элемент, содержащий весь контент страницы. Все остальные элементы HTML-документа должны быть вложены в этот тег. [1](https://sky.pro/wiki/html/osnovy-html-struktura-html-dokumenta/)
3. **<head>**. Элемент, содержащий метаданные о документе, такие как заголовок страницы и ссылки на стили. Здесь также можно указать кодировку документа, подключить внешние стили и скрипты. [1](https://sky.pro/wiki/html/osnovy-html-struktura-html-dokumenta/)
4. **<title>**. Тег, определяющий заголовок страницы, который отображается на вкладке браузера. Этот заголовок также используется поисковыми системами для индексации страницы. [1](https://sky.pro/wiki/html/osnovy-html-struktura-html-dokumenta/)
5. **<body>**. Элемент, содержащий видимый контент страницы. Всё, что отображается на веб-странице, находится внутри этого тега. – туда мы пишем то что мы хотим отобразить на странице

**Правила написания** – все теги должны открываться и закрываться - <p>Hello world</p>, За исключением <input>

Правила написания тегов.

Теги всегда пишутся в треугольных скобках. Сначала идет открывающийся тег, он состоит из треугольных скобок и названия самого тега. Внутри тега находиться какой-либо контент (текст). Далее тег нужно закрыть. Закрывающийся тег выглядит так же как открывающийся, но перед названием тега ставиться слэш.  
 Пример написания тега:

Бывают теги которые не нужно закрывать, для их работы нужно только открыть тег. Таких тегов очень мало, о них мы поговорим позже. Практически все теги необходимо закрывать, если не закрыть тег, который по своим правилам требует закрытия, то это будет грубейшая ошибка. Один не закрытый тег может сделать так, что вся ваша html страница будет отображаться браузером некорректно.

Правило закрытия тегов.

Часто бывает, что внутри тега, помимо текста располагаются еще другие теги. В этом случае теги закрываются в зеркальном порядке, то есть первым будет закрыт тот тег, который был открыт последним.  
 Пример правильного закрытия тегов:

Параметры тегов.

С помощью параметров (атрибутов) тега мы можем задать нужный нам цвет или размер текста, находящегося внутри этого тега. Значение параметра пишется в кавычках.  
 Пример написания тега с параметром (атрибутом):

Параметров у тега может быть несколько. Например цвет и размер. В этом случае параметры указываются через пробел.

2. Структура веб-страницы, теги и содержание мета-данных. Структура

HTML тегов.

<https://developer.mozilla.org/ru/docs/Learn_web_development/Core/Structuring_content/Structuring_documents>

**Метаданные в HTML** — это **данные (информация) о данных**. [1](https://html5css.ru/tags/tag_meta.php) Они не отображаются на странице, но интерпретируются браузерами и поисковыми системами. [2](https://basicweb.ru/html/metadannie.php)

Метаданные включают описание страницы, ключевые слова, автора документа, последнее изменение и другую информацию. Они могут использоваться браузерами (например, чтобы определить, как отображать содержимое или перезагрузить страницу), поисковыми системами (для индексации по ключевым словам) или другими веб-сервисами. [1](https://html5css.ru/tags/tag_meta.php)

Тег <meta> содержит метаданные о документе HTML и всегда размещается внутри тега <head>. [2](https://basicweb.ru/html/metadannie.php)

В общей форме метаданные записываются так: в теге meta указывается атрибут name (имя) и связанный с ним атрибут content (содержимое), в котором описывается какой-либо аспект веб-страницы.

Метаданные в HTML — это **информация, которая описывает содержимое веб-страницы для поисковых систем**. Они не отображаются на странице, но интерпретируются браузерами и поисковыми системами. [1](https://codelab.pro/kurs-po-html-metategi/)[2](https://basicweb.ru/html/metadannie.php)

**Некоторые типы метаданных в HTML**:

* **Тег заголовка (title)**. Определяет заголовок веб-страницы и отображается в поисковой выдаче в виде кликабельного заголовка. [1](https://codelab.pro/kurs-po-html-metategi/)
* **Тег описания (description)**. Содержит краткое описание содержимого веб-страницы. [1](https://codelab.pro/kurs-po-html-metategi/)
* **Тег ключевых слов (keywords)**. Предоставляет поисковым системам список ключевых слов, которые описывают содержимое страницы. [1](https://codelab.pro/kurs-po-html-metategi/)
* **Тег роботов (robots)**. Сообщает поисковым системам, следует ли индексировать страницу или переходить по ссылкам на странице. [1](https://codelab.pro/kurs-po-html-metategi/)
* **Тег автора (author)**. Используется для указания авторства веб-страницы. [1](https://codelab.pro/kurs-po-html-metategi/)
* **Тег кодировки (charset)**. Задает кодировку символов для HTML-документа. Это гарантирует, что веб-страница правильно отображается в разных браузерах и на разных устройствах.

**Типичная структура HTML-тега** включает следующие элементы:

1. **Открывающий тег**. Все теги представляются в угловых скобках, в открывающем, если требуется, перечисляются атрибуты. [1](https://smartiqa.ru/courses/web/lesson-2-html)
2. **Закрывающий тег**. Присутствует не во всех тегах, идентифицируется косой чертой. [1](https://smartiqa.ru/courses/web/lesson-2-html)
3. **Атрибут, свойство**. Название свойства и его значение предпочтительно в двойных кавычках, возможно использовать и одинарные. [1](https://smartiqa.ru/courses/web/lesson-2-html)
4. **Содержимое**. Внутреннее содержимое тега, обычно в виде текста либо других вложенных тегов.

3. Принципы интеграции HTML с каскадными таблицами стилей и

кодом JavaScript

**Некоторые способы подключения CSS к HTML-документу**:

* **Подключение внешней таблицы стилей**. Для этого создают файл с расширением .css и названием на английском языке (например, main.css или style.css). Затем загружают его на сайт и подключают к HTML-документу в секцию HEAD. [4](https://www.pandoge.com/stati-i-sovety/kak-podklyuchit-css-kaskadnye-tablicy-stiley-k-html)
* **Подключение внутренней таблицы стилей**. В этом случае нет файла стилей, а свойства таблицы вставляют на страницу. Конструкцию можно вставить в секцию HEAD или BODY, при этом на странице в разных её частях может быть несколько таких конструкций. [4](https://www.pandoge.com/stati-i-sovety/kak-podklyuchit-css-kaskadnye-tablicy-stiley-k-html)
* **Подключение встроенной таблицы стилей**. Такой вариант применим ко всем элементам веб-страницы. Если во внешних или внутренних стилях не используют свойство «!important», то встроенное свойство является приоритетным по исполнению.

Делается это либо через styles внутри тега, либо через

<link href=»URL\_адрес\_CSS\_файла» rel=»stylesheet» type=»text/css»>

Код js подключается через <script></stript>

<script type="module" src="main.js"></script>

4. Технология разработки веб-приложений и сайтов МодельПредставление-Контроллер (MVC), причины необходимости применения и

предназначение каждого блока.

MVC (Model-View-Controller) — это архитектурный шаблон для

разработки программного обеспечения, особенно веб-приложений и сайтов.

**MVC (Модель-Представление-Контроллер)** — это подход к организации кода, основанный на принципе разделения ответственности между тремя компонентами: моделью, представлением и контроллером. [5](https://elbrusboot.camp/blog/chto-takoie-mvc/)

**Причины необходимости применения MVC**:

* **Упрощение разработки и поддержки**. Разделение приложения на компоненты позволяет вносить изменения в один из них, не затрагивая остальные. Это упрощает обслуживание и снижает вероятность ошибок. [5](https://elbrusboot.camp/blog/chto-takoie-mvc/)
* **Повышение читаемости кода**. Код организован по функциям каждого конкретного блока, что облегчает его понимание и доработку. [1](https://education.yandex.ru/journal/model-view-controller)
* **Организация работы разных программистов**. Каждый занимается своим компонентом, при этом разработчик не должен вникать в чужой код. [2](https://blog.skillfactory.ru/glossary/mvc/)

**Предназначение каждого блока MVC**:

1. **Модель (Model)**. Это основная логика приложения. Отвечает за данные, методы работы с ними и структуру программы. Модель реагирует на команды из контроллера и выдаёт информацию и/или изменяет своё состояние. [2](https://blog.skillfactory.ru/glossary/mvc/)
2. **Представление (View)**. Задача компонента — визуализация информации, которую он получает от модели. View отображает данные на уровне пользовательского интерфейса, например, в виде таблицы или списка. [2](https://blog.skillfactory.ru/glossary/mvc/)
3. **Контроллер (Controller)**. Обеспечивает взаимодействие с системой: обрабатывает действия пользователя, проверяет полученную информацию и передаёт её модели. Контроллер определяет, как приложение будет реагировать на действия пользователя. Также контроллер может отвечать за фильтрацию данных и авторизацию.

5. Структура каркаса (фреймворка) разработки веб-приложений и сайтов

на языке программирования Python (Django)

6. Принципы создания динамических HTML страниц при использовании

Python (Django), методы интеграции HTML с серверной частью веб-сайта.

Всё храниться в templates. Фрагменты сайта разбиваются на блоки, которые можно переиспользовать и менять подставляя переменные внутри них.

7. Организация данных веб-сайта через модели на стороне сервера,

взаимодействие веб-сайта с базами данных.

Делается это через models там задается структура бд которую мы хотим создать и использовать. Взаимодействие страницы с бд происходит через, views которые мы прописываем во views.py