**Тестовое задание.**

Тестовое задание состоит из нескольких задач по различным темам. Необходимо решить хотя бы одну задачу до технического собеседования. Решение задач допускается продолжить после собеседования. Все задачи имеют разную сложность и разное количество баллов за решение задачи. Чем больше баллов набрано за решение задач, тем лучше.

Решение задач необходимо предоставить в виде воспроизводимого отчета (например: .ipynb), если не указано иное. Все файлы решения необходимо упаковать в архив zip. Приветствуется описание приведенных решений и их предпосылок в блокноте решения.

**Задача 1.** *4 балла.*

Необходимо подготовить (предобработать) запись с микрофона плохого качества для дальнешей передачи в модель распознавания речи. Обработка должна вести к повышению качества распознавания (большее соответствие произнесенным словам). Необходимо подготовить решение осуществляющее предобработку аудиозаписей и их дальнейшее распознавание.

**Входные данные:**

Аудиозаписи: download\_14.mp4, download\_16.mp4

**Выходные данные:**

Последовательность слов

**Задача 2.** *3 балла.*

На массиве изображений с автомобилями необходимо определить их цвет в пространстве RGB. Если на изображении несколько автомобилей, необходимо определить цвет для каждого.

Используя открытые источники сформировать и подготовить выборку изображений с автомобилями.

**Входные данные:**

Изображение

**Выходные данные:**

Список кортежей, каждый из которых содержит значения каналов RGB (три целых числа от 0 до 255)

**Задача 3.** *7 баллов.*

Необходимо распознать весь печатный текст на изображении, которое представляет собой фото или скан документа. В качестве примера допускается использование изображений в свободном доступе.

**Входные данные:**

Изображение

**Выходные данные:**

Список строк

**Задача 4.** *6 баллов.*

Используя открытые источники и наборы данных решить задачу классификации намерений пользователя. В качестве входных данных поступают реплики пользователя (на русском языке), которые необходимо отнести к одному из классов. Код должен быть организован в соответствиями с требованиями архитектуры, используя ООП парадигму. Библиотека реализации на усмотрение соискателя.

Дополнением будет реализация поиска «слотов» в рамках интента или реализация механизма их извлечения.

**Задача 5.** *10 баллов.*

Создайте тестовый стенд для API по распознаванию речи. Стенд состоит из источника аудио и потребителя аудио.

Сервис потребитель аудио принимает bytes аудиопотока и транскрибирует речь, сохраняя ее в файл. Реализация на усмотрение соискателя. Требуется обеспечить единовременную транскрибацию, как минимум, 3-х аудио-потоков. Ориентируйтесь на переиспользование уже инстанцированной модели, а не создание новой при каждом запросе.

Источник аудио представляет собой скрипт читающий аудио файл (например, файлы из задания 1), открывающий соединение по RPC или Web Socket и отправляющий в канал содержимое файла (bytes) пакетами по 1000 фреймов.

Код должен быть организован в соответствиями с требованиями архитектуры, используя ООП парадигму. Реализация на усмотрение соискателя.

В качестве решения приложите ссылку на репозиторий содержащий исходный код на Python. Требуемая версия не ниже 3.10.

Приветствуется создание документации к приведенному решению.

Приветствуется создание временного фронтенда с использованием средств прототипирования, например, библиотека Streamlit.

Приветствуется упаковка решения в контейнер Docker.

**Задача 6.** *8 баллов.*

Создайте API к одному из предыдущих заданий. API должно содержать не менее двух выходных эндпоинтов с асинхронным доступом. Код должен быть организован в соответствиями с требованиями архитектуры, используя ООП парадигму. Библиотека реализации на усмотрение соискателя.

В качестве решения приложите ссылку на репозиторий содержащий исходный код на Python. Требуемая версия не ниже 3.10.

Приветствуется реализация процесса прерывания выполнения задачи по ее идентификатору (например, при получении идентификатора задания по эндпоинту */cancel* эта задача отменяется и пользователю возвращается статус задачи)

Приветствуется создание документации к приведенному решению

Приветствуется создание временного фронтенда с использованием средств прототипирования, например, библиотека Streamlit.

Приветствуется упаковка решения в контейнер Docker.

**Задача 7. Пример бизнес задачи.** *9 баллов.*

Данное задание обсуждается на техническом собеседовании в разделе кейс-интервью. Предполагается решение этого задания после прохождения собеседования. Ознакомление с заданием позволит заранее подумать и сформулировать варианты решения, которые обсуждаются на собеседовании.

**Входные данные:**

В файле ***train.csv*** (разделитель - «;») представлен набор данных о визитах клиентов на сервис.

Пояснение признаков в ***train.csv:***

* АвтомобильМодель — Модель автомобиля
* АвтомобильМодельБренд — Марка автомобиля
* АвтомобильVIN — обезличенный VIN-номер автомобиля
* Значение — Пробег автомобиля на дату визита (показания одометра)
* АвтомобильГодВыпуска — Год выпуска автомобиля
* Период — Дата визита
* АвтомобильМодельМежсервисныйПробегТО — Пробег данной марки автомобиля между ТО
* ТО — Метка, которая показывает, является ли данный визит ТО или нет (например ремонт)

В файле содержится около 100 тыс. записей, около 10 тыс. уникальных автомобилей.

**Необходимо:**

Для каждого автомобиля из обучающего набора данных спрогнозировать дату следующего ТО и пробег автомобиля на эту дату.

Для тестирования предсказаний предоставлен файл ***test.csv*** (разделитель - «;»), в котором имеются ответы (реальная дата ТО и пробег автомобиля на эту дату).

**Выходные данные:**

Точность предсказания, полученная на тестовом наборе данных. Необходимо выбрать метрику качества и обосновать ее.