Добрый день!

*Еще раз благодарим тебя за интерес к стажировке в М.Тех. И поздравляем с прохождением на третий этап нашего отбора.*

*Перед тобой задание для* вакансии «Младший исследователь данных (DS NLP)».

***Подробнее с описанием вакансии ты можешь ознакомиться по ссылке из ее названия (зажав Ctrl и кликнув).***

*Вернемся к заданию. На его выполнение у тебя есть 72 часа с момента отправки письма.*

*Мы просим прислать решение в файлах в формате .doc, .docx, .pdf, .tss и других, собранных в архив. Ссылки на внешние ресурсы также размести в отдельном файле. Файлы назови по следующему принципу: Фамилия_Имя_Название_вакансии.* ***Задания сформулированы исчерпывающе и не требуют дополнительных уточнений. Пожалуйста, отвечай на них так, как понимаешь****.*

*Если у тебя возникнут организационные вопросы по стажировке, можешь задать их, написав на почту* ***Alexey.stolyarov@mvideo.ru*** *(Алексей)*

*Желаем удачи!*

# Тестовое задание

**Бизнес-кейс:**
Одним из приоритетов нашего бизнеса является создание лучшего клиентского опыта, поэтому для компании важно понимать текущий уровень CSI (customer satisfaction index) и его динамику (для оценки эффекта изменений в бизнес-процессах).
С этой целью, разработан автоматический механизм сбора обратной связи по заказам. Кроме общей оценки удовлетворённости, клиенты оставляют обратную связь в виде текстового комментария/отзыва в свободной форме.
Вам необходимо обучить модель классификации негативных отзывов, которая позволит получать детальную аналитику по проблемам клиентов в автоматическом режиме.

**Описание датасета:**

Dataset представляет собой подвыборку негативных комментариев клиентов по заказам с типом получения «самовывоз из магазина» с разметкой на 8 классов проблем.

Файл M.Tex_ТЗ_Датасет_DS_NLP.xlsx
text - текст отзыва клиента
class - метка класса на которую надо обучиться
ID - колонка нужна только для проверяющих, не используйте её в задании

**Постановка задачи и оформление результатов:**

Построить модель классификации (multiclass, 8 классов) для отзывов клиентов. Для этого проведите исследование с обучением различных моделей, используя известные вам подходы. Визуализируйте промежуточные и итоговые результаты, замерьте метрики и сделайте финальные выводы.

Работу нужно оформлять в jupyter notebook, снабдив комментариями.
Метрика для оценки - f1.
Готовая модель должна быть сохранена в каком-либо формате.
В отдельном файле model.py должна храниться функция:

```
def get_result(text: pd.Series) -> pd.Series:
    pass
```

Функция принимает колонку с отзывами, на выходе отдаёт колонку с предсказанными классами. Колонка с классами должна иметь наименование - class_predicted. Функция должна выполнять предобработку текста (если она есть), вызывать обученную модель из файла, предсказывать классы и отдавать их.

Также, финальная посылка результатов должна обязательно содержать файл requirements.txt с перечнем библиотек и версиями, которые вы использовали. Файл должен быть в таком виде, чтобы командой:
*pip install -r requirements.txt*
устанавливались все ваши зависимости в отдельное окружение.

**Внимание!** Перед отправкой готовых результатов, внимательно проверяйте и тестируйте работоспособность.
Для этого:
- создайте с помощью requirements новое окружение и кернел
- запустите на нем свой юпитер ноутбук и итоговую функцию для проверки корректной выдачи результатов