

Добрый день!

Данные задания необходимо сделать в **одном** Jupyter Python Notebook на платформе [COLAB](#) и предоставить результаты работы ссылкой, приложив в текстовый файл на платформу.

В рамках вашего **одного** COLAB ноутбука , укажите разделение на задания, чтобы преподавателю можно было понять какой блок к какому заданию относиться.

### **Задание №1**

Создайте модель на основе LSTM для прогнозирования будущих значений временного ряда, такого как акции компании, погода или продажи продукта.

Подготовка данных: Выберите любой временной ряд (**ВНИМАНИЕ!** используйте данные которые вы сами обнаружите в интернете или на платформе Kaggle), который хотите анализировать. Произведите разделение на обучающую и тестовую выборки, а также нормализацию данных, если необходимо.

Построение модели LSTM: Создайте архитектуру сети с одним или несколькими слоями LSTM.

Обучение модели: Обучите модель на обучающей выборке, используя подходящую функцию потерь и оптимизатор.

Прогнозирование и оценка: Произведите прогнозирование на тестовой выборке и оцените качество прогнозов с использованием подходящих метрик, таких как среднеквадратичная ошибка (MSE).

Визуализация: Визуализируйте реальные и прогнозируемые значения временного ряда на графиках для наглядного сравнения.

### **Задание №2**

Используйте модель BERT для классификации настроения [ОТЗЫВОВ](#).

Загрузите набор данных, состоящий из текстов отзывов и их меток настроения (положительное, негативное). Проведите необходимую предобработку данных, включая токенизацию и создание входных данных для BERT.

Использование BERT: Загрузите предварительно обученную модель BERT и настройте ее для вашей задачи классификации.

Обучение модели: Разделите данные на обучающую и тестовую выборки и обучите модель на обучающей выборке.

Оценка модели: Проанализируйте эффективность модели на тестовой выборке, используя различные метрики качества, такие как точность, полноту и F1-меру.

Интерпретация результатов: Проведите анализ ошибок модели, попытайтесь выявить общие тенденции или особенности, на которых модель допускает ошибки.

Визуализация внимания: (Дополнительно) Визуализируйте веса внимания трансформера в различных слоях и попытайтесь интерпретировать, на что модель обращает внимание при классификации.

### **Задание №3**

Используя Python библиотеку `cv2`, `RetinaNET` и вашу фотографию, содержащую любые объекты / вашего питомца, решите задачу Object Detection (определение объектов на фотографии).