

# Классификация сельскохозяйственных культур<sup>2</sup>

baton

Сидоров Артём  
Гребенник Артур

# Задача



**Построение** модели классификации сельскохозяйственных культур, произрастающих на территории Хабаровского края на основе временных рядов значений оптических каналов и вегетационного индекса **NDVI**



**Данные:** сглаженные временные ряды вегетационных индексов **NDVI** и отдельных спутниковых каналов, рассчитанных с помощью оптических снимков, полученные со спутников **Sentinel-2A/B**

# Предварительная обработка

6 различных датасетов

Импортирование и предобработка

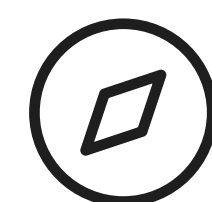
6-ти csv файлов: NDVI, NIR, SWIR,  
GREEN, BLUE, RED

Удаление столбца `index`

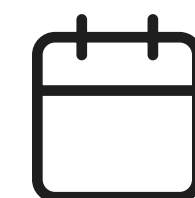
# Структура данных

7 тысяч строк  
в каждом датасете

26 дней\*



Регион сбора данных -  
Хабаровский край



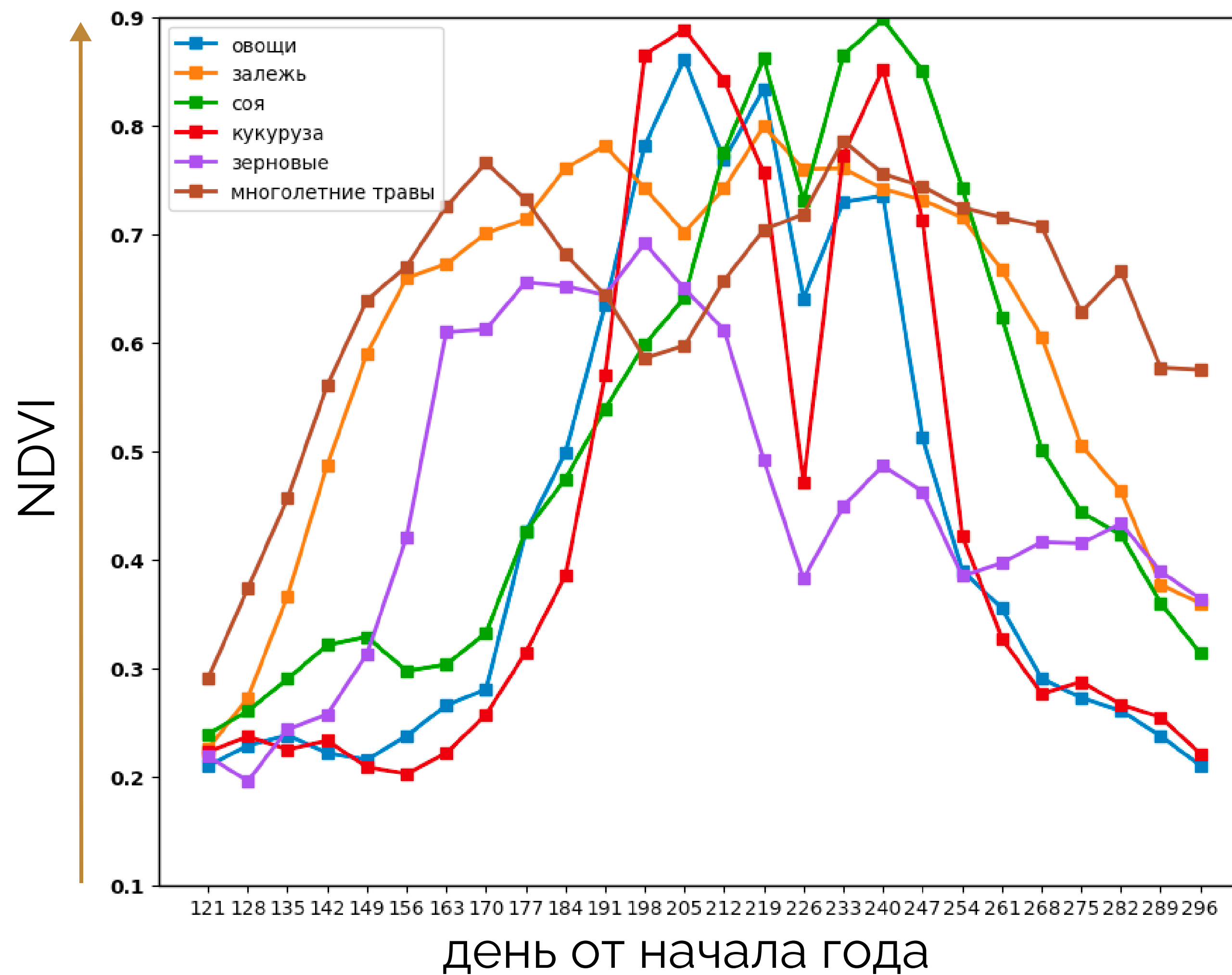
Период снятия данных  
начало мая - конец октября  
Периодичность - неделя



Характеристики: разделение  
по дням, оптическим каналам,  
культуре

\*26 недель по 1 дню в каждой

# Предварительный анализ



# Обработка данных

7 261

итоговый размер выборки

**Заполнение** пустых значений средним по определенному дню и культуре

**Очистка** выбросов, предельных значений

**Дополнительное сглаживание** некоторых дней

# Подготовка данных

12

новых переменных  
были добавлены/изменены

1

## culture

**Изначально:** овощи, залежь, соя, кукуруза, зерновые, многолетние травы.

**Изменено:** 6 столбцов, для каждой культуры, где 0 - нет, 1 - да

2

## {day}\_{Example}

**Добавлено:** 5 новых формул расчета NDWI по различным оптическим каналам

```
df_features[f'{day}_NDWI'] = (df_nir[day] - df_swir[day]) / (df_nir[day] + df_swir[day])
df_features[f'{day}_NEW'] = df_ndvi[day] * df_features[f'{day}_NDWI']
df_features[f'{day}_NDWI1'] = (df_green[day] - df_nir[day]) / (df_green[day] + df_nir[day])
df_features[f'{day}_NDVI'] = (df_nir[day] - df_red[day]) / (df_red[day] + df_nir[day])
df_features[f'{day}_NDBI'] = (df_swir[day] - df_nir[day]) / (df_swir[day] + df_nir[day])
```

# Обработка обучающих данных

156

итоговое количество признаков,  
обучающего датасета

**OneHotEncoder()**

**Преобразование** категориальных  
переменных в числовые значения

**StandardScaler()**

**Преобразование** данных для повышения  
надежности, интерпретируемости и  
производительности модели



# Выбор модели



открытая библиотека, написанная на языке Python и обеспечивающая взаимодействие с искусственными нейронными сетями

# Архитектура модели машинного обучения

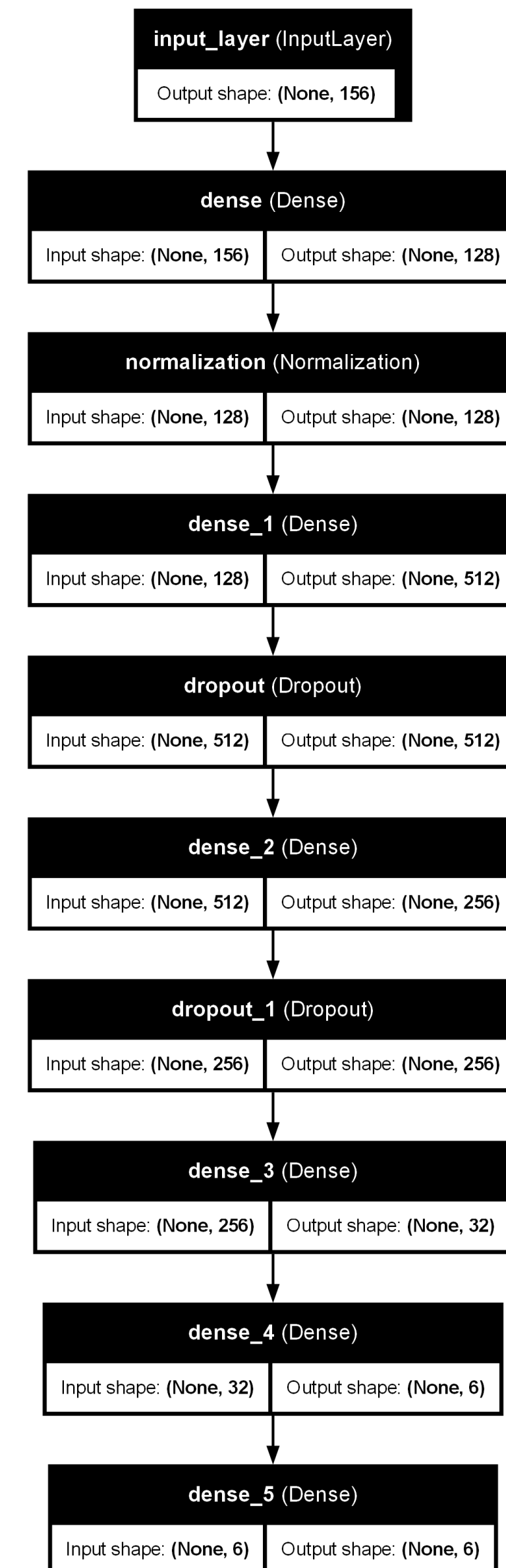
**Модель** принимает на вход вектор размером **156**, который представляет собой набор признаков.

## Слои:

- Первый слой: 128 нейронов, ReLU
- Нормализация
- Второй слой: 512 нейронов, ReLU
- Dropout: 50%
- Третий слой: 256 нейронов, ReLU
- Второй Dropout: 50%
- Четвёртый слой: 32 нейрона, ReLU
- Выходной слой: 6 нейронов, softmax

## Компиляция:

- Оптимизатор: Adamax (0.001).
- Функция потерь: categorical\_crossentropy
- Метрика: f1\_score



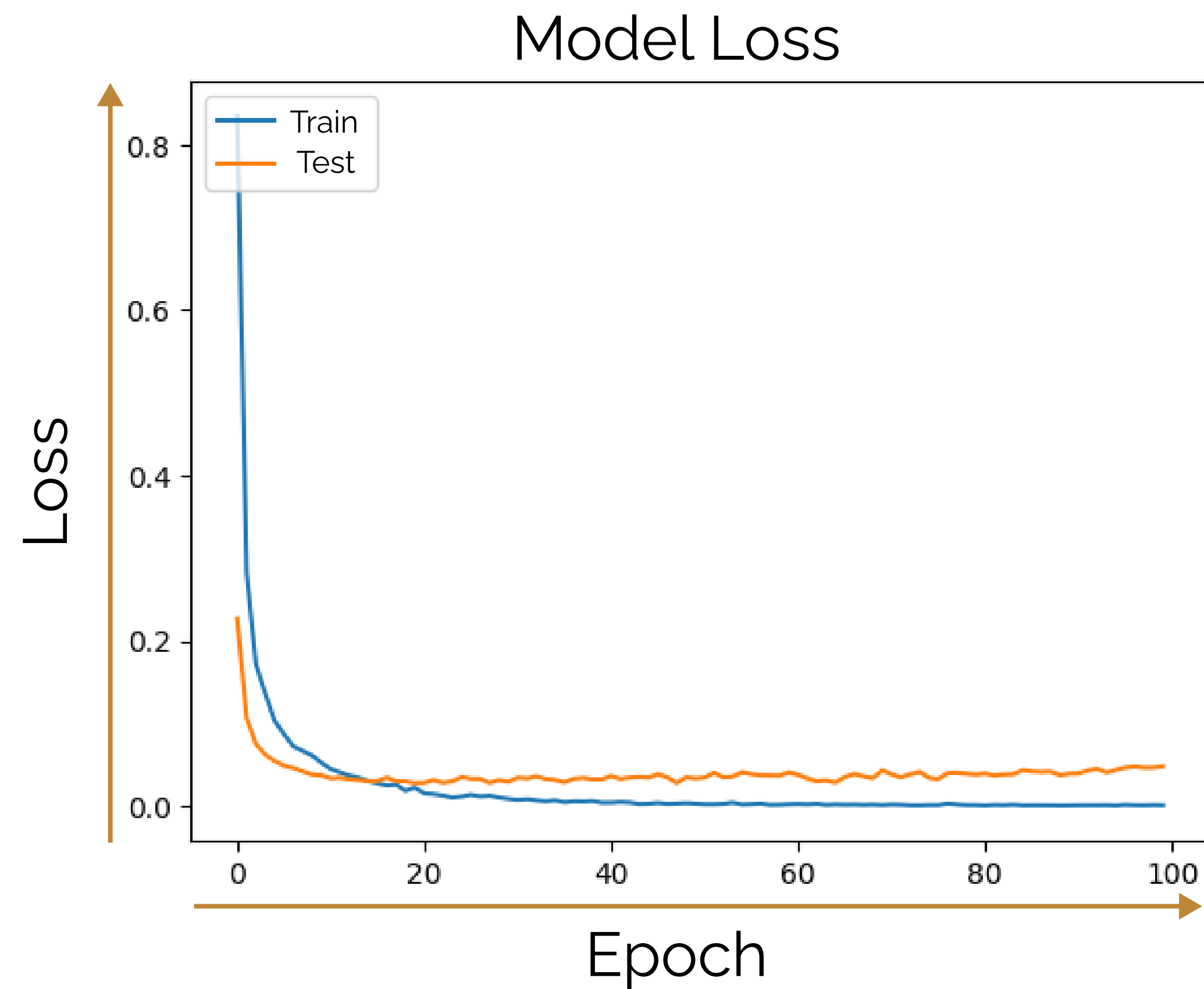
# Модель машинного обучения

5 слоев

100 epochs

0.9999 f1\_score\*





\*на тестовой выборке



# Ограничения

- ↘ | Модель может работать некорректно в **других** регионах
- 🕒 | Нельзя обобщить на **2024** год из-за неопределенности времени

# Перспективы

-  Прогнозирование в режиме реального времени
-  Интеграция с другими источниками данных
-  Обучение и самосовершенствование
-  Сотрудничество и обмен знаниями

