МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Московский государственный технический университет

имени Н.Э. Баумана

(национальный исследовательский университет)»

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**по курсу**

«Data Science»

Слушатель Семёнов Дмитрий Александрович

Москва, 2023

## Содержание

[Содержание](#_rszce6mcxe3e) 2

[Введение](#_5tjxt8b5zbuu) 3

1. Аналитическая часть 5
   1. Постановка задачи 5
   2. Описание используемых методов 6
   3. Разведочный анализ данных 9
2. Практическая часть 12
   1. Предобработка данных 12
   2. Разработка и обучение модели 15
   3. Тестирование модели 16
   4. Нейронная сеть 18
   5. Репозиторий 26

## 

## 

## 

## 

## 

## 

## 

## 

## 

## Введение

Данная выпускная квалификационная работа была выполнена в рамках курса “Data Science” МГТУ им. Н.Э. Баумана.

В наши дни малый и средний бизнес являются самыми активно

развивающимися сферами экономики. От уровня их развития зависит гибкость и приспосабливаемость экономики страны к нестабильной экономической обстановке.

Большую часть организаций малого и среднего бизнеса составляют торговые предпринимательства. Для того чтобы быть конкурентоспособными организациями они должны обладать современными инструментами для ведения бизнеса. В настоящее время существует большое множество наборов таких инструментов, но не все организации могут себе позволить приобретение таких инструментов. Причиной тому высокая стоимость и необходимость экспертных знаний при работе с ними.

Один из важных факторов успешной деятельности торговых организации является точное прогнозирование объёмов продаж. С помощью качественно составленного прогноза можно распределять эффективно ресурсы внутри организации и увеличить эффективность бизнеса. Также существует необходимость обработки большого количества номенклатуры.

Становится актуальным вопрос об автоматизации рутинных действий и оперативного контроля запасов. Одним из способов решения данной задачи является использование методов машинного обучения.

Результатом решения задач прогнозирования продаж становятся снижение товарного запаса и повышение оборачиваемости материальных ресурсов.

## Аналитическая часть

### 1.1 Постановка задачи

Цель исследования: выявление наиболее эффективной модели

прогнозирования прибыли на примере данных компании по продажам инженерной сантехники с использованием методов машинного обучения.

Характеристика датасета: данные были представлены в виде excel файлов со статистикой продаж инженерной сантехники сети магазинов состоящей из 23 точек розничных продаж в городе Санкт-Петербурге за период пять лет с 2018 по 2022 года и были преобразованы в один csv файл. Интервал временного ряда составляет один день. По результатам разведочного анализа обнаружены выбросы обусловленные некорректным ведение отчетности, пропусков не обнаружено.

В отчете о продажах имеются следующие колонки:

* дата продажи с уточнением до дня;
* магазин в котором была совершена продажа;
* номенклатурная позиция;
* количество проданной номенклатурной позиции;
* себестоимость;
* выручка;
* прибыль;
* прибыль выраженная в процентах.

### 1.2 Описание используемых методов

Для решения поставленной задачи были выбраны методы:

* LinearRegression (линейная регрессия)
* RandomForestRegressor (случайный лес)
* XGBoost (градиентный бустинг)

LinearRegression - обычная линейная регрессия методом наименьших квадратов. Соответствует линейной модели с коэффициентами w = (w1, …, wp), чтобы минимизировать остаточную сумму квадратов между наблюдаемыми целями в наборе данных и целями, предсказанными линейным приближением.

## Преимущества:

* Простая и простая реализация.
* Космическое комплексное решение.
* Быстрое обучение.
* Значение коэффициентов θ дает предположение о значимости признака.

## Недостатки:

* Применимо, только если решение является линейным. Во многих реальных сценариях это может быть не так.
* Алгоритм предполагает, что входные остатки (ошибка) распределены нормально, но не всегда могут быть выполнены.
* Алгоритм предполагает, что входные функции взаимно независимы (без коллинеарности).

RandomForestRegressor - универсальный алгоритм машинного обучения, суть которого состоит в использовании ансамбля решающих деревьев. Само по себе решающее дерево предоставляет крайне невысокое качество предсказания, но из-за большого их количества результат значительно улучшается. Также это один из немногих алгоритмов, который можно использовать в абсолютном большинстве задач.

Преимущества:

* Имеет высокую точность предсказания, которая сравнима с результатами градиентного бустинга.
* Не требует тщательной настройки параметров, хорошо работает из коробки.
* Практически не чувствителен к выбросам в данных из-за случайного семплирования (random sample).
* Редко переобучается. На практике добавление деревьев только улучшает композицию.

## Недостатки:

* Для реализации алгоритма случайного дерева требуется значительный объем вычислительных ресурсов.
* Большой размер моделей.
* Построение случайного леса отнимает больше времени, чем деревья решений или линейные алгоритмы.
* Алгоритм склонен к переобучению на зашумленных данных.
* В отличие от более простых алгоритмов, результаты случайного леса сложнее интерпретировать.
* В отличие от линейной регрессии, Random Forest не обладает возможностью [экстраполяции](https://ru.wikipedia.org/wiki/%D0%AD%D0%BA%D1%81%D1%82%D1%80%D0%B0%D0%BF%D0%BE%D0%BB%D1%8F%D1%86%D0%B8%D1%8F). Это можно считать и плюсом, так как в случае выбросов не будет экстремальных значений.

XGBoost - алгоритм машинного обучения, основанный на дереве поиска решений и использующий фреймворк градиентного [бустинга](https://ru.wikipedia.org/wiki/%D0%91%D1%83%D1%81%D1%82%D0%B8%D0%BD%D0%B3_(%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BC%D0%B0%D1%88%D0%B8%D0%BD)#%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D1%8B_%D0%B1%D1%83%D1%81%D1%82%D0%B8%D0%BD%D0%B3%D0%B0). В задачах предсказания, которые используют неструктурированные данные (например, изображения или текст), искусственная нейронная сеть превосходит все остальные алгоритмы или фреймворки. Но когда дело доходит до структурированных или табличных данных небольших размеров, в первенстве оказываются алгоритмы, основанные на дереве поиска решений.

### 

### 

### Рисунок 1 - Оценка качества моделей

### 1.3 Разведочный анализ данных

### Для разведочного анализа данных были использованы методы описательной статистики.

Датасет был проверен на наличие пропусков в значениях (метод DataFrame.isna().sum()).

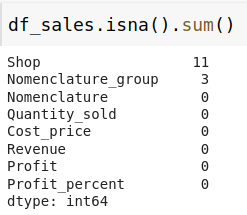


Рисунок 2 - Проверка на наличие пропусков

Были обнаружены незначительные пропуски в двух колонках, в одиннадцати случаях не указан магазин в котором совершалась продажа и в трех случаях не указана номенклатурная группа товара, так как в нашем случае эти колонки не учитываются в анализе, было принято решения ничего не удалять из датасета.

Также с помощью метода DataFrame.describe(), было обнаружено, что датасет содержит отрицательные значения, в нашем случае в отчете так представлена информация о возвратах товара покупателями и в дальнейшем при разработке модели мы будем изменять временной шаг беря сумму за неделю, что таким образом компенсирует отрицательные значения и даст более объективную картину о фактических значениях прибыли.

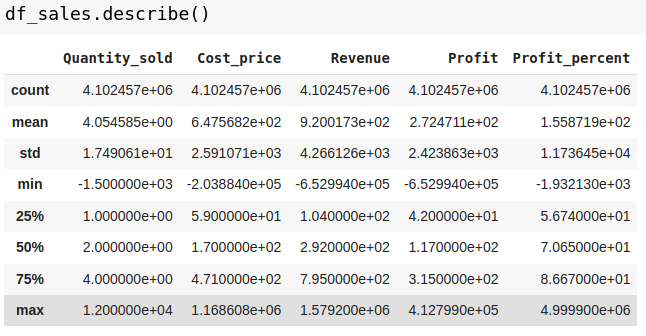


Рисунок 3 - Применение метода DataFrame.describe()

С помощью построение гистограммы и boxplot для нашей целевой переменной Profit (прибыль) мы можем увидеть распределение близкое к нормальному и наличие значительных выбросов.

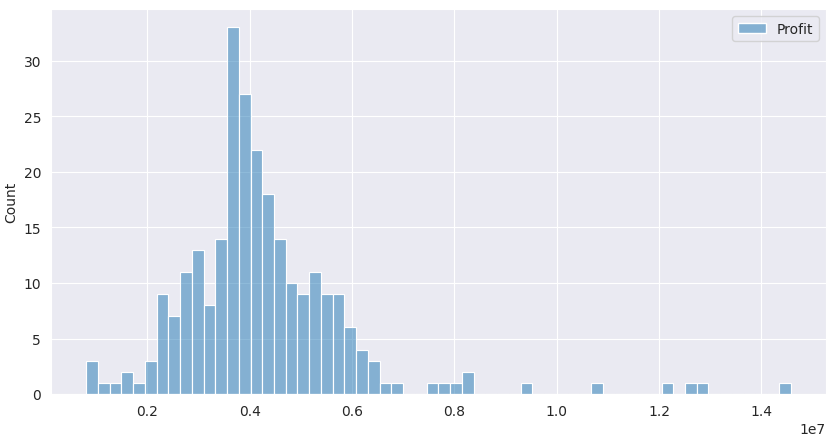


Рисунок 4 - Гистограмма целевой переменной Profit

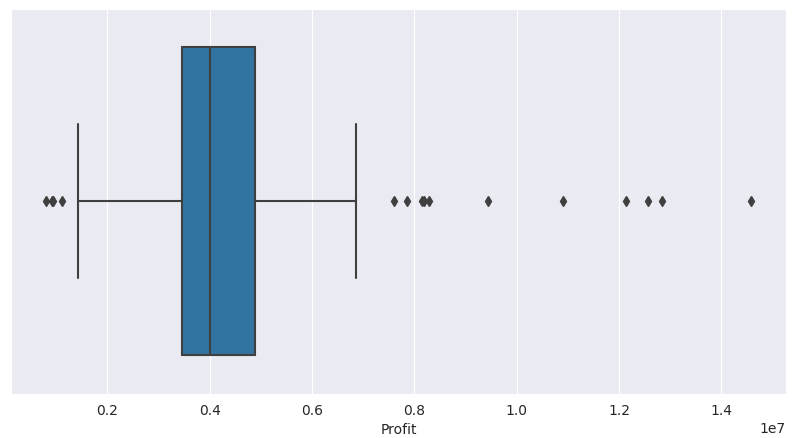


Рисунок 5 - boxplot для целевой переменной Profit

Выбросы обусловлены ошибкой отчета о продажах, так как при уточнении информации таких значений продаж в начале 2022 года, в соответствии с рисунком 6, не было, что говорит о необходимости корректировки этих значений.

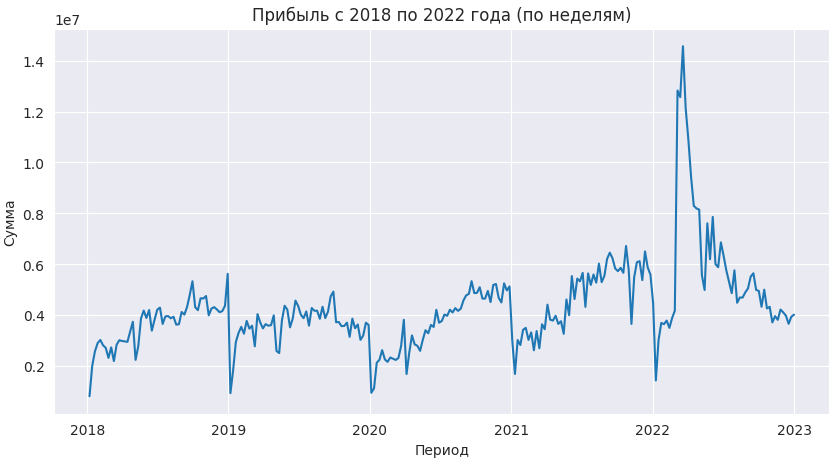


Рисунок 6 - Временной ряд целевой переменной Profit

## Практическая часть

### 2.1 Предобработка данных

Предобработка данных осуществлялась на основании разведочного анализа данных, который показал наличие выбросов, которые лучше всего демонстрирует boxplot на рисунке 5.

Выбросы было принято решение сгладить средним значением за аналогичные по счету недели в году, были взяты именно средние значения, а не медиана, так как выбросы довольно значительны и присутствую только за март и апрель 2022 года и вызваны ошибкой ведения отчетности, по этой причине медиана бы сгладила выбросы не значительно.

На рисунке 7 представлен код с помощью которого было произведена замена выбросов на средние значения.

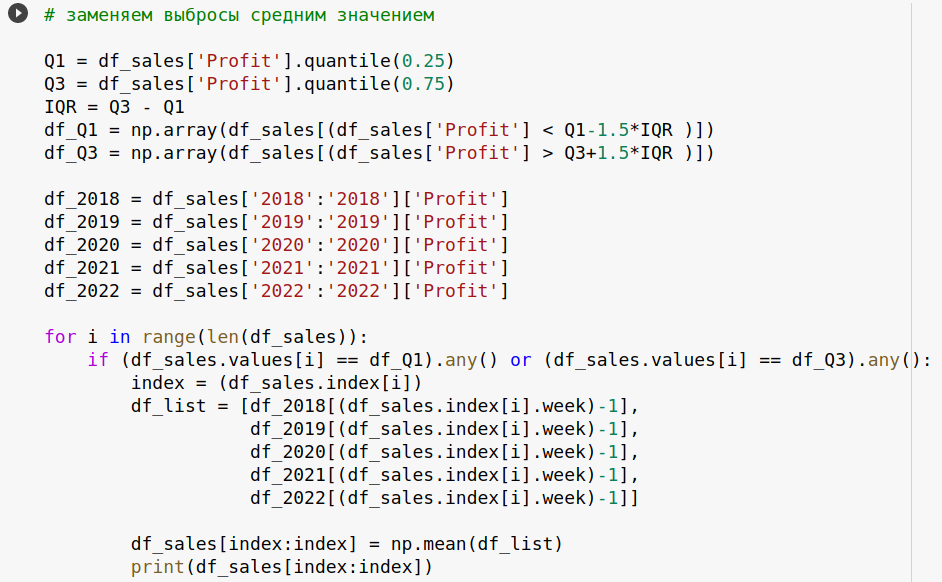


Рисунок 7 - Код для сглаживающий выбросы

Рисунки 8, 9 и 10 иллюстрируют результат разового применения кода для сглаживания выбросов средним значением.

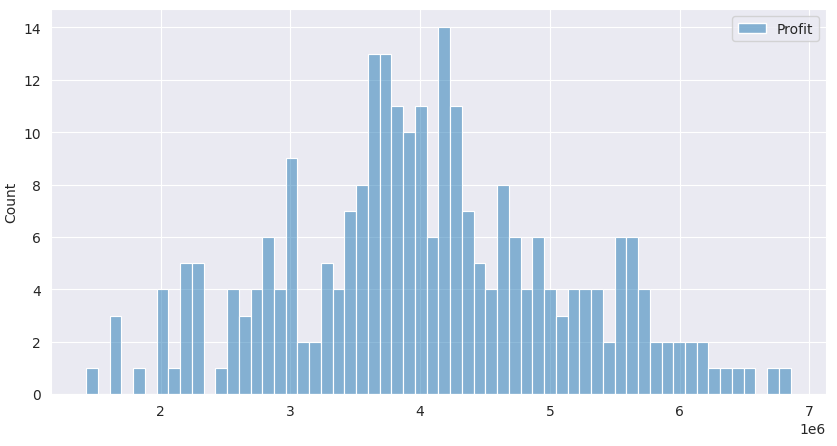


Рисунок 8 - Гистограмма после сглаживания выбросов

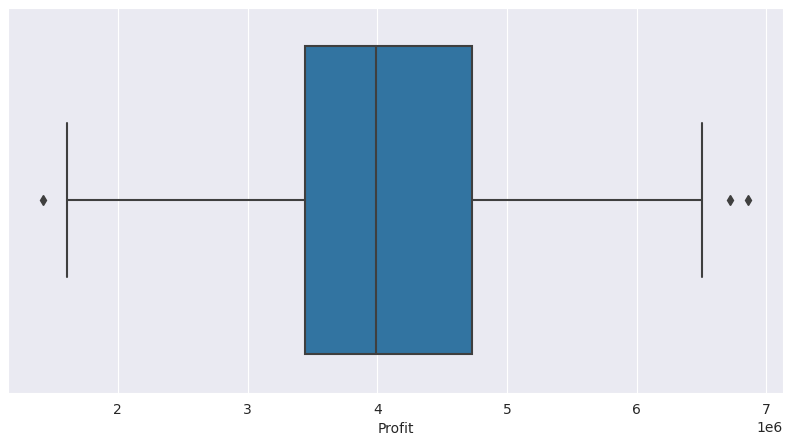


Рисунок 9 - boxplot после сглаживания выбросов

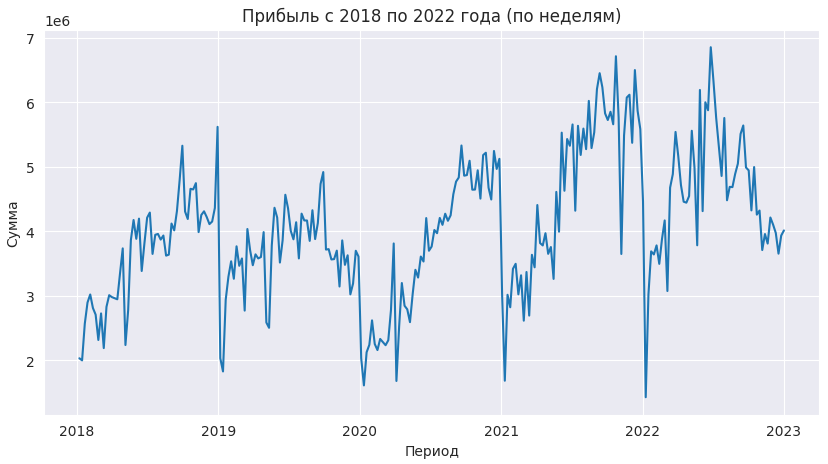


Рисунок 10 - временной ряд после сглаживания выбросов

### 

### 

### 2.2 Разработка и обучение модели

В качестве моделей для прогнозирования прибыли были использованы:

* LinearRegression (линейная регрессия)
* RandomForestRegressor (случайный лес)
* XGBoost (градиентный бустинг)

Для построения моделей была написана функция prepareData, с помощью которой в датасет были добавлены лаги исходного временного ряда в качестве признаков и датасет был разбит на тренировочную и тестовую выборки.

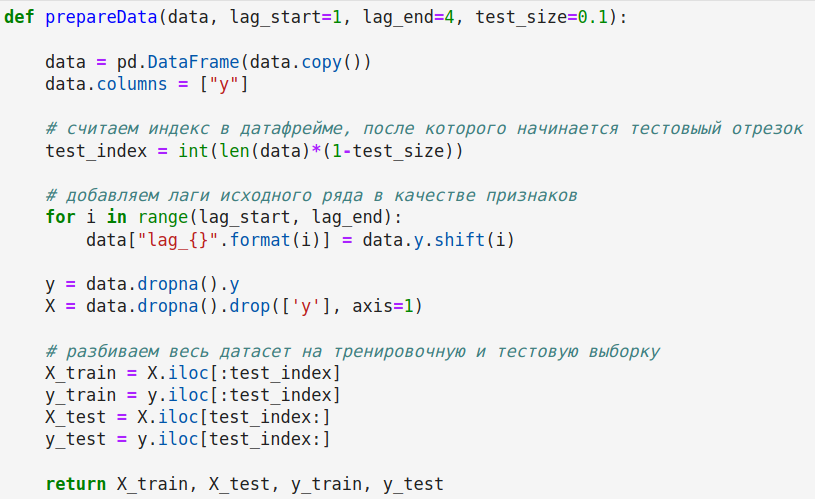


Рисунок 11 - Код для преобразования датасета

При обучении моделей были использованы три лага исходного временного ряда, при обучении случайного леса и градиентного бустинга был осуществлен подбор гиперпараметров посредствам поиска по сетке с перекрестной проверкой.

Результаты оценки качества разработанных моделей представлена на рисунке 1.

### 2.3 Тестирование модели

Тестирование модели осуществлялось на тестовой выборке, которая составляла 10 процентов от всего датасета, что выражалось в последних двадцати четырех неделях временного ряда, обучающая выборка состояла из двухсот тридцати четырех недель.

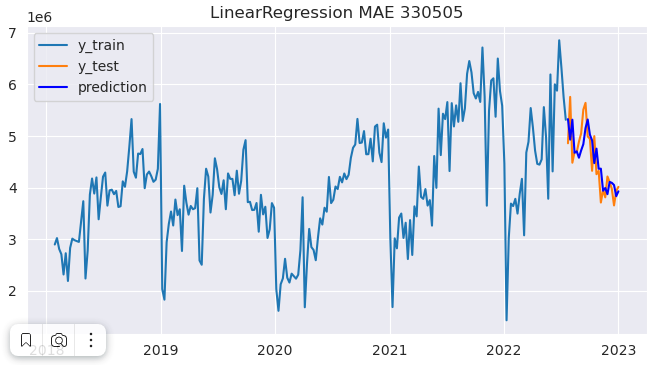


Рисунок 12 - Линейная регрессия

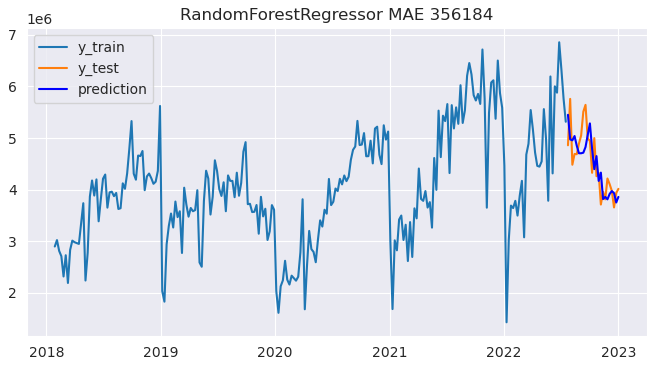


Рисунок 13 - Случайный лес

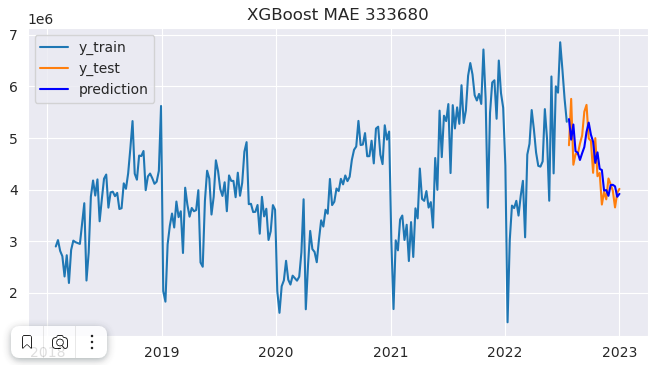


Рисунок 14 - Градиентный бустинг

### 2.4 Нейронная сеть

Рекуррентная нейронная сеть (RNN) — это тип нейронной сети, хорошо подходящий для данных временных рядов. RNN обрабатывают временной ряд шаг за шагом, сохраняя внутреннее состояние от шага к шагу.

В этом работе мы будете использовать слой RNN под названием Long Short-Term Memory ( [tf.keras.layers.LSTM](https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM?hl=ru) ).

Перед обучением нейронной сети важно масштабировать функции. Нормализация — это распространенный способ масштабирования: вычесть среднее значение и разделить на стандартное отклонение каждого признака.

Среднее значение и стандартное отклонение следует вычислять только с использованием обучающих данных, чтобы модели не имели доступа к значениям в проверочных и тестовых наборах.

Также можно утверждать, что модель не должна иметь доступа к будущим значениям в тренировочном наборе во время обучения и что эта нормализация должна выполняться с использованием скользящих средних. Наборы тестирования гарантируют, что мы получим (в некоторой степени) честные метрики. В целях простоты в данной работе используется простое среднее значение.

Модели в данной работе будут делать набор прогнозов на основе окна последовательных выборок из данных.

В зависимости от задачи и типа модели может потребоваться создание различных окон данных.

Например, чтобы сделать один прогноз на 24 часа вперед, учитывая 24-часовую историю, мы можем определить окно следующим образом:

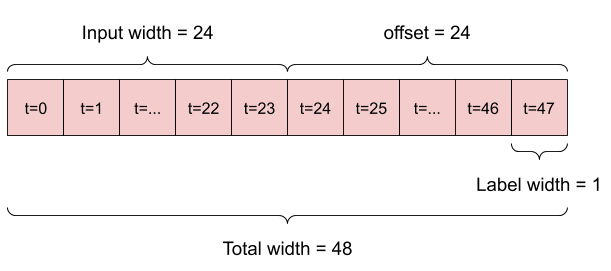


Рисунок 15 - Пример окна данных

Модель, которая делает прогноз на один час вперед, учитывая шесть часов истории, нуждалась бы в таком окне:

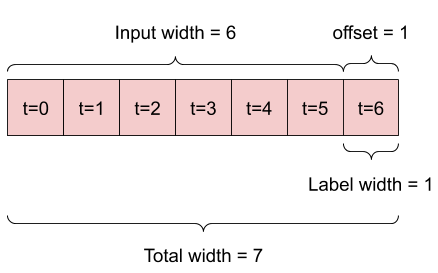


Рисунок 16 - Пример окна данных

Учитывая список последовательных входных данных, метод univariate\_data или multivariate\_data преобразует их в окно входных данных и окно меток.

Пример на Рисунке 16 , который мы определили ранее, будет разбит следующим образом:

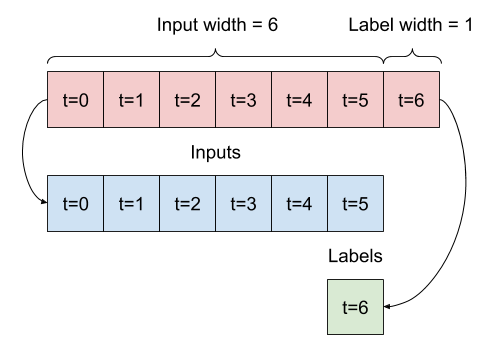


Рисунок 17 - Пример окна данных

Важным аргументом конструктора для всех слоев Keras RNN, таких как [tf.keras.layers.LSTM](https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM?hl=ru), является аргумент return\_sequences. Этот параметр может настроить слой одним из двух способов:

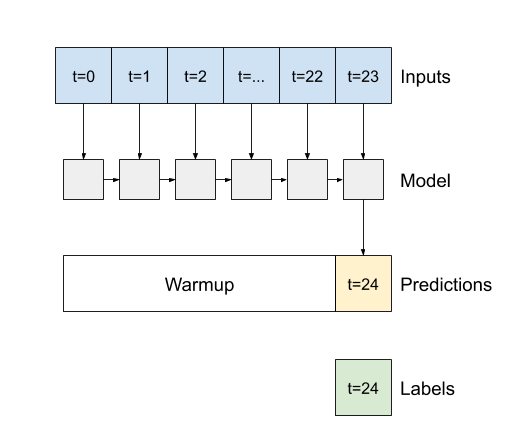
* Если False по умолчанию, слой возвращает только выходные данные последнего временного шага, давая модели время, чтобы прогреть свое внутреннее состояние, прежде чем делать один прогноз:

Рисунок 18 - return\_sequences (False)

* Если True , слой возвращает результат для каждого входа. Это полезно для:
  + Укладка слоев RNN.
  + Обучение модели на нескольких временных шагах одновременно.

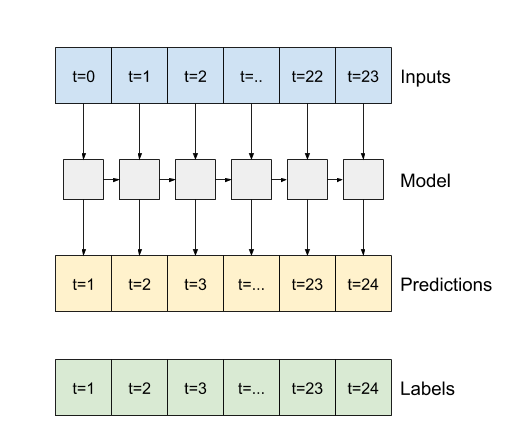


Рисунок 19 - return\_sequences (True)

Простая LSTM модель.

В данном случае мы делаем прогноз на одну неделю вперед учитывая двенадцать недель. Пример окна данных на рисунке 16

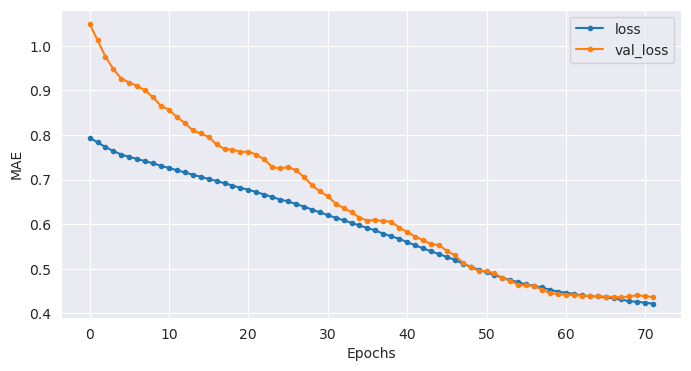
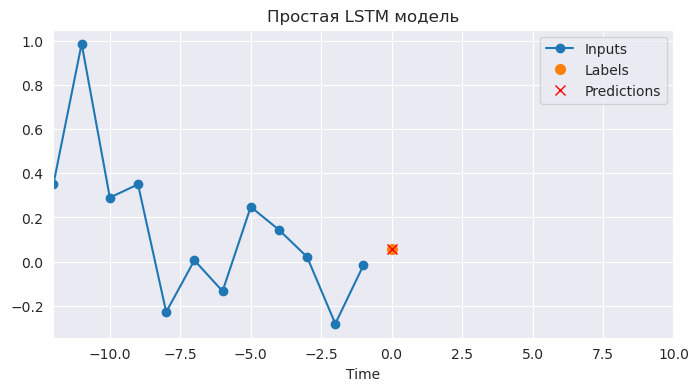


Рисунок 21 - Обучение модели

Рисунок 22 - Простая LSTM модель

### Точечное прогнозирование.

Здесь мы делаем прогноз также учитывая двенадцать недель, но делаем прогноз не на следующую по счету неделю, а на пятую неделю, пропуская четыре предыдущих. Пример окна данных на рисунке 15

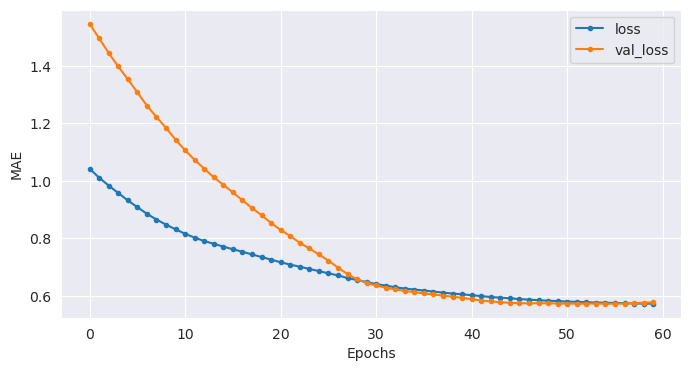


Рисунок 24 - Обучение модели

Рисунок 25 - Точечное прогнозирование

Прогноз на несколько шагов.

При таком подходе, мы делаем прогноз сразу на несколько следующих недель за тренировочными данными, которые также составляют 12 недель.

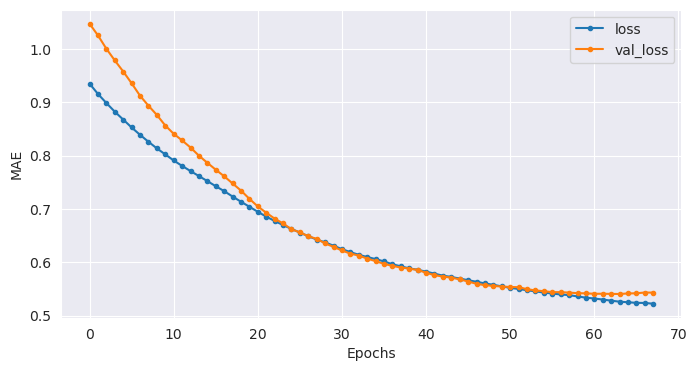


Рисунок 27 - Обучение модели

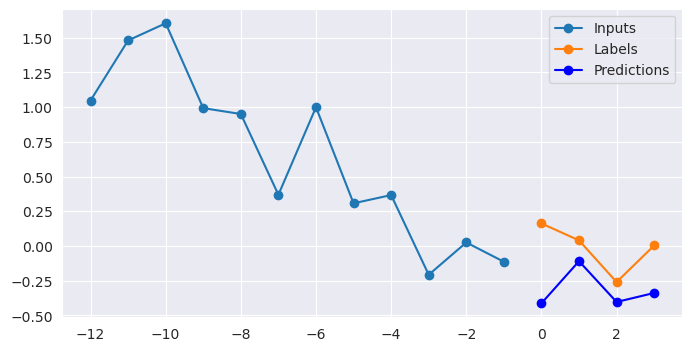


Рисунок 28 - Прогноз на несколько шагов

### 2.5 Репозиторий

Страница слушателя на github:

https://github.com/tendw/Time\_series\_model.git