

Федеральное государственное автономное образовательное учреждение  
высшего образования «Национальный исследовательский университет  
«Высшая школа экономики»

Факультет компьютерных наук  
Основная образовательная программа  
Прикладная математика и информатика

**КУРСОВАЯ РАБОТА**  
**ИССЛЕДОВАТЕЛЬСКИЙ ПРОЕКТ НА ТЕМУ**  
**"ПРОГНОЗИРОВАНИЕ ВРЕМЕННЫХ РЯДОВ С ПОМОЩЬЮ**  
**ДИНАМИЧЕСКИХ АВТОРЕГРЕССИОННЫХ МОДЕЛЕЙ"**

Выполнил студент группы 192, 3 курса,  
Лоптев Сергей Евгеньевич

Руководитель КР:  
научный сотрудник Зехов Матвей Сергеевич

Москва 2022

# Содержание

<b>1</b>	<b>Введение</b>	<b>4</b>
<b>2</b>	<b>Обзор литературы</b>	<b>5</b>
<b>3</b>	<b>Описание предлагаемого метода</b>	<b>8</b>
3.1	Формулы . . . . .	8
3.2	Гиперпараметры . . . . .	9
3.3	Обучение . . . . .	10
3.4	Детали реализации . . . . .	10
3.5	Подбор гиперпараметров . . . . .	11
<b>4</b>	<b>Описание поставленного эксперимента</b>	<b>17</b>
4.1	Данные . . . . .	17
4.2	Постановка эксперимента . . . . .	18
4.3	Проведение эксперимента . . . . .	20
<b>5</b>	<b>Анализ полученных результатов</b>	<b>21</b>
5.1	Сравнение моделей . . . . .	21
5.1.1	Периодичность . . . . .	21
5.1.2	Категория источника . . . . .	22
5.1.3	Сезонность . . . . .	23
5.1.4	Стационарность . . . . .	23
5.1.5	Горизонт прогнозирования . . . . .	24
5.2	Выявление благоприятных временных рядов . . . . .	25
5.2.1	Периодичность . . . . .	26
5.2.2	Категория источника . . . . .	27
5.2.3	Сезонность . . . . .	27
5.2.4	Стационарность . . . . .	27
5.2.5	Горизонт прогнозирования . . . . .	28



# Аннотация

Существует множество популярных моделей, используемых в прогнозировании временных рядов. В данной работе мы реализуем не самые популярные модели — *динамические авторегрессионные модели*. Мы статистически сравниваем данные модели с базовыми моделями — ARIMA, ETS, PROPHET и наивной на выборке M4. Кроме обычной динамической регрессии вводится динамическая регрессия с сезонностью; для неё также проводится статистическое сравнение. Кроме того, проводится исследование особенностей временных рядов, на которых динамические модели выдают лучшее качество. Мы показываем, что динамические модели в целом не хуже, чем базовые методы, работают в прогнозировании временных рядов. При этом на качество работы динамических моделей может влиять периодичность временного ряда, сезонность и горизонт прогнозирования.

*Ключевые слова:* динамическая регрессия, DLS, динамические наименьшие квадраты, ETS, ARIMA, PROPHET, наивная модель, временные ряды, прогнозирование временных рядов.

There are a lot of popular models in time series forecasting. In this paperwork we implement not so popular *dynamic autoregression models*. We conduct statistical comparison with baselines: ARIMA, ETS, PROPHET and naive models on the M4 dataset. Apart from dynamic regression we implement dynamic regression with seasonality; we conduct statistical tests for it as well. We show that dynamic models generally have a worse performance than the baseline. At the same time, periodicity and seasonality of time series may influence quality of dynamic models predicts, as well as forecasting horizon.

*Keywords:* dynamic regression, DLS, dynamic least squares, ETS, ARIMA, PROPHET, naive model, time series, time series forecasting.

# 1 Введение

Прогнозирование временных рядов — это достаточно часто встречаемая в индустрии задача. Она решается, например, в финансах, бизнесе и IT, причём свойства прогнозируемых рядов сильно варьируются. Задача прогнозирования временного ряда зачастую сводится к перебору известных моделей и подбору гиперпараметров. В этой статье мы сравним работу динамических регрессионных моделей с такими моделями, как ETS, ARIMA, PROPHET, наивная модель на большом количестве реальных временных рядов. Кроме того, мы классифицируем ряды, на которых динамическая регрессия показывает себя лучше остальных моделей. Таким образом, задача бьётся на несколько конкретных подзадач: оптимальная реализация градиентного спуска для динамических моделей на языках Python и C++, применение динамических и описанных выше моделей на большом количестве реальных временных рядов, и оценка полученных результатов.

Динамические регрессионные модели были достаточно детально изучены теоретически. Например, были выведены свойства оценок параметров с помощью динамического метода, а также проводилось сравнение динамических моделей с обычной линейной регрессией на синтетических данных ([Hartley M. J. \(1972\)](#), [Johnston et al. \(1974\)](#)). В последнее время было выпущено несколько статей с реализацией динамических моделей ([Gandrud, C. et al. \(2015\)](#), [Williams, L. K., Whitten, G. D. \(2011\)](#)), но все они имеют свои недостатки с точки зрения задачи, которую мы ставим. Таким образом, наша постановка задачи имеет практическую ценность, так как все работы о данной модели рассматривали её в основном именно в теоретическом ключе.

В результате проведения работы мы получили, что динамические модели в целом справляются с предсказанием не хуже, чем остальные представленные модели. Мы делили ряды на кластеры по периодичности, источнику, сезонности и стационарности; в среднем по каждому признаку процент побед динамической регрессии (то есть, лучшая MAE, чем у остальных моделей) со-

ставляет около 20-30%. С точки зрения классификации рядов можно сделать вывод, что периодичность и сезонность ряда могут влиять на качество динамических моделей относительно базовых методов. Кроме того, было сделано наблюдение о том, что ошибка динамических моделей относительно базовых методов больше на многошаговом прогнозе (то есть, на дальний шаг), чем на одношаговом, и ещё больше, если мы пытаемся прогнозировать весь путь до этого дальнего шага.

Код работы доступен по ссылке <https://github.com/beastSL/coursework-3>.

## 2 Обзор литературы

Как уже было упомянуто, мы собираемся сравнивать новый метод с базовыми методами. Коротко опишем их.

- ARIMA (и метод её оценки) была представлена в [Box G., Jenkins G. \(1970\)](#). Эта модель — улучшение модели ARMA, позволяющее эффективно прогнозировать нестационарные (с точки зрения тренда) ряды. Компонента AR (AutoRegressive) названия означает, что в качестве регрессоров мы используем лагированные значения ряда; компонента I (Integrated) означает, что вместо сырых значений ряда мы возьмём разности соседних значений, возможно, несколько раз, компонента MA означает, что целевая переменная линейно зависит от ошибок за прошлые моменты времени. Модель ARIMA имеет три гиперпараметра  $p$ ,  $d$  и  $q$ , и представляет собой следующую формулу:

$$\Delta^d y_t = c + \sum_{i=1}^p a_i \Delta^d y_{t-i} + \sum_{j=1}^q b_j \varepsilon_{t-j} + \varepsilon_t.$$

- ETS — модель, представленная в [Brown R. G. \(1956\)](#) и улучшенная и расширенная в [Holt C. C. \(1957\)](#). В самом базовом виде ETS задаётся

следующими уравнениями:

$$\hat{y}_1 = y_0$$

$$\hat{y}_t = \alpha y_{t-1} + (1 - \alpha)\hat{y}_{t-1} = \hat{y}_{t-1} + \alpha(y_{t-1} - \hat{y}_{t-1}),$$

где  $\alpha \in (0, 1)$ . Эта формула может быть расширена для учёта тренда и сезонности тренда, а также стандартной ошибки.

- Модель PROPHET от Facebook была представлена в [Taylor S. J. \(2017\)](#). Общая формула для этой модели:

$$y_t = g_t + s_t + h_t + \varepsilon_t,$$

где  $g_t$  — компонента, отвечающая за тренд (модель насыщающего роста или кусочно-линейная),  $s_t$  отвечает за тренд (аппроксимируется рядами Фурье, может включать несколько сезонностей одновременно),  $h_t$  отвечает за выходные дни (скалярное произведение вектора изменений ряда в соответствующий выходной на вектор индикаторов выходного дня).

- Наивная модель — модель, прогноз которой на все шаги равен последнему значению обучающей части временного ряда.

Динамические модели были достаточно детально исследованы с теоретической точки зрения. Разберём важные статьи:

- В работе [Hartley M. J. \(1972\)](#) были непосредственно введены динамические модели (причём как включающие исключительно эндогенные переменные, так и экзогенные); была исследована оптимизация таких моделей и выведены формулы для функций потерь MSE и формула градиента в случае одного регрессора. Также была доказана асимптотическая несмещённость и состоятельность оценок параметров в случае независимых ошибок и асимптотическая несмещённость в случае

линейно зависимых ошибок. Кроме того, в данной статье приводится динамическая форма модели системы одновременных уравнений.

- В работе [Calzolari G. \(1987\)](#) выводится распределение и оценка ковариационной матрицы остатков прогноза для динамической модели системы одновременных уравнений.
- В работе [Johnston et al. \(1974\)](#) динамическая регрессионная модель тестируется против OLS на различных искусственно синтезированных стационарных рядах. Затем делается несколько любопытных выводов, которые мы протестируем в данной работе. Перечислим эти выводы:
  - Определяющие факторы для эффективности прогнозирования — это смещение и разброс оценки параметров, а также длина ряда (то есть, на коротких рядах DLS начинает работать лучше OLS).
  - Ошибка оценки параметра растёт с ростом  $\alpha$ .
  - DLS лучше предсказывает более волатильные ряды, чем OLS.

Однако главные недостатки работы [Johnston et al. \(1974\)](#) по сравнению с нашей — это тестирование на стационарных искусственно синтезированных рядах. Такое тестирование имеет косвенное отношение к использованию DLS в реальных задачах, так как выводы, построенные на теоретических рядах, нуждаются в подтверждении на реальных данных. Кроме того, параметр  $\alpha$  в данной работе был оценён обычным GridSearch, что может быть не вполне эффективно.

- Кроме того, существуют работы с демонстрацией кода для использования DLS: [Gandrud, C. et al. \(2015\)](#), [Williams, L. K., Whitten, G. D. \(2011\)](#). Однако в данных работах код написан на языке R, который не так часто используется в реальном промышленном программировании. В нашей работе весь код будет написан на языках Python и C++, что позволяет использовать наработки для использования в реальных задачах.



## 3 Описание предлагаемого метода

### 3.1 Формулы

Реализованная модель достаточно подробно описана в работе [Hartley M. J. \(1972\)](#). Выглядит она так: Главной особенностью этих моделей является то, что они используют прогнозы прошлых моментов времени вместо лагированных значений зависимой переменной в качестве регрессоров, то есть ряд влияет на оценённую модель лишь своим первым значением. В общем случае формула записывается следующим образом:

$$y_t = \alpha_0 + \alpha_1 y_{t-1} + \dots + \alpha_p y_{t-p} + \varepsilon_t.$$

Также можно представить оцениваемое уравнение (здесь важно подчеркнуть, что  $y_0$  можно либо как-то оценить, либо взять из самого ряда):

$$\hat{y}_t = \hat{\alpha}_0 + \hat{\alpha}_1 \hat{y}_{t-1} + \dots + \hat{\alpha}_p \hat{y}_{t-p}.$$

В статье [Hartley M. J. \(1972\)](#) для оценки параметра предлагается решать полиномиальное уравнение большой степени, но на практике такой подход оказывается неэффективен, поэтому мы хотели бы оценивать параметры с помощью градиентного спуска. Учиться будем на метрику MSE; тогда функционал потерь будет выглядеть как

$$\begin{aligned} L(y, \alpha) &= \frac{1}{|\text{learning\_steps}|} \sum_{i \in \text{learning\_steps}} (\hat{y}_{t+i} - y_{t+i})^2 = \\ &= \frac{1}{|\text{learning\_steps}|} \sum_{i \in \text{learning\_steps}} (\hat{\alpha}_0 + \hat{\alpha}_1 \hat{y}_{t+i-1} + \dots + \hat{\alpha}_p \hat{y}_{t+i-p} - y_{t+i})^2, \end{aligned}$$

где гиперпараметры `learning_steps` — шаги, на которых мы будем обучаться, а  $p$  — глубина авторегрессии. Теперь нужно найти градиент функции потерь MSE по вектору  $\alpha$ . Нетрудно заметить, что (по правилу дифференцирования

произведения) градиент для одного из шагов обучения будет выглядеть как

$$\nabla \hat{y}_t = 2(\hat{y}_t - y_t) \left( \begin{pmatrix} 1 \\ \hat{y}_{t-1} \\ \vdots \\ \hat{y}_{t-p} \end{pmatrix} + \begin{pmatrix} 0 & | & \nabla \hat{y}_{t-1} & | & \dots & | & \nabla \hat{y}_{t-p} \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_p \end{pmatrix} \right).$$

Важно подчеркнуть, что здесь  $\hat{y}_{t-i}$  — это либо прошлые уже сделанные прогнозы модели, либо (например, если мы прогнозируем первый шаг) значения, взятые из обучающей выборки. Та же логика с  $\nabla \hat{y}_{t-i}$  — это либо уже посчитанные градиенты для прошлых прогнозов модели, либо нулевые векторы. Тогда общий градиент будет выглядеть как

$$\nabla \hat{y} = \frac{1}{|\text{learning\_steps}|} \sum_{i \in \text{learning\_steps}} \nabla \hat{y}_{t+i}.$$

Аналогично считается градиент для динамической регрессии с сезонностью — нужно просто добавить нужные регрессоры в формулы.

## 3.2 Гиперпараметры

Для начала опишем параметры, которые наша функция принимает на вход:

- **fh** — горизонт прогнозирования. Он важен нам на этапе обучения, поэтому мы просим указывать его уже в конструкторе.
- **sp** — период сезонности (либо **None** в случае, если сезонности не предполагается).
- **learning\_steps** — на какие шаги обучаться.
- **ar\_depth** — число авторегрессионных шагов, которые будут использоваться в модели.

- `seas_depth` — число сезонных авторегрессионных шагов, которые будут использоваться в модели.
- `fit_intercept` — нужно ли обучать отступ.
- `max_iter` — максимальное число итераций для обучения
- `tolerance` — достаточная точность оптимизации. Пусть  $f_0$  — норма изначального градиента,  $f_k$  — норма текущего градиента. Тогда оптимизация остановится, если  $\frac{f_0^2}{f_k^2} < \text{tolerance}$ .

### 3.3 Обучение

Опишем один шаг в градиентном спуске. Градиент считается следующим образом. Пусть на вход подаётся временной ряд  $y$ , на котором мы хотим обучиться. Введём две величины: пусть `num_steps` — максимальный номер шага, на который нам нужно сделать прогноз, а `max_depth` — максимальный номер лага, являющийся регрессором в нашей модели. Построим обучающую выборку с помощью скользящего окна с длиной `num_steps + max_depth` и шагом 1. Далее посчитаем на каждом элементе выборки градиент, усредним и отнормируем на длину. Это и будет градиент на обучающей выборке. Далее с помощью заранее указанного способа выберем длину шага (она может быть константной, выбираться процедурой бэктрекинга под условия Армихо, либо процедурой линейного поиска под условия Вульфа, либо по формуле  $\alpha_k = \lambda \cdot \left(\frac{s_0}{s_0+k}\right)^p$ ). После этого обновляем наш вектор параметров по антиградиенту с выбранной длиной шага.

### 3.4 Детали реализации

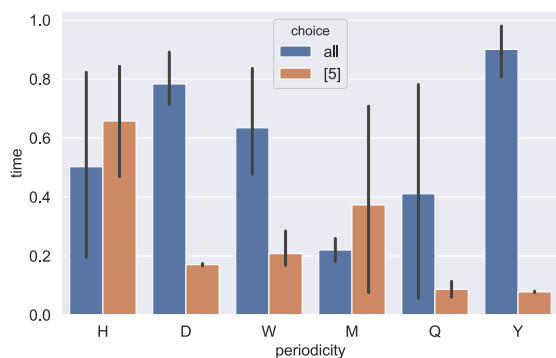
Изначально код был написан полностью на языке Python. Однако из-за медленной работы моделей и большого количества вычислений, необходимых для проведения экспериментов, было решено воспользоваться языком C++ и библиотекой Boost.Python. Благодаря такому решению скорость работы кода

увеличилась примерно в 100 раз и проведение экспериментов стало возможным. Для того, чтобы гарантировать верную работу написанного кода, были также проведены юнит-тесты.

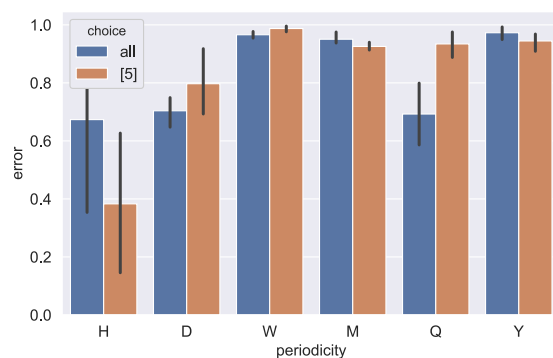
### 3.5 Подбор гиперпараметров

Также мной был проведён мини-эксперимент с целью подбора правильных гиперпараметров динамической регрессии по умолчанию. Для этого были взяты ряды H1, D1, W1, M1, Q1 и Y1 из выборки M4 и на этих рядах тестировал скорость сходимости метода и ошибку на тестирующей выборке в зависимости от значения очередного гиперпараметра. Каждый ряд обучался по 5 раз, чтобы получить более статистически значимые результаты как по времени работы, так и по качеству (стартовые параметры почти во всех экспериментах выбирались случайно). По очереди перебирались гиперпараметры в порядке `learning_steps`, `ar_depth`, `seas_depth`, `fit_intercept`, `tolerance`, `max_iter`, `starting_params` (стартовый вектор параметров в оптимизации), `lr_finder` (метод поиска длины шага), для каждого подбирал лучшее, то есть, дающее максимальное качество при минимальных временных затратах, значение, и фиксировал его для следующих гиперпараметров. Опишем результаты по порядку:

- 1) Значение гиперпараметра `learning_steps` выбиралось из двух опций: либо обучаемся на всех `num_steps` шагах, либо обучаемся на шагах, на которые будем прогнозировать, то есть на шагах из `fh`. По результатам эксперимента оказалось, что мы достигаем примерно равного качества на этих опциях, но времени быстрее обучать модель на шагах из `fh`. Результаты тестирования можно посмотреть на графиках [3.1](#).



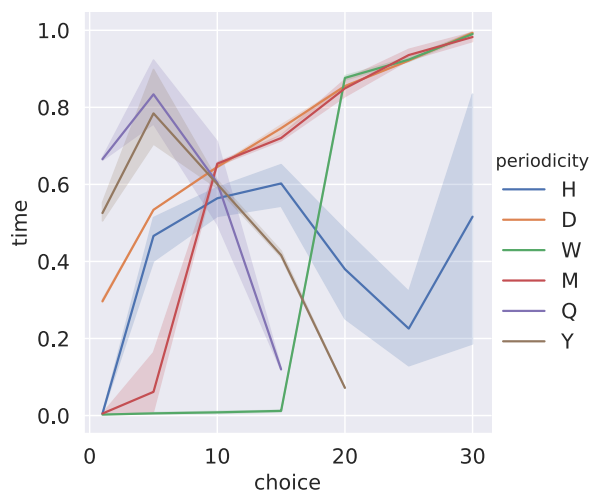
(a) Время до сходимости против выбора обучающих шагов



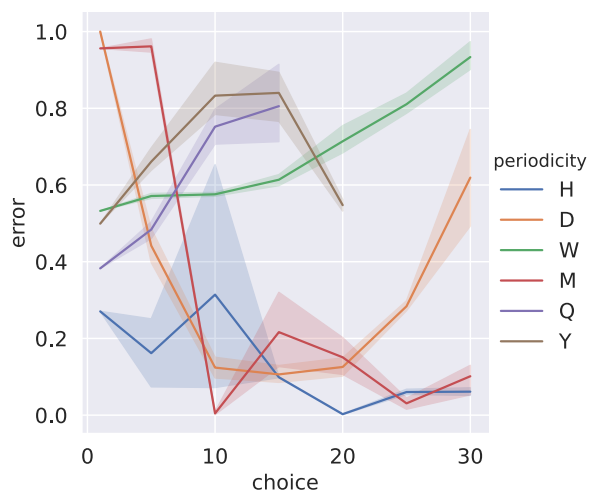
(b) MSE на тестирующей выборке против выбора обучающих шагов

Рис. 3.1: Сравнение моделей с различным выбором обучающих шагов

2) Значение гиперпараметра `ar_depth` подбиралось по сетке [1, 5, 10, 15, 20, 25, 30]. Как видно на графиках 3.2, на половине выбранных рядов лучший MSE случается при `ar_depth` = 1, поэтому именно это значение было зафиксировано. Тем не менее, в случае `ar_depth` нужно понимать, что этот гиперпараметр нужно подбирать отдельно для каждого временного ряда.



(a) Время до сходимости против глубины авторегрессии

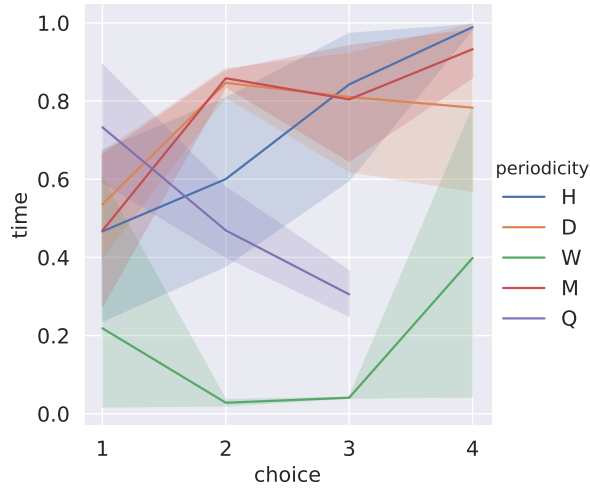


(b) MSE на тестирующей выборке против глубины авторегрессии

Рис. 3.2: Сравнение моделей с различной глубиной авторегрессии

3) Для тестирования гиперпараметра `seas_depth` отдельно для каждой периодичности, кроме годовой (годовые данные в этом тестировании не участвовали в силу того, что они обычно не сезонны) были выбра-

ны некоторые стандартные периоды сезонности – для часовых данных 24, для дневных 30, для недельных 4 (месяц), для месячных 12, для квартальных 4. После этого гиперпараметр `seas_depth` перебирался по сетке [1, 2, 3, 4]. Как результат получаем (с графиков 3.3), что лучший `seas_depth` = 1.

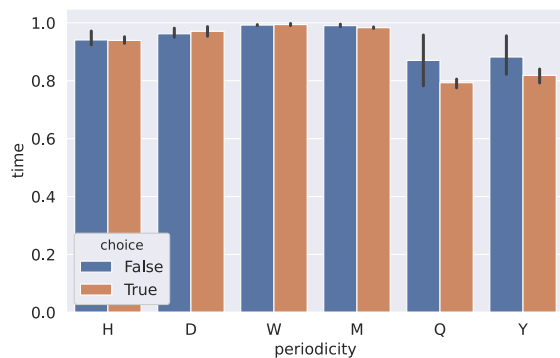


(a) Время до сходимости против глубины сезонной авторегрессии

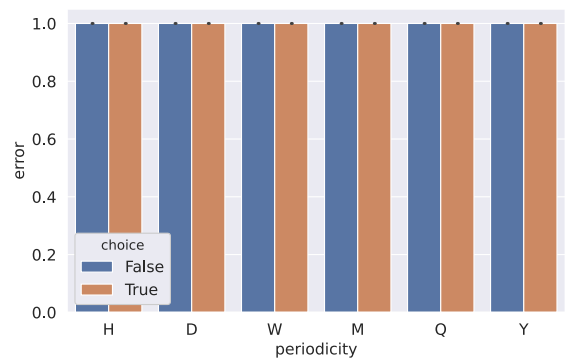
(b) MSE на тестирующей выборке против глубины сезонной авторегрессии

Рис. 3.3: Сравнение моделей с различной глубиной сезонной авторегрессии

4) Понятно, что гиперпараметр `fit_intercept` выбирался среди значений [True, False]. По графикам 3.4 видно, что модели почти не отличаются по MSE, но при этом немного быстрее обучается модель, в которой отступ обучается, поэтому решили обучать отступ.



(a) Время до сходимости против выбора обучения отступа



(b) MSE на тестирующей выборке против выбора обучения отступа

Рис. 3.4: Сравнение моделей с обученным отступом и без

5) Параметр `tolerance` подбирался по логарифмической сетке от 1 до  $10^{-8}$ . По графикам 3.5 видно, что MSE на тестирующей выборке для 66% рядов выросло, когда мы хоть немного начали обучать модель, и из-за этого лучшим выбором, возможно, был бы `tolerance` = 1. Однако это бы значило, что мы совсем ничего не обучаем, поэтому было решено взять `tolerance` всё-таки поменьше. Видно, что также на 66% рядов качество на тестирующей выборке немного растёт при `tolerance` =  $10^{-2}$ , поэтому было решено взять именно такую точность.

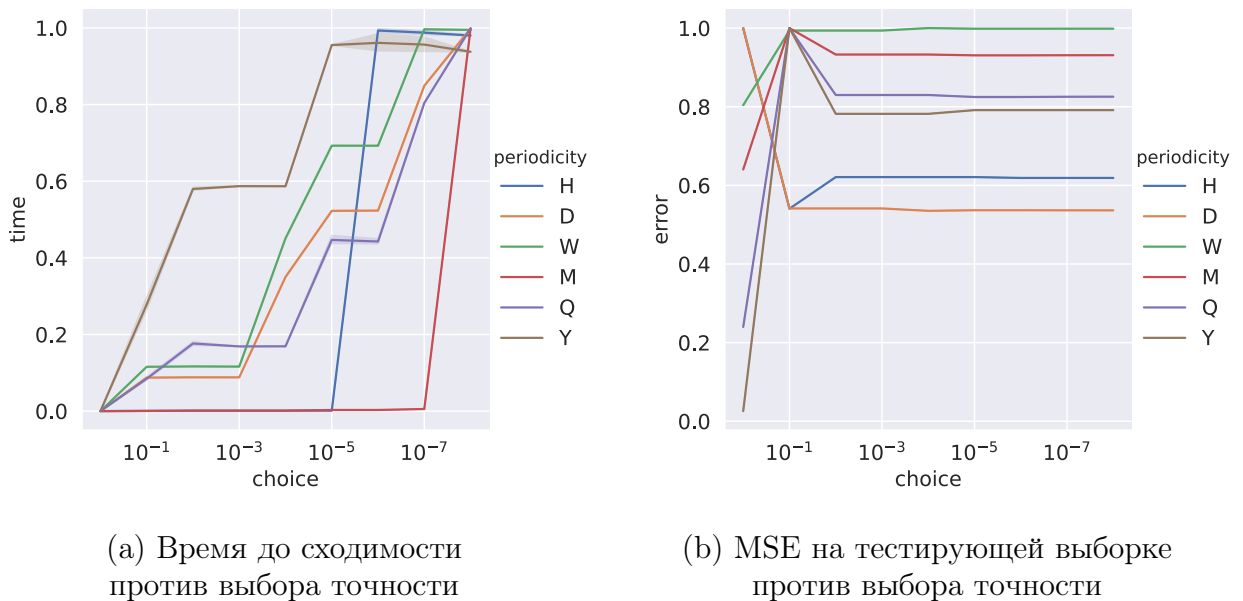


Рис. 3.5: Сравнение моделей, обученных с разной точностью

- 6) Задача параметра `max_iter` — быть достаточно большим, чтобы большинство рядов сходилось к заданной точности, но не быть слишком большим, чтобы модель не зависала на “плохих” рядах. В ходе эксперимента было выяснено, что все 6 рядов сходятся к заданной точности за 10 итераций, таким образом, это значение было выбрано как значение по умолчанию.
- 7) Мы разобрались с гиперпараметрами самой модели, теперь надо разобратся с параметрами обучения. Первый гиперпараметр — это метод выбора стартового вектора параметров. Опции для этого гиперпараметра такие:

- Выбирать стартовые параметры случайно (и отнормировать, чтобы сумма была равна 1);
- Сделать их равными частным автокорреляциям значений ряда с соответствующими лагами;
- Сделать равным частным автокорреляциям и отнормировать.

Так как бесполезно сравнивать эти методы с  $\text{ar\_depth} = 1$  (нормирование приводит к тому, что первый и третий методы совпадают), хотя это и сделано на графиках 3.6, я попробовал сравнить их с  $\text{ar\_depth} = 15$  на графиках 3.7. Получилось, что с точки зрения качества выгоднее всего выбирать стартовый набор параметров случайно.

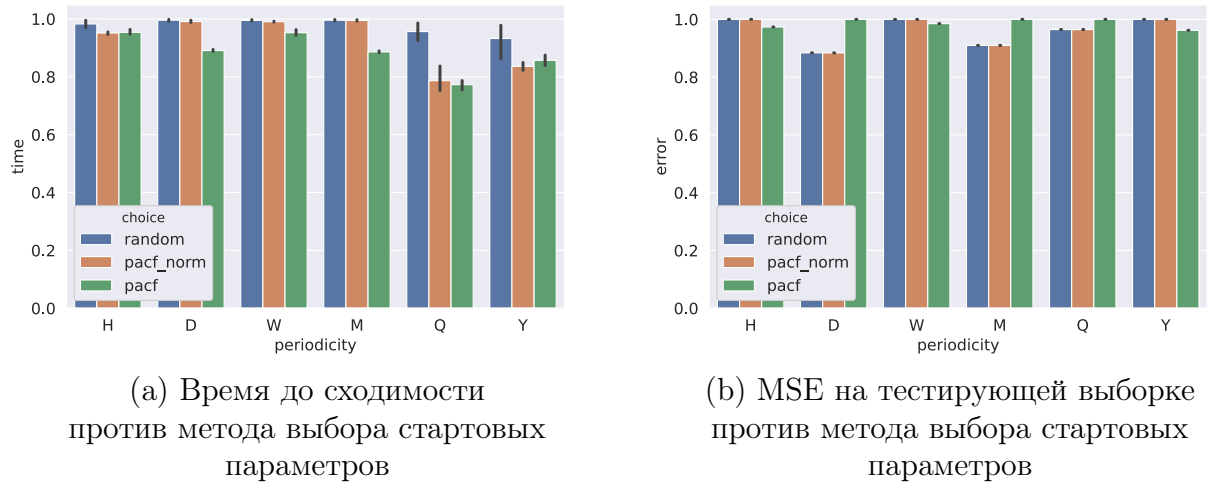
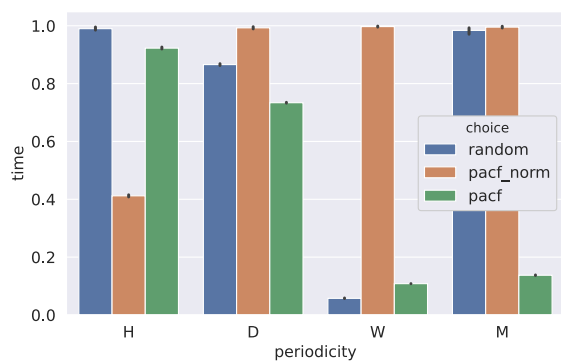


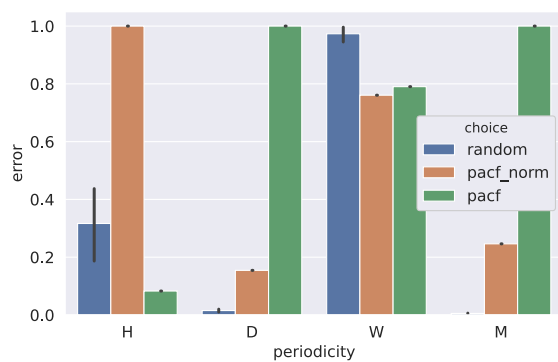
Рис. 3.6: Сравнение моделей с различным методом выбора стартовых параметров ( $\text{ar\_depth} = 1$ )

- 8) Также были отобраны метод выбора длины шага как гиперпараметр по умолчанию. Этот гиперпараметр, в теории, никак не должен влиять на качество модели, а только на время сходимости, поэтому я построил график времени сходимости против выбора длины шага: 3.8. Оказалось, что для динамических моделей быстрее всего выбирать длину шага по формуле  $\alpha_k = \lambda \cdot \left( \frac{s_0}{s_0 + k} \right)^p$ , где  $\lambda = 5 \cdot 10^{-3}$ ,  $s_0 = 1$ ,  $p = 0.5$ .





(a) Время до сходимости против метода выбора стартовых параметров



(b) MSE на тестирующей выборке против метода выбора стартовых параметров

Рис. 3.7: Сравнение моделей с различным методом выбора стартовых параметров ( $\text{ar\_depth} = 15$ )

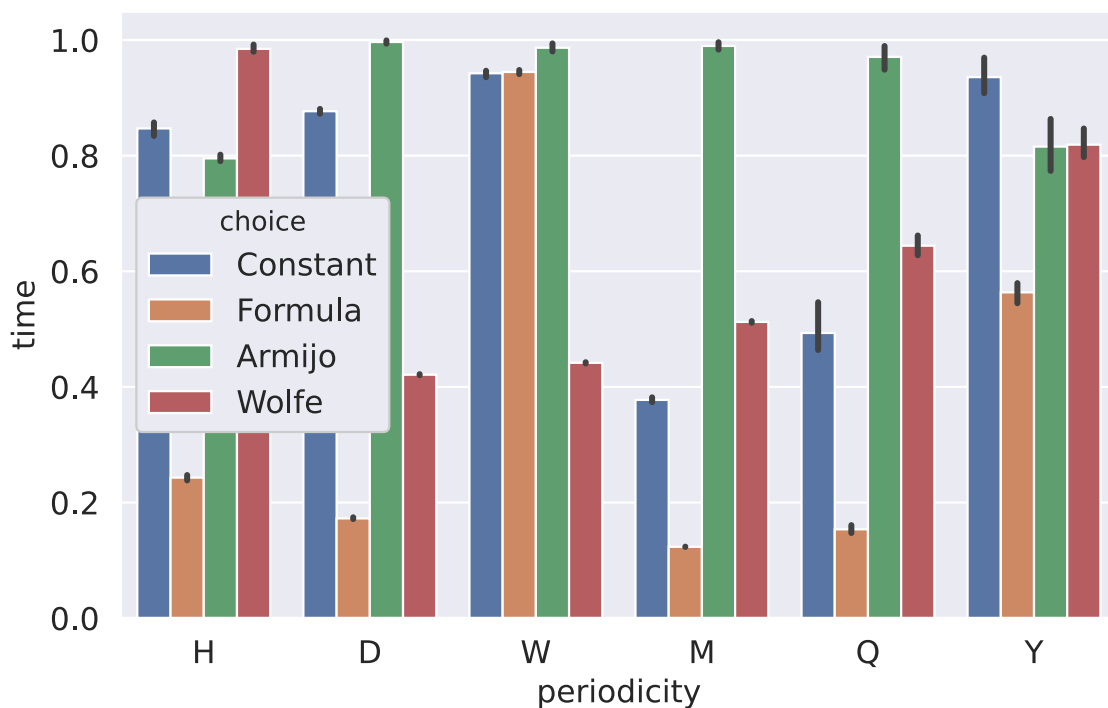


Рис. 3.8: Время до сходимости моделей с различным методом выбора длины шага

## 4 Описание поставленного эксперимента

### 4.1 Данные

Для сравнения динамических моделей с базовыми методами были использованы временные ряды из выборки М4. Данная выборка содержит 100000 рядов разной периодичности и из разных источников. В деталях можно отобразить разбиение рядов по категориям с помощью таблицы: [4.1](#).

Таблица 4.1: Категоризация рядов в выборке М4

	Demographic	Finance	Industry	Macro	Micro	Other	Total
Yearly	1088	6519	3716	3903	6538	1236	23000
Quarterly	1858	5305	4637	5315	6020	865	24000
Monthly	5728	10987	10017	10016	10975	277	48000
Weekly	24	164	6	41	112	12	359
Daily	10	1559	422	127	1476	633	4227
Hourly	0	0	0	0	0	414	414
Total	8708	24354	18798	19402	25121	3437	100000

Данные предобрабатывались следующим образом. Мы хотели получить разбиение рядов по нескольким признакам, а именно периодичности, источнику, наличию сезонности (считалось, что ряд сезонный, если сила сезонности в алгоритме разложения STL ([Cleveland R. B. et al. \(1990\)](#)) больше или равна 0.2) и стационарности ряда по тренду [image.pdf](#)(стационарность тестировалась с помощью теста KPSS ([Kwiatkowski D. et al. \(1992\)](#))). При этом для каждого “кластера” нам нужно было провести статистические тесты о доле, чтобы сравнить нашу модель с какой-то другой. Для этого нам нужно было, чтобы в каждом кластере было достаточно много рядов, например, от 200. Но при этом мы не хотели тратить слишком много времени на вычисления, поэтому все кластеры, которые содержали больше 200 рядов, обрезались до 200 рядов. Итоговую выборку и разбиение на кластеры можно описать этой таблицей: [4.2](#). В ней бинарные значения в ячейках означают, был ли данный кластер взят в выборку. Столбцы у/н означают, являются ли ряды в соответствующем кластере стационарными; строки у/н – являются ли они

сезонными. Например, единица в строке Yearly/y и столбце Demographic/n означает, что в анализ был включён кластер сезонных нестационарных рядов с годовой периодичностью из демографических данных.

Таблица 4.2: Расширенная категоризация рядов в выборке M4

		Demographic		Finance		Industry		Macro		Micro		Other		Total
		y	n	y	n	y	n	y	n	y	n	y	n	
Yearly	y	0	1	1	1	1	1	1	1	1	1	1	1	11
	n	0	1	0	1	0	1	0	1	0	1	0	0	5
Quarterly	y	1	1	1	1	1	1	1	1	1	1	0	1	11
	n	0	1	0	1	1	1	1	1	1	1	0	0	8
Monthly	y	1	1	1	1	1	1	1	1	1	1	0	0	10
	n	0	1	1	1	1	1	1	1	1	1	0	0	9
Weekly	y	0	0	0	0	0	0	0	0	0	0	0	0	0
	n	0	0	0	0	0	0	0	0	0	0	0	0	0
Daily	y	0	0	0	1	0	1	0	0	0	1	0	1	4
	n	0	0	0	0	0	0	0	0	0	0	0	0	0
Hourly	y	0	0	0	0	0	0	0	0	0	0	0	0	0
	n	0	0	0	0	0	0	0	0	0	0	0	1	1
Total		2	6	4	7	5	7	5	6	5	7	1	4	59

## 4.2 Постановка эксперимента

Данный эксперимент ставит перед собой две основные задачи:

- 1) Статистическое сравнение динамических моделей с базовыми методами как в целом, так и на отдельных кластерах рядов;
- 2) Выявление свойств рядов и горизонтов прогнозирования, на которых динамические регрессионные модели выдают лучшее качество относительно базовых методов.

Чтобы выполнить эти задачи, для начала для каждого ряда из собранных кластеров и для каждого горизонта прогнозирования были запущены следующие модели:

- Базовые, то есть наивная, ETS, ARIMA, PROPHET;

- Обычная динамическая регрессия, причём глубина авторегрессии перебиралась от 1 до 5, то есть всего таких моделей было запущено по пять на каждый ряд;
- Сезонная динамическая регрессия, причём глубина авторегрессии перебиралась от 1 до 5, а глубина сезонной авторегрессии — от 1 до 2, то есть всего таких моделей было запущено по десять на каждый ряд.

При этом горизонты прогнозирования перебирались следующие:

- Одношаговый;
- Многошаговый (при чём номер самого дальнего шага определялся периодичностью ряда — так, для часовых данных это 24, для дневных — 30, для месячных — 12, для квартальных — 4, а для годовых — 10);
- Прогноз на дальний шаг (последний среди соответствующего многошагового прогноза).

По вышеописанным правилам (как для горизонта прогнозирования) выбирался и период сезонности для сезонных динамических моделей; при этом для годовых данных сезонные динамические модели не запускались. После запуска моделей динамических регрессий для них брался минимум ошибки по разным параметрам. То есть, на выходе в столбцах таблицы должны быть базовые методы, а также динамическая регрессия и сезонная динамическая регрессия. Для базовых методов брались следующие реализации (все из библиотеки `sktime`): для ETS — `sktime.forecasting.ets.AutoETS`, для ARIMA — `sktime.forecasting.arima.AutoARIMA`, для PROPHET — `sktime.forecasting.fbprophet.Prophet`, для наивной модели — `sktime.forecasting.naive.NaiveForecaster`. Дополнительно можно сказать о том, что если ряд оказывался слишком коротким, чтобы можно было эффективно обучиться, то он не рассматривается.

Таким образом, после применения моделей и некоторых преобразований данных мы получим таблицу, в которой для каждого кластера и каждой мо-

дели лежит список ошибок модели на рядах из данного кластера. Мы считаем следующие статистики. Во-первых, для каждой модели мы считаем процент побед на рядах из данного кластера (то есть процент рядов, которые эта модель лучше всего предсказывает), а также по наибольшему числу побед выбираем лучшую модель на данном кластере. Во-вторых, для каждого базового метода мы проверяем гипотезу о доле, то есть гипотезу о том, что с вероятностью 0.5 динамическая модель показывает не худшее качество, чем рассматриваемый базовый метод, с уровнем значимости 0.95. Можно сказать, что таким образом мы проверяем гипотезу о том, что динамическая модель работает не хуже, чем рассматриваемый базовый метод. Опишем это более формально. Пусть доля побед динамической регрессии на рядах из кластера равна  $P$ . Тогда мы проверяем следующие гипотезы:

$$\begin{cases} H_0 : P = 0.5 \\ H_1 : P < 0.5 \end{cases}$$

В-третьих, на всех данных, кроме годовых, мы проверяем эти гипотезы и для сезонной динамической регрессии.

### 4.3 Проведение эксперимента

Вычисления в данном эксперименте запускались одновременно на двух машинах – это Apple MacBook Pro (2021) с чипом Apple M1 Pro и оперативной памятью 32 GB на 9 потоках, а также на суперкомпьютере департамента прикладной экономики НИУ ВШЭ с оперативной памятью 251 GB на 47 потоках. Вычисления были выполнены в сумме примерно за 42 часа. Параллельно здесь запускались именно процессы обучения моделей, это было выполнено с помощью библиотеки `multiprocessing`.

## 5 Анализ полученных результатов

### 5.1 Сравнение моделей

В данном разделе мы будем строить столбчатые диаграммы, на которых столбцами будут доли неотвергнутых гипотез о доле среди подмножества кластеров, объединённых по какому-то из признаков. Для начала нарисуем такие графики на всех кластерах для динамической и сезонной динамической регрессий: 5.1, 5.2. Видим, что обе модели имеют большой процент неотвер-

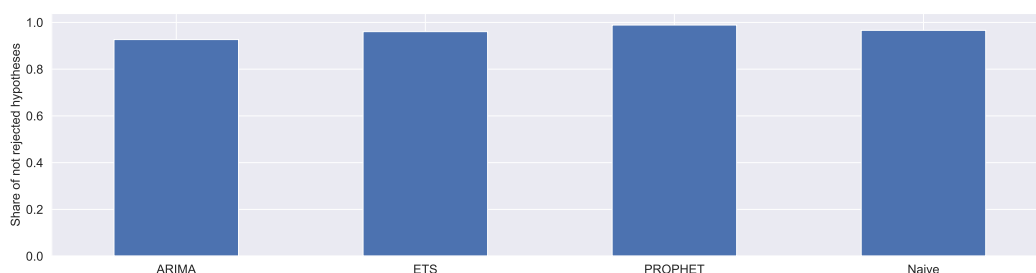


Рис. 5.1: Доли неотвергнутых гипотез о том, что динамическая регрессия не хуже рассматриваемого базового метода

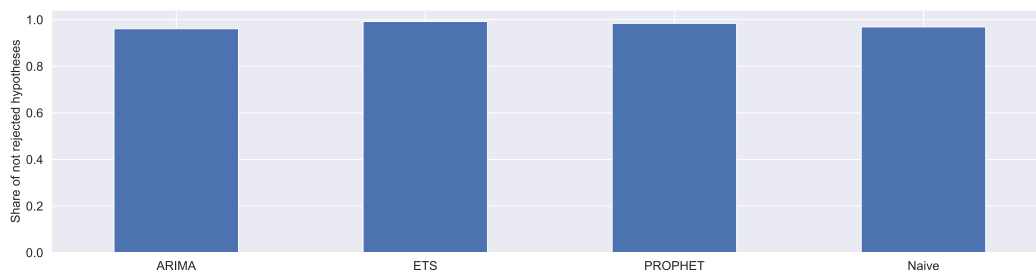


Рис. 5.2: Доли неотвергнутых гипотез о том, что сезонная динамическая регрессия не хуже рассматриваемого базового метода

гаемых гипотез, что говорит об успешной работе этих моделей. Рассмотрим теперь сравнение динамических моделей с базовыми методами в зависимости от различных признаков:

#### 5.1.1 Периодичность

Изобразим столбчатые диаграммы с отдельным измерением периодичности: 5.3, 5.4. Мы видим, что общая тенденция сохраняется, но на часовых

рядах (вероятно, самых волатильных) качество моделей относительно базовых методов сильно падает.

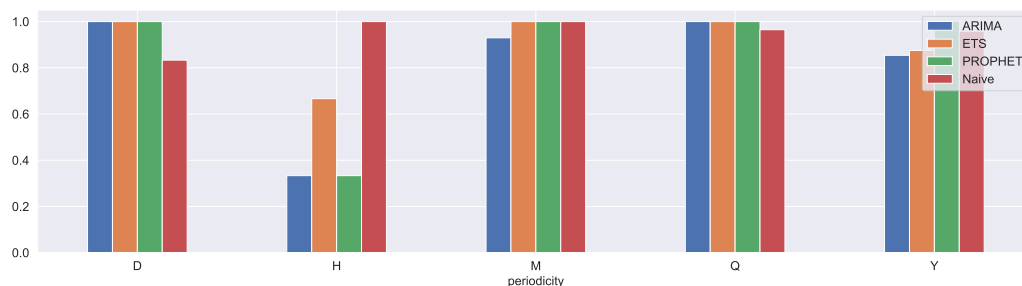


Рис. 5.3: Доли неотвергнутых гипотез о том, что динамическая регрессия не хуже рассматриваемого базового метода в зависимости от периодичности рядов

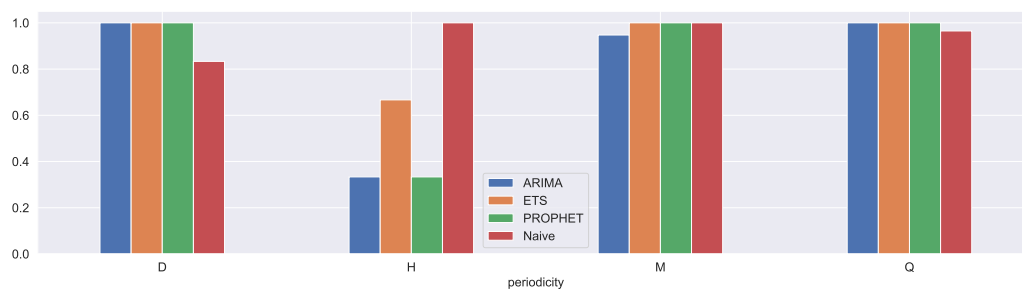


Рис. 5.4: Доли неотвергнутых гипотез о том, что сезонная динамическая регрессия не хуже рассматриваемого базового метода в зависимости от периодичности рядов

### 5.1.2 Категория источника

Изобразим графики с отдельным измерением категории источника: 5.5, 5.6. Мы видим падение качества по категории Other. Сложно судить, с чем это может быть связано, но одна из гипотез — достаточно большая часть рядов в категории Other — часовые, и из-за этого в целом категория более волатильна, чем остальные. Кроме того, можем заметить, что методы ARIMA и ETS примерно в 20% случаев дают лучшее качество, чем динамические регрессии, на макроэкономических рядах.

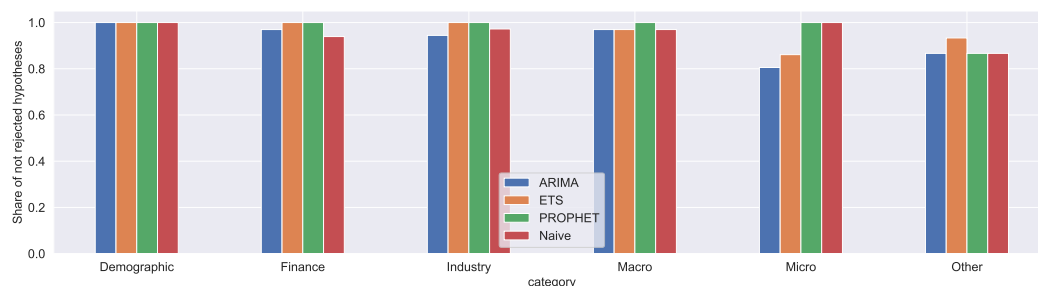


Рис. 5.5: Доли неотвергнутых гипотез о том, что динамическая регрессия не хуже рассматриваемого базового метода в зависимости от категории рядов

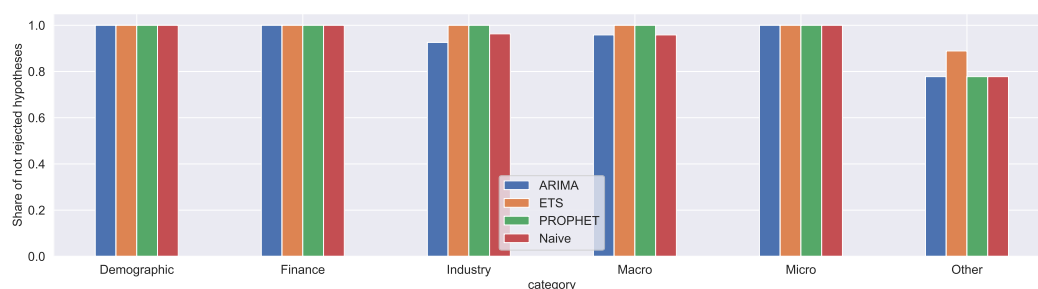


Рис. 5.6: Доли неотвергнутых гипотез о том, что сезонная динамическая регрессия не хуже рассматриваемого базового метода в зависимости от категории рядов

### 5.1.3 Сезонность

Нарисуем результаты на рядах в зависимости от сезонности: [5.7](#), [5.8](#). Наверное, единственное, что выделяется из этого графика — это то, что на несезонных данных ARIMA начинает работать относительно динамических моделей лучше примерно на 10%, чем на сезонных. Кроме того, можно отметить, что на сезонных данных сезонная динамическая регрессия статистически оказывается лучше почти всех базовых моделей в 100% случаях.

### 5.1.4 Стационарность

Изобразим графики с отдельным измерением стационарности: [5.9](#), [5.10](#). На мой взгляд, между двумя диаграммами больших изменений нет. Как в случае стационарных рядов, так и в случае нестационарных, динамические модели почти всегда работают не хуже базовых методов.



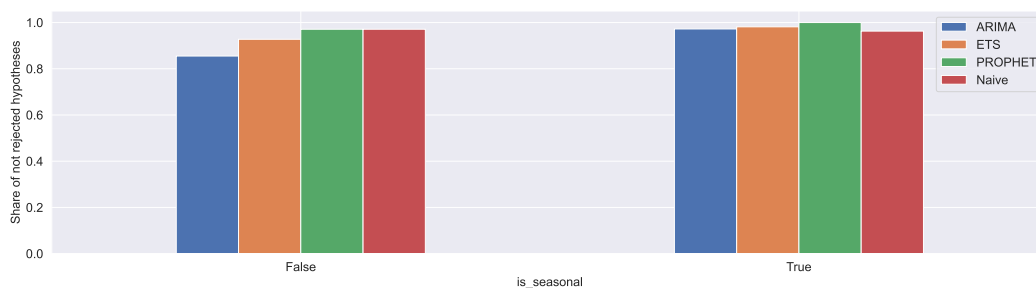


Рис. 5.7: Доли неотвергнутых гипотез о том, что динамическая регрессия не хуже рассматриваемого базового метода в зависимости от сезонности рядов

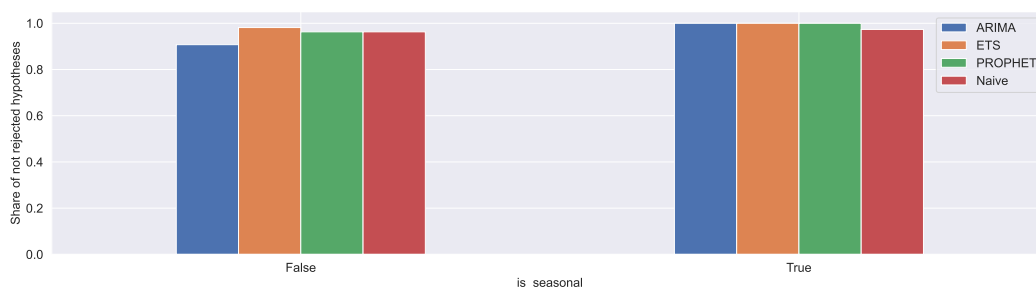


Рис. 5.8: Доли неотвергнутых гипотез о том, что сезонная динамическая регрессия не хуже рассматриваемого базового метода в зависимости от сезонности рядов

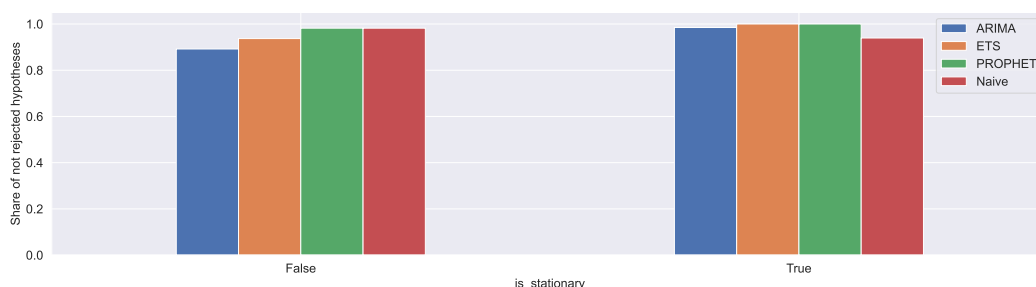


Рис. 5.9: Доли неотвергнутых гипотез о том, что динамическая регрессия не хуже рассматриваемого базового метода в зависимости от стационарности рядов

### 5.1.5 Горизонт прогнозирования

Нарисуем графики с отдельным измерением для горизонта прогнозирования: 5.11, 5.12. На обоих графиках мы видим, что лучшие результаты относительно базовых методов динамические модели показывают именно на дальнем шаге. При этом хуже всего они работают, если нужно до этого дальнего шага спрогнозировать весь путь.

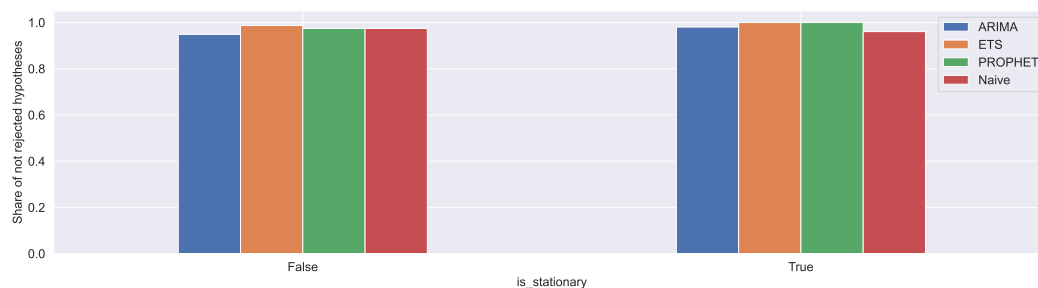


Рис. 5.10: Доли неотвергнутых гипотез о том, что сезонная динамическая регрессия не хуже рассматриваемого базового метода в зависимости от стационарности рядов

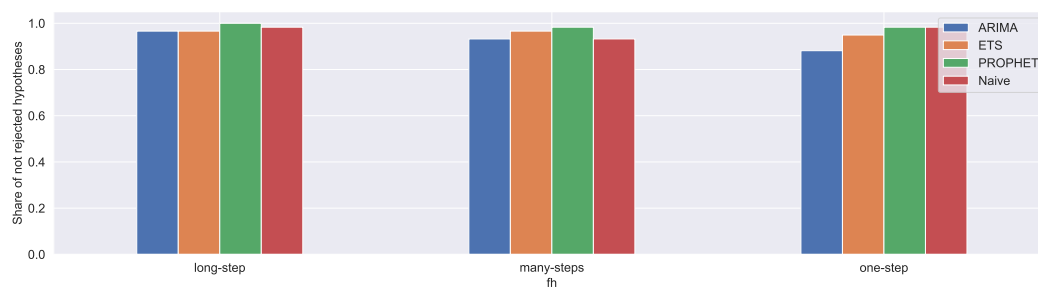


Рис. 5.11: Доли неотвергнутых гипотез о том, что динамическая регрессия не хуже рассматриваемого базового метода в зависимости от горизонта прогнозирования

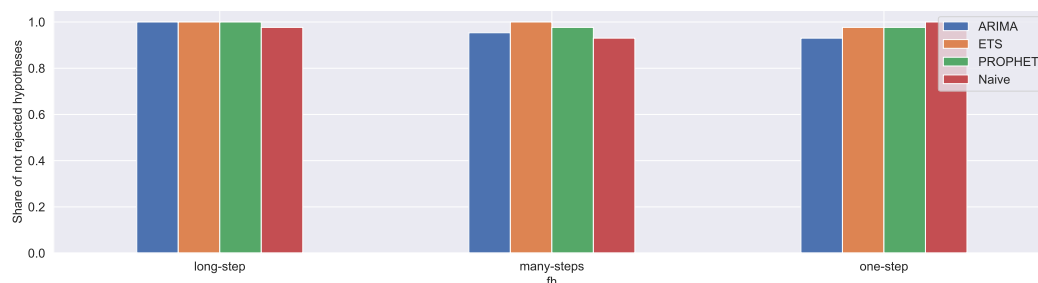


Рис. 5.12: Доли неотвергнутых гипотез о том, что сезонная динамическая регрессия не хуже рассматриваемого базового метода в зависимости от горизонта прогнозирования

Таким образом, во всех рассмотренных случаях динамические модели работают в основном не хуже, чем рассмотренные базовые методы.

## 5.2 Выявление благоприятных временных рядов

Теперь постараемся найти свойства временных рядов, на которых динамические модели справляются лучше не какой-то конкретной модели, а сразу

всех моделей в целом. Для этого мы будем для каждого множества кластеров изображать круговую диаграмму с процентом побед для каждой модели. Начнём с круговой диаграммы по всем кластерам: 5.13. На данной диаграмме

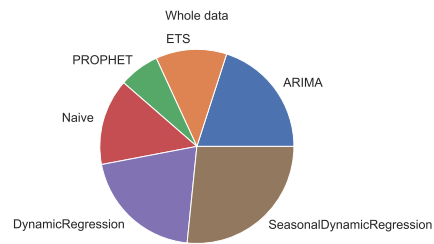


Рис. 5.13: Проценты побед рассматриваемых моделей

примерно 30% побед у сезонной динамической регрессии, и таким образом её можно признать самой сильной моделью из рассмотренных. У обычной динамической регрессии на всех кластерах около 23% побед, что тоже является достойным показателем. Рассмотрим более подробно влияние различных признаков на качество реализованных моделей:

### 5.2.1 Периодичность

Изобразим проценты побед в зависимости от периодичности рядов: 5.14. На графике мы видим, что периодичность может повлиять на качество таких моделей: на часовых данных они начинают работать хуже относительно в основном модели PROPHEIT. Самую стабильную работу на изображённом графике можно видеть при дневных, квартальных и годовых данных.

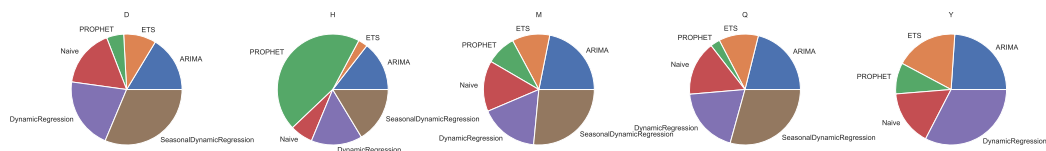


Рис. 5.14: Проценты побед рассматриваемых моделей в зависимости от периодичности рядов

### 5.2.2 Категория источника

Нарисуем результаты в зависимости от категории рядов: 5.15. Мы видим, что корреляции между источниками данных и особенностями работы динамических моделей не обнаруживается.

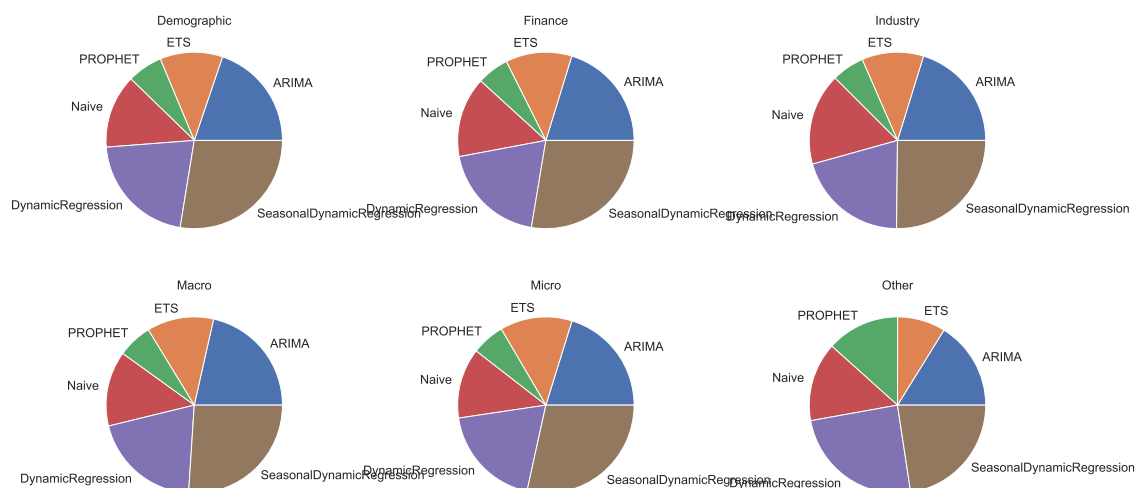


Рис. 5.15: Проценты побед рассматриваемых моделей в зависимости от категории рядов

### 5.2.3 Сезонность

Выведем графики процента побед моделей в зависимости от сезонности рядов: 5.16. Мы действительно видим, что в случае сезонности обрабатываемых рядов качество динамических моделей вырастает до 50% побед совместно у динамической и сезонной динамической регрессий. Таким образом, этот фактор может повлиять на качество прогнозов.

### 5.2.4 Стационарность

Изобразим результаты с разделением по стационарности: 5.17. Мы видим, что этот фактор очень слабо влияет на качество динамических моделей, так как две диаграммы практически одинаковы.



Рис. 5.16: Проценты побед рассматриваемых моделей в зависимости от сезонности рядов



Рис. 5.17: Проценты побед рассматриваемых моделей в зависимости от стационарности рядов

### 5.2.5 Горизонт прогнозирования

Наконец, покажем работу моделей относительно горизонта прогнозирования: 5.18. Мы видим, что хуже всего динамические модели справляются спрогнозировать весь путь до дальнего шага, а лучше всего получается прогнозировать на один шаг с больше чем 50% побед.

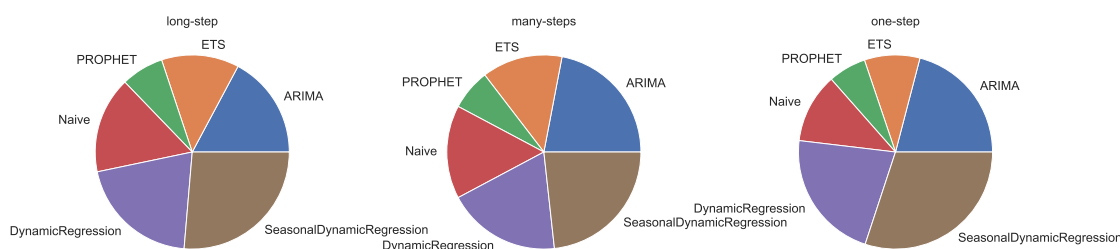


Рис. 5.18: Проценты побед рассматриваемых моделей в зависимости от горизонта прогнозирования

Таким образом, динамические модели действительно можно признать достаточно сильными. Однако необходимо сказать, что здесь есть некоторый сдвиг, заключающийся в том, что в эксперименте более тщательно прово-

дился подбор гиперпараметров для динамических моделей, чем для базовых методов. Таким образом, у динамических моделей изначально было некоторое преимущество. Основные факторы, которые могут влиять на качество их работы — это периодичность графика, его сезонность и горизонт прогнозирования.

## 6 Заключение

В завершение работы можно ещё раз подчеркнуть, что динамические модели способны показывать достаточно сильное качество и уверенно обходить базовые методы, которые были представлены в данной статье. Также были выявлены оптимальные свойства рядов для прогнозирования с помощью динамических моделей: в идеале ряд должен быть дневным, квартальным или годовым, он должен быть сезонным, а также горизонт прогнозирования не должен включать в себя большой путь.

Из направления дальнейшей работы можно указать реализацию более эффективного способа оценки параметров, например, L-BFGS; можно добавить в модель экзогенные переменные; поработать над качеством кода, в том числе провести дополнительный рефакторинг и добавить юнит-тесты; добавить сравнение с методами из машинного обучения, например, с градиентным бустингом.

## Список литературы (или источников)

1. Hartley, M. J. Optimum Simulation Path Estimators. In International Economic Review (1972).
2. Calzolari, G. Forecast Variance in Dynamic Simulation of Simultaneous Equation Models. In Econometrica (1987).
3. Gandrud, C., Williams, L. K., & Whitten, G. D. Dynamic Simulations of Autoregressive Relationships in R with dynsim. (2015).
4. Williams, L. K., Whitten, G. D. Dynamic Simulations of Autoregressive Relationships. In The Stata Journal (2011).
5. Johnston, H. N., Klein, L., & Shinjo, K. Estimation and Prediction in Dynamic Econometric Models. In Essays in honor of Jan Tinbergen (1974). London: Macmillan.
6. Brown R. G. Exponential Smoothing for Predicting Demand. (1956).
7. Holt C. C. Forecasting Trends and Seasonal by Exponentially Weighted Averages. In Office of Naval Research Memorandum (1957).
8. Box, G., Jenkins, G. Time Series Analysis: Forecasting and Control. In San Francisco: Holden-Day (1970).
9. Taylor, S. J, Letham, B. Forecasting at Scale. In PeerJ Preprints (2017).
10. Cleveland, R. B., Cleveland, W. S, McRae, J. E., Terpenning, I. STL: A Seasonal-Trend Decomposition Procedure Based on LOESS. In Journal of Official Statistics (1990).
11. Kwiatkowski, D., Phillips, P. C. B., Schmidt, P., Shin, Y. Testing the null hypothesis of stationarity against the alternative of a unit root. In Journal of Econometrics (1992).