Оглавление

Глава 1. Теоретический обзор

1.1. Временные ряды

Временной ряд – ряд значений одной и той же переменной, полученных в результате измерений, произведенных в последовательные моменты (периоды) времени. Временной ряд обычно интерпретируется как одна из возможных реализаций набора зависимых случайных величин, представляющей собой случайный (стохастический) процесс с дискретным временем.

Обозначения:

- Y_t – значение переменной Y в момент времени t
- Выборка: $Y_1, \dots, Y_t$ – T наблюдений случайной переменной Y_t
- k-ый лаг переменной Y_t – это Y_{t-k}
- Первая разность переменной: $\Delta Y_t = Y_t - Y_{t-1}$
- Темп прироста переменной: $\frac{\Delta Y_t}{Y_{t-1}}$
- Лаговый оператор: $LY_t = Y_{t-1} (L^k Y_t = Y_{t-k})$

Стационарность временного ряда означает неизменность его вероятностных характеристик во времени (в которое этот ряд наблюдается). Именно неизменность поведения во времени позволяет строить прогнозы стационарных временных рядов на основе их предыстории.

Выделяют два типа стационарности: стационарность в узком смысле (строгая) и в широком (слабая).

Временной ряд называется *стационарным в узком смысле (строго стационарным)*, если совместное распределение m наблюдений $Y_{t_1}, Y_{t_2}, \dots, Y_{t_m}$ не зависит от сдвига по времени (то есть совпадает с распределением $Y_{t_1+k}, Y_{t_2+k}, \dots, Y_{t_m+k}$ для любых $m, t_1, \dots, t_m, k$).

Временной ряд называется *стационарным в широком смысле (слабо стационарным)*, если для всех t:

$$E(Y_t) = \mu < \infty$$

$$V(Y_t) = \gamma_0$$

$$Cov(Y_t, Y_{t-k}) = \gamma_k$$

(μ, γ_0 и γ_k не зависят от t)

Коэффициент автокорреляции k-го порядка: $\rho_k = Corr(Y_t, Y_{t-k}) = \frac{\gamma_k}{\gamma_0}$, ρ_k как функция от k называется автокорреляционной функцией (ACF), ее график называется *коррелограммой*. Функция и ее график используются для анализа свойств временного ряда.

1.2. Одномерные модели временных рядов

В одномерных моделях временных рядов текущее значение ряда зависит только от его предыстории, поэтому такие модели удобны для целей прогнозирования.

Рассмотрим важные примеры одномерных случайных процессов.

1.2.1. Белый шум

Процессом белого шума мы будем называть последовательность независимых и одинаково распределенных случайных величин с нулевым математическим ожиданием и дисперсией σ^2: $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{t-1}, \varepsilon_t, \dots$

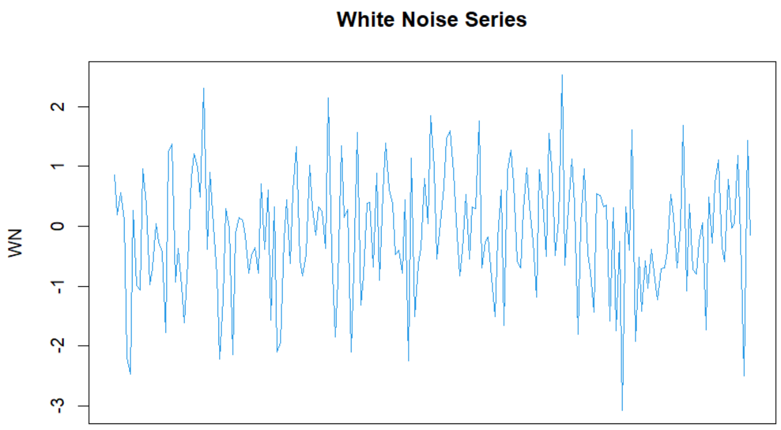

Рисунок 1. Белый шум.

Временной ряд, соответствующий процессу белого шума, ведет себя крайне нерегулярным образом из-за некоррелированности при $t \neq s$ случайных величин X_t и X_s. В связи с этим, процесс белого шума не подходит для непосредственного моделирования эволюции большинства временных рядов, встречающихся в жизни. В то же время, такой процесс является базой для построения более реалистичных моделей временных рядов.

1.2.2. Процесс авторегрессии (AR-процесс)

Одной из широко используемых моделей временных рядов является процесс авторегрессии. Временной ряд является авторегрессионным, если текущее значение может быть получено с использованием предыдущих значений того же ряда. Текущее значение – средневзвешенное прошлых значений.

- AR(1) – процесс авторегрессии 1-го порядка

$$y_t = \delta + \theta y_{t-1} + \varepsilon_t, |\theta| < 1, \varepsilon_t \text{ – белый шум}$$

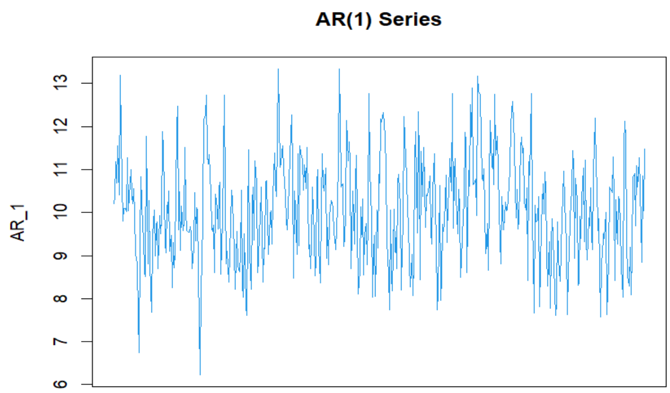

Рисунок 2. AR(1) процесс с $\theta = 0.5$

- AR(p) – процесс авторегрессии p-го порядка

$$y_t = \delta + \theta_1 y_{t-1} + \dots + \theta_p y_{t-p} + \varepsilon_t, |\theta_k| < 1,$$

$$\varepsilon_t \text{ – белый шум, } y_{t-p} \text{ – лаги, } k = 1, \dots, p$$

Характеристическое уравнение:

$$1 - \theta_1 z - \dots - \theta_p z^p = 0$$

Условие стационарности: все корни характеристического уравнения по модулю больше единицы ($|z_j| > 1$)

1.2.3. Процесс скользящего среднего (МА-процесс)

Процесс, в котором оценка прогнозируемых членов ряда линейно зависит от текущего и прошлых значений, а также некоторого стохастического члена, который отражает вероятностный характер модели.

- MA(1) – процесс авторегрессии 1-го порядка

$$y_t = \delta + \varepsilon_t + \alpha\varepsilon_{t-1}, \varepsilon_t \text{ – белый шум}$$

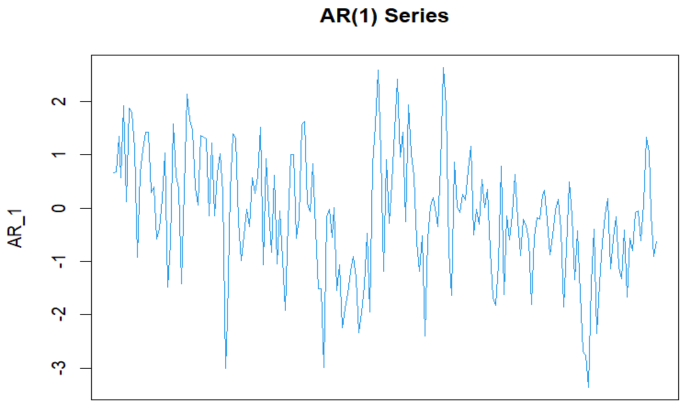

Рисунок 3. AR(1) процесс с $\theta = 0.5$

- MA(q) – процесс авторегрессии q-го порядка

$$y_t = \delta + \varepsilon_t + \alpha_1\varepsilon_{t-1} + \dots + \alpha_q\varepsilon_{t-q}, \ \varepsilon_t \text{ – белый шум}$$

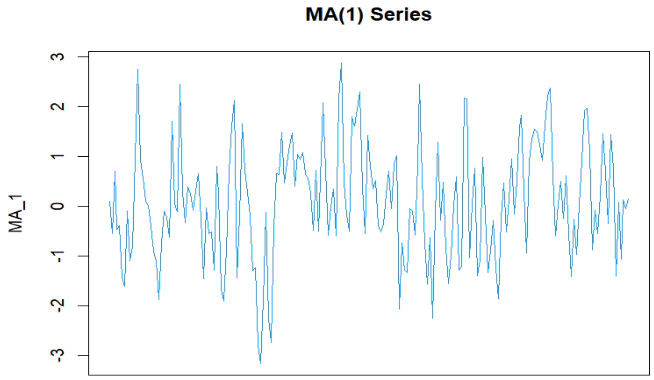

Рисунок 4. MA(1) процесс с $\alpha = 0.5$

Процесс стационарен при любых значениях $\alpha_1, \dots, \alpha_q$

1.2.4. Процесс ARMA(p, q)

Модель ARMA обобщает две более простые модели временных рядов — модель авторегрессии (AR) и модель скользящего среднего (MA).

$$y_t = \delta + \theta_1 y_{t-1} + \dots + \theta_p y_{t-p} + \varepsilon_t + \alpha_1 \varepsilon_{t-1} + \dots + \alpha_q \varepsilon_{t-q}$$

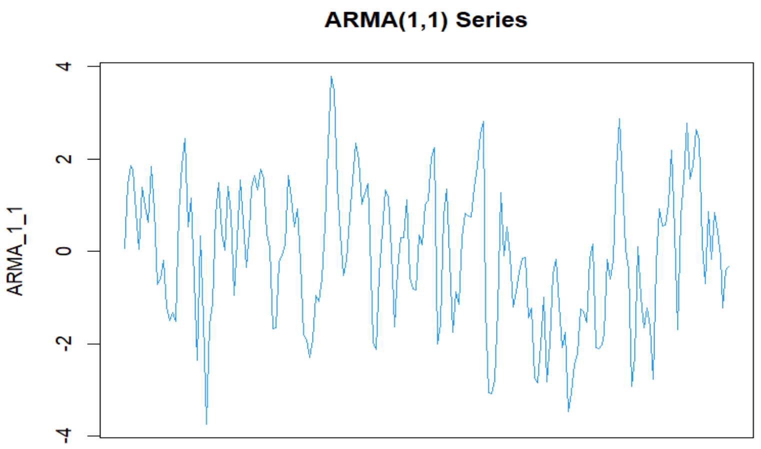

Рисунок 5. ARMA(1,1) процесс с $\theta = \alpha = 0.5$

Разложение Вольда. Любой стационарный процесс может быть представлен в виде бесконечного ряда членов белого шума, с коэффициентами, образующими абсолютно сходящийся числовой ряд.

Таким образом, нет фундаментальной разницы между AR, MA и ARMA представлением временного ряда. Выбор между этими представлениями – вопрос удобства и лаконичности.

1.2.5. Процесс случайного блуждания (Random Walk)

$$y_t = \delta + y_{t-1} + \varepsilon_t$$

Первая разность для этого процесса:

$$\Delta y_t = \delta + \varepsilon_t$$

Процесс случайного блуждания не стационарен, однако его первая разность является стационарным процессом. На практике многие финансовые или макроэкономические переменные также не стационарны сами по себе, но стационарны в разностях.

1.2.6. Процесс $ARIMA(p, d, q)$

Порядок интегрированности временного ряда. Если процесс y_t не стационарен, его первые, вторые, …, $(d-1)$-ые разности не стационарны, а d-ая разность $\Delta^d y_t$ – стационарна, то процесс называется интегрированным d-го порядка.

Если процесс является интегрированным d-го порядка, и его разность d-го порядка описывается процессом $ARMA(p, d, q)$, то исходный процесс называется интегрированным процессом авторегрессии со скользящим средним в остатках $ARIMA(p, d, q)$.

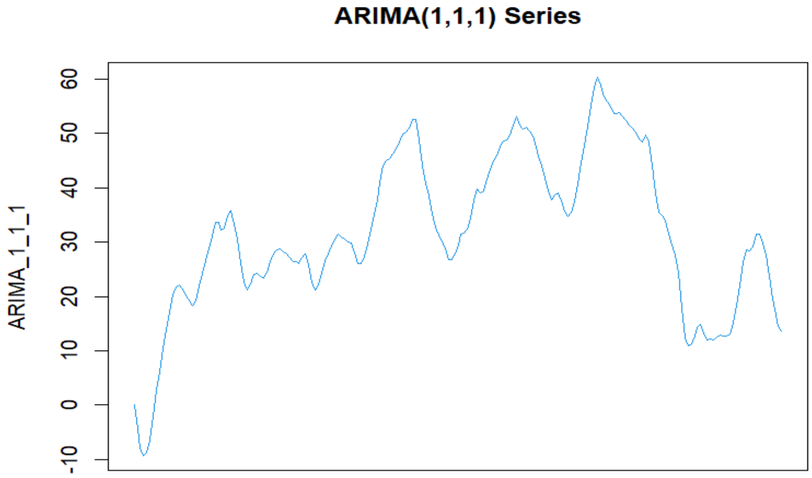

Рисунок 6. $ARIMA(1,1,1)$ процесс с $\theta = 0.6$, $\alpha = 0.7$

1.3. Прогнозирование

Главная цель построения ARIMA моделей – прогнозирование будущих значений экономических переменных.

Предположим, что сейчас момент времени T. Нам доступна информация о y_T, y_{T-1}, y_{T-2}… Нас интересует предсказание y_{T+h}, то есть предсказание на h шагов вперед.

Для того чтобы сделать предсказание, необходимо убедиться в том, что ряд является стационарным. Для этого полезно: 1) смотреть на график временного ряда, 2) использовать формальные статистические тесты.

1.3.1. Расширенный тест Дики-Фуллера (Augmented DF-test, ADF-test)

$$y_t = \theta_1 y_{t-1} + \dots + \theta_p y_{t-p} + \varepsilon_t$$

H_0: ряд является нестационарным, содержит единичный корень

H_1: ряд является стационарным процессом AR(p)

Оценим уравнение:

$$\Delta y_t = b y_{t-1} + c_1 \Delta y_{t-1} + \dots + c_{p-1} \Delta y_{t-p+1} + \varepsilon_t$$

через метод наименьших квадратов и проверим значимость b при помощи t-статистики. Расчетное значение статистики: $\hat{\tau} = \frac{\hat{b}}{se(\hat{b})}$

Сравниваем расчетное значение с критическим значением из специальных таблиц Дики и Фуллера. Если расчетное значение по модулю больше критического, то гипотеза H_0 отвергается $\Rightarrow$ делаем вывод о том, что ряд стационарен.

Также можно осуществлять ADF-тест с добавлением константы и тренда.

1.3.2. Критерий KPSS (Kwiatkowski–Phillips–Schmidt–Shin)

Альтернативным тестом для проверки стационарности является KPSS-тест.

H_0: ряд является тренд-стационарным

H_1: ряд является нестационарным

Оцениваем регрессию:

$$y_t = \delta + \varphi t + \varepsilon_t$$

Вычисляем остатки $e_1, e_2, \dots, e_T$

Вычисляем вспомогательные суммы (T штук):

$$S_t = \sum_{m=1}^{T} e_m$$

Вычисляем расчетное значение статистики:

$$KPSS = \sum_{t=1}^{T} \frac{S_t^2}{\widehat{\sigma^2}}$$

где $\widehat{\sigma^2}$ – оценка дисперсии случайной ошибки

Сравниваем расчетное значение с критическим. Если расчетное значение по модулю меньше критического, то гипотеза H_0 принимается ⇒ делаем вывод о том, что ряд тренд-стационарен

1.4. Анализ ACF и PACF для стационарного ряда

Эмпирическая автокорреляционная функция временного ряда (ACF) – выборочный аналог теоретической автокорреляционной функции, рассчитывается на основе выборочных коэффициентов корреляции:

$$ACF(k) = \widehat{\rho_k} = \widehat{Corr}(y_t, y_{t-k})$$

Эмпирическая частная автокорреляционная функция временного ряда (PACF) рассчитывается на основе выборочных частных коэффициентов корреляции.

Определим выборочный частный коэффициент корреляции k-го порядка как МНК-оценку для θ_k в модели $AR(k)$:

$$PACF(k) = \widehat{\theta_k}$$

1.4.1. Процесс AR(p)

1. ACF будет бесконечна по протяженности, и только в пределе при $k \to \infty$ сходится к нулю.

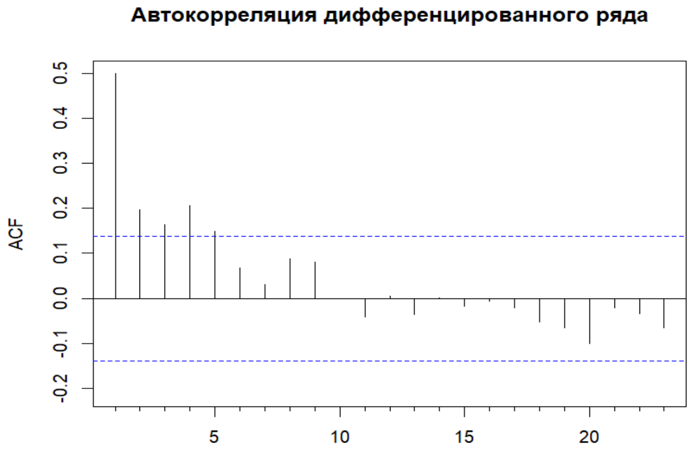

Рисунок 7. ACF процесса $AR(1)$ с $\theta = 0.5$

2. PACF равна (или близка) к нулю для лагов, больших чем p.

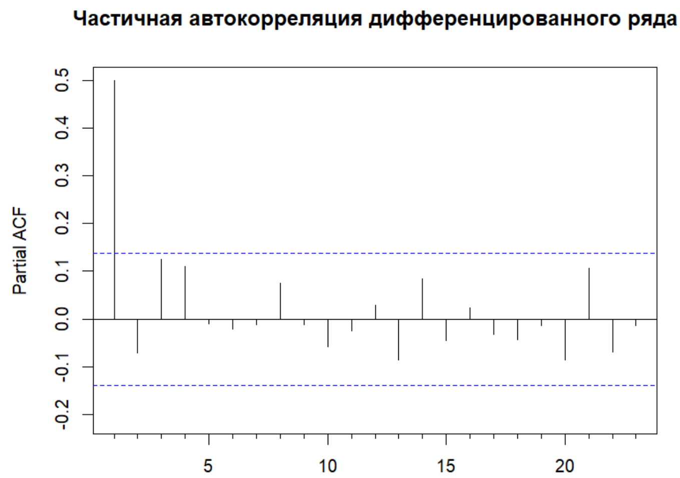

Рисунок 8. $PACF$ процесса $AR(1)$ с $\alpha = 0.5$

1.4.2. Процесс MA(q)

1. ACF равна (или близка) к нулю для лагов, больших чем q.

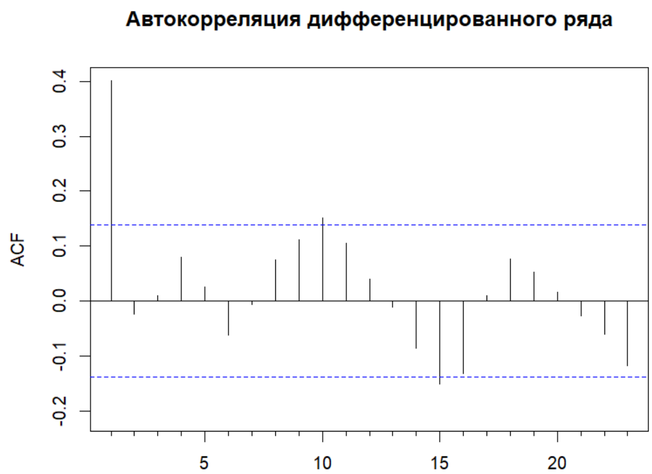

Рисунок 9. ACF процесса $MA(1)$ с $\theta = 0.5$

2. PACF бесконечна по протяженности, и только в пределе при $k \to \infty$ сходится к нулю.

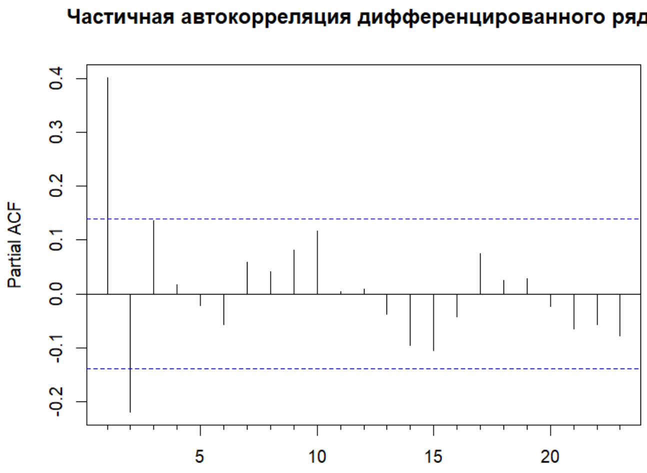

Рисунок 10. $PACF$ процесса $MA(1)$ с $\alpha = 0.5$

1.4.3. Процесс ARMA(p, q)

Если ни один из процессов AR(p) и MA(q) не подходит, судя по ACF и PACF, то это процесс ARMA(p, q).

1.5. Методология Бокса – Дженкинса

1.5.1. Определение порядка интегрированности ряда и переход к стационарным разностям

1. Тестируем ряд на стационарность, используя тесты, которые обсуждались ранее

2. Если ряд оказался стационарным, то переходим к шагу 2. Если нет, то переходим к разностям ряда и снова тестируем стационарность

3. Повторяем шаги 1-2 до тех пор, пока не получим стационарный ряд

4. Таким образом, на этом шаге определяется параметр d модели ARIMA(p,d,q), то есть порядок интегрированности ряда

5. Далее в рамках шагов 2 и 3 следует работать со стационарными разностями ряда

Помимо взятия разностей (дифференцирования) ряд также можно логарифмировать, чтобы стабилизировать дисперсию. Дифференцирование в свою очередь может помочь устранить или снизить определенные типы трендов и сезонности.

1.5.2. Анализ ACF и PACF

- На шаге 2 следует построить и проанализировать графики ACF и PACF для рассматриваемого временного ряда

- Анализ коррелограмм позволяет сделать предварительные предположения о возможных порядках авторегрессии p и скользящего среднего q

- Эмпирические ACF и PACF не обязаны в точности совпадать с теоретическими, но должны быть похожи на них

- По возможности рекомендуется использовать экономичные модели: $p + q \leq 3$ (если нет сезонной компоненты)

1.5.3. Оценивание и проверка адекватности модели

- Для каждой из выбранных на втором шаге моделей оцениваются их параметры
- Обычно оценивание производится при помощи ММП (метода максимального правдоподобия) (для AR моделей состоятельные оценки также дает обычный метод наименьших квадратов)
- Каждая из моделей проверяется на адекватность на основе критериев, представленных далее
- Наилучшая из моделей выбирается в качестве итоговой для использования на четвертом шаге

<u>Критерии адекватности ARMA модели</u>

1. Значимость коэффициентов модели
2. Анализ остатков модели: остатки должны быть белым шумом ⇒ должны иметь нулевую автокорреляцию ⇒ все элементы ACF для ряда остатков должны незначимо отличатся от нуля
3. Информационные критерии:
    - тестирование гипотезы о равенстве нулю отдельного коэффициента автокорреляции
        - $H_0: \rho_k = 0$
        - тестовая статистика: $\widehat{\rho_k} \sim N\left(0, \frac{1}{T}\right)$
        - если $|\widehat{\rho_k}| < \frac{1{,}96}{\sqrt{T}}$, то при уровне значимости 5% гипотеза H_0 принимается
    - тест Льюинга – Бокса
        - $H_0: \rho_1 = \rho_2 = \cdots = \rho_K = 0$
        - $\tilde{Q} = T(T+2)\sum_{i=1}^{K} \frac{\widehat{\rho_i}^2}{T-i} \sim \chi^2(K-p-q)$, p и q – параметры ARIMA модели
    - информационный критерий Шварца (Байесовский информационный критерий)
        - $SIC = lnT\frac{p+q}{T} + \ln\left(\frac{\sum e_t^2}{T}\right)$
        - p и q — параметры ARIMA модели, если в модель включена константа, то вместо $p + q$ следует использовать $p + q + 1$
        - можно использовать для сравнения разных моделей с одинаковой зависимой переменной

- следует выбирать модель с наименьшим значением критерия
- можно использовать не только для ARIMA, но и для любых других моделей временных рядов

- информационный критерий Акаике
  - $AIC = 2\frac{p+q}{T} + \ln\left(\frac{\sum e_t^2}{T}\right)$
  - работает аналогично критерию Шварца, однако используется реже, так как асимптотически приводит к выбору перепараметризированных моделей

1.5.4. Прогнозирование

Последний шаг методологии. Осуществляется после выбора наилучшей модели в соответствии с описанными выше, в пункте 1.3.

Также полезно проверить ошибки: MAE (средняя квадратичная ошибка), RMSE (корень из среднеквадратичной ошибки) и MAPE (средняя абсолютная ошибка в процентах). Под ошибкой понимается $e(t) = Z(t) - \hat{Z}(t)$, где $Z(t)$ – фактическое значение временного ряда, а $\hat{Z}(t)$ – спрогнозированное. Формулы перечисленных оценок можно записать в следующем виде:

1. $MSE = \frac{1}{N}\sum_{t=1}^{N}\left(Z(t) - \hat{Z}(t)\right)^2$
2. $RMSE = \sqrt{MSE}$
3. $MAPE = \frac{1}{N}\sum_{t=1}^{N}\frac{|Z(t)-\hat{Z}(t)|}{Z(t)} \cdot 100\%$ - применяется для временных рядов, фактические значения которых значительно больше 1.

Точность прогнозирования – понятие обратно пропорциональное *ошибке прогнозирования*. Если ошибка прогнозирования велика, то точность мала и наоборот, если ошибка прогнозирования мала, то точность велика.

1.6. Bagging

Бэггинг (bootstrap aggregating) представляет собой метод улучшения качества прогнозирования модели и снижения ее дисперсии. Это достигается путем создания нескольких независимых моделей на основе выборок данных с повторением, что помогает снизить вероятность переобучения.

Исходная выборка разделяется на подмножества для обучения базовых алгоритмов. Для агрегирования результатов базовых моделей в прогнозных задачах используется усреднение с использованием среднего арифметического или средневзвешенного значения.

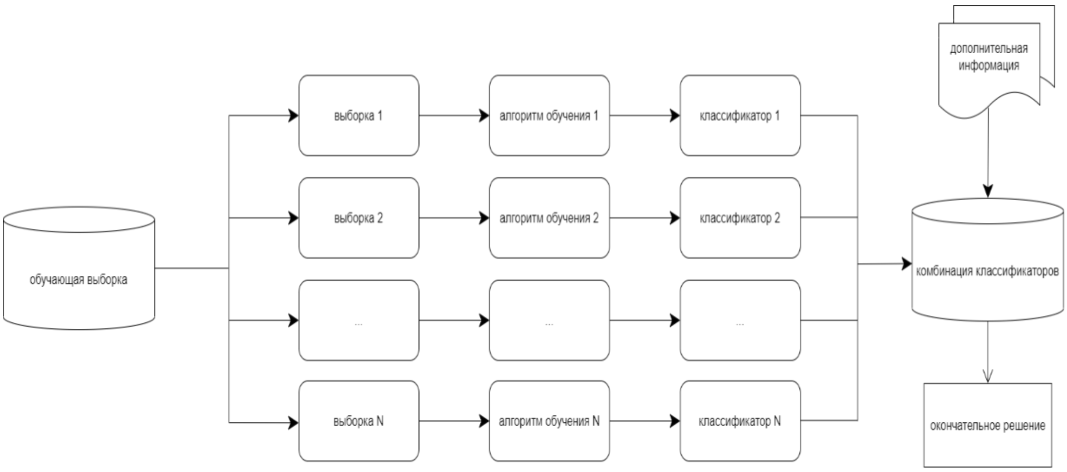

Рисунок 11. Bagging

1.7. Boosting

Бустинг – это метод, используемый для уменьшения количества ошибок при построении прогнозов. Он улучшает точность прогнозирования временных рядов и производительность моделей путем преобразования слабых классификаторов, таких как деревья решений, для создания единой сильной модели.

Деревья решений – это структуры данных, разделяющие исходный набор данных на меньшие подмножества в зависимости от их характеристик. Идея заключается в том, что деревья решений многократно разделяют данные, пока не останется только один класс. Например, дерево может задать ряд вопросов с ответами «да» или «нет» и разделить данные на категории при каждом шаге.

Отличие бустинга от бэггинга

Бустинг и бэггинг — два распространенных ансамблевых метода, повышающих точность прогнозирования. Основное различие между ними — метод обучения. В случае с бэггингом повышают точность слабых моделей, параллельно обучая некоторые из них на различных наборах данных. Бустинг же обучает слабые модели последовательно.

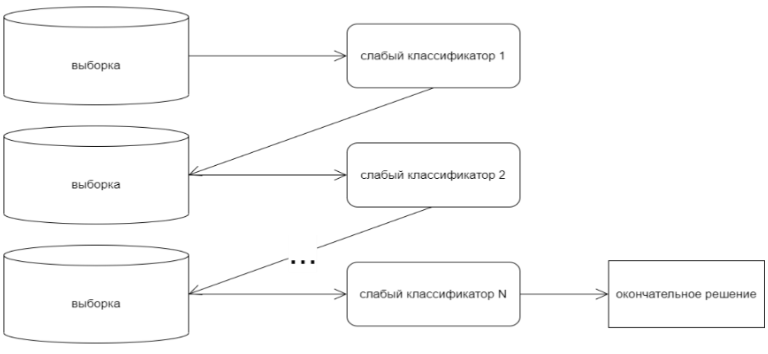

Рисунок 12. Boosting

Адаптивный бустинг (AdaBoost) – одна из самых ранних моделей бустинга. Он адаптируется и самостоятельно корректирует классификаторы в каждой итерации бустинга.

AdaBoost изначально присваивает одинаковый вес каждому набору данных. Затем он автоматически корректирует веса точек выборки после каждого шага на дереве решений. Элементы, которые были классифицированы неверно, приобретают больший вес в следующей итерации. Процесс повторяется до тех пор, пока остаточная ошибка или разница между фактическими и прогнозируемыми значениями не опустится ниже допустимого уровня.

AdaBoost менее чувствителен, чем другие алгоритмы бустинга, но не так эффективен при корреляции между признаками или использовании данных большой размерности.

Градиентный бустинг (GB) – похож на AdaBoost: он также представляет собой метод последовательного обучения. Разница между AdaBoost и GB в том, что GB не присваивает неправильно классифицированным элементам больший вес. Вместо этого он оптимизирует функцию потерь через последовательное генерирование базовых моделей, в результате чего текущая базовая модель всегда становится эффективнее предыдущей. В отличие от AdaBoost, метод GB пытается сразу генерировать точные результаты, а не исправлять ошибки. По этой причине метод GB дает более точные результаты. Градиентный бустинг подходит и для задач классификации, и для регрессии.

1.8. Bootstrap методы

Bootstrap – метод, имитирующий поведение оценки $\hat{\theta}$ с помощью случайного независимого ресемплинга последовательных наблюдений. Группировка в блоки используется для сохранения исходной структуры временного ряда внутри блока. Предназначен для работы с общими стационарными процессами.

1.8.1. NBB

Non-overlapping Block Bootstrap (NBB) – bootstrap метод, разделяющий выборку $\boldsymbol{X_t} = (X_1, X_2, \dots, X_n)$ размера n на непересекающиеся между собой блоки $(X_1, \dots, X_l), (X_{l+1}, \dots, X_{2l}), \dots, (X_{n-l+1}, \dots, X_n)$ длиной $l \in \mathbb{N}$. Далее метод осуществляет независимые "вытягивания" k $(n = kl)$ блоков с возвращением, причем точки начала блоков независимо равномерно распределены на множестве $\{1, l+1, \dots, n-l+1\}$ всех возможных начальных точек.

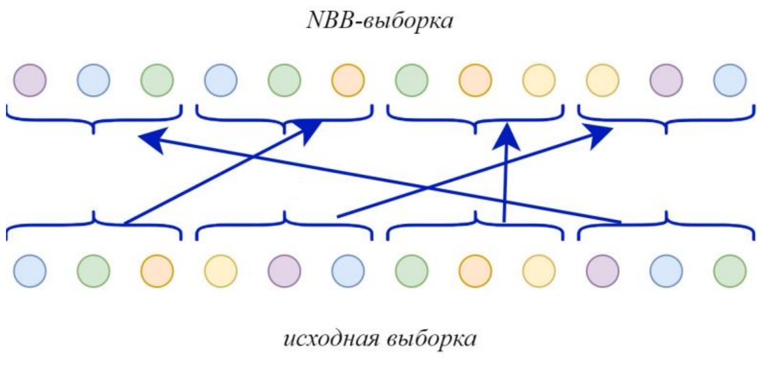

Рисунок 13. Non-overlapping Block Bootstrap

1.8.2. MBB

Moving Block Bootstrap – bootstrap метод, разделяющий выборку $\boldsymbol{X_t} = (X_1, X_2, \dots, X_n)$ размера n на пересекающиеся блоки $(X_1, \dots, X_l), (X_2, \dots, X_{l+1}), \dots, (X_{n-l+1}, \dots, X_n)$ размера l $(n = kl)$. Далее метод осуществляет действия, аналогичные методу NBB, точки начала блоков независимо равномерно распределены на множестве $\{1, 2, \dots, n-l+1\}$ всех возможных начальных точек.

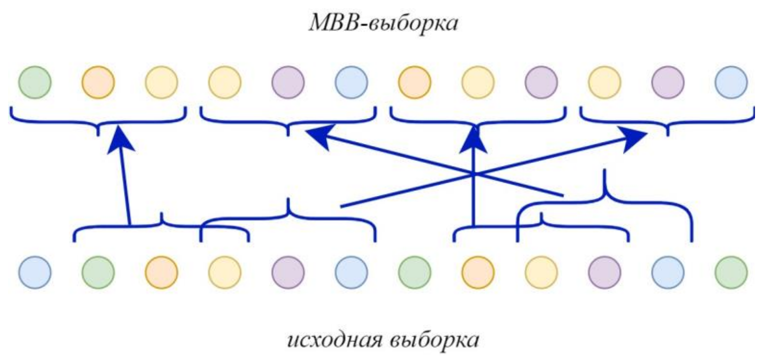

Рисунок 14. Moving Block Bootstrap

1.8.3. Выбор длины блока

Оптимальная длина блока – параметр настройки блочного бутстрапа – зависит от: процесса, порождающего данные, бутстрапируемой статистики и цели использования бутстрапа (например, оценивание смещения, дисперсии или распределения).

Метод HHJ (Hall, Horowitz & Jing) применим для выбора оптимальной длины блока при оценивании распределения. В методе рассматривается поведение блочного бутстрапа при разных значениях длины блоков для подвыборок размера $m \ll n$ и получают оптимальную длину блока для размера подвыборки m. Затем оцененная оптимальная длина блока выводится путем экстраполяции Ричардсона до размера исходной выборки n. Этот метод требует спецификации размера подвыборки m, что менее критично, чем выбор длины блока. Подобная техника использования подвыборок является очень общей, но может быть не очень эффективной. В частности, если оценка $\hat{\theta}$ сильно нелинейная, свойства метода в подвыборках могут быть очень плохими.

Длины блоков для bootstrap методов стоит выбирать по минимальному значению $RMSE$.

Глава 2. Практическая реализация

2.1. Индекс реальной месячной зарплаты

2.1.1. Анализ исходных данных

Из базы ВШЭ получаем временной ряд WAG_M – месячные показатели индекса реальной зарплаты с января 1993 года по январь 2023. Для того чтобы далее сравнить прогноз с реальными значениями, разбиваем данные на две выборки: 29 лет и 1 год. Первая необходима построения моделей – обучающая выборка, а вторая – тестовая, для сравнения.

Рисунок 15. Данные исходной выборки

Из графика можно сделать очевидный вывод о том, что процесс нестационарный, присутствуют тренд и сезонность. Разложим ряд на составные части для большей наглядности:

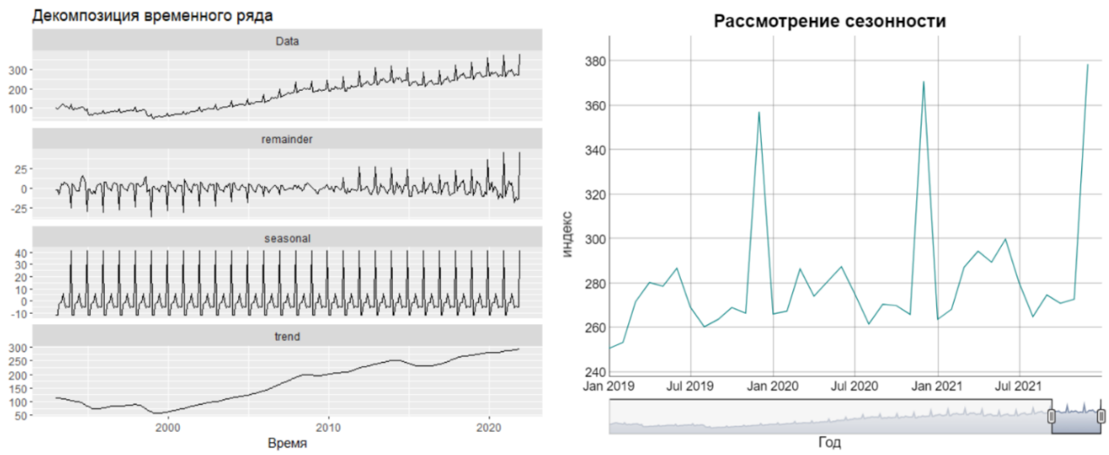

Рисунок 16. Декомпозиция исходного ряда. График сезонной компоненты

Сезонность хорошо выражена, значит, для прогнозирования нужно будет строить модели, учитывающие сезонные параметры.

Стационарность ряда проверяем по тесту Дики-Фуллера. Переходим к временной разности следующего порядка, пока он не даст стационарный ряд.

Тест Дики-Фуллера выполняется с помощью функции adf.test(), которая на вход принимает одномерный временной ряд, а на выходе возвращает логическое значение, равное 1, если нулевая гипотеза отвергается в пользу альтернативной, и 0 – иначе. Выполним его при 5%-ном уровне значимости. Если $p - value\ <\ 0.05$, ряд стационарен.

```
        Augmented Dickey-Fuller Test

data:  learn
Dickey-Fuller = -3.3939, Lag order = 12, p-value = 0.05544
alternative hypothesis: stationary
```

Рисунок 17. Результат работы функции $adf.test()$

Из значения p-value можем сделать вывод, что ряд нестационарен. Продифференцируем его.

```
        Augmented Dickey-Fuller Test

data:  diff(learn)
Dickey-Fuller = -8.6863, Lag order = 7, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(diff(learn), alternative = "stationary") :
  p-value smaller than printed p-value
> kpss.test(diff(learn))

        KPSS Test for Level Stationarity

data:  diff(learn)
KPSS Level = 0.18267, Truncation lag parameter = 5, p-value = 0.1

Warning message:
In kpss.test(diff(learn)) : p-value greater than printed p-value
```

Рисунок 18. Результат работы функции $adf.test()$ и функции $kpss.test()$ для продифференцированного ряда

Видим предупреждение о том, что $p - value\ <\ 0.01$ в ADF-тесте. Значит достаточно временной разности первого порядка. Тест $KPSS$ также принимает гипотезу о стационарности.

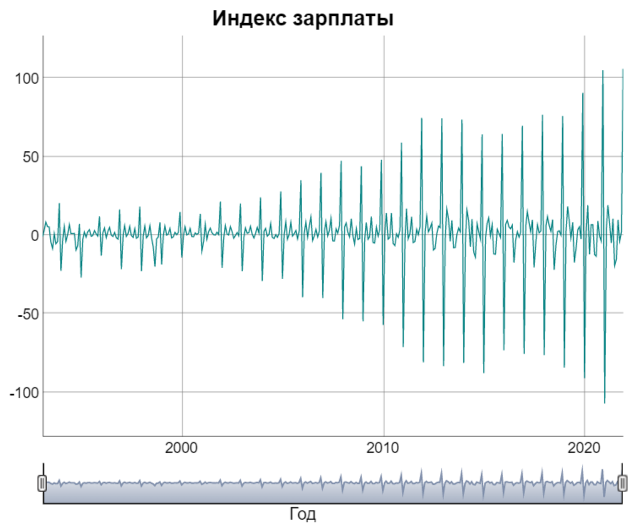

Рисунок 19. Ряд, приведенный к стационарному виду

График полученного в результате дифференцирования ряда вполне похож на стационарный процесс.

2.1.2. Построение автоматической ARIMA модели

Пользуемся командой $auto.arima()$ для автоматического подбора модели $automodel$. Затем выводим информацию об этой модели.

```
Series: learn
ARIMA(1,1,2)(2,1,0)[12]

Coefficients:
         ar1      ma1     ma2     sar1    sar2
      0.6841  -0.8366  0.1709  -0.2119  0.0832
s.e.  0.2521   0.2568  0.0587   0.0604  0.0627

sigma^2 = 23.27:  log likelihood = -1000.41
AIC=2012.81   AICc=2013.07   BIC=2035.7
```

Рисунок 20. Результат вывода $automodel$

Процесс распознался, как нестационарный с порядком дифференцирования, равным 1, что соответствует результату теста Дики-Фуллера. При этом это процесс с авторегрессионным коэффициентом, двумя коэффициентами скользящего среднего и двумя сезонными авторегрессионными коэффициентами.

Создаем прогноз $auto\_prediction$ по модели на 12 месяцев вперед с помощью команды $forecast()$. И строим график прогноза с помощью $plot()$.

```
> auto_prediction
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Jan 2022       273.8579 267.6759 280.0400 264.4033 283.3126
Feb 2022       277.8766 269.7730 285.9802 265.4832 290.2700
Mar 2022       297.1491 287.2697 307.0285 282.0398 312.2584
Apr 2022       298.6162 287.0923 310.1401 280.9919 316.2405
May 2022       296.8652 283.8145 309.9159 276.9058 316.8245
Jun 2022       306.4332 291.9600 320.9065 284.2984 328.5681
Jul 2022       288.9386 273.1347 304.7424 264.7686 313.1085
Aug 2022       273.5905 256.5364 290.6445 247.5086 299.6723
Sep 2022       283.6794 265.4460 301.9129 255.7938 311.5651
Oct 2022       280.0815 260.7308 299.4323 250.4871 309.6759
Nov 2022       280.6168 260.2036 301.0300 249.3974 311.8361
Dec 2022       387.4443 366.0171 408.8715 354.6742 420.2144
```

Рисунок 21. Результат работы функции $auto\_prediction$: значения прогноза

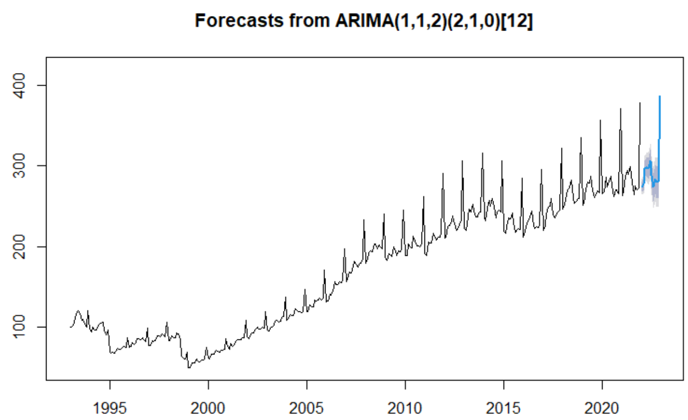

Рисунок 22. Результат работы функции $plot()$ - график прогноза

2.1.3. Подбор коэффициентов модели с помощью коррелограмм

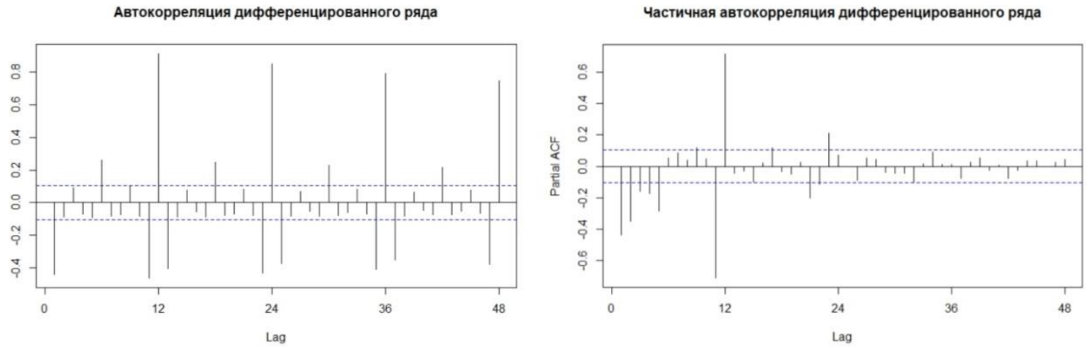

Рисунок 23. Коррелограммы модели

Из-за наличия сезонности и тренда на коррелограммах наблюдаются выбросы. Оценим сверху параметры p и q $ARIMA$-модели, как $p = 6$ и $q = 5$ – последние по счету значимые лаги. Значимые лаги на коррелограммах дальше $p = 6$ и $q = 5$ связаны с сезонностью ряда. Нужно также учитывать, что большие значения параметров будут сильно усложнять модель.

Подберем параметры вручную, выберем лучшую модель, опираясь на значения ошибок MAE и $RMSE$, а также критериев AIC, BIC и $AICc$.

Таблица 1. Различные $ARIMA$-модели и значения их ошибок и критериев

p	q	AIC	BIC	AICc	RMSE	MAE
0	0	2016.718	2028.161	2016.791	17.80576	15.8672
0	1	2012.136	2027.392	2012.257	17.62427	15.74751
0	2	2012.169	2031.239	2012.351	18.0018	16.16072
0	3	2012.925	2035.809	2013.181	18.16007	16.32905
0	4	2014.855	2041.554	2015.197	18.08488	16.25586
0	5	2016.358	2046.871	2016.799	18.06804	16.22536
1	0	2011.372	2026.628	2011.493	17.70346	15.84024
1	1	2013.194	2032.264	2013.376	17.77878	15.91871
1	2	2012.811	2035.696	2013.068	18.09905	16.26244
1	3	2014.811	2041.51	2015.154	18.09869	16.26221
1	4	2016.707	2047.22	2017.149	18.13905	16.30896
1	5	2017.292	2051.619	2017.846	17.53828	15.69654
2	1	2013.304	2036.189	2013.56	18.08124	16.24931
2	2	2014.956	2041.655	2015.298	18.02305	16.19112
2	3	2016.037	2046.55	2016.479	18.15203	16.32088
2	4	2017.634	2051.961	2018.188	18.15372	16.34129
2	5	2019.409	2057.551	2020.088	18.0432	16.22459
3	0	2012.666	2035.551	2012.922	18.08789	16.25671
3	1	2014.593	2041.292	2014.935	18.14136	16.3088
3	2	2008.036	2038.549	2008.477	19.1182	17.06843
3	4	2014.314	2052.455	2014.993	18.28563	16.48816
3	5	2011.186	2053.141	2012.003	17.99667	16.0603
4	0	2014.645	2041.344	2014.987	18.11772	16.28766
4	1	2016.385	2046.898	2016.827	18.08143	16.25578
4	3	2011.381	2049.523	2012.06	19.35429	17.31238

4	5	2023.372	2069.142	2024.341	18.0614	16.24818
5	0	2016.527	2047.04	2016.968	18.13023	16.2927
5	1	2018.177	2052.504	2018.731	17.94154	16.10932
5	2	2019.457	2057.599	2020.136	18.17476	16.36121
5	3	2011.419	2053.374	2012.236	18.26814	16.31479
5	4	2023.48	2069.249	2024.448	18.11356	16.30404
5	5	2013.104	2062.688	2014.238	18.5983	16.61782
6	0	2018.427	2052.754	2018.98	18.04904	16.21001
6	1	2019.974	2058.116	2020.653	17.77806	15.95251
6	4	2003.167	2052.751	2004.301	24.86748	23.20084
6	5	2015.383	2068.781	2016.696	19.66764	17.67358

По значениям критериев Акаике и Шварца и $RMSE$ и MAE, лучшими моделями оказались:

Таблица 2. Лучшие, из подобранных $ARIMA$-моделей

p	q	AIC	BIC	AICc	RMSE	MAE	LogLik
1	5	2017.292	2051.619	2017.846	**17.53828**	**15.69654**	-999.646
6	4	**2003.167**	2052.751	**2004.301**	24.86748	23.20084	**-988.5834**
1	0	2011.372	**2026.628**	2011.493	17.70346	15.84024	-1001.686

Среди лучших, из рассмотренных, моделей выберем модели $ARIMA(1,1,5)(2,1,0)[12]$ и $ARIMA(6,1,4)(2,1,0)[12]$, поскольку значения их информационных критериев и ошибок лучше, чем у третьей модели. Также у этих моделей больше значения функции правдоподобия.

2.1.4. Прогноз и сравнение с реальными данными

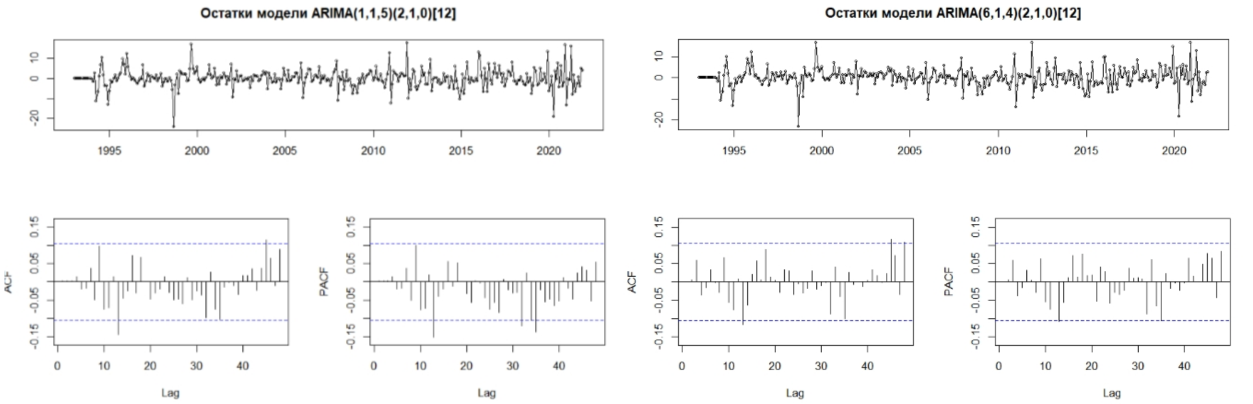

Рисунок 24. Остатки выбранных моделей

Теперь создадим прогноз по подобранным моделям, построим их график и наложим эти графики на график реальных значений, из тестовой выборки.

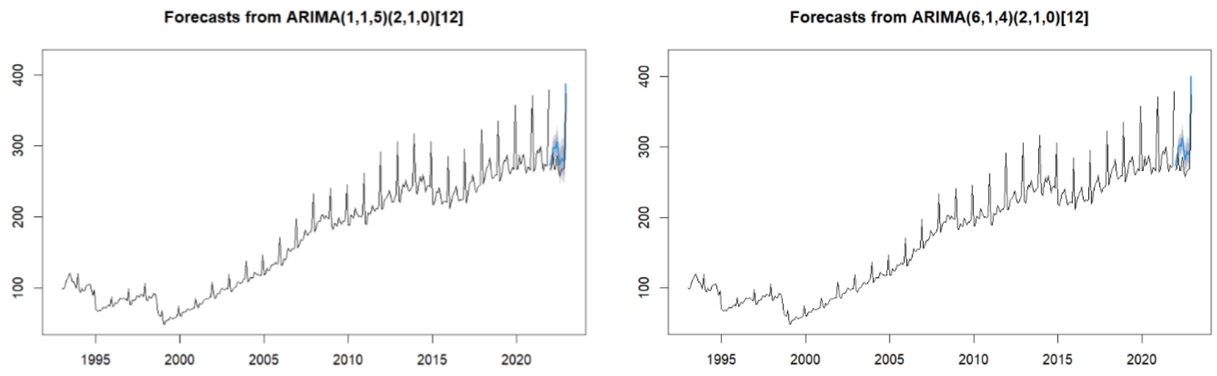

Рисунок 25. Результат работы функции $plot()$ - графики прогнозов, совмещенные с графиком реальных значений

Прогнозы более-менее повторяют поведение тестовой выборки, но есть некоторые неточности. Скорее всего это связано с тем, что функция правдоподобия имеет не очень большие значения, либо в выборке имеются выбросы.

2.1.5. MBB

Применим алгоритм Moving Block Bootstrap к стационарному временному ряду d_learn (первой разности изначального ряда). Оптимальная длина блока, была найдена в результате работы алгоритма HHJ и равна 2.

График полученной, бутстрапированной, выборки $series$, наложенный на $dlearn$:

Рисунок 26. Результат работы алгоритма MBB

```
> summary(series)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-107.200   -2.585    0.650    1.301    6.500  105.000
> summary(d_learn)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-107.2000   -2.4600    1.0300    0.8032    5.7900  105.8000
```

Рисунок 27. Основные характеристики рядов $series$ и d_learn

Характеристики этих рядов не сильно отличаются, но все же значения медианы и среднего значения разные. Это может быть связано с тем, что выборка не очень большая.

2.1.6. NBB

Применим алгоритм Non-overlapping Block Bootstrap к стационарному временному ряду d_learn (первой разности изначального ряда). Длина блока, как и в методе MBB, равна 2.

График полученной, бутстрапированной, выборки $series$, наложенный на $dlearn$:

Рисунок 28. Результат работы алгоритма MBB

```
> summary(series)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-107.2000   -2.4650    1.0300    0.4797    5.7625  105.0000
> summary(d_learn)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-107.2000   -2.4600    1.0300    0.8032    5.7900  105.8000
```

Рисунок 29. Основные характеристики рядов $series$ и d_learn

Характеристики этих рядов ближе, чем в результате работы алгоритма MBB, поэтому можно сделать вывод, что для данного ряда NBB лучше сохраняет характеристики.

2.1.7. Заключение по эксперименту

Ряд был проанализирован. По нему были построены ARIMA модели, которые сравнили по RMSE, MAE, AIC, AICc, BIC, а также функции правдоподобия. Были выбраны лучшие модели, для которых были построены прогнозы на 1 год.

Сравнив прогнозы для выбранных по перечисленному списку параметров модели с прогнозом для модели, построенной функцией $auto.arima()$, можно сделать вывод, что они практически совпадают.

Выполнено бутстрапирование рядов алгоритмами MBB и NBB, в результате работы которых были получены ряды с практически совпадающими характеристиками.

2.2. Годовые показатели коэффициента рождаемости

2.2.1. Анализ исходных данных

Из базы ВШЭ получаем временной ряд POPFER_Y – годовые показатели коэффициента рождаемости с 1991 по 2020 год. Для того чтобы далее сравнить прогноз с реальными значениями, аналогично предыдущему ряду, разбиваем данные на две выборки: 26 лет и 3 года.

Рисунок 30. Данные исходной выборки

Из графика сложно сделать очевидный вывод о стационарности процесса. Проверим ее по тесту Дики-Фуллера. Будем переходить к временной разности следующего порядка, до тех пор, пока не получится стационарный ряд.

```
	Augmented Dickey-Fuller Test

data:  learn
Dickey-Fuller = -0.65677, Lag order = 2, p-value = 0.9622
alternative hypothesis: stationary
```

Рисунок 31. Результат работы функции $adf.test()$

Из значения p-value можем сделать вывод, что ряд нестационарен. Продифференцируем его.

```
	Augmented Dickey-Fuller Test

data:  diff(learn)
Dickey-Fuller = -0.31895, Lag order = 2, p-value = 0.9828
alternative hypothesis: stationary

> kpss.test(diff(learn))

	KPSS Test for Level Stationarity

data:  diff(learn)
KPSS Level = 0.28161, Truncation lag parameter = 2, p-value = 0.1

Warning message:
In kpss.test(diff(learn)) : p-value greater than printed p-value
```

Рисунок 32. Результат работы функций $adf.test(diff())$ и $kpss.test(diff())$

Судя по тесту $KPSS$ ряд стационарен, но ADF отвергает гипотезу о стационарности. Продолжаем дифференцирование.

```
	Augmented Dickey-Fuller Test

data:  diff(diff(learn))
Dickey-Fuller = -4.7788, Lag order = 2, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(diff(diff(learn)), alternative = "stationary") :
  p-value smaller than printed p-value
> kpss.test(diff(diff(learn)))

	KPSS Test for Level Stationarity

data:  diff(diff(learn))
KPSS Level = 0.4263, Truncation lag parameter = 2, p-value = 0.06582
```

Рисунок 33. Результат работы функции $adf.test(diff(diff()))$

После взятия 2-ой разности видим предупреждение о том, что $p - value < 0.01$ в ADF-тесте. Тест $KPSS$ также принимает гипотезу о стационарности. Значит достаточно временной разности второго порядка.

Рисунок 34. Ряд, приведенный к стационарному виду

График ряда, полученного в результате второго дифференцирования, похож на стационарный процесс.

2.1.2. Построение автоматической ARIMA модели

Пользуемся командой $auto.arima()$ для автоматического подбора модели $automodel$. Затем выводим информацию об этой модели.

```
Series: learn
ARIMA(1,1,0)

Coefficients:
         ar1
      0.6202
s.e.  0.2019

sigma^2 = 0.2835:  log likelihood = -20.24
AIC=44.47   AICc=44.99   BIC=46.99
```

Рисунок 35. Результат вывода $automodel$

Процесс распознался, как нестационарный с порядком дифференцирования, равным 1. При этом это процесс с одним авторегрессионным коэффициентом.

Создаем прогноз $auto\_prediction$ по модели с помощью команды $forecast()$. И строим график прогноза с помощью $plot()$.

```
> auto_prediction
     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
2018      10.631691 9.949370 11.31401 9.588171 11.67521
2019      10.093148 8.794026 11.39227 8.106313 12.07998
2020       9.759133 7.872577 11.64569 6.873895 12.64437
2021       9.551969 7.122493 11.98145 5.836406 13.26753
2022       9.423482 6.496893 12.35007 4.947651 13.89931
2023       9.343792 5.962110 12.72547 4.171956 14.51563
2024       9.294367 5.494400 13.09433 3.482819 15.10591
2025       9.263712 5.077090 13.45033 2.860826 15.66660
2026       9.244699 4.698427 13.79097 2.291776 16.19762
2027       9.232907 4.350049 14.11577 1.765220 16.70059
```

Рисунок 36. Результат работы функции $auto\_prediction$: значения прогноза

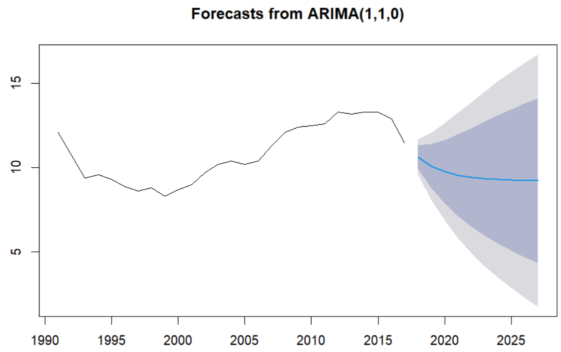

Рисунок 37. Результат работы функции $plot()$ - график прогноза

2.1.3. Подбор коэффициентов модели с помощью коррелограмм

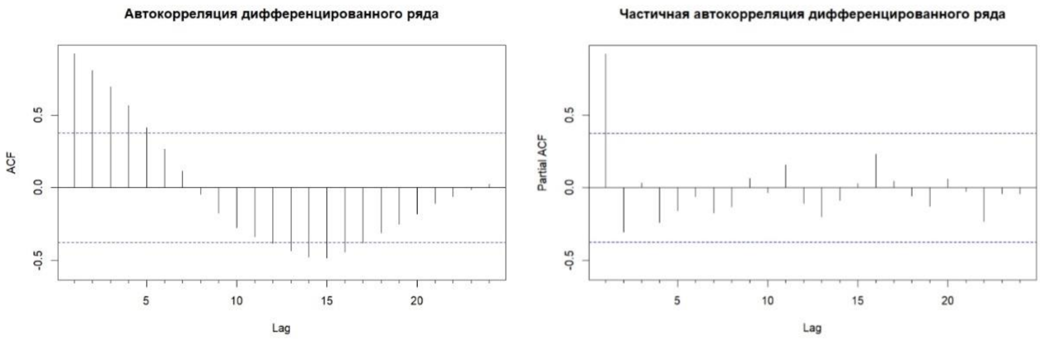

Рисунок 38. Коррелограммы модели

Оценим сверху параметры p и q $ARIMA$-модели, как $p = 5$ и $q = 1$. Параметр d будем рассматривать от 0 до 2.

Подберем параметры вручную, выберем лучшую модель, опираясь на значения ошибок MAE и $RMSE$, а также критериев AIC, BIC и $AICc$.

Таблица 3. Различные $ARIMA$-модели и значения их ошибок и критериев

p	d	q	AIC	BIC	AICc	RMSE	MAE
0	0	0	108.5454	111.1371	109.0454	0.73832	0.61358
0	0	1	82.95889	86.84641	84.00237	0.770678	0.718937
0	1	0	49.93424	51.19234	50.10091	1.317826	1.233333
0	1	1	46.41034	48.92654	46.93208	0.829816	0.687778
0	2	0	43.50554	44.72442	43.67945	1.717556	1.566667
0	2	1	43.9754	46.41315	44.52085	0.918275	0.822787
1	0	0	56.07288	59.96039	57.11636	1.281627	1.199379
1	0	1	52.19849	57.38184	54.01667	0.856169	0.707745
1	1	0	44.47317	46.98936	44.99491	0.156745	0.105343
1	1	1	46.47264	50.24693	47.56355	0.158058	0.112142
1	2	0	44.3792	46.81695	44.92466	1.299078	1.171453
1	2	1	45.75122	49.40785	46.89408	0.774498	0.704016
2	0	0	48.96416	54.14751	50.78234	0.243246	0.217253
2	0	1	44.9018	51.38098	47.75894	0.123558	0.112704
2	1	0	46.47273	50.24702	47.56364	0.157835	0.11099

2	1	1	46.00717	51.03956	47.91193	0.262865	0.259365
2	2	0	46.02951	49.68614	47.17237	0.957035	0.860171
2	2	1	47.53168	52.40718	49.53168	0.568804	0.516041
3	0	0	50.70951	57.1887	53.56666	0.338812	0.335246
3	0	1	50.11356	57.88858	54.31356	0.329679	0.329406
3	1	0	48.42222	53.4546	50.32698	0.144393	0.105615
3	1	1	47.31805	53.60854	50.31805	0.155952	0.139544
3	2	0	45.96032	50.83583	47.96032	0.281964	0.266968
3	2	1	44.96638	51.06076	48.12427	0.407545	0.364425
4	0	0	52.01219	59.78721	56.21219	0.279499	0.25762
4	0	1	51.98849	61.05935	57.88323	0.267274	0.261979
4	1	0	48.96232	55.2528	51.96232	0.290252	0.276748
4	1	1	47.65067	55.19925	52.07173	0.279188	0.252218
4	2	0	46.68911	52.78348	49.847	0.627795	0.575898
4	2	1	46.84379	54.15704	51.51045	0.272673	0.247353
5	0	0	49.25589	58.32674	55.15062	0.153358	0.136986
5	0	1	50.05322	60.41992	58.05322	0.12422	0.111565
5	1	0	49.24547	56.79405	53.66652	0.152802	0.136815
5	1	1	49.94956	58.75624	56.17179	0.256433	0.236518
5	2	0	47.60835	54.9216	52.27501	0.128484	0.104692
5	2	1	49.0841	57.61623	55.67233	0.122533	0.121708

По значениям критериев Акаике и Шварца и $RMSE$ и MAE, лучшими моделями оказались:

Таблица 4. Лучшие, из подобранных $ARIMA$-моделей

p	d	q	AIC	BIC	AICc	RMSE	MAE	LogLik
0	2	0	**43.50554**	**44.72442**	**43.67945**	1.717556	1.566667	-20.75277
5	2	0	47.60835	54.9216	52.27501	0.128484	**0.104692**	-17.80417
5	2	1	49.0841	57.61623	55.67233	**0.122533**	0.121708	**-17.54205**

Выберем модели $ARIMA(5,2,0)$ и $ARIMA(5,2,1)$. Значения их информационных критериев не сильно больше, чем у первой модели, а по остальным параметрам ($RMSE$, MAE и значение функции правдоподобия) они лучше.

2.1.4. Прогноз и сравнение с реальными данными

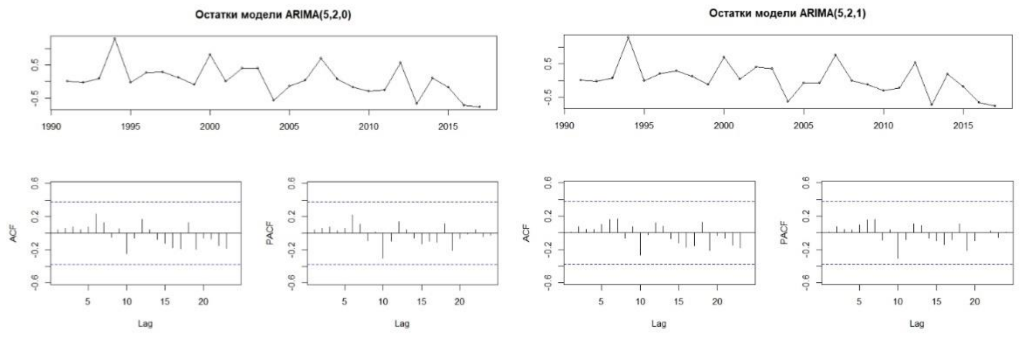

Рисунок 39. Остатки выбранных моделей

Теперь создадим прогнозы по подобранным моделям, построим их график и наложим эти графики на график реальных значений, из тестовой выборки.

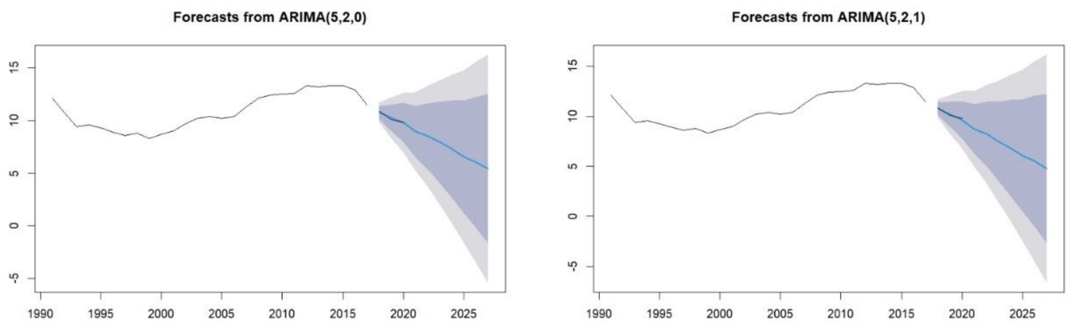

Рисунок 40. Результат работы функции $plot()$ - графики прогнозов, совмещенные с графиком реальных значений

Прогноз (синяя линия) обеих моделей достаточно хорошо повторяет поведение тестовой выборки (черная линия), есть небольшие неточности в начале. Скорее всего это связано с небольшим количеством данных в выборке.

2.1.5. MBB

Применим алгоритм Moving Block Bootstrap к стационарному временному ряду d_learn (второй разности изначального ряда). Оптимальная длина блока, была найдена в результате работы алгоритма HHJ и равна 1.

График полученной, бутстрапированной, выборки $series$, наложенный на $dlearn$:

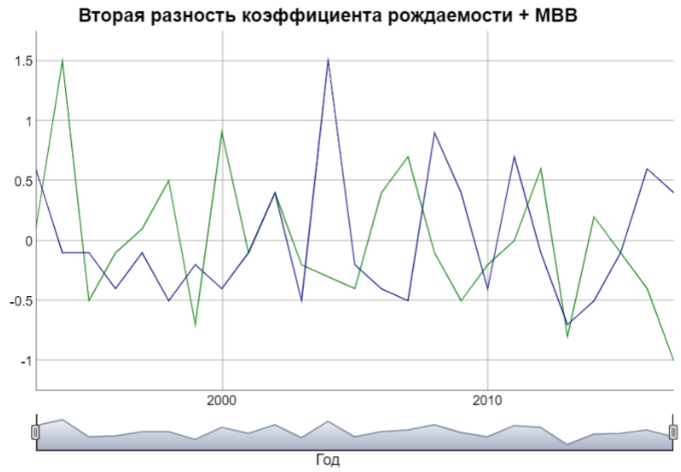

Рисунок 41. Результат работы алгоритма MBB

```
> summary(series)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -0.800  -0.400  -0.100   0.008   0.500   1.500
> summary(d_learn)
    POPFER_Y
 Min.   :-1.0
 1st Qu.:-0.4
 Median :-0.1
 Mean   : 0.0
 3rd Qu.: 0.4
 Max.   : 1.5
```

Рисунок 42. Основные характеристики рядов $series$ и d_learn

Характеристики этих рядов отличаются, но не так уж сильно. Медиана совпала, среднее практически не изменилось.

2.1.6. NBB

Применим алгоритм Non-overlapping Block Bootstrap к стационарному временному ряду d_learn (первой разности изначального ряда). Длина блока, как и в методе MBB, равна 1.

График полученной, бутстрапированной, выборки $series$, наложенный на $dlearn$:

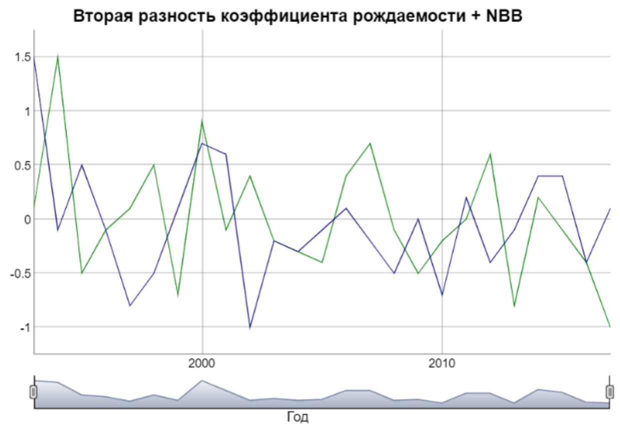

Рисунок 43. Результат работы алгоритма MBB

```
> summary(series)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 -1.000  -0.400  -0.100  -0.032   0.200   1.500 
> summary(d_learn)
    POPFER_Y   
 Min.   :-1.0  
 1st Qu.:-0.4  
 Median :-0.1  
 Mean   : 0.0  
 3rd Qu.: 0.4  
 Max.   : 1.5  
```

Рисунок 44. Основные характеристики рядов $series$ и d_learn

Ситуация с характеристиками ряда в результате работы алгоритма NBB не отличается от прошлого случая, потому что оптимальная длина блока равна 1 (выборка мала).

2.1.7. Заключение по эксперименту

Ряд был проанализирован. По нему были построены ARIMA модели, которые сравнили по RMSE, MAE, AIC, AICc, BIC, а также функции правдоподобия. Были выбраны две лучшие модели, для которых были построены прогнозы на 3 года.

Прогнозы для выбранных по перечисленному списку параметров моделей достаточно хорошо совпадают с тестовой выборкой.

Выполнено бутстрапирование рядов алгоритмами MBB и NBB, в результате работы которых были получены ряды с практически совпадающими характеристиками, причем из-за оптимальной длины блока, равной единице, значения характеристик по результатам работы этих двух алгоритмов не различались.

2.3. Ежемесячные доходы компании

2.3.1. Градиентный бустинг

Набор данных, временной ряд: https://www.kaggle.com/datasets/podsyp/time-series-starter-dataset – месячные показатели дохода некоторой компании с 2015 по 2020 год.

Сделаем предсказание для этого ряда с помощью градиентного бустинга (функции $xgb()$ из пакета $xgboost$).

Для начала избавимся от неопределенных значений в ряде и удалим последний столбец со сведениями о средней зарплате по регионам, т.к. для нас он неинтересен. Далее разобьем данные на две выборки: 52 месяца – обучающую, и 12 месяцев – тестовую. Обучим выборку с параметрами: функция потерь - reg:squarederror, максимальная глубина деревьев – 6, метрика, используемая для оценки качества модели - $RMSE$. В результате выполнения 100 итерация градиентного бустинга, получаем следующий прогноз:

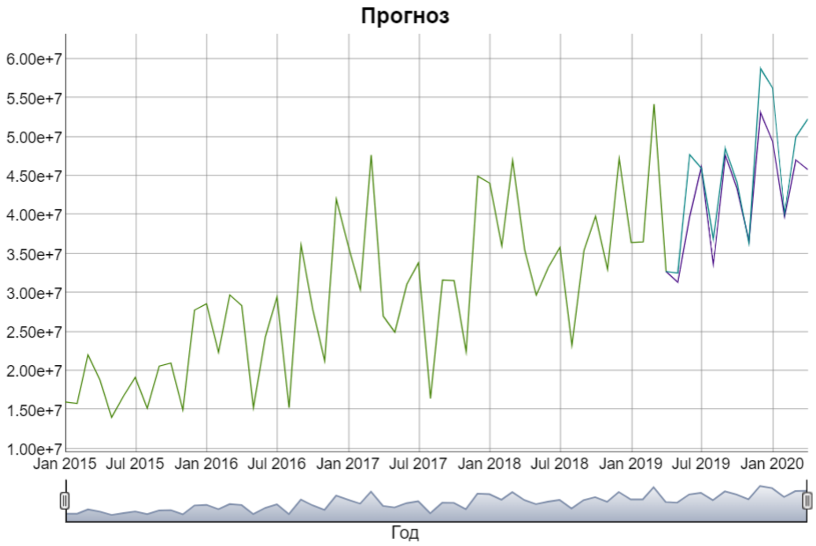

Рисунок 45. Прогноз с помощью градиентного бустинга

Полученный прогноз довольно хорошо совпадает с тестовой выборкой, имеются некоторые неточности, связанные с выбросами.

2.3.2. Заключение по эксперименту

Градиентный бустинг достаточно неплохо справился с задачей прогнозирования. В общем случае необходимо учитывать особенности временного ряда, чтобы прогноз был более точным.

Заключение

В ходе выполнения практической работы были рассмотрены нестационарные временные ряды:

- Ежемесячные показатели индекса зарплаты
- Ежегодные показатели коэффициента рождаемости

Каждый из этих рядов был проанализирован: разбит на обучающую и тестовую выборки. В обучающей были выявлены сезонность и тренд. Была проверена стационарность с помощью тестов Дики-Фуллера и Квятковского-Филлипса-Шмидта-Шина. Построена и проверена автоматическая модель с помощью функции $auto.arima()$.

Для первого ряда, поскольку в нем была явно выражена сезонность, были построены $ARIMA$-модели с сезонными параметрами. Несезонные параметры p и q подбирались среди значений от 0 до 6 и от 0 до 5 соответственно. Во втором ряде параметры p, d, q подбирались от 0 до 5, от 0 до 2 и от 0 до 1 соответственно.

Из этих моделей для каждого ряда были выбраны лучшие, причем, стоит отметить, что модели более высоких порядков обладали показателями лучше: функция правдоподобия – больше, значения критериев и ошибок – меньше. Затем по этим моделям были построены прогнозы и графики сравнения с действительными значениями из тестовых выборок и сделаны выводы по каждому из прогнозов.

Для каждого из рядов были выполнены алгоритмы MBB и NBB, по результатам выполнения которых можно сделать вывод, что основные характеристики рядов не поменялись.

Также для ряда, содержащего информацию про ежемесячные доходы некоторой компании, был выполнен градиентный бустинг, и с помощью него построен прогноз на год вперед.

Список литературы

1. ЕДИНЫЙ АРХИВ ЭКОНОМИЧЕСКИХ И СОЦИОЛОГИЧЕСКИХ ДАННЫХ // Статистическая база по макроэкономике РФ // Основные макроэкономические показатели. Оперативная информация // BRDATA // Индекс реальной зарплаты, месячный, цепной, с поправкой на сезонность (сглаженный) (WAG_M_SA) // [Электронный ресурс]. Режим доступа: http://sophist.hse.ru/hse/nindex.shtml (дата обращения 17.03.2023)

2. ЕДИНЫЙ АРХИВ ЭКОНОМИЧЕСКИХ И СОЦИОЛОГИЧЕСКИХ ДАННЫХ // Статистическая база по макроэкономике РФ // Население и трудовые ресурсы // Коэффициент рождаемости // Годовые показатели (POPMOR_Y) // [Электронный ресурс]. Режим доступа: http://sophist.hse.ru/hse/nindex.shtml (дата обращения 23.03.2023)

3. Hyndman, R. J., Forecasting: Principles and Practice[Текст]/ Hyndman R. J., Athanasopoulos G. Monash University, Australia. – 2018 - 149 с.

4. Box, G. E. P. et al. Time series analysis: forecasting and control.[Текст]/ G. E. P. Box, Gwilym M. Jenkins 2015 - John Wiley & Sons - 784 с.

5. Peng, R. D. R programming for data science. [Текст]/ R. D. Peng – Leanpub, 2016 – С. 86 181.

6. Доугерти, К. Введение в эконометрику.[Текст]/ К. Доугерти – 2004 - 465 с.

7. Орлов, А. И. Прикладная статистика [Текст]/ А. И. Орлов. - Издательство "Экзамен 2004. - 656 с.

Приложение 1

```
library("forecast")
library("tseries")
library("dplyr")
library("dygraphs")
library("ggfortify")
library("stats")
library("caret")
library("blocklength")
library("stats")
library("devtools")
library("plotly")
library("sophisthse")
library("Metrics")
data <- sophisthse("WAG_M_SA")
head(data)
data <- ts(data = data[,"WAG_M"],
            start = 1993,
            frequency = 12)
learn <- window(data, end = c(2021, 12)) # обучающий период
test <- window(data, start = c(2022, 1), end = c(2022, 12)) # тестовый период
dygraph(learn, main = "Индекс зарплаты",
        xlab = "Год",
        ylab = "") %>%
dyLegend(show = "follow") %>%
dyRangeSelector()
```

```
autoplot(stl(learn, s.window="periodic")) +
  labs(title = "Декомпозиция временного ряда",
       x = "Время")
dygraph(learn, main = "Рассмотрение сезонности",
        xlab = "Год",
        ylab = "индекс") %>%
  dyLegend(show = "follow") %>%
  dyRangeSelector(dateWindow = c("2019-01-01", "2022-01-01"))
dygraph(learn, main = "Индекс зарплаты",
```

```
        xlab = "Год",
        ylab = "") %>%
  dyLegend(show = "follow") %>%
  dyRangeSelector()
```

```
adf.test(learn, k=12, alternative = "stationary")
model <- auto.arima(learn)
model
adf.test(diff(learn), alternative = "stationary")
kpss.test(diff(learn))
dygraph(diff(learn), main = "Индекс зарплаты",
        xlab = "Год",
        ylab = "") %>%
  dyLegend(show = "follow") %>%
  dyRangeSelector()
```

```
automodel = auto.arima(learn, seasonal = TRUE)
automodel
auto_prediction <- forecast(automodel, h = 12)
auto_prediction
plot(auto_prediction)
```

```
Acf(diff(learn), lag.max = 48, main='Автокорреляция дифференцированного ряда'
)
Pacf(diff(learn), lag.max = 48, main='Частичная автокорреляция дифференцирова
нного ряда')
p_values <- c(0, 1, 2, 3, 4, 5, 6)
q_values <- c(0, 1, 2, 3, 4, 5)

RMSE <- c()
MAE <- c()
pp <- c()
qq <- c()
aic_arr <- c()
bic_arr <- c()
```

```r
aicc_arr <- c()

for (p in p_values) {
  for (q in q_values) {
    res <- tryCatch(
    {
      arima_model <- Arima(learn, order=c(p,1,q), seasonal=c(2,1,0))
      aic_arr = append(aic_arr, arima_model$aic)
      bic_arr = append(bic_arr, arima_model$bic)
      aicc_arr = append(aicc_arr, arima_model$aicc)
      pred = predict(arima_model, n.ahead = 12)
      RMSE = append(RMSE, rmse(test, pred$pred))
      MAE = append(MAE, mae(test, pred$pred))
      pp = append(pp, p)
      qq = append(qq, q)
    }, error = function(cond){return(NA)})
  }
}


length((aicc_arr))
DF = data.frame(pp, qq, aic_arr, bic_arr, aicc_arr, RMSE, MAE)
ar1 <- Arima(learn, order=c(1,1,5), seasonal=c(2,1,0))
ar2 <- Arima(learn, order=c(6,1,4), seasonal=c(2,1,0))
ar3 <- Arima(learn, order=c(1,1,0), seasonal=c(2,1,0))
tsdisplay(residuals(ar1), lag.max = 48, main = "Остатки модели ARIMA(6,1,4)(2,1,0)[12]")
tsdisplay(residuals(ar2), lag.max = 48, main = "Остатки модели ARIMA(1,1,5)(2,1,0)[12]")
ar1$loglik
ar2$loglik
ar3$loglik
forecast1 <- forecast(ar1, h = 12)
forecast2 <- forecast(ar2, h = 12)
plot(forecast1)
lines(test)
plot(forecast2)
lines(test)
```

```
# Moving Blocks Bootstrap
d_learn <- diff(learn)
block_size <- hhj(d_learn)$"Optimal Block Length"
reps <- 1000
data_size <- length(d_learn)
d_learn
mbb_v <- rep(NA,reps)
for(i in 1:reps) {
  series <- rep(NA,data_size)
  for(j in 1:ceiling(data_size/block_size)) {
    endpoint <- sample(block_size:data_size, size=1)
    #print(endpoint)
    series[(j-1)*block_size+1:block_size] <- d_learn[endpoint-(block_size:1)+
1]
  }
  series <- series[1:data_size]
  mbb_v[i] <- cor(series[-1],series[-data_size])
}

series <- ts(data = series,
                  start = c(1993, 2),
                  frequency = 12)
series
salaries <- cbind(d_learn, series)
dygraph(salaries, main = "Первые разности индекса зарплаты + MBB",
        xlab = "Год",
        ylab = "") %>%
  dyLegend(show = "follow") %>%
  dyRangeSelector()

summary(series)
summary(d_learn)
```

```
# Non-Overlapping Blocks Bootstrap
block_size <- 2
reps <- 1000
```

```
data_size <- length(d_learn)
ceiling(data_size/block_size)
N <- data_size/block_size
nbb_v <- rep(NA,reps)

for(i in 1:reps) {
  series <- rep(NA,data_size)
  used <- c(1:N)
  for(j in 1:N) {
    block_num <- sample(used, size=1)
    used <- used[! used %in% c(block_num)]
    series[(j-1)*block_size+1:block_size] <- d_learn[(block_num-1)*block_size
+1:block_size]
  }
  series <- series[1:data_size]
  nbb_v[i] <- cor(series[-1],series[-data_size])
}

data_ds
series <- ts(data = series,
             start = c(1993, 2),
             frequency = 12)
series
salaries <- cbind(d_learn, series)
dygraph(salaries, main = "Первые разности индекса зарплаты + NBB",
        xlab = "Год",
        ylab = "") %>%
  dyLegend(show = "follow") %>%
  dyRangeSelector()
summary(series)
summary(d_learn)
```

Приложение 2

```
library("forecast")
library("tseries")
library("dplyr")
library("dygraphs")
library("ggfortify")
library("stats")
library("caret")
library("blocklength")
library("stats")
library("devtools")
library("plotly")
library("sophisthse")
library("Metrics")

# Загрузка данных (для примера были взяты данные с сайта sophist.hse.ru о еже
годной рождаемости)
data <- sophisthse("POPFER_Y")

head(data)
class(data)
```

```
learn <- window(data, end = c(2017)) # обучающий период
test <- window(data, start = c(2018), end = c(2020)) # тестовый период
dygraph(learn, main = "Коэффициент рождаемости",
        xlab = "Год",
        ylab = "") %>%
  dyLegend(show = "follow") %>%
  dyRangeSelector()
```

```
adf.test(learn)
adf.test(diff(learn))
kpss.test(diff(learn))
adf.test(diff(diff(learn)), alternative = "stationary")
kpss.test(diff(diff(learn)))
dygraph(diff(diff(learn)), main = "Вторая разность коэффициента рождаемости",
```

```
        xlab = "Год",
        ylab = "") %>%
  dyLegend(show = "follow") %>%
  dyRangeSelector()
```

```
automodel = auto.arima(learn)
automodel
auto_prediction <- forecast(automodel)
auto_prediction
plot(auto_prediction)
```

```
Acf(learn, lag.max = 24, main='Автокорреляция дифференцированного ряда')
Pacf(learn, lag.max = 24, main='Частичная автокорреляция дифференцированного
ряда')
p_values <- c(0, 1, 2, 3, 4, 5)
d_values <- c(0, 1, 2)
q_values <- c(0, 1)

RMSE <- c()
MAE <- c()
pp <- c()
dd <- c()
qq <- c()
aic_arr <- c()
bic_arr <- c()
aicc_arr <- c()

for (p in p_values) {
  for (d in d_values) {
    for (q in q_values) {
      #print(p)
      #print(q)
      res <- tryCatch(
        {
          arima_model <- Arima(learn, order=c(p,d,q))
          aic_arr = append(aic_arr, arima_model$aic)
          bic_arr = append(bic_arr, arima_model$bic)
```

```
          aicc_arr = append(aicc_arr, arima_model$aicc)
          pred = predict(arima_model, n.ahead = 3)
          RMSE = append(RMSE, rmse(test, pred$pred))
          MAE = append(MAE, mae(test, pred$pred))
          pp = append(pp, p)
          dd = append(dd, d)
          qq = append(qq, q)
        }, error = function(cond){return(NA)})
      }
    }
}
DF = data.frame(pp, dd, qq, aic_arr, bic_arr, aicc_arr, RMSE, MAE)
DF
ar1 <- arima(learn, order=c(0, 2, 0))
ar2 <- arima(learn, order=c(5, 2, 0))
ar3 <- arima(learn, order=c(5, 2, 1))
tsdisplay(residuals(ar1), lag.max = 48, main = "Остатки модели ARIMA(0,2,0)")
tsdisplay(residuals(ar2), lag.max = 24, main = "Остатки модели ARIMA(5,2,0)")
tsdisplay(residuals(ar3), lag.max = 48, main = "Остатки модели ARIMA(5,2,1)")
ar1$loglik
ar2$loglik
ar3$loglik
forecast1 <- forecast(ar2)
forecast2 <- forecast(ar2)
plot(forecast1)
lines(test)
plot(forecast2)
lines(test)
```

```
# Moving Blocks Bootstrap
d_learn <- diff(diff(learn))
block_size <- hhj(d_learn)$"Optimal Block Length"
reps <- 1000
data_size <- length(d_learn)
mbb_v <- rep(NA,reps)
for(i in 1:reps) {
```

```
  series <- rep(NA,data_size)
  for(j in 1:ceiling(data_size/block_size)) {
    endpoint <- sample(block_size:data_size, size=1)
    #print(endpoint)
    series[(j-1)*block_size+1:block_size] <- d_learn[endpoint-(block_size:1)+
1]
  }
  series <- series[1:data_size]
  mbb_v[i] <- cor(series[-1],series[-data_size])
}
series <- ts(data = series,
             start = c(1993))
series
salaries <- cbind(d_learn, series)
dygraph(salaries, main = "Вторая разность коэффициента рождаемости + MBB",
        xlab = "Год",
        ylab = "") %>%
  dyLegend(show = "follow") %>%
  dyRangeSelector()
summary(series)
summary(d_learn)
```

```
# Non-Overlapping Blocks Bootstrap
block_size <- 1
reps <- 1000
data_size <- length(d_learn)
N <- data_size/block_size
nbb_v <- rep(NA,reps)

for(i in 1:reps) {
  series <- rep(NA,data_size)
  used <- c(1:N)
  for(j in 1:N) {
    block_num <- sample(used, size=1)
    used <- used[! used %in% c(block_num)]
    series[(j-1)*block_size+1:block_size] <- d_learn[(block_num-1)*block_size
+1:block_size]
  }
```

```
  print(length(used))
  series <- series[1:data_size]
  nbb_v[i] <- cor(series[-1],series[-data_size])
}
series <- ts(data = series,
             start = c(1993))
series
salaries <- cbind(d_learn, series)
dygraph(salaries, main = "Вторая разность коэффициента рождаемости + NBB",
        xlab = "Год",
        ylab = "") %>%
  dyLegend(show = "follow") %>%
  dyRangeSelector()
```

```
summary(series)
summary(d_learn)
```

Приложение 3

```r
library("caret")
library("tidyverse")
library("xgboost")
library("tseries")
library("dplyr")
library("dygraphs")
library("forecast")

data <- read_csv("Month_Value_1.csv")
View(data)

data_framed = as.data.frame(data)[-5]

df = data_framed
df
summary(df)
df$Period <- as.Date(df$Period, "%m.%d.%Y")

typeof(df$Period)
df <- na.omit(df)
# разделим данные
train <- df[1:52,]
test <- df[52:64,]

dtrain <- xgb.DMatrix(data = as.matrix(train[,3:ncol(train)]), label = train[
,2])
dtest <- xgb.DMatrix(data = as.matrix(test[,3:ncol(test)]), label = test[,2])

params <- list(
  objective = "reg:squarederror",
  max_depth = 6,
  eta = 0.3,
  nthread = 4,
  eval_metric = "rmse"
)
```

```
model <- xgb.train(
  params = params,
  data = dtrain,
  nrounds = 100
)

preds <- predict(model, dtest)

preds
test
train_ts <- ts(data = train[,2],
              start = c(2015, 1),
              frequency = 12)

preds_ts <- ts(data = preds,
             start = c(2019, 4),
             frequency = 12)
test_ts <- ts(data = test[,2],
              start = c(2019, 4),
              frequency = 12)

revenue <- cbind(train_ts, preds_ts, test_ts)
dygraph(revenue, main = "Прогноз",
        xlab = "Год",
        ylab = "") %>%
  dyLegend(show = "follow") %>%
  dyRangeSelector()
```