|  |
| --- |
|  |

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ МОРДОВИЯ**

**Государственное бюджетное профессиональное образовательное учреждение Республики Мордовия «Инсарский аграрный техникум»**

Учебно-методическая разработка

на тему:

Основы работы с временными рядами на **платформе для аналитики данных и машинного обучения**

**Knime Analytics Platform**

Автор учебно-методической разработки:

Т.Е. Шабанова – преподаватель математики и информатики

ГБПОУ РМ «Инсарский аграрный техникум»

город Инсар, 2023 год

**РЕФЕРАТ**

Учебно-методическая разработка содержит 37 страниц печатного текста, в том числе 36 рисунков. Для написания работы были использованы 6 источников: 2 книжных издания и 3 электронных ресурса.

KONSTANZ INFORMATION MINER (KNIME), KNIME ANALYTICS PLATFORM, ВРЕМЕННЫЕ РЯДЫ, WORKFLOW, МОДЕЛЬ ARIMA, **ТРЕНД**

Объектом исследования являются Knime Analytics Platform.

Цель работы: изучить основы работы с временными рядами в Knime Analytics Platform.

В процессе работы проводилось изучение специальной литературы по направлению исследования и выполнение практической работы по изучению способов ввода данных и их первичному анализу в Knime Analytics Platform.

В результате исследования было изучено несколько важных особенностей платформы Knime Analytics Platform и проведен самостоятельный элементарный анализ данных, взятых с сайта https://hh.ru о вакансии "программист" в Саранске.

Степень внедрения – частичная.

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ………………………………………………………………….…. 5

1 Основные сведения о Knime Analytics Platform……………………….…… 7

2 Способы ввода данных и первичный анализ в Knime Analytics Platform .. 9

3 Методы анализа временных рядов Knime Analytics Platform…………….. 23

ЗАКЛЮЧЕНИЕ………………………………………………………………... 36

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ……………………….…. 37

**ВВЕДЕНИЕ**

 Konstanz Information Miner (KNIME) – бесплатная платформа для интеграции данных с открытым исходным кодом, их анализа и отчетности.

Эта платформа объединяет различные компоненты для интеллектуального анализа данных  и машинного обучения с помощью концепции модульной конвейерной обработки данных.

Разработку этого программного обеспечения начали инженеры-программисты Констанцского университета (ФРГ) под руководством Майкла Бертольда  в январе 2004 года. Основная идея создания заключалась в том, чтобы создать модульную, масштабную и открытую платформу обработки данных, при помощи которой можно было бы легко интегрировать различные модули загрузки, обработки, преобразования, анализа и визуального исследования данных без опоры на какую-либо конкретную область применения. Она должна была стать фундаментом для сотрудничества в сфере исследования больших данных, а также служить интеграционной основой для различных других проектов анализа данных.

В работе предполагается применять платформу KNIME для анализа времнных рядов. Актуальность выбранной темы заключается в том, что анализ временных рядов в настоящее время является одной из востребованных сфер аналитики. Ключевой частью анализа временных рядов является изучение основ прогнозирования с помощью соответствующих моделей.

Цель работы: изучить основу работы с временными рядами в Knime Analytics Platform.

Задачи учебно-методической разработки:

* изучить литературу по рассматриваемой теме,
* ознакомиться с Knime Analytics Platform,
* провести элементарный анализ данных в Knime Analytics Platform,
* ознакомиться теоретическими аспектами анализа данных на основе динамических моделей временных рядов,
* рассмотреть методы анализа временных рядов Knime Analytics Platform.

**1 Основные сведения о Knime Analytics Platform**

KNIME Analytics Platform представляет собой бесплатное программное обеспечение с открытым исходным кодом для создания приложений и сервисов Data Science.

В KNIME Analytics Platform есть возможность создавать визуальные рабочие процессы (Workflow) состоящие из узлов (node) с понятным графическим интерфейсом в стиле drag and drop (перетаскивания) и последующей настройкой конфигурации узлов.

KNIME Analytics Platform обеспечивает чтение данных из различных источников, их преобразование и фильтрацию, сам анализ данных, визуализацию и экспорт, то есть полный цикл анализа данных.

В бесплатном приложении для анализа и обработки данных Knime процесс программирования осуществляется через создание рабочего процесса (Workflow). Workflow состоит из узлов которые выполняют ту или иную функцию (например, чтение данных, трансформация, визуализация и т.п.). Узлы, соответственно, соединяются между собой стрелками, которые показывают направление движения данных.

Следует отметить, что:

* В дистрибутив KNIME входит большое количество сторонних библиотек (например JFreeChart), которые становятся доступны в виде узлов. Полная версия KNIME со всеми плагинами «весит» почти 2 гигабайта.
* При покупке KNIME-Server будет доступно исполнение workflow на сервере и предоставление доступа к результатам работы через REST API.
* В открытом доступе размещена большая библиотека примеров.
* Можно осуществлять работу с Hadoop и другими BigData источниками.

В KNIME Analytics Platform наглядно видно как преобразуются данные, по каким они перемещаются веткам, что вообще происходит с данными. KNIME Analytics Platform позволяет как изучать реализацию существующих узлов, так и дописывать свои узлы, увеличивая ими библиотеку.

**2 Способы ввода данных и первичный анализ в Knime Analytics Platform**

При написании учебно-методической разработки были решены следующие задачи:

1. проведен элементарный анализ данных, находящихся в файле Excel и построена «ящичковая диаграмма»;
2. проведен анализ данных с сайта https://hh.ru о вакансии "программист" в Саранске.

Для выполнения задания №1 был создан рабочий процесс Workflow. Перенеся на рабочий стол и воспользовавшись узлом (node) Excel Reader, были считаны данные с excel файла (рисунок 2.1).

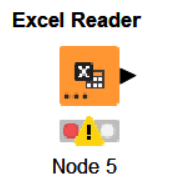


Рисунок 2.1

Для перевода светофора во второе положение была выполнена настройка узла (node): указан  диапазон с числовыми значениями, не включая текст (рисунок 2.2).

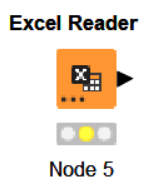


Рисунок 2.2

Это означает, что процесс еще не выполняется, но данные уже подключены. Чтобы переключить светофор во второе положение, была нажата на node правая кнопка мыши и выбран пункт из выпадающего меню Execute (рисунки 2.3 и 2.4).

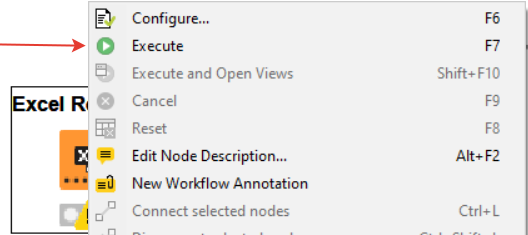


Рисунок 2.3



Рисунок 2.4

Для просмотра результата в выпадающем меню необходимо перейти в File Table (рисунок 2.5).

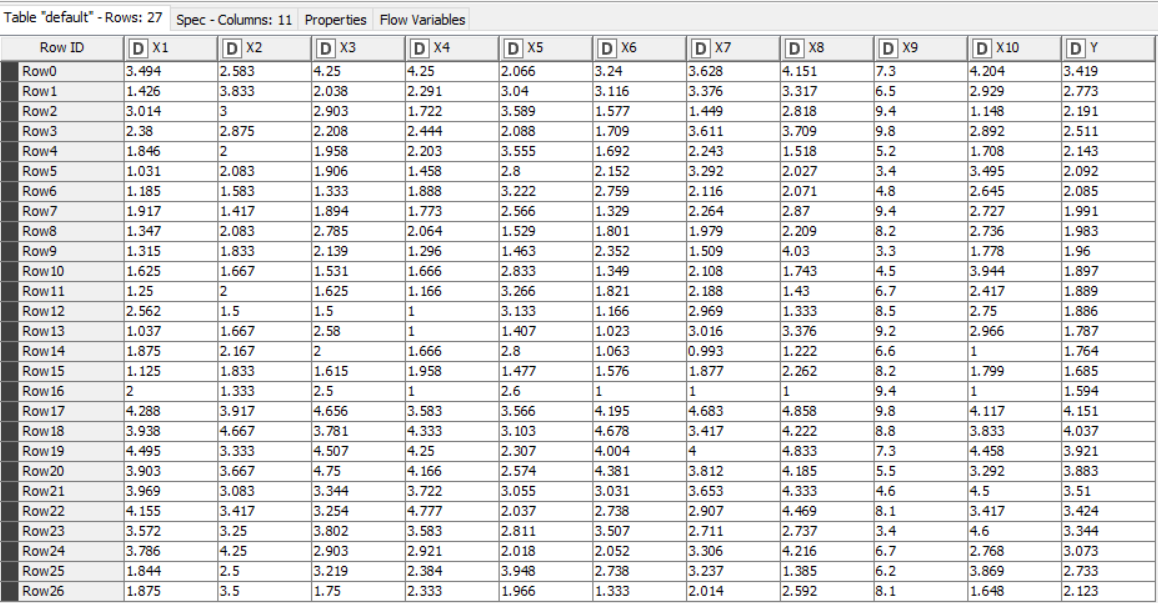


Рисунок 2.5

Тут же во второй вкладке можно посмотреть типы столбцов, минимальный/максимальный элементы и другие характеристики.

Для вывода «ящичковой диаграммы» был выбран узел (node) Box Plote и соединен с Excel Reader. Светофор переходит во второе положение (рисунок 2.6).

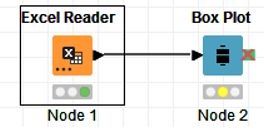


Рисунок 2.6

Настраивааем конфигурацию процесса (рисунок 2.7).

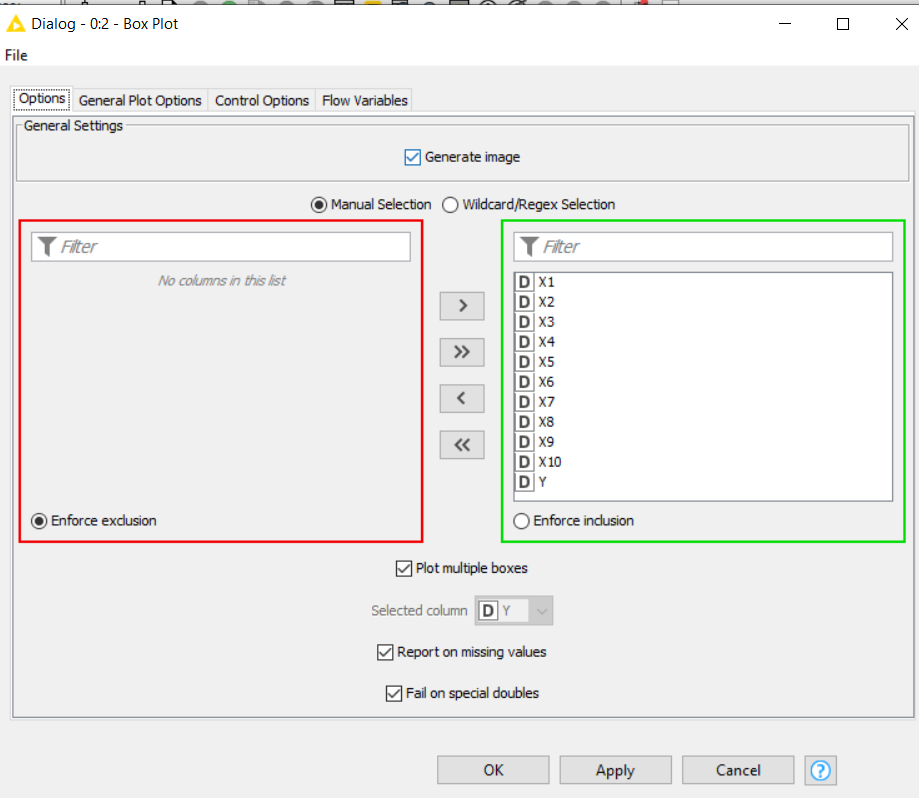


Рисунок 2.7

В окне конфигурации можно выбрать необходимые фильтры. Так же необходимо поставить галочку перед Generate image, чтобы сгенерировать рисунок. Для просмотра «ящичковой диаграммы» нажимаем правой кнопкой мыши и Box Plote Image и получаем следующую визуализацию (рисунки 2.8 и 2.9).

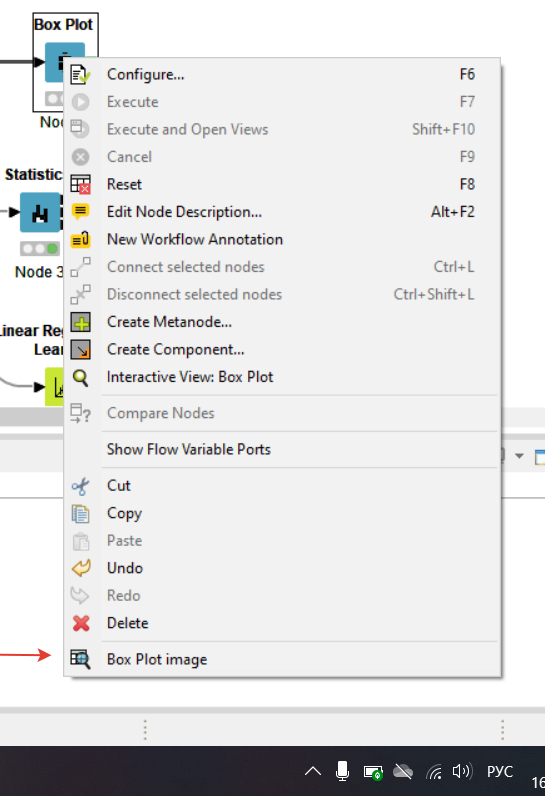


Рисунок 2.8

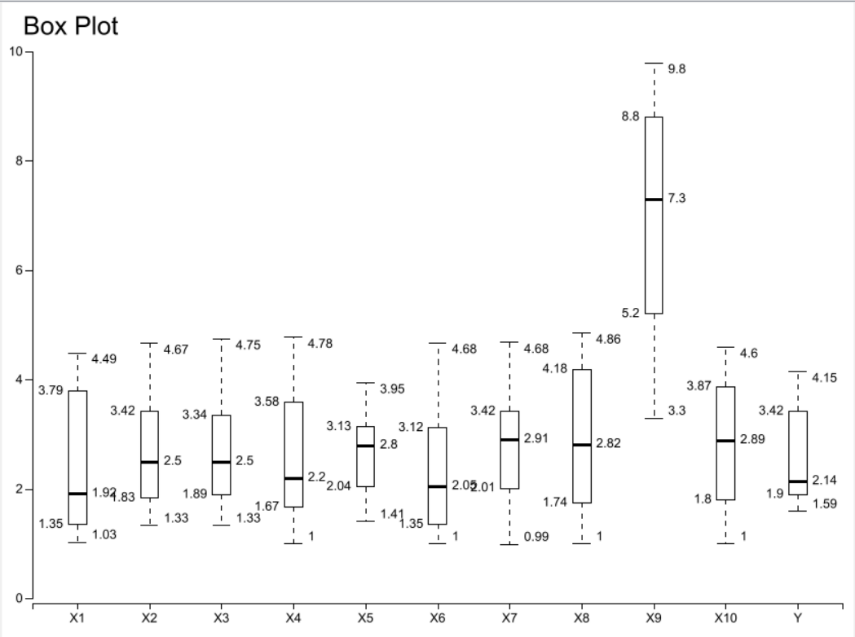


Рисунок 2.9

Используя узел Statistics, выполним элементарный статистический анализ данных из Excel файла (рисунок 2.10).

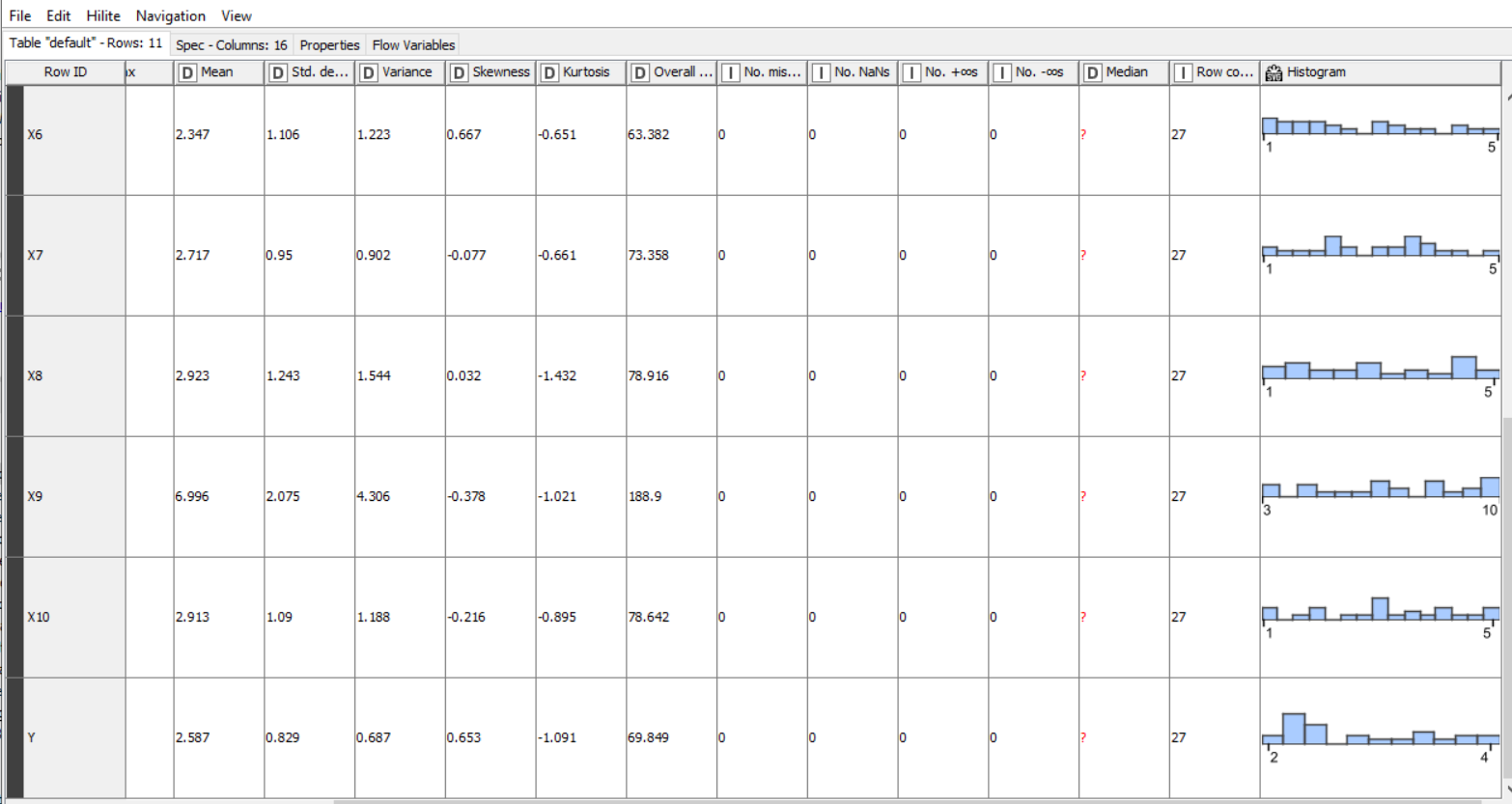


Рисунок 2.10

Далее, используя узел Linear Regression Learner, найдем коэффициенты линейной модели регрессии (рисунок 2.11).

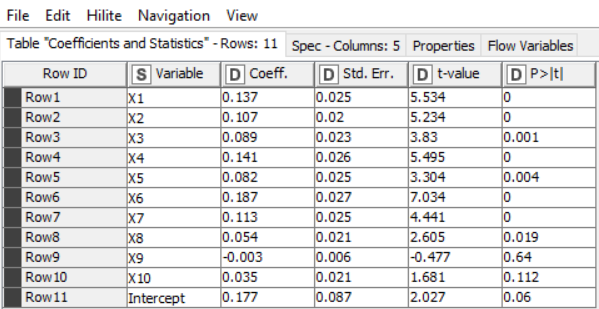


Рисунок 2.11

Рабочий процесс выполнения задания №1 представлены на рисунке 2.12.

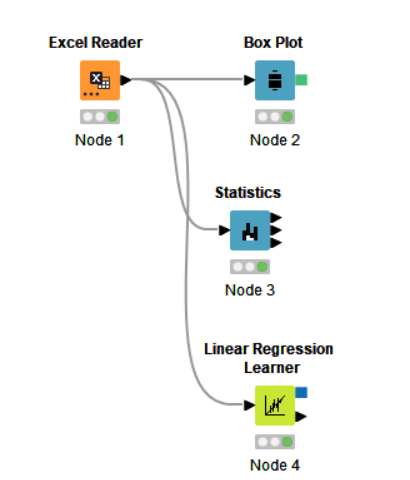


Рисунок 2.12

Для выполнения задания №2 извлекаем данные с сайта https://hh.ru о вакансии "программист" города Саранск и анализируем их.

Был создан рабочий процесс Workflow. Затем, перенеся на рабочий стол и воспользовавшись узлом (node) GET Request, считаны данные с сайта (рисунок 2.13).



Рисунок 2.13

По умолчанию для этого узла светофор горит желтым.

Это означает, что процесс еще не выполняется. Чтобы переключить светофор во второе положение, нажимаем на node правой кнопкой мыши и выбрем Configure. В строчку URL внесем код на Python (рисунок 2.14).

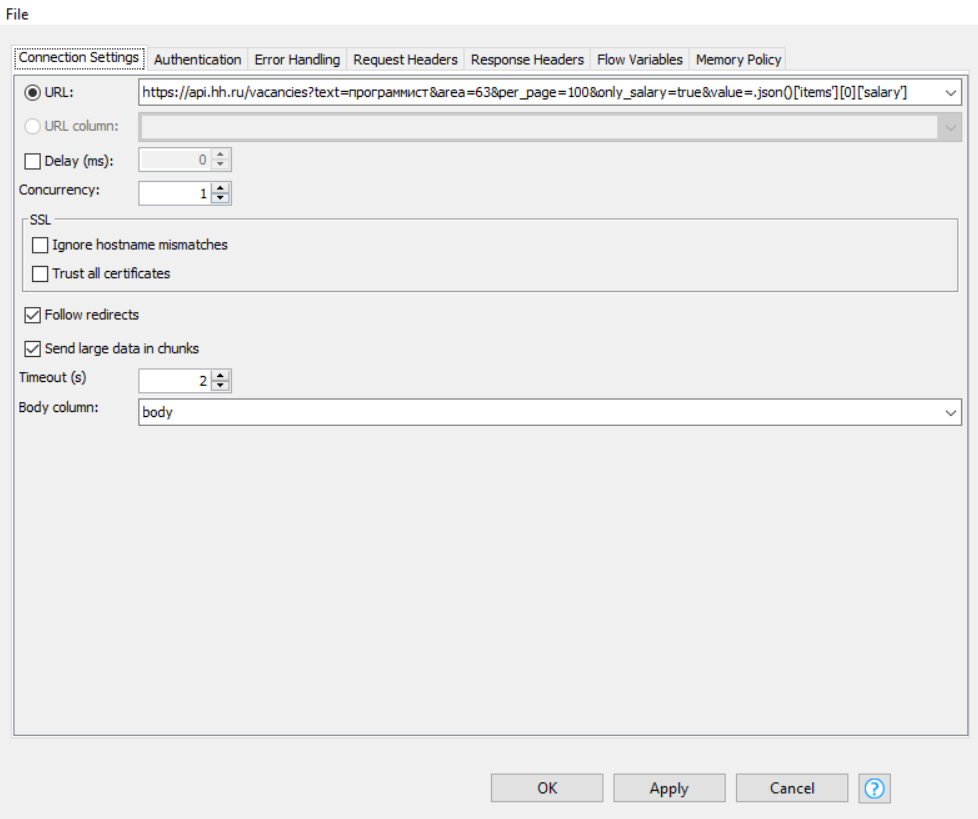


Рисунок 2.14

Нажимаем ОК снова правой кнопкой мыши на узел и выбраем Execute. После этого светофор переходит в третье положение – это означает, что процесс выполнен (рисунок 2.15).

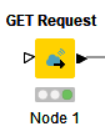


Рисунок 2.15

Для просмотра результата нажимаем правой кнопкой мыши и переходим в Get results (рисунок 2.16).

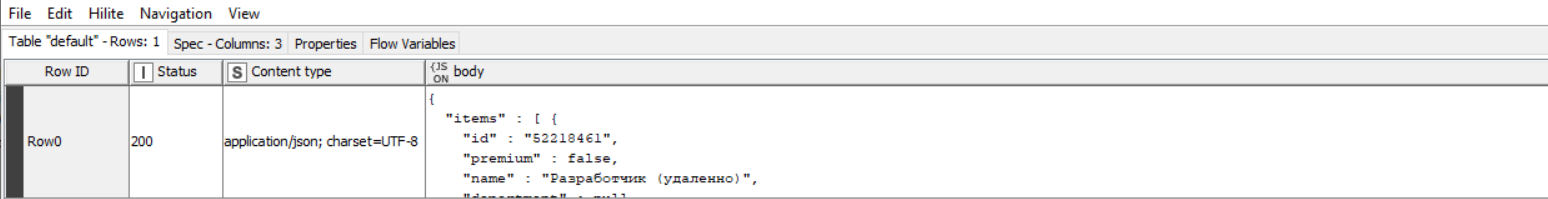


Рисунок 2.16

В выведенном окне видно, что полученный json попал в одно единственное поле. Необходимо извлечь 100 вакансий. Для этого воспользуемся узлом Json Path (рисунок 2.17).



Рисунок 2.17

Чтобы JSON Path извлек данные, нужно соединить его с GET Request (рисунок 2.18).

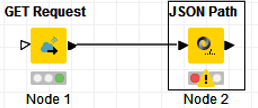


Рисунок 2.18

Нажимаемо Json Path правой кнопкой мыши, выбираем configure для получения таблицы с id вакансии с названиями и зарплатой.

Во вкладке Setting выбираем «id» и нажимаю по нему ПКМ и далее Add JSONPath, так же поступаем с «name». Далее находим «salary» и поступаем так же со строчкой «from». Так же, чтобы работать не только с одной вакансией, а со всеми, заменил 0 после ['items'] на \* (рисунок 2.19).

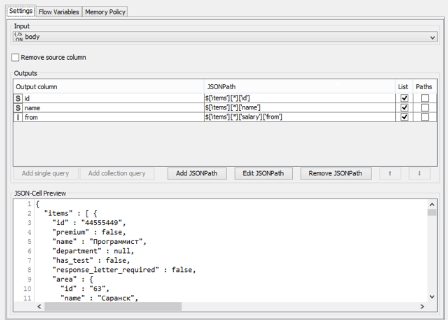


Рисунок 2.19

Нажимаем ОК, снова правой кнопкой мыши на JSON Path и выбираем Execute. Чтобы посмотреть результат, нажимаем правой кнопкой мыши и далее Table. Теперь появились списки со всеми «id», «name» и «salary». Нужно разделить эти списки на части, для этого воспользуюсь Ungroup. И так же соединяем его с предыдущим узлом (рисунок 2.20).

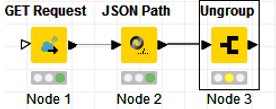


Рисунок 2.20

Можно сразу выполнить этот процесс и посмотреть результат. Нажимаем по Ungroup ПКМ и на Execut, далее повторяю действие с ПКМ и выбираю Data Table. В появившемся окне, видем, что и теперь есть отдельные колонки и «id», «name» и «from». И что самое главное – отделили вакансии из общего массива (рисунок 2.21).

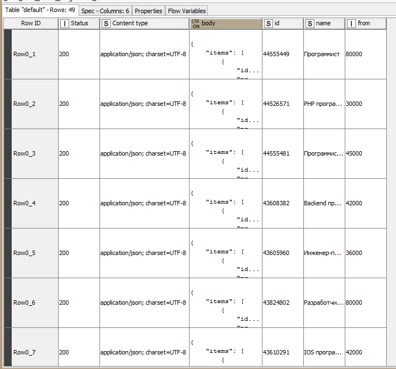


Рисунок 2.21

Для вывода «ящичковой диаграммы» перенес на рабочий стол (node) Box Plote и соединил Ungroup🡪Box Plot. Светофор переходит во второе положение (рисунок 2.22).

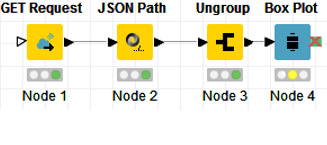


Рисунок 2.22

Настраиваем конфигурацию процесса. В окне конфигурации оставляем только «from» (рисунок 2.23).

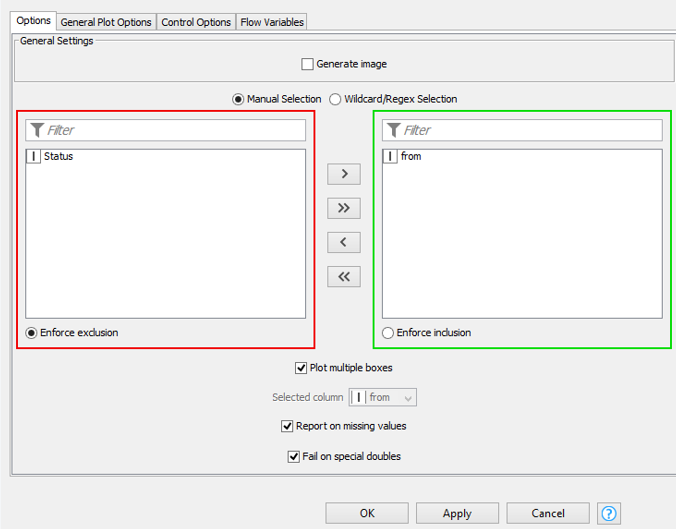


Рисунок 2.23

Нижний квартиль равен 30000, верхний 80000, медиана 42000, минимальное значение 10000, максимальное 150000 и нет аномальных значений (рисунок 2.24).

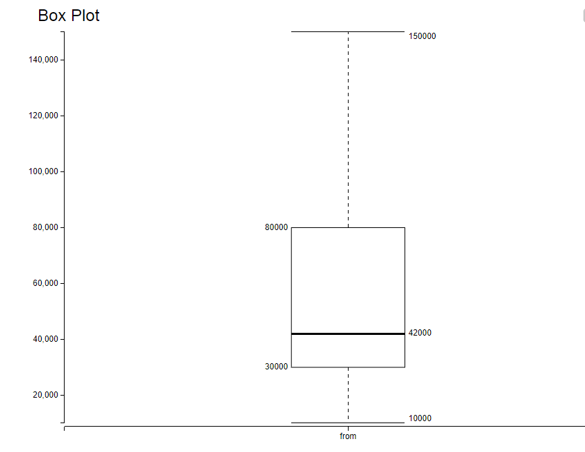


Рисунок 2.24

Используя узел Statistics, выполняем элементарный статистический анализ.

Далее создаем Pie/Donut Chart для построение круговой диаграммы.

С помощью узла Missing Value заполняем пропущенные значения полей средними значениями.

Далее, используя узел Linear Regression Learner, строим линейную регрессию зависимости верхнего предела предлагаемой зарплаты от нижнего (рисунок 2.25).

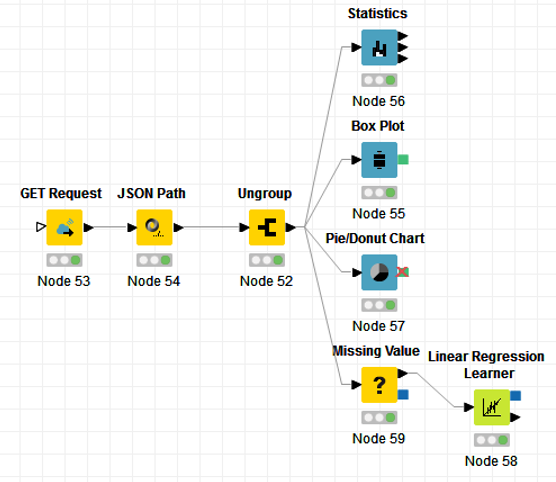


Рисунок 2.25

**3 Методы анализа временных рядов Knime Analytics Platform**

Анализ временных рядов в настоящее время является одной из востребованных сфер аналитики.

Для анализа временных рядов существует большое количество различных методов. Наиболее распространенными из них  являются следующие:

1. Корреляционный анализ. Он позволяет выявить существенные периодические зависимости и их лаги (задержки) внутри одного процесса (автокорреляция) или между несколькими процессами (кросскорреляция);
2. Спектральный анализ. Позволяет  находить периодические и квазипериодические составляющие временного ряда;
3. Модели авторегрессии и скользящего среднего, которые  оказываются особенно полезными для описания и прогнозирования процессов, проявляющих однородные колебания вокруг среднего значения;
4. Прогнозирование. Позволяет на основе подобранной  модели поведения временного рада предсказывать его значения в будущем;
5. Сглаживание и фильтрация. Они предназначены для преобразования временных рядов для удаления из них высокочастотных или сезонных колебаний.

Для изучения основных шагов при выполнении анализа временных рядов с помощью платформы KNIME Analytics используем определенные компоненты в KNIME. Они содержат готовые узлы для легкого выполнения задач анализа временных рядов.

Узел KNIME, который инкапсулирует функциональность рабочего процесса KNIME, например обучение модели ARIMA – это компонент. Компоненты можно использовать повторно и совместно использовать локально, через сервер KNIME или на концентраторе KNIME.

Компоненты для анализа временных рядов охватывают различные задачи от агрегирования и проверки сезонности во временных рядах до построения модели авторегрессионной интегрированной скользящей средней (ARIMA) и проверки остатков модели. Эти компоненты используют интеграцию KNIME Python, расширяя аналитические возможности платформы KNIME Analytics для анализа временных рядов с помощью модуля statsmodels в Python. Однако код выполняется только в фоновом режиме, и можно определить параметры для каждой задачи, как и для любого другого узла KNIME: в диалоговом окне конфигурации компонента.

Временные ряды имеют различные приложения источники и приложения: данные датчиков смарт-часов для анализа тренировки, ежедневные данные о продажах для прогнозирования спроса и многое другое. Все эти временные ряды отличаются: своей детализацией, регулярностью, чистотой и т.д. Возможно также, что данные временных рядов недоступны через регулярные промежутки времени, но могут быть собраны только из случайных точек событий. Но общим для всех этих типов данных временных рядов является то, что они собираются из одного и того же источника с течением времени.

**3.1 Регуляризация и очистка временных рядов**

Как только данные временных рядов собраны, необходимо сделать их равномерно распределенными с подходящей степенью детализации, чистыми и непрерывными. Требуемые задачи зависят от цели аналитики и исходной формы данных.

**3.2 Сортировка**

Временные ряды должны быть отсортированы по времени. Необходимо агрегировать их по метке времени, если данные содержат более одной записи на метку времени.  Можно дополнительно агрегировать данные с предпочтительной степенью детализации, если нас интересуют временные ряды с другой степенью детализации, а не то, что имеем сейчас в данных.

**3.3 Недостающие значения**

При отсутствии некоторых временных меток, необходимо ввести их во временной ряд, чтобы сделать его равным. Недостающие записи иногда являются частью динамики временных рядов.Эти недостающие значения можно заплнить, например, с помощью линейной интерполяции или значений скользящего среднего. Лучший метод для заполнения недостающих значений зависит от регулярной динамики данных. Если отсутствующие значения отсутствуют не случайным образом, их можно заменить фиксированным значением, которое в этом случае будет равно 0. А если пропущенные значения случайны и они встречаются достаточно далеко в прошлом, можно использовать данные после пропущенного значения и пропустить более старые данные.

**3.4 Нерегулярные данные**

Сглаживание данных является хорошим способом обработки колебаний и выбросов. Можно использовать несколько методов – [скользящее среднее](https://en.wikipedia.org/wiki/Moving_average) и [экспоненциальное сглаживание](https://en.wikipedia.org/wiki/Exponential_smoothing#:~:text=Exponential%20smoothing%20is%20a%20rule,exponentially%20decreasing%20weights%20over%20time.). Но иногда временные ряды показывают очень нерегулярное явление. Тогда можно попробовать сделать временные ряды более регулярными, например, извлечь их подмножество (на рисунке 3.4.1 рассматриваетсяя продажи только одного продукта вместо продаж всего супермаркета).

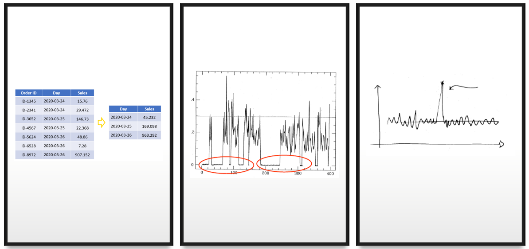


Рисунок 3.4.1

**3.5 Изучение и преобразование временных рядов**

 Различные графики и статистика выявляют долгосрочные и краткосрочные формы и временные отношения во временном ряду, которые можно использовать для лучшего понимания динамики процесса и прогнозирования его будущего развития.

**3.6 Визуальное исследование временных рядов**

Графики на рисунке к 3.6.1 показывают возможное направление, регулярные и нерегулярные колебания, выбросы, разрывы или поворотные точки во временном ряду. На сезонном графикеможно проверить каждый сезонный цикл (год) отдельно, если наблюдается регулярная закономерность во временных рядах, например, ежегодную сезонность продаж напитков. Если нужно узнать, что происходит в течение сезонов, например, какова медиана продаж в летние месяцы и насколько, и в какую сторону продажи меняются каждый месяц, можно проверить эти виды динамики на условном прямоугольном графике. График лага показывает взаимосвязь между текущими значениями и прошлыми значениями, например, продажи сегодня и продажи неделей ранее.

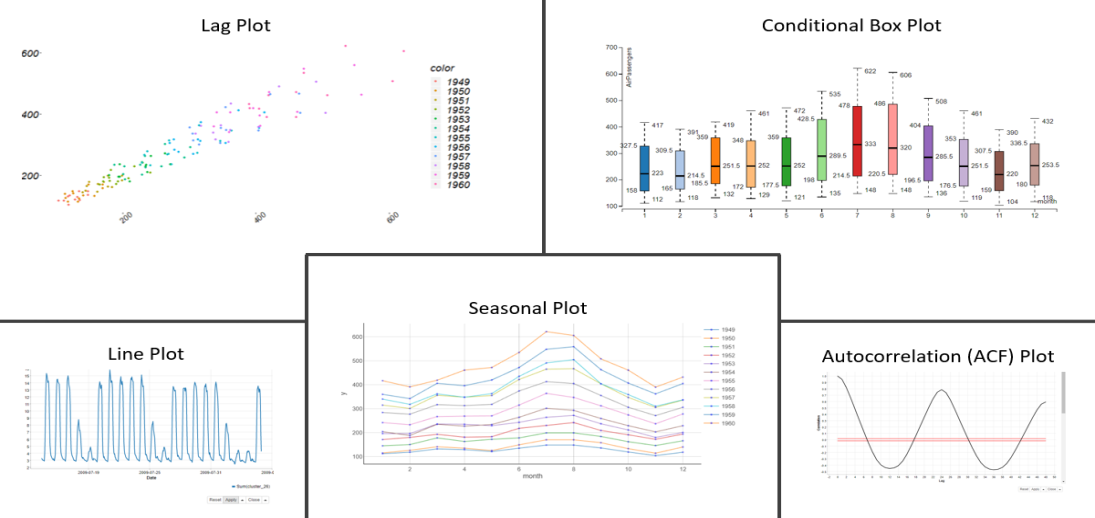


Рисунок 3.6.1

**3.7 Проверка сезонности и тренда**

Прежде чем приступить к моделированию агрегированных, очищенных данных, лучше изучить его визуально. Распространенные модели предполагают стационарный временной ряд. Это говорит о том, что его статистика остается неизменной с течением времени. Поэтому следует разложить временной ряд на его тренд и сезонность, и поместить модель в ее нерегулярную часть.

Можно проверить сезонность во временных рядах на автокорреляционном графике (ACF). Если на графике регулярные пики и минимумы – это говорят о сезонности во временном ряду, которую можно устранить, распределив данные по лагу с наибольшей корреляцией.  Используем компонент сезонности Inspect (рисунок 3.7.1), чтобы найти этот локальный максимум.  Компонент Remove Seasonality «Удалить сезонность» применяем, чтобы удалить сезонность на локальном максимуме.  Повторив эту процедуру (рисунок 3.7.2), можно устранить вторую, третью и другие сезонности.

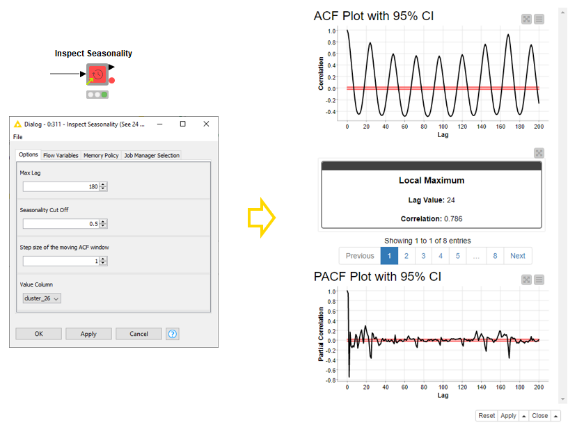


Рисунок 3.7.1

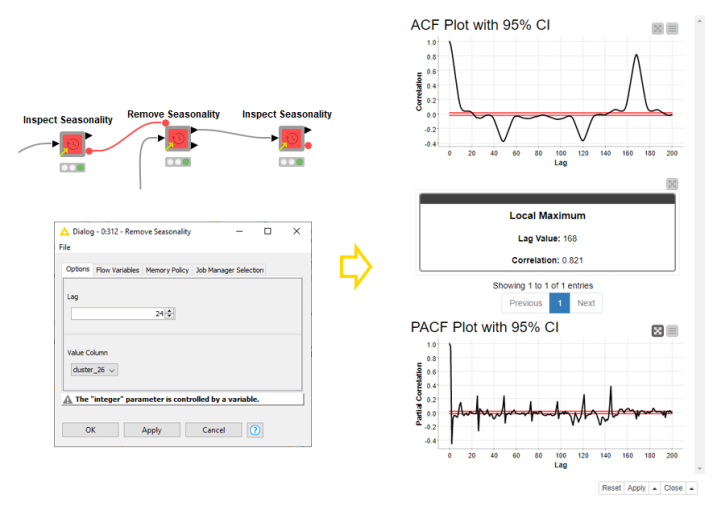


Рисунок 3.7.2

**3.8 Дифференцирование для удаления трендов**

Общее направление, которого «придерживаются» данные временного ряда в течение длительного периода – это т**ренд**.  Когда происходит долгосрочное увеличение или уменьшение данных, тогда тренд существует.

Тренд можно смоделировать с помощью линейной регрессионной модели, но остатки также образуют параболу (рисунок 3.8.1).

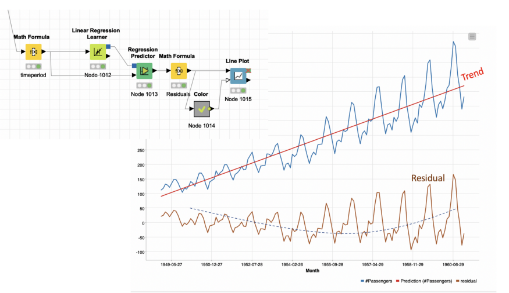


Рисунок 3.8.1 - Исходный временной ряд и остаток после удаления линейного тренда

Возможно удалить тренд с помощью полиномиальной подгонки (рисунок 3.8.2).

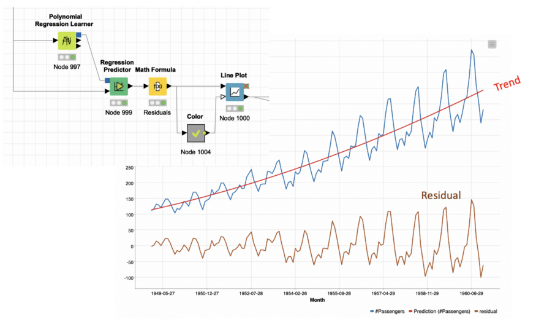


Рисунок 3.8.2 – Исходный временной ряд и остаток после удаления полиномиального тренда

Теперь остатки выглядят более неподвижными. Однако, если есть шаблон в данных, который повторяется через регулярные промежутки времени. Это сезонный эффект, который также должен быть устранен.

**3.9 Классическая декомпозиция временных рядов**

Разложение временного ряда на его тренд, сезонность и остаток – классическая декомпозиция. Она обеспечивает хороший ориентир для прогнозирования. Остаточная, оставшаяся часть временного ряда должна быть [стационарной](https://en.wikipedia.org/wiki/Stationary_process) и может быть предсказана, например, моделью ARIMA.  Но могут потребоваться некоторые дополнительные преобразования, такие как дифференцирование первого порядка или логарифмическое преобразование исходного временного ряда, если остаточный ряд не является стационарным.

Во-первых, если временной ряд показывает направление, тренд, его можно детрендировать, например, при помощи подгонки регрессионной модели по данным или вычисления скользящего среднего значения.

Во - вторых, если временной ряд показывает сезонность (регулярное колебание), тогда временной ряд может быть скорректирован для него. Можно найти лаг, где происходит основная сезонность, на графике автокорреляции временного ряда.

При помощи дифференцирования данных в лаге, где происходит основной всплеск, сезонность может быть скорректирована.  Повторив процедуру для скорректированного (дифференцированного) временного ряда, можно скорректировать вторую сезонность в данных. Когда стационарный временной ряд готов к моделированию, например, моделью ARIMA, можно выполнить окончательную проверку на стационарность, например, [с помощью теста Юнга-бокса](https://en.wikipedia.org/wiki/Ljung%E2%80%93Box_test) (рисунок 3.9.1).

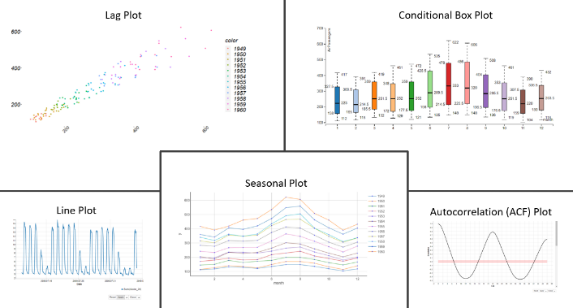


Рисунок 3.9.1

**3.10 Моделирование и оценка временных рядов**

Моделирование остаточной части временного ряда, содержащей его нерегулярную динамику, возможно с помощью моделей ARIMA, моделей машинного обучения, нейронных сетей и многих их вариаций. Этими моделями часто моделируют остаточную часть временного ряда, потому что она стационарна. Но декомпозиция временных рядов необходима не всегда. Некоторые модели, например, сезонная модель ARIMA, также работают для моделирования нестационарных временных рядов.

**3.11 Модели ARIMA**

[Модель ARIMA](https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average) (AutoRegressive Integrated Moving Average) – это модель линейной регрессии между текущими и прошлыми значениями (AR-часть), а также между текущими и прошлыми ошибками прогноза (MA-часть).  Данные дифференцируются, если модель имеет ненулевую I-часть, чтобы сделать ее стационарной. Базовые модели ARIMA предполагают, что временные ряды являются стационарными, а стационарные временные ряды не имеют предсказуемых закономерностей в долгосрочной перспективе. Снижение точности долгосрочных прогнозов можно увидеть в увеличении доверительных интервалов прогнозов.  Для обучения моделей ARIMA наличие большего количества данных не всегда удачно: они могут сделать оценку параметров модели ARIMA трудоемкой, а также преувеличить разницу между истинным процессом и процессом модели.

#### 3.12 Построение моделей ARIMA

#### Когда необходимо построить модель ARIMA, следует определить три параметра, которые определяют, какие временные структуры захватываются моделью. Эти параметры определяют взаимосвязь между текущим значением и запаздывающими значениями (порядок AR), между текущей ошибкой прогноза и запаздывающими ошибками прогноза (порядок MA) и степенью дифференцирования, необходимой для того, чтобы сделать временные ряды стационарными (порядок I). Лучшие параметры AR, I и MA могут быть определены путем изучения (частичных) графиков автокорреляции временных рядов.

#### Если определили порядки AR, I и MA, можно обучить модель ARIMA. Но определение значений параметров не всегда просто. Тогда можно дать максимальные значения для этих параметров, обучить несколько моделей ARIMA и выбрать наилучшую производительность на основе [информационного критерия](https://en.wikipedia.org/wiki/Akaike_information_criterion) AIC или BIC. Это можно сделать с [помощью компонента Auto ARIMA Learner](https://kni.me/w/_ROfECTX-hZ84VW4). Оба компонента выводят модели, их сводную статистику и остатки прогнозирования в выборке (рисунок 3.12.1).

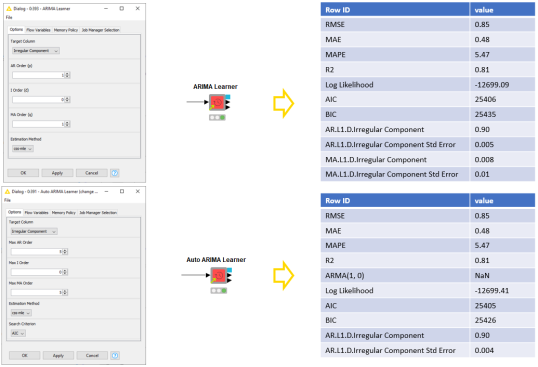


Рисунок 3.12.1

**3.13 Модели машинного обучения**

Модели машинного обучения используют запаздывающие значения в качестве столбцов предиктора и игнорируют временную структуру между целевым столбцом и столбцами предиктора. Они также могут идентифицировать долгосрочные паттерны и поворотные точки в данных, если в обучающих данных представлено достаточно данных для установления этих паттернов. Следует, что чем больше неравномерностей показывают данные, тем больше данных требуется для обучения модели. При применении модели машинного обучения рекомендуется моделировать остаток. Иначе можно построить модель, которая сложнее классической модели декомпозиции, но не изучает ничего нового.

**3.14 Выбор модели**

Некоторые явления трудно прогнозировать, поэтому лучше пойти на более простую модель и не вкладывать ресурсы в моделирование того, что невозможно точно спрогнозировать.

Производительность модели не единственный критерий. Если важные решения основаны на результатах модели, ее интерпретируемость может быть более нужной. Тем не менее, нейронная сеть может проиграть против простой классической модели декомпозиции, хотя она прогнозирует немного лучше.

Повысить точность прогноза может добавление экспликативных (поясняющих) переменных в модель. Но прогнозировать экспликативные переменные в такой модели также необходимо, и возрастающая сложность модели не всегда приводит к лучшей точности.

**3.15 Оценка модели**

Следующим шагом после обучения модели является ее оценка. Набор тестов для прогнозирования в выборке является самим обучающим набором, поэтому процесс модели подгоняется к данным, которые использовались для обучения модели. Для прогнозирования вне выборки тестовый набор следует за обучающим набором во времени.Одной из рекомендуемых метрик ошибок для оценки модели временных рядов является средняя абсолютная процентная ошибка ([MAPE](https://en.wikipedia.org/wiki/Mean_absolute_percentage_error)), т.к. она обеспечивает ошибку в универсальной шкале в процентах от фактического значения. Но эта метрика не определена, если истинное значение равно нулю. Можно использовать и другие метрики ошибок, такие как среднеквадратичная ошибка (RMSE). Использовать [R-квадрат](https://en.wikipedia.org/wiki/Coefficient_of_determination) часто не рекомендуется в виду того, что вместо моделирования всей изменчивости в прошлом основное внимание уделяется прогнозированию будущей систематической изменчивости целевого столбца, метрика R-squared не соответствует контексту анализа временных рядов.

**3.16 Динамическое прогнозирование**

Если есть модель, которая не может обеспечить точные прогнозы в долгосрочной перспективе, динамическое развертывание часто повышает точность прогноза вне выборки. При динамическом развертывании одновременно прогнозируется только одна точка в будущем, и прошлые данные обновляются этим значением прогноза для создания следующего прогноза (рисунок 3.16.1).

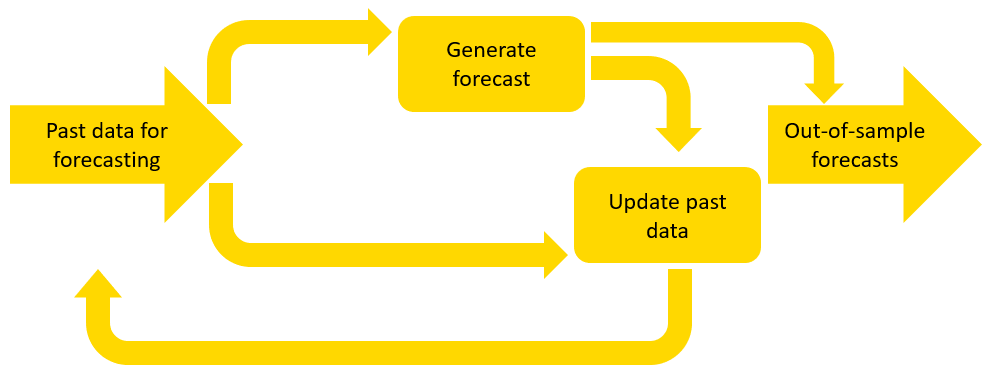
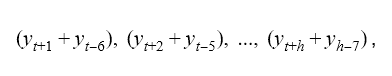


Рисунок 3.16.1 – В динамическом развертывании одновременно генерируется только один прогноз, и этот прогноз добавляется к прошлым данным, которые используются для создания следующего прогноза на один момент времени вперед

**3.17 Восстановление тренда и сезонности**

Следует восстановить тренд и/или сезонность в прогнозах, если происходит разложение временных рядов перед прогнозированием.  Нужно восстановить сигнал, добавляя значения в лаге, где происходит сезонность, если происходит коррекция сезонности путем дифференцирования данных. Для прогнозных значений  восстановление сезонности потребовало бы следующего расчета:

*yt*+1, *yt*+2,..., *yt*+*h*:



где *t* – последняя временная точка в обучающих данных, а *h* – горизонт прогноза.

Чтобы восстановить вторую сезонность, повторяется шаг, описанный выше, для восстановленного временного ряда. Если потребовалось бы восстановить компонент тренда во временном ряду, применили бы регрессионную модель, представляющую тренд, к восстановленному временному ряду.

**ЗАКЛЮЧЕНИЕ**

В результате данного исследования были выполнены следующие задачи:

* собраны и изучены основные сведения о Knime Analytics,
* реализованы различные способы ввода данных и первичный анализ в Knime Analytics Platform,
* проведен элементарный анализ данных с сайта https://hh.ru о вакансии «программист» в Саранске,
* изучены некоторые методы анализа временных рядов, в том числе с использованием Knime Analytics Platform.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Афанасьев В. Н. Анализ временных рядов и прогнозирование: учебник / В. Н. Афанасьев, М. М. Юзбашев. – Москва : Финансы и статистика, 2001. – 231 с. – ISBN 5-279-02419-8. – Текст: электронный.
2. Введение в анализ временных рядов : учебное пособие для вузов / Н. В. Артамонов, Е. А. Ивин, А. Н. Курбацкий, Д. Фантаццини. – Вологда : Изд-во ВолНЦ РАН , 2021. – 135 с. – ISBN 978-5-93299-496-2. – Текст: электронный.
3. Сайт / Time Series Analysis with KNIME — an introduction  
   URL: https://medium.com/mlearning-ai/time-series-analysis-.. – Режим доступа: для авториз. пользователей. – Текст: электронный.
4. Сайт / Обзор Knime Analytics Platform — open source системы для анализа данных URL: https://habr.com/ru/post/320500/ – Режим доступа: для авториз. пользователей. – Текст: электронный.
5. Сайт / Обзор Building a Time series analysis application  
   URL: https://www.knime.com/blog/building-a-time-series-ana.. – Режим доступа: для авториз. пользователей. – Текст: электронный.
6. Сайт / Применение Фреймворка H2O в обработке временных рядов URL https://cyberleninka.ru/article/n/primenenie-freymvor.. – Режим доступа: для авториз. пользователей. – Текст: электронный.