

УДК 004.942, 519.6**Специализированная платформа анализа экономических моделей****Костиков Юрий Александрович**

Кандидат физико-математических наук,
заведующий кафедрой 812,
Московский авиационный институт
(национальный исследовательский университет)
125993, Российская Федерация, Москва, Волоколамское шоссе, 4;
e-mail: jkostikov@mail.ru

Мокряков Алексей Викторович

Кандидат физико-математических наук,
доцент кафедры 813,
Московский авиационный институт
(национальный исследовательский университет)
125993, Российская Федерация, Москва, Волоколамское шоссе, 4;
e-mail: mokryakovav@mati.ru

Романенков Александр Михайлович

Кандидат технических наук,
доцент кафедры 812,
Московский авиационный институт
(национальный исследовательский университет)
125993, Российская Федерация, Москва, Волоколамское шоссе, 4;
e-mail: romanaleks@gmail.com

Аннотация

Оценка эффективности работы экономической модели является одной из актуальных проблем в настоящее время. Часто при упрощении модели рядом параметров можно пренебречь, однако достаточно трудно оценить, насколько адекватной она останется. Для того, чтобы получить данную оценку, возникла необходимость определить, насколько изменилось поведение модели, как поменялись данные и их характер. Найти ответы на поставленные вопросы позволяют сравнительный и статистический анализы. Для исследования этой проблемы была реализована платформа, которая может по некоторым признакам оценить влияние внесенных изменений в модель и сравнить с помощью методов статистического анализа измененную модель с ее исходной версией. Данная система предоставляет пользователю возможность ввода параметров для запуска процесса моделирования, а также осуществляет параллельное выполнение поставленных пользователем задач. Также платформа дает полную свободу пользователю использовать в качестве инструментов аналитики, не считая уже имеющихся в системе, собственноручно разработанные модули. Платформа имеет надежную и вариативную систему хранения

данных. Специально разработанный протокол позволяет сохранять состояние системы во время ее работы, что дает возможность возобновить задачу, если та была не завершена при закрытии окна платформы.

Для цитирования в научных исследованиях

Костиков Ю.А., Мокряков А.В., Романенков А.М. Специализированная платформы анализа экономических моделей // Экономика: вчера, сегодня, завтра. 2019. Том 9. № 5А. С. 218-230.

Ключевые слова

Платформа, математическая модель, экономическая модель, автоматизация, статистический анализ.

Введение

В современном мире существует множество нетривиальных задач, которые могут быть решены с помощью новейших информационно-коммуникационных технологий. Большое количество из этих задач для своего решения требуют моделирования определенного физического, технологического или экономического процесса. Создание реальных условий для моделирования подобных процессов бывает очень сложным, а зачастую даже невозможным. На помощь приходят модели, созданные на основе информационных технологий, которые могут симитировать практически любой процесс, будь то физический, математический процесс или, к примеру, имитация работы небольшого предприятия. Такие модели имеют ряд преимуществ: они относительно «недорогие» в плане разработки; достаточно гибкие для изменений; могут максимально приблизить результаты работы к реальным условиям. Однако разработка такой модели требует очень много трудозатрат и занимает немало времени. Зачастую к уже реализованным моделям необходимо применить оптимизацию, не нарушая принципов ее работы. Подобная оптимизация является очень сложным процессом, а чтобы проверить, осталась ли модель эффективной, необходимо провести много тестов и анализов. Помимо оптимизации, существует еще процесс доработки модели, при котором в исследуемую систему вносят изменения, обусловленные дополнительными параметрами. Этот процесс требует еще больше времени и внимательности при анализе результатов. Перечисленные недостатки определяют общую актуальную проблему при разработке и тестировании исследуемых моделей.

Для решения этой проблемы была поставлена задача разработать программный комплекс, целью которого является упрощение разработки моделей, а также автоматизированное тестирование моделей для оценки их «адекватности». Данную задачу необходимо разделить на несколько подзадач, что позволит упростить и структурировать решение основной задачи [Малышева, Мокряков, Ненахов, 2016].

Первая подзадача – это формирование функционала разрабатываемого программного комплекса. В зависимости от насыщения функционала системы, будут решаться последующие подзадачи, связанные с разработкой архитектуры приложения, удобного пользовательского интерфейса, хранилища данных и т.д. Функционал системы должен позволять создавать проекты с различными версиями исследуемой модели, предоставлять пользователю возможность редактирования входных параметров, использовать несколько способов анализа

исследуемого объекта, а также в автоматизированном режиме выполнять параллельный запуск и анализ нескольких версий модели с различными входными параметрами. К прочему функционалу необходимо отнести возможность предоставления пользователю нескольких форматов представления результатов работы программного комплекса, например в графическом представлении в виде гистограмм, диаграмм и графиков или в общедоступных структурированных форматах XLS, JSON [Bassett, 2015], XML.

Исходя из описанного выше функционала системы, второй подзадачей является создание универсальной, динамической архитектуры программного комплекса [Мокряков, Ненахов, Ненахова, 2017]. Каркас, созданный на основе этой архитектуры, будет исполнять роль центральной оси для дальнейшей разработки платформы. Архитектура должна быть динамически расширяемой для поддержания возможности использования нескольких вариантов анализа и тестирования данных. Соответственно, каждый объект в системе должен быть независим от других. Более того, все элементы системы должны быть заменяемыми в случае необходимости расширения функционала. Так как платформа подразумевает несколько различных этапов выполнения, то архитектура данного приложения должна иметь свойство сохранения состояния, что позволит сохранять результаты работы программы на различных этапах ее исполнения.

Еще одной важной подзадачей является определение математического аппарата системы, который будет выполнять анализ полученных результатов модели [Малышева, Мокряков, Ненахов, 2016]. Математический аппарат должен быть достаточно простым в реализации и устойчивым к изменениям в системе анализа. Но самым главным требованием к математическому аппарату является его эффективность. Результат работы математического аппарата должен демонстрировать результат выполнения исследуемой модели. Для решения поставленной подзадачи наиболее эффективно подходят методы математической статистики, которые позволяют определить значимость, а также зависимость используемых параметров в модели. В частности, необходимо рассмотреть регрессионный и корреляционный анализы для возможности их реализации в математическом аппарате.

Не менее важной подзадачей является создание удобного пользовательского интерфейса, который предоставит пользователю очевидный алгоритм действий для запуска тестирования и анализа различных версий модели в платформе. Интерфейс должен быть интуитивно понятен, а также удовлетворять современным требованиям дизайна приложений.

Так как система предусматривает поддержку нескольких версий исследуемых объектов, то необходимо выделить еще одну подзадачу, которая формулируется как определение взаимодействия платформы с хранилищем данных. Хранилище данных должно эффективно хранить результаты работы приложения, а также различную метаинформацию, которая необходима для корректной работы платформы.

Математический аппарат

На основе описанного функционала разрабатываемой системы и поставленной задачи необходимо определить математический аппарат, который будет являться ядром платформы. Выбор математического аппарата очень важен для программного комплекса, так как именно эффективность математических методов будет определять общую эффективность всей платформы при анализе результатов ее работы. Учитывая то, что анализ данных будет осуществляться в отдельном модуле системы, необходимо выбрать математический аппарат,

достаточно простой в реализации и не требующий особых вычислительных мощностей ЭВМ. Исходя из функционала системы, анализ полученных данных будет сводиться к задаче определения степени зависимости между различными переменными, присутствующими в исследуемом объекте и сравнении полученных результатов с эталоном для оценки критичности их отклонения. Подобные задачи обычно решаются с помощью математической статистики. В современной математической статистике существует множество методов, которые позволяют дать оценку степени связи между двумя переменными.

Одним из самых популярных и известных методов математической статистики является метод множественной регрессии. Задачей множественной линейной регрессии является построение линейной модели связи между набором непрерывных предикторов и непрерывной зависимой переменной [Дрейпер, Смит, 2007; Наследов, 2013]. Помимо метода множественной регрессии, существует еще один метод математической статистики, который может решать рассматриваемые задачи, – корреляционный анализ. Целью корреляционного анализа является выявление оценки силы связи между случайными величинами (признаками), которые характеризует некоторый реальный процесс.

Данные методы математической статистики отлично подходят для реализации задачи анализа данных в разрабатываемом программном комплексе. Однако использование этих методов по отдельности не несет большой эффективности при тестировании реальных задач. Рассмотренные выше методы необходимо скомбинировать и провести анализ полученных результатов от каждого из них методом сравнительного анализа. Сравнительный анализ на основании данных, полученных из корреляционного анализа и множественной регрессии, сможет дать более точную оценку степени зависимости исследуемых параметров модели. На основании этих результатов сравнительный анализ сможет дать оценку адекватности полученных данных путем сравнения значений с выбранным эталоном. При сравнении значений будут учитываться степень взаимосвязи, а также характер взаимосвязи различных компонентов. По каждому компоненту будет получена информация о его значимости в модели и о том, насколько он адекватен по отношению к соответствующему эталону.

Основной функционал платформы

Платформа для автоматизированного тестирования и анализа математических и имитационных моделей включает в себя достаточно богатый функционал, который обеспечивает эффективную работу всего программного комплекса. Концепция данной работы была реализована в отдельной работе [Малышева, Мокряков, Ненахов, 2016].

Перед запуском приложения необходимо прежде всего настроить конфигурационный файл платформы. В частности, конфигурационный файл представляет собой файл формата JSON. JSON – это простой формат обмена данными, удобный для чтения и написания как человеком, так и компьютером [Bassett, 2015]. JSON – это текстовый формат, полностью независимый от языка реализации, но он использует соглашения, знакомые программистам C-подобных языков. В структуре JSON определены следующие сущности:

1. Коллекция пар ключ/значение. В разных языках эта концепция реализована как объект, запись, структура, словарь, хэш, именованный список или ассоциативный массив.
2. Упорядоченный список значений. В большинстве языков это реализовано как массив, вектор, список или последовательность.

Эти сущности являются универсальными для многих популярных языков

программирования. Поэтому формат JSON используется во многих современных информационных системах как стандарт обмена данными.

Разрабатываемая платформа оперирует несколькими основными сущностями. Первая сущность – это проект. Проект представляет собой некий объект, который содержит имя, описание и вложенный список версий. Проект – это верхняя часть иерархии в системе платформы. Проект включает в себя объект исследования, то есть саму модель, параметры запуска модели и инструменты анализа результатов выполнения модели. Пользователь может создавать, редактировать и удалять проекты. По логике платформы два проекта – это две разные исследуемые модели, однако пользователь неограничен в создании проектов и может создавать абсолютно любые проекты.

Для поддержания версионности и более удобного восприятия исследуемого объекта в платформе присутствует еще одна сущность – версия. Версия – это атомарный объект программного комплекса, который находится ниже проекта в иерархии системы. Можно сказать, что версия – это составная часть проекта. Версия содержит имя, описание, модель как объект исследования, список параметров для запуска модели, список анализаторов в порядке обработки данных, результаты работы модели, которые необходимы для дальнейшей обработки.

Еще одним типом сущности в системе является анализатор. Анализатор представляет собой инструмент анализа данных, которые платформа получает при запуске модели с определенными параметрами. Анализатор содержит такие свойства, как имя, описание и сам инструмент обработки. Анализатор напрямую связан с версией. У одной версии может быть несколько анализаторов. Причем важен порядок анализаторов, так как данные будут поступать последовательно от одного анализатора к другому. Анализатор представляет собой независимый объект в системе и может быть также отредактирован, удален или загружен. Данная сущность в качестве инструмента анализа использует исполняемый файл, загружаемый в систему. Так как анализатор использует исполняемый файл, то совершенно неважно, на каком языке был написан данный инструмент. Нужно лишь учесть некоторые требования при разработке инструмента. Во-первых, исполняемый файл должен быть формата exe или jar. Во-вторых, исполняемый файл в качестве входного параметра должен принимать файл формата JSON, который имеет определенную структуру. Входной файл будет содержать данные, которые нужно исследовать, а также дополнительные данные для формирования результата работы. В результате своей работы программа должна создавать файл, путь которого указан во входном файле, в формате JSON с определенной структурой, которая включает результаты анализа и другую служебную информацию для работы платформы.

Одной из основных сущностей системы является модель. Модель – это сущность системы, которая содержит объект исследования, то есть математическую или имитационную модель. Модель содержится в версии в единственном экземпляре. Данная сущность, так же как и остальные сущности в платформе, имеет определенные свойства: имя, описание и исполняемый файл модели. Исполняемый файл модели имеет формат exe и так же, как и исполняемый файл анализатора, независим от языка разработки. Для того чтобы платформа была более универсальной, чтобы можно было запускать различного рода модели, необходимо сделать определенную стандартизацию разработки моделей для этой платформы. Данная стандартизация – это не ограничение, а лишь общее правило для создания моделей. В качестве входных данных исполняемый файл модели должен принимать файл с параметрами запуска в формате JSON. Этот файл содержит параметры запуска модели, а также метаинформацию,

которая нужна для создания результирующего файла. Результатом работы программы модели является результирующий файл формата JSON с абсолютным путем, указанным во входном файле модели.

Помимо описанных выше основных сущностей, в платформе присутствуют еще несколько сущностей, которые необходимы для эффективной работы платформы. Эти сущности позволяют платформе поддерживать весь основной функционал, а также придают ей такое важное свойство, как сохранение состояния выполнения.

Как уже упоминалось ранее, программный комплекс представляет из себя систему, которая имеет модульную архитектуру. Это означает, что платформа состоит из нескольких относительно независимых между собой модулей, связанных между собой специально разработанным протоколом. Одним из таких модулей является обозреватель проектов. Обозреватель проектов позволяет пользователю просматривать детальную информацию о проектах и версиях модели. Также модуль организует удобный графический интерфейс в виде древовидной иерархии проектов и версий, а также отдельную форму для отображения конкретно выбранного объекта.

Таким образом, можно посмотреть детальную информацию по версии одного из проектов, например имя, описание, параметры модели и список анализаторов в порядке обработки данных. Также обозреватель позволяет создавать новые проекты и новые версии проектов. При создании нового проекта или версии данный модуль создает необходимые файлы в файловой системе пользователя или, если указана база данных, создаются записи в заданных коллекциях.

При создании проектов и версии необходимо указать обязательные параметры этих объектов, иначе объекты не будут созданы. Проект в системе может существовать без версий, однако версия проекта не может существовать без модели исследуемого объекта. Также при создании версии пользователю предлагается ввести входные параметры для модели.

Чтобы упростить ввод параметров для модели, была реализована следующая схема. Указывается имя параметра, его шаблонное значение, затем указывается диапазон слева и справа от шаблона значения. Диапазон значения означает максимальный и минимальный порог изменения шаблонного значения. Затем указывается шаг – значение, на которое изменяется изначальное значение для достижения своего минимального и максимального порога. Существуют модели, которым для работы необходимо очень большое число параметров, поэтому в модуле предусмотрена массовая загрузка данных из разных форматов файлов. Так, модуль позволяет загружать параметры из файлов с форматом JSON, XML, CSV. Также в модуле предусмотрена возможность редактирования уже введенных параметров, а также их удаления.

Помимо параметров, нужно выбрать исполняемый файл самой модели, которая будет исследоваться, а также указать ее имя и описание, чтобы пользователь в дальнейшем мог определить, что за модель тестируется. Также при выборе модели указывается еще один важный параметр, который называется эталон. Модель-эталон – это тот объект, с которым сравниваются все последующие версии проекта. Именно эталон задает степень адекватности системы.

Наконец, обозреватель проектов также дает возможность выбрать из списка уже существующих в системе анализаторов те, которые будут анализировать объект исследования. Каждый анализатор можно выбрать и посмотреть его описание, которое содержит принцип его работы. Если пользователь понимает, что данный анализатор его устраивает, то он может добавить его к версии. Так можно добавлять различное количество анализаторов. При добавлении анализаторов создается специальный список. В этом списке очень важен порядок

анализаторов, так как именно в этом порядке будут поступать последовательно данные от одного анализатора к другому. В связи с этим обозреватель проектов имеет функционал, который позволяет менять местами анализаторы при редактировании и создании версий. Редактирование проектов и версий предоставляется тем же функционалом, что и при создании этих сущностей. Обозреватель проектов также имеет возможность запускать на тестирование и анализ несколько версий проектов.

Подобным образом в платформе реализован и модуль под названием «обозреватель анализаторов». Обозреватель анализаторов предоставляет функционал для создания, удаления и редактирования анализаторов в системе. Процесс создания представляет собой загрузку исполняемого файла в платформу, а также задания ему имени и описания. При открытии модуля пользователю будет представлен список уже загруженных в платформу анализаторов. Данный список можно редактировать. Выбрав любой анализатор, можно изменить его имя, описание и исполняемый файл. Также можно удалять анализаторы. При создании каждого анализатора платформа создает специальные файлы в локальной файловой системе или в базе данных, если таковая указана.

По постановке задачи платформа должна исследовать модель, запущенную с различными значениями параметров. Соответственно, платформа должна уметь запускать несколько процессов моделирования с различными входными данными и обрабатывать полученные результаты. В качестве реализации подобного функционала был создан модуль исполняющего сервиса. Данный модуль способен запускать параллельно несколько процессов с различными входными данными. В частности, модуль запускает несколько экземпляров исследуемой модели равной числу сочетаний ее входных параметров. После завершения работы каждого экземпляра модели модуль в автоматизированном режиме собирает результаты их работы и делает предварительную их обработку для передачи следующему модулю. Данный модуль использует несколько парадигм параллельной обработки данных. Так как форматы и структуры входных и выходных файлов модели стандартизированы, то модуль универсально подходит для работы с любыми моделями, которые реализованы для данной платформы.

Следующий модуль, куда передаются данные от исполняющего сервиса, называется «аналитика и обработка». Данный модуль принимает в качестве входного файла специально подготовленный JSON, полученный от исполняющего сервиса. Модуль подготавливает список анализаторов, который должен пройти этот файл. Далее модуль последовательно запускает несколько анализаторов, которые были указаны в тестируемой версии. Входными данными следующего анализатора являются выходные данные предыдущего. Таким образом, модуль передает данные по цепочке от одного анализатора к другому. В конечном итоге будет получен файл, который предварительно будет обработан и передан следующему модулю. Основная суть этого модуля – поддержание порядка выполнения анализаторов и поддержка целостности обработки. В том модуле происходит основная аналитика и обработка данных, на основании которых можно будет дать оценку адекватности модели. Разработка анализаторов для этого модуля – задача непростая, ведь для решения подобных задач используется достаточно мощный математический аппарат.

После аналитики подготовленные данные отдаются на обработку в модуль обработки результатов. В этом модуле происходит обработка и разбор входного файла. Данные из входного файла являются источником при проведении сравнительного анализа версии рассматриваемого проекта с эталоном проекта. В сравнительном анализе происходит сравнение значений отклонений зависимых переменных эталона и тестируемой версии. На основе

полученных данных можно сделать вывод о критичности внесенных изменений в модель данной версии по сравнению с эталоном.

Сравнительный анализ модуля обработки результатов отображается в виде гистограммы. Такая форма является наиболее удобной для зрительной оценки сравнения результатов. Также модуль предоставляет возможности графического отображения результатов корреляционного и регрессионного анализов. Данные представляются в виде графиков функций, которые представляют собой зависимости параметров модели друг от друга, а также функций зависимости зависимой и независимой переменной. На основе выводимой информации можно дать оценку степени зависимости переменных и насколько сильно они влияют на конечный результат. Оценка может быть негативной, положительной, нейтральной и критичной.

Кроме того, модуль предоставляет возможность просмотреть полный отчет о результатах работы тестируемой модели, а также о результатах работы каждого анализатора на определенном этапе тестирования. Отчет можно выгрузить в удобном для пользователя формате, например в XLS. Как только данные появляются в модуле обработки результатов, они сохраняются в локальном хранилище данных или в базе данных в зависимости от настроек системы. Это делается, чтобы пользователь мог посмотреть результаты работы системы в любой момент времени и сравнить их с различными эталонами. Подобная схема работы делает систему более эффективной и удобной для пользователя.

Несложно заметить, что при запуске тестирования и аналитики определенной версии проекта модель проходит несколько этапов обработки данных:

- подготовка модели к запуску;
- формирование файла с параметрами запуска экземпляров модели;
- параллельный запуск нескольких экземпляров модели;
- сбор результатов выполнения экземпляров модели и предварительная их обработка;
- подготовка и запуск анализаторов;
- обработка результатов работы анализаторов и отображение конечного результата.

Каждый этап занимает определенное время на выполнение, особенно этапы запуска моделей и анализаторов. Существует вероятность того, что на определенном этапе что-то пойдет не так и этап завершится неудачей. В этом случае пользователю было бы удобно следить за прогрессом выполнения тестирования и аналитики. Для решения этой задачи была разработана система мониторинга за выполняемыми задачами. После запуска определенной версии данный проект и его версия включаются в список мониторинга. После этого уже на вкладке мониторинга можно будет выбрать наблюдаемый проект и его версии и следить за прогрессом выполнения задачи.

Как уже говорилось ранее, система является модульной, динамически расширяемой. Модули в системе относительно независимы друг от друга. Для того чтобы обеспечить передачу данных между этими модулями, а также обеспечить поддержку динамической расширяемости системы, был разработан специальный протокол взаимодействия между модулями. Этот протокол обеспечивает не только надежную передачу данных между модулями, но и сохранение состояния системы на определенных этапах ее выполнения. Это сделано, чтобы в случае непредвиденного завершения работы платформы во время выполнения тестирования и анализа прогресс был сохранен и при последующем запуске платформы система продолжила выполнять свои задачи с тех этапов, на которых она остановилась. В качестве атомарных объектов передачи данных используются файлы формата JSON, а также сериализованные объекты для сохранения состояния выполнения этапов. Протокол сохраняет в сериализованных файлах

специальную метаинформацию и промежуточные результаты для каждого отдельного этапа. Если на момент запуска платформы в системе присутствуют файлы-протоколы, которые свидетельствуют о незавершенности тестирования и анализа некоторых задач, то пользователю будет предложено возобновить приостановленные задачи. Если пользователь откажется, то данные будут удалены, а история запуска будет очищена. Помимо поддержки сохранения состояния, протокол обеспечивает целостность передачи данных между модулями, проверяя файлы формата JSON на целостность структуры. Если структура будет нарушена, то платформа сообщит об этом пользователю и завершит задачу тестирования.

Рассматриваемая архитектура позволяет использовать систему не только для тестирования и анализа математических и имитационных моделей, но и для задач потоковой обработки данных. Такие задачи можно решить, к примеру, с помощью наращивания цепочки анализаторов, с помощью которых будут обрабатываться данные последовательно от одного исполняемого файла к другому. Передачу данных и их целостность обеспечит протокол, который способен хранить состояние системы. Еще одним плюсом этой платформы является ее простая и открытая архитектура. Данную архитектуру могут дополнить своими идеями и другие программисты для более узких и специальных задач.

Тестирование платформы на реальных данных

Теперь, когда имеется представление о работе данной платформы, рассмотрим подробнее конкретный пример. В качестве примера работы с Системой мы рассмотрели эконометрическое моделирование стоимости квартир в г. Москве (район Замоскворечье). Данные представлены агентством недвижимости «Миэль» и базой данных по аренде и продаже жилой, коммерческой и загородной недвижимости «Realty.dmir.ru». Как уже было описано ранее, данная платформа дает возможность оценки не только четко заданной математической модели, но и модели с данными, имеющими корреляционную зависимость вместо прямой математической зависимости.

Для проведения эконометрического моделирования рынка квартир на основе предложений о продаже квартир была построена выборка, содержащая 75 наблюдений (таблица 1).

Оценка цены 1 м² квартиры производится по следующим факторам: общая площадь квартиры; тип дома (монолитный/панельный); наличие балкона; площадь кухни; тип жилья (новостройка/вторичное жилье).

Помимо общего отчета, пользователь может посмотреть интересующие диаграммы рассеяния, выбрав абсциссу (независимую переменную) и ординату (зависимую переменную). Соответствующий коэффициент корреляции можно посмотреть в отчете по названию, отображенном в легенде диаграммы рассеяния.

Таблица 1 - Тестовые данные по квартирам

| № наблюдения | Стоимость 1 м ² квартиры | Площадь квартиры | Тип дома | Наличие балкона | Площадь кухни | Тип жилья |
|--------------|-------------------------------------|------------------|----------|-----------------|---------------|-----------|
| 1 | 360 000 | 80 | 0 | 0 | 25,00 | 0 |
| 2 | 388 015 | 110 | 0 | 1 | 23,00 | 0 |
| 3 | 328 393 | 127 | 0 | 0 | 30,00 | 0 |
| 4 | 319 000 | 135 | 0 | 1 | 20,00 | 0 |
| 5 | 343 600 | 76 | 0 | 0 | 16,00 | 0 |
| 6 | 360 000 | 75 | 0 | 1 | 16,00 | 0 |

| № наблюдения | Стоимость 1 м ² квартиры | Площадь квартиры | Тип дома | Наличие балкона | Площадь кухни | Тип жилья |
|--------------|-------------------------------------|------------------|----------|-----------------|---------------|-----------|
| ... | ... | ... | ... | ... | ... | ... |
| 65 | 208 600 | 49 | 1 | 0 | 10,00 | 1 |
| 66 | 307 850 | 75 | 1 | 1 | 13,00 | 1 |
| 67 | 263 600 | 55 | 1 | 0 | 6,50 | 1 |
| 68 | 307 260 | 51 | 0 | 1 | 10,00 | 0 |
| 69 | 264 600 | 108 | 0 | 0 | 15,00 | 0 |
| 70 | 255 430 | 46 | 1 | 1 | 12,00 | 1 |
| 71 | 294 290 | 53 | 1 | 0 | 15,00 | 0 |
| 72 | 327 800 | 61 | 0 | 0 | 9,00 | 1 |
| 73 | 333 600 | 74 | 0 | 0 | 15,00 | 1 |
| 74 | 200 200 | 90 | 1 | 1 | 9,00 | 0 |
| 75 | 495 640 | 78 | 0 | 1 | 30,00 | 0 |

Результаты работы корреляционного и регрессионного анализов для данной версии модели представлены далее в таблицах 2 и 3.

Таблица 2 - Результаты корреляционного анализа

| | Стоимость 1 м ² квартиры | Площадь квартиры | Тип дома | Наличие балкона | Площадь кухни | Тип жилья |
|-------------------------------------|-------------------------------------|------------------|----------|-----------------|---------------|-----------|
| Стоимость 1 м ² квартиры | 1,000 | -0,146 | -0,631 | 0,119 | 0,337 | -0,528 |
| Площадь квартиры | -0,146 | 1,000 | -0,364 | -0,034 | 0,505 | -0,073 |
| Тип дома | -0,631 | -0,364 | 1,000 | 0,505 | -0,338 | 0,468 |
| Наличие балкона | 0,119 | -0,034 | -0,066 | 1,000 | -0,001 | 0,052 |
| Площадь кухни | 0,337 | 0,505 | -0,338 | -0,001 | 1,000 | -0,421 |
| Тип жилья | -0,528 | -0,073 | 0,468 | 0,052 | -0,421 | 1,000 |

Таблица 3 - Результаты регрессионного анализа

| Шаг 1 | Коэффициенты | Шаг 2 | Коэффициенты |
|-----------------|--------------|---------------|--------------|
| Константа | 399882,01 | Константа | 406194,23 |
| Общая площадь | -1231,89 | Общая площадь | -1246,79 |
| Тип дома | -95966,92 | Тип дома | -97336,47 |
| Наличие балкона | 9115,73 | Площадь кухни | 3406,99 |
| Площадь кухни | 3369,28 | Тип жилья | -14996,45 |
| Тип жилья | -16243,08 | | |

Из результатов корреляционного анализа видно, что корреляционная зависимость стоимости 1 м² и фактора «Наличие балкона» слабая (наличие балкона в небольшой степени влияет на стоимость 1 м²). Попробуем в новой версии модели не учитывать этот параметр, чтобы получить новые результаты с большей точностью.

Дадим экономическую интерпретацию коэффициентов модели:

1. Коэффициент при факторе «**площадь квартиры**» является статистически значимым.

Жилая площадь квартиры существенно влияет на ее цену.

2. Коэффициент при фиктивной переменной «тип дома» показывает, что при прочих равных условиях цена 1 м² в монолитном доме на 97336,47 руб. выше, чем в панельном доме.
3. Коэффициент при факторе «площадь кухни» является также статистически значимым. При увеличении площади кухни на 1 м² цена квартиры при прочих равных условиях возрастает в среднем на 3406,99 руб.

Заключение

В работе разработана система для автоматизированного тестирования и анализа экономических, математических и имитационных моделей, которая представляет собой универсальную платформу, помогающую быстро и качественно провести оценку адекватности разрабатываемой модели. Платформа должна быть достаточно гибкой, масштабируемой, относительно простой для использования и эффективной для задач, которые поставит пользователь. Система должна быть надежной и при этом поддерживать возможность сохранения данных о предыдущих запусках системы. Результатом работы такой платформы должны стать аналитические результаты, которые позволят легко провести оценку адекватности исследуемого объекта.

В результате решения поставленной задачи был создан программный комплекс, который позволяет использовать моделирующие приложения, разработанные на любом языке программирования как объекты исследования. Данная система предоставляет пользователю возможность ввода параметров для запуска процесса моделирования, а также осуществляет параллельное выполнение поставленных пользователем задач. Также платформа дает полную свободу пользователю использовать в качестве инструментов аналитики, не считая уже имеющихся в системе, собственноручно разработанные модули. Исполняемые объекты моделей и анализаторов должны лишь удовлетворять достаточно простым требованиям системы. Платформа имеет надежную и вариативную систему хранения данных. Специально разработанный протокол позволяет сохранять состояние системы во время ее работы, что дает возможность возобновить задачу, если та была не завершена при закрытии окна платформы.

Библиография

1. Bassett L. Introduction to JavaScript Object Notation: A To-the-Point Guide to JSON. O'Reilly, 2015.
2. Дрейпер Н., Смит Г. Прикладной регрессионный анализ. М.: Вильямс, 2007. 912 с.
3. Костиков Ю.А., Павлов В.Ю., Романенков А.М., Терновсков В.Б. Адаптивная архитектура программно-аппаратного комплекса хранения и обработки данных // Экономика: вчера, сегодня, завтра. 2017. Том 7. № 9А. С. 192-207.
4. Костиков Ю.А., Павлов В.Ю., Романенков А.М., Терновсков В.Б. Модель графического представления информации в программном комплексе обработки экспериментальных данных // Современная наука: актуальные проблемы теории и практики. Серия «Естественные и технические науки». 2018. № 1.
5. Костиков Ю.А., Павлов В.Ю., Романенков А.М., Терновсков В.Б. Модуль графического представления информации в программном комплексе обработки экспериментальных данных // Экономика: вчера, сегодня, завтра. 2017. Том 7. № 10А. С. 118-125.
6. Мальшева М.С., Мокряков А.В., Ненахов Е.В. Концепция платформы для анализа математических и имитационных моделей // XLII Международная молодежная научная конференция «Гагаринские чтения – 2016». М.: МАИ (НИУ), 2016. С. 224-225.
7. Мальшева М.С., Мокряков А.В., Ненахов Е.В. Методы статистического анализа значимости внесенных изменений в математическую модель // XLIII Международная молодежная научная конференция «Гагаринские чтения – 2016». М.: МАИ (НИУ). 2016. С. 225-226.

8. Мокряков А.В., Ненахов Е.В., Ненахова М.С. Платформа для автоматизированного анализа математических и имитационных моделей // XLIII Международная молодежная научная конференция «Гагаринские чтения – 2017». М.: МАИ (НИУ), 2017. С. 781-782.
9. Наследов А. IBM SPSS Statistics 20 и AMOS: профессиональный статистический анализ данных. СПб.: Питер, 2013.
10. Уорсли Дж., Дрейк Дж. PostgreSQL. Для профессионалов. СПб.: Питер, 2003.

Specialized platform for analysis of economic models

Yurii A. Kostikov

PhD in Physical and Mathematical Sciences,
Head of the Department 812,
Moscow Aviation Institute (National Research University),
125993, 4 Volokolamskoe highway, Moscow, Russian Federation;
e-mail: jkostikov@mail.ru

Aleksei V. Mokryakov

PhD in Physical and Mathematical Sciences,
Head of the Department 813,
Moscow Aviation Institute (National Research University),
125993, 4 Volokolamskoe highway, Moscow, Russian Federation;
e-mail: mokryakovav@matl.ru

Aleksandr M. Romanenkov

PhD in Technical Sciences,
Associate Professor of the Department 812,
Moscow Aviation Institute (National Research University),
125993, 4 Volokolamskoe highway, Moscow, Russian Federation;
e-mail: romanaleks@gmail.com

Abstract

Evaluation of the effectiveness of an economic model is one of the urgent problems. Often, while simplifying a model, a number of parameters can be neglected, but it is rather difficult to estimate how adequate a model will remain. In order to obtain this estimate, it is necessary to determine how the behavior of the model has changed, how the data and their character have changed. Comparative and statistical analyzes help to find answers to the questions. To investigate this problem, a platform was implemented, which can, by some attributes, evaluate the impact of the changes made to the model and compare the modified model with its original version using statistical analysis methods. This system provides the user with the opportunity to enter parameters to start the modeling process, and performs parallel execution of the tasks set by the user. In addition, the platform gives the user complete freedom to use handcrafted models as analytics tools, apart from the modules existing in the system. The platform has a reliable and variable data storage system.

A specially designed protocol allows you to save the state of the system during its operation, which makes it possible to restore the task if it was not completed when the platform window was closed.

For citation

Kostikov Yu.A., Mokryakov A.V., Romanenkov A.M. (2019) Spetsializirovannaya platformy analiza ekonomicheskikh modelei [Specialized platform for analysis of economic models]. *Ekonomika: vchera, segodnya, zavtra* [Economics: Yesterday, Today and Tomorrow], 9 (5A), pp. 218-230.

Keywords

Platform, mathematical model, economic model, automatization, statistical analysis.

References

1. Bassett L. (2015) Introduction to JavaScript Object Notation: A To-the-Point Guide to JSON. O'Reilly.
2. Dreiper N., Smit G. (2007) Prikladnoi regressionnyi analiz [Applied regression analysis]. Moscow: Vil'yams Publ.
3. Kostikov Yu.A., Pavlov V.Yu., Romanenkov A.M., Ternovskov V.B. (2018) Mo-del' graficheskogo predstavleniya informatsii v programmnom komplekse obrabotki eksperimental'nykh dannykh [Model of graphical representation of information in the software package for processing experimental data]. *Sovremennaya nauka: aktual'nye problemy teorii i praktiki. Seriya "Estestvennye i tekhnicheskie nauki"* [Modern Science: Actual Problems of Theory and Practice. Series "Natural and Technical Science"], 1.
4. Kostikov Yu.A., Pavlov V.Yu., Romanenkov A.M., Ternovskov V.B. (2017) Mo-dul' graficheskogo predstavleniya informatsii v programmnom komplekse obrabotki eksperimental'nykh dannykh [Module of graphical representation of information in the software package of experimental data processing]. *Ekonomika: vchera, segodnya, zavtra* [Economy: Yesterday, Today, Tomorrow], 7 (10A), pp. 118-125.
5. Kostikov Yu.A., Pavlov V.Yu., Romanenkov A.M., Ternovskov V.B. (2017) Adaptivnaya arkhitektura programmno-apparatnogo kompleksa khraneniya i obrabotki dannykh [Adaptive architecture of software and hardware complex for data storage and processing]. *Ekonomika: vchera, segodnya, zavtra* [Economy: yesterday, today, tomorrow], 7 (9A), pp. 192-207.
6. Malysheva M.S., Mokryakov A.V., Nenakhov E.V. (2016) Kontsepsiya platformy dlya analiza matematicheskikh i imitatsionnykh modelei [The concept of a platform for analyzing mathematical and simulation models]. XLII Mezhduna-rodnyaya molodezhnaya nauchnaya konferentsiya "Gagarinskie chteniya – 2016" [Proc. Int. Conf. "Gagarin Readings - 2016"]. Moscow: Moscow Aviation Institute (National Research University), pp. 224-225.
7. Malysheva M.S., Mokryakov A.V., Nenakhov E.V. (2016) Metody statisticheskogo analiza znachimosti vnesennykh izmenenii v matematicheskuyu model' [Methods of statistical analysis of the significance of the changes made in the mathematical model]. XLIII Mezhdunarodnaya molodezhnaya nauchnaya konferentsiya "Gagarinskie chteniya – 2016" [Proc. Int. Conf. "Gagarin Readings – 2016"]. Moscow: Moscow Aviation Institute (National Research University), pp. 225-226.
8. Mokryakov A.V., Nenakhov E.V., Nenakhova M.S. (2017) Platforma dlya avtoma-tizirovannogo analiza matematicheskikh i imitatsionnykh modelei [Platform for automated analysis of mathematical and simulation models]. XLIII Mezhdunarodnaya molodezhnaya nauchnaya konferentsiya "Gagarinskie chteniya – 2017" [XLIII International Youth Scientific Conference "Gagarin Readings - 2017"]. Moscow: Moscow Aviation Institute (National Research University), pp. 781-782.
9. Nasledov A. (2013) IBM SPSS Statistics 20 i AMOS: professional'nyi statisticheskii analiz dannykh [IBM SPSS Statistics 20 and AMOS: Professional Statistical Data Analysis]. Saint Petersburg: Piter Publ.
10. Uorsli Dzh., Dreik Dzh. (2003) PostgreSQL. Dlya professionalov [PostgreSQL. For professionals]. Saint Petersburg: Piter Publ.