

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

**АЛГОРИТМ И ПРОГРАММА ДЛЯ ПРОГНОЗА ПОТРЕБЛЕНИЯ
ГАЗА НА ОСНОВЕ ЗАВИСИМОСТИ ОТ СКОРОСТИ ИЗМЕНЕНИЯ
ТЕМПЕРАТУРЫ ВОЗДУХА**
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы
направления 02.03.03 Математическое обеспечение и администрирование
информационных систем
факультета компьютерных наук и информационных технологий
Аванесяна Армена Самвеловича

Научный руководитель:
проф. кафедры информатики
и программирования, д.т.н. _____ Фалькович А.С.

подпись, дата

Зав. кафедрой:
канд. ф.-м. наук _____ Огнева М.В.

подпись, дата

Саратов 2018

ВВЕДЕНИЕ

Актуальность темы. Успешность работы организаций и предприятий зависит от многих факторов. Проанализировав все факторы, можно спрогнозировать развитие событий, вероятность возможного успеха и выстроить успешную стратегию ведения дел. В свою очередь, основываясь на результатах прогнозирования, открываются и закрываются предприятия, заключаются сделки, формируются векторы движения компаний. Таким образом в современном мире прогнозирование является неотъемлемой составляющей успешной стратегии ведения дел. Прогнозирование лежит в основе многих отраслей экономики и обуславливает актуальность изучения вопросов связанных с ним.

Цель бакалаврской работы. Цель работы заключается в проведении анализа результатов, полученных при различных уравнениях регрессионной зависимости, различных интервалах осреднения и методах осреднения температуры, и выбор самого оптимального из них. Задача, которую необходимо решить для достижения поставленной цели, состоит в разработке программы, осуществляющей прогнозирование потребления газа с учетом изменения температуры.

Поставленная цель определила **следующие задачи:**

1. изучить методы прогноза потребления газа
2. изучить основы регрессионного анализа
3. сформировать уравнения регрессионной зависимости
4. решить систему уравнений и получить регрессионные параметры
5. разработать алгоритм расчета параметров регрессионной зависимости потребления газа с учетом изменения температуры
6. разработать на языке Java программную реализацию алгоритма формирования прогноза с учетом изменения температуры для различных интервалов осреднения по потреблению газа и температуры;
7. разработать графическое приложение на языке Java, которое на основе программной реализации алгоритма отображает для каждого из способов

прогнозирования его средне медианное отклонение от действительных величин

8. провести анализ полученных результатов исследования и составить заключение о самом оптимальном способе прогнозирования

Методологические основы алгоритма и программа для прогноза потребления газа на основе зависимости от скорости изменения температуры воздуха представлены в работах А. Аффифи, С. Эйзен [1], Кремер Н.Ш., Путко Б.А. [2], Елисеева И.И., Курышева С.В., Костеева Т.В [3], Гмурман В.Е [5], Герберт Шилдт [11], Brian Goetz, Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes, Doug Lea [14].

Теоретическая значимость бакалаврской работы заключается в разработанном алгоритме прогноза потребления газа на основе регрессионной зависимости, а так же в разработанных уравнениях регрессии, которые позволяют данную зависимость построить.

Практическая значимость бакалаврской работы. В ходе выполнения практической части бакалаврской работы было разработано графическое приложение, которое позволяет рассчитать прогнозы потребления газа на основе различных уравнениях регрессионной зависимости, различных интервалах осреднения и методах осреднения температуры. И позволяет, опираясь на полученные результаты, сделать заключение о самом оптимальном способе прогнозирования.

Структура и объём работы. Бакалаврская работа состоит из введения, 5 глав, заключения, списка использованных источников и приложения. Общий объем работы – 74 страницы, из них 68 страниц – основное содержание, включая 19 рисунков и 1 график, цифровой носитель в качестве приложения, список использованных источников информации – 22 наименования.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первая глава «Регрессионный анализ» посвящена теоретическим основам регрессионного анализа.

В пункте 1.1 «Основные понятия» рассматривается понятие регрессии, ее происхождение. Термин регрессия (лат. regressio – обратное движение, отход) был введен английским ученым Фрэнсисом Гальтоном в 1886 году для выражения обнаруженной им зависимости между ростом родителей и детей. Так термин регрессия появился в математической статистике.[1] В регрессионном анализе рассматривается связь между одной переменной, называемой зависимой, и несколькими другими, называемыми независимыми.[2] Рассказывается о существовании двух видов: парной и множественной. Приводятся формулы, описывающие обе модели. Соответственно $y = f(x)$ (1) и $y = f(x_1, x_2, \dots, x_k)$ (2). [3] Объясняется, что регрессионный анализ предназначен для организации сбора, систематизации и обработки статистических данных с целью их удобного представления, интерпретации и получения научных и практических выводов.[4][5]

В пункте 1.2 «Линейная регрессия» приводится уравнение, описывающее модель линейной регрессии: $y = a + bx + \varepsilon$ (4)

где y – фактическое значение результирующего признака;

$a + bx$ – теоретическое значение результирующего признака;

a, b – искомые регрессионные параметры;

ε – случайная величина, характеризующая отклонения реального значения результирующего признака от теоретического, найденного по уравнению регрессии.[6]

В пункте 1.3 «Метод наименьших квадратов» описан классический подход к решению уравнения линейной регрессии, а именно метод наименьших квадратов (МНК).[7] Здесь приводится последовательность действий, которые позволяют решить уравнение линейной регрессии и найти регрессионные параметры.[8]

Таким образом, в главе 1 рассматривается весь математический аппарат, который требуется для дальнейшего прогнозирования объемов потребления газа на основе регрессионной зависимости.

Вторая глава «Модель прогноза потребления газа на основе регрессионной зависимости» посвящена построению линейных уравнений регрессии для прогноза потребления газа.

В пункте 2.1 «Построение уравнения формирования суточного прогноза на основе температуры воздуха, ее производной и объемах уже поставленного газа» приводится уравнение для прогноза на потребления газа для i -ых суток $V_i = a\bar{t}_i^\circ + b \frac{d\bar{t}_i^\circ}{d\tau} + c$ (11) на основе температуры воздуха, ее производной и объемах уже поставленного газа. Здесь же описываются формулы для нахождения регрессионных параметров a, b, c .

В пункте 2.2 «Построение уравнения формирования суточного прогноза на основе температуры воздуха, ее производной, знаке производной и объемах уже поставленного газа» приводится уравнение для прогноза на потребления газа для i -ых суток $V_i = a\bar{t}_i^\circ + \left(\frac{b+c}{2} + \frac{b-c}{2} \operatorname{sgn} \frac{d\bar{t}_i^\circ}{d\tau} \right) \frac{d\bar{t}_i^\circ}{d\tau} + d$ (19) на основе температуры воздуха, ее производной, знаке производной и объемах уже поставленного газа. Здесь же приводятся подробные математические выкладки для нахождения регрессионных параметров a, b, c и d .

В пункте 2.3 «Алгоритм формирования суточного прогноза» рассказывается о 7 способах нахождения осредненной температуры воздуха за i -ые сутки \bar{t}_i° . Рассмотрим эти периоды:

1. одни предшествующие и текущие сутки
2. одни предшествующие, текущие и одни последующие сутки
3. двое предшествующих суток
4. двое предшествующих и текущие сутки
5. двое предшествующих, текущие и одни последующие сутки
6. трое предшествующих и текущие сутки
7. трое предшествующих, текущие и одни последующие сутки

Так же, для лучшего восприятия приведено графическое представление алгоритма формирования суточного прогноза поставки/потребления газа на двое газовых суток для объектов, на которых потребление газа имеет/не имеет четко выраженную зависимость от дня недели и зависит от температуры воздуха. В качестве примера взят период осреднения температуры – двое предшествующих, текущие и одни последующие сутки.

Третья глава «Применяемые инструменты и технологии программирования» посвящена описанию инструментов и технологий, которые были применены в ходе создания приложения.

В пункте 3.1 «Язык программирования» кратко рассказывается об истории создания языка Java и его основных концепциях. Так же рассказывается об инструментах разработчика, необходимых для создания приложений на данном языке: JDK (англ. Java Development Kit – набор для разработки на языке Java). Данный комплект распространяется бесплатно компанией Oracle Corporation. JDK включает в себя Java компилятор (javac), стандартные библиотеки классов Java, документацию и исполнительную систему JRE (англ. Java Runtime Environment – среда исполнения Java). Виртуальная машина JVM (англ. Java Virtual Machine – среда исполнения Java) входит в состав JRE. [9]

В пункте 3.2 «Среда разработки» рассказывается о среде разработки, которая была применена в ходе разработки приложения, а так же дана информация о распространяемых версиях: JetBrains выпускает IntelliJ IDEA в версиях Community Edition и Ultimate Edition. В свою очередь Ultimate Edition имеет 30 дневный пробный период. Community Edition является полностью бесплатной версией и поддерживает разработку на Java SE (англ. Java Standard Edition – стандартная версия платформы Java). Ultimate Edition предназначена для Java EE (англ. Java Enterprise Edition – набор спецификаций описывающий архитектуру серверной платформы) разработки.[10]

В пункте 3.3 «Многозадачное программирование» рассказывается о понятиях многозадачности процессов и многозадачности потоков. Для

операционной системы процесс представляет собой наименьшую единицу кода, которую диспетчеризует планировщик операционной системы. При многозадачности на основе потоков речь идет об уровне одной программы. Так, многопоточность воспринимается программой как наименьшая единица диспетчеризуемого кода. Из этого следует, что программа может выполнять несколько потоков одновременно. Примером многопоточности может служить браузер, который позволяет одновременно просматривать веб-страницы и скачивать какой-либо файл.

В подпункте 3.3.1 «Модель потоков исполнения Java» дается краткая информация о модели потоков с исполняющей системе Java, а так же о достоинствах данной модели. Один поток может быть приостановлен без остановки других частей программы. Например, время ожидания при чтении данных из сети или вводе пользователем данных может быть выгодно использовано в любом другом месте программы. [11]

В подпункте 3.3.2 «Многопоточная разработка в Java» рассказывается о практических основах многопоточного программирования в Java. В частности о методах `wait` и `notify` базового класса `Object`, позволяющие приостановить и пробудить потоки. [12]. Так же здесь рассказывается об основных методах класса `Thread`, который предоставляет возможность в отдельных потоках исполнять объекты интерфейса `Runnable`. [13]

В подпункте 3.3.3 «Concurrency API. Высокоуровневая работа с потоками» рассказывается об инструментах для высокоуровневой работы с многопоточностью, а именно о пакете библиотек `java.util.concurrent`. [14]

В подпункте 3.3.4 «Интерфейсы – исполнители Concurrency API» рассматриваются интерфейсы `Executor` и `ExecutorService`. В частности описываются методы интерфейса `ExecutorService` для высокоуровневой работы с потоками. Такие как **`submit`**, **`invokeAll`**, **`invokeAny`**, **`shutdown`**, **`shutdownNow`**. Так же приводится пример создания объект типа `ExecutorService` на заданное количество потоков:

```
ExecutorService service = Executors.newFixedThreadPool(3);
```

Создан ExecutorService на 3 потока. [14][15]

В подпункте 3.3.4 «Интерфейсы – задачи возвращающего типа Concurrency API» рассказывается об интерфейсе Callable, позволяющем не только описать блок кода для исполнения в отдельном потоке, но и получить некоторый результат работы данного блока. Для этого используется интерфейс Future, позволяющий впоследствии по завершению работы блока кода получить искомый результат. Так же рассматриваются основные методы интерфейса Future: get и cancel. [14][16]

В пункте 3.4 «Stream API» рассказывается об инструментах Stream API, позволяющих эффективно работать с коллекциями данных в функциональном стиле. Рассказывается о ключевых различиях старых методов работы с коллекциями от новых, в частности о том, что с помощью Stream API можно избежать написания лишнего кода, не имеющего отношения к бизнес логике.[17] Так же описываются некоторые методы для работы с коллекциями: **filter, findFirst, forEach, collect, sorted.**[18]

В пункте 3.5 «Платформа JavaFX» рассказывается о библиотеке инструментов для разработки графических приложений. JavaFX дает возможность разработки приложений, работающих на персональных компьютерах, в браузерах или на мобильных устройствах. [19] Описаны тонкости при работе с JavaFX, в частности, что компоненты JavaFX, согласно документации, обязаны менять свои свойства и состояния из потока JavaFX. В случае, если изменения произойдут из другого потока, следует применять статический метод runLater из класса Platform. Данный метод получает в качестве параметра экземпляр Runnable, который впоследствии будет выполнен в потоке JavaFX.[20] Здесь же рассказывает о приложении SceneBuilder, упрощающее процесс конструирования оконного приложения. Данное графическое приложение позволяет перетаскивать требуемые элементы из набора компонентов на панель. Результат работы сохраняется в виде fxml-файла, который в дальнейшем будет использован программой для построения и добавления элементов в окно приложения.[21][22]

Четвертая глава «Программная реализация» посвящена описанию разработанного приложения.

В пункте 4.1 «Реализация алгоритма» рассказывается о базовом абстрактном классе Equation, в котором описаны все методы для вычислений регрессионных параметров и прогнозов объемов потребления газа. В качестве примера берется алгоритм для прогноза потребления газа, который основывается на температуры воздуха, ее производной и объемах уже поставленного газа. Приводятся скриншоты таких методов как `getMean`, `getDerivativeI`, `getSummationT`, `getSummationDT` и `getSummationT_DT`. Описываются их предназначения и даются пояснения о переменных, с которыми ведется работа. Рассказывается о семи наследниках класса Equation, которые переопределяют метод `getMean`. Именно в данном методе реализуются различные способы нахождения осредненной температуры.

В пункте 4.2 «Реализация графического приложения» рассказывается о том, что при разработке графического приложения был применен шаблон проектирования MVC (англ. Model View Controller – Модель Представление Контроллер). Отмечается, что в программе в роли view выступает оконное приложение. Для взаимодействия графического представления и результатов работы алгоритма используется класс ReportUnit. Каждый объект этого класса содержит информацию о способе исследования и о полученном результате. Поля этого класса соответствуют столбцам в результирующей таблице, которая показывается пользователю по завершению вычислений. Данный класс является моделью. Описывается функционал, в частности, что приложение позволяет пользователю импортировать данные, на их основе проводить вычисления, отображать результаты и экспортировать их в виде *html* – таблицы. Для обработки возникающих событий, как например нажатие кнопки, используется класс Controller. В нем описаны объект, которые будут «слушаться» обработчиками и сами обработчики. А то, к какому объекту относится тот или иной обработчик описывается в *fxml* – файле.

В пункте 4.3 «Реализация многопоточности» рассказывается о том, каким способом достигается вычисление результатов в многопоточном режиме. А именно, что базовый класс Equation реализовывает интерфейс Callable и переопределяет метод *call*. Указанное в данном методе будет выполняться в отдельном потоке, а по завершению будет возвращен результат вычислений. Класс EquationsExecutor с помощью класса Executors создает объект ExecutorService, который будет содержать заданное число потоков. Данным потокам на выполнение отдаются наследники класса Equation с определенными параметрами осреднения. Так же отмечается, что в графическом приложении применяется многопоточность для предотвращения «замерзания» пользовательского интерфейса. Оно может возникать по причине того, что графическое приложения выполняется в одном потоке, и передача в этот поток тяжелых вычислений может повлечь долгую реакцию со стороны приложения, так как все последующие события будут обработаны после завершения текущих тяжелых вычислений. Для решения этой проблемы создается новый фоновый поток, в котором будут выполняться вычисления, а графическое приложение будет оставаться «отзывчивым».

Пятая глава «Анализ полученных данных» содержит анализ результатов работы приложения.

В пункте 5.1 «Искомый результат» рассказывается о том, что спрогнозированные данные сравнивались с фактическими. Для этого были применены два подхода к построению регрессионной зависимости. Первый зависел от осредненной температуры, ее производной и уже потребленных/поставленных объемов газа. А второй, помимо всех зависимостей первого, еще и имел зависимость от знака производной. В результате работы приложения были найдены абсолютные значения разностей прогнозируемого и фактического значения. Данные заносились в таблицу. Которая впоследствии сортировалась по возрастанию с целью нахождения наименьшего медианного расхождения спрогнозированного значения от фактического. Таким образом, делались выводы о самом оптимальном

интервале осреднения при построении регрессионной зависимости для прогнозирования объемов поставки/потребления газа. Исходя из полученных результатов, был сделан вывод, что наилучший результат в ≈ 47.5 единиц отклонения достигается при выборе 7 суток в качестве интервала осреднения при использовании уравнения регрессионной зависимости независимого от знака производной.

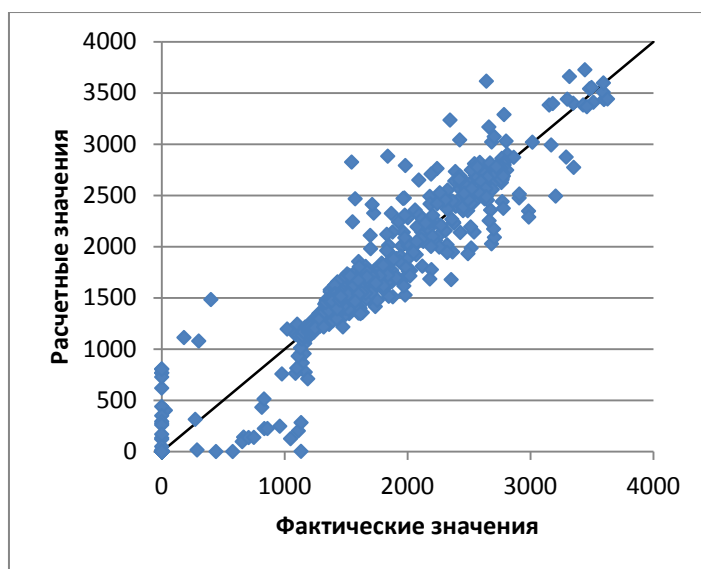


График 1. Сравнение фактических и расчетных значений для первых последующих суток при осреднении температуры за трое суток (двое предшествующих, текущие и последующие) и прогнозе на основе анализа потребления газа за 7 суток

ЗАКЛЮЧЕНИЕ

В результате данной работы была разработана программа с графическим интерфейсом, которая рассматривает различные способы нахождения осредненной температуры и осуществляет прогнозирование на основе регрессионных формул, сравнивает полученные результаты с фактическими и выводит средне медианные отклонения от фактических результатов для каждого из рассмотренных интервалов осреднения.

Программа написана на языке Java в среде IntelliJ IDEA. Исходные данные загружаются из файла с помощью окна импортирования файла с данными из приложения, результаты расчета каждого потока выполнения записываются в таблицу, которая содержит информацию о том какая формула была применена, какой был интервал осреднения, а так же какой был способ нахождения осредненной температуры и средне медианное отклонение. Так же, программа предоставляет возможность выгрузить полученные результаты в формате *html*, где информация так же представлена в виде таблицы. С помощью программы были построены регрессионные зависимости потребления газа от температуры воздуха и на их основе проведены ретроспективные прогнозные расчеты для реальных значений потребления газа и температуры воздуха.

Исходя из результатов работы программы, самой оптимальной формулой для подсчета является регрессионная модель, зависящая от температуры и ее производной. В свою очередь при прогнозировании с помощью этого способа наилучший результат достигается путем вычислений основанных на интервале осреднения в 7 суток и использовании в качестве метода осреднения, метод, основанный на осреднении температуры за двое прошлых, текущие и одни последующие сутки. Таким образом, все задачи, поставленные для достижения цели, решены, и цель достигнута. По тематике бакалаврской работы был представлен доклад «Алгоритм и программа для прогноза потребления газа на основе зависимости от скорости изменения температуры воздуха».

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. А. Аффифи, С. Эйзен, Статистический анализ: Подход с использованием ЭВМ. Пер. с англ. М.: «Мир», 1982, с.141–142
2. Кремер Н.Ш., Путко Б.А., Эконометрика: Учебник для вузов. М.: ЮНИТИ-ДАНА, 2003. с.17–19
3. Елисеева И.И., Курышева С.В., Костеева Т.В. и др., Эконометрика: Учебник. Под ред. И.И.Елисеевой. 2-е изд., перераб. и доп. М.: «Финансы и статистика», 2005. с.51–53
4. Поллард Дж. Справочник по вычислительным методам статистики. Пер. с англ. – М.: Финансы и статистика, 1982. - 344 с.
5. Гмурман В.Е. Теория вероятностей и математическая статистика. - 9-е изд. - М.: Высшая школа, 2003. – 480 с.
6. Дрейпер Н., Смит Г. Прикладной регрессионный анализ. Т. 1. - М.: Финансы и статистика, 1986. - 368 с.
7. Айвазян С.А., Енюков И.С., Мешалкин Л.Д. Прикладная статистика : Исследование зависимостей. - М: Финансы и статистика, 1985. - 488 с.
8. Справочник по прикладной статистике. / Под ред. Ллойда Э., Ледермана У. Т. 1. - М: Финансы и статистика, 1989. - 512 с.
9. Кей С. Хорстманн, Гари Корнелл. Java. Библиотека профессионала, том 1. Основы. 9-е издание ООО «И.Д.Вильямс», 2014. – 864с.
10. IntelliJ IDEA: The Java IDE for Professional Developers by JetBrains. [Электронный ресурс] URL: <https://www.jetbrains.com/idea/> Дата обращения: 07.05.18
11. Герберт Шилдт. Java 8. Полное руководство. 9-е изд.: Пер. с англ. М.: ООО «И.Д.Вильямс», 2015, с. 287–295
12. Java Multithreading. [Электронный ресурс] URL: https://www.tutorialspoint.com/java/java_multithreading.htm Дата обращения: 07.05.18

13. Брюс Эккель. Философия Java. Библиотека программиста. 4-е изд. СПб.: Питре, 2009, с. 557–561
14. Brian Goetz, Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes and Doug Lea. Java Concurrency in Practice. 1st Edition. 2006, Яз. англ. с.112 – 126
15. Многопоточное программирование в Java 8. Часть первая. Параллельное выполнение кода с помощью потоков. [Электронный ресурс] URL: <https://tproger.ru/translations/java8-concurrency-tutorial-1/> Дата обращения: 07.05.18
16. Java Callable and Future Tutorial. [Электронный ресурс] URL: <https://www.callicoder.com/java-callable-and-future-tutorial/> Дата обращения: 10.05.18
17. Stream API. Введение в Stream API. [Электронный ресурс] URL: <https://metanit.com/java/tutorial/10.1.php> Дата обращения: 10.05.18
18. Raoul–Gabriel Urma, Mario Fusco, and Alan Mycroft. Java 8 in Action: Lambdas, streams, and functional-style programming. 1st Edition, 2014, Яз. англ. с. 28–32
19. Cay S. Horstmann. Java SE 8 for the Really Impatient. –Indiana. 2014, Яз. англ. с. 69–97
20. Carl Dea, Mark Heckler, Gerrit Grunwald, Jose Pereda, Sean M Phillips. JavaFX 8 Introduction by Example. Second Edition. Apress. 2014. Яз. англ. –409с
21. Scene Builder – Gluon [Электронный ресурс] URL: <http://gluonhq.com/products/scene-builder/> Дата обращения: 16.05.18
22. Кей С. Хорстманн, Гари Корнелл. Java. Библиотека профессионала, Том 2. Расширенные средства. 9-е издание ООО «И.Д.Вильямс», 2014. – 1008с.