

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

**ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ ПОСТРОЕНИЯ ФОРМУЛ НА ОСНОВЕ
ЛИНЕЙНОГО РЕГРЕССИОННОГО АНАЛИЗА ЭМПИРИЧЕСКИХ
ДАННЫХ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы

направления 02.03.03 Математическое обеспечение и администрирование
информационных систем

факультета компьютерных наук и информационных технологий

Васьковцова Романа Александровича

Научный руководитель:

проф. кафедры ИиП, д. т. н.

Фалькович. А.С.

подпись, дата

Зав. кафедрой:

к.ф.-м.н., доцент

Огнева. М.В.

подпись, дата

Саратов 2021

ВВЕДЕНИЕ

Актуальность работы заключается в часто возникающей необходимости оперативно подобрать формулу, описывающую эмпирические данные. Например, социологические исследования зависимости количества преступлений от индекса Джинни, средних доходов домохозяйств показали высокий уровень корреляции, а значит и взаимосвязь уровня доходов в регионе с совершаемыми преступлениями. Показатель тесной связи позволяет произвести регрессионный анализ данной зависимости для прогнозирования совершаемых преступлений для индивидуума с определенным доходом.

Цель бакалаврской работы - разработка web-приложения для построения и вывода формул функций для эмпирических данных, с возможностью последовательной эксплуатации для проведения вычислений по данным формулам.

Поставленная цель определила **следующие задачи**:

- 1) Изучение методов аппроксимации и построения эмпирических функций
- 2) Реализация алгоритмов регрессионного анализа для эмпирических данных
- 3) Реализация Web-приложения для анализа зависимости предоставляемых данных и вывода аппроксимирующей формулы

Методологические основы построения формул на основе линейного анализа представлены в работах Бернулли, Пуассона, Боровикова.

Практическая значимость бакалаврской работы. Реализация быстрого открытого интернет сервиса (веб-сайта) для моментального анализа зависимости предоставляемых числовых рядов и построения на основе анализа эмпирической или приближенной функции заданного линейного или нелинейного вида.

Структура и объём работы. Бакалаврская работа состоит из введения, 4 разделов, заключения, списка использованных источников и 54 приложений. Общий объем работы – 135 страниц, из них 51 страница – основное содержание, включая 14 рисунков и 1 таблицу, цифровой носитель в качестве приложения, список использованных источников информации – 20 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Основные определения» посвящен основным определениям и терминам, которые используются в выпускной квалификационной работе.

Второй раздел «Сравнение интерполяции сплайнами и регрессионного алгоритма построения функции» посвящен алгоритмам аппроксимации и построения функции по заданным эмпирическим данным, их сравнению, выделению сильных и слабых сторон предоставленных алгоритмов, а также выбору одного из них на последующую реализацию в проекте. При рассмотрении двух алгоритмов было выявлено, что фиксированная заданная форма сплайн интерполяции слишком велика, а при линейной регрессии вычисления коэффициентов предоставляют простой и удобный для использования формат формулы, который можно выбрать и для нелинейного представления функции самостоятельно из степенных, экспоненциальных или логарифмических функций. В связи с достоинствами скорости вычисления, возможности прогнозирования за пределом

эмпирических данных, а также общего объема построенной функции для дальнейшей реализации был выбран алгоритм линейного регрессионного анализа. В разделе рассмотрены формулы коэффициентов корреляции, сам алгоритм регрессионного анализа, а также простые линейные и нелинейные зависимости и соответственно преобразования данных для линейного анализа в случае построения нелинейной функции.

Третий раздел «Web-приложение» посвящен описанию технологии для реализации приложения для заданной цели выпускной квалификационной работы. Рассмотрена трехслойная архитектура всего проекта, которая составлена из построения сборок логики расчетов, сборки доступа к базе данных, самой базе данных, и интерфейса пользователя – сборки самого веб-приложения. Описание выбранных технологий веб-приложения: ASP.NET MVC и RestAPI. Описание выбранной модели безопасности через авто генерируемые ключи и хэширование пароля пользователя в базе данных. Процесс авторизации через логин пароль на стороне приложения и процесс авторизации через AccessKey и SecretAccessKey на стороне внешнего клиента RestAPI приложения по HTTP протоколу и передачи модели, подходящей для RestAPI.

Четвертый раздел «Реализация Web-приложения для вывода формул» посвящен реализации всего проекта, включая вычисления, построение формул, построения системы безопасности, разработку по паттерну трехслойной архитектуры. Первый шаг реализации проекта – создание сборки Entities, содержащей в себе модели, для работы, имитирующие контейнеры эмпирических данных, и коэффициенты, которые по ним строятся. Факторы коэффициента, тип Enum выбранной функции, модель для быстрой регрессии, модель для регрессии на сохранение работы, модель, содержащая ответ сервера, модель, содержащая входные данные для работы сервера: CorrelationCoefficient.cs User.cs SubInfo.cs RegressionData.cs RegressionModel.cs FRSCClient.cs QuickRegression.cs CorrelationFactor.cs.

Следующий шаг реализации — это сборки интерфейсов логики, в которых без реализации указаны основные методы — их названия, тип возвращаемых значений, от интерфейсов наследуются классы реализации логики, через интерфейсы и объекты экземпляров классов и будет вызываться функционал на стороне веб-приложения, чтобы не засорять саму сборку ASP.NET. ICorrelationService.cs основные методами которого являются следующие: CorrellationCoefficient
CountCorrellationCoefficient(List<double> x, List<double> y);

Сервис логики пользователя IUserService.cs:

int AddUser(User user) – добавление нового пользователя при регистрации.

void DeleteUserById(int userId) – удаление пользователя по id

bool Login(User user) – авторизация пользователя на сайте

int GetUserIdByUsername(string username) – получить идентификатор пользователя по его имени пользователя

User GetUserById(int id) – получить весь объект пользователя по его идентификатору

SubInfo GetSubInfo(int userId) – получить информацию о подписке пользователя по его идентификатору

void AddSubInfo(int userId) – добавить данные о подписке пользователя по его идентификатору

int ApiAuth(string accessKey, string secretAccessKey) – авторизация по секретным ключам для проверки пользователя в RestAPI приложении.

Сервис логики IRegressionAnalyzeService.cs:

RegressionModel CountForecast(double xArg, RegressionData data) – расчёт прогноза по заданной регрессионной модели

RegressionModel CountMultipleForecast(List<double> xArgs, RegressionData data) – расчёт множественного прогноза данных

IEnumerable<RegressionData> GetRegressionDataById(int userId) – получить регрессионные данные по идентификатору пользователя

`RegressionData GetRegressionDataById(int dataId)` – получить регрессионные данные по идентификатору

`int AddNewRegressionData(RegressionData data)` – добавить новые регрессионные данные

`void UpdateRegressionData(RegressionData data)` – обновить регрессионные данные

`RegressionData GetRegressionFromFile(int dataId)` – получить регрессионную модель из файла по идентификатору

`void UploadRegressionInFile(RegressionData data)` – загрузить регрессионную модель в файл

`RegressionData CalculateRegression(RegressionData data)` – рассчитать регрессию

Далее в реализации проекта идет слой доступа к базе данных, на основе технологии ADO.Net и Microsoft.Sql.SDK для построения командных запросов в базу данных SQLEXPRESS.

На слое `DataAccessLayer` повторяются большинство методов из интерфейса логики, каждая реализация логики вызывает интерфейс `DataAccessLayer`, который в свою очередь вызывает реализацию ADO.Net и отправляет запросы в базу данных. Для подобной архитектуры вызовов используется шаблон технологии `IoC – inversion of control`, которого в .Net Framework можно добиться за счет `Ninject – Фреймворка для построения зависимостей интерфейсов` на этапе компиляции и сборки проекта. В дальнейшем на этапе разработки слоя пользовательского интерфейса используется ASP.Net MVC – `Model View Controller`. Основная часть, отвечающая за backend – `Controller`.

Это специальный механизм в приложении, который обрабатывает запрос таким образом, что, определяя входные параметры, модель, тип HTTP запроса выбирается соответствующий метод в контроллере, который вызывает запрос в логику через экземпляр соответствующего ему интерфейса. В качестве ответа пользователю он вызывает страницу

представления - View. View – являет собой html верстку и страницу типа razor5 в которую можно передать модель – сущность, которая создавалось на первом этапе проекта, хранящие определенные данные. Razor5 созданная под сущность может получить доступ к ее полям и обработки компилятором кода написанного на ней с директивой @.

Таким образом на странице представления можно гибко обозначать, какими данными может оперировать страница, и что можно показывать пользователю. Под каждый метод и контроллер Get запроса есть своя страница представления, под метод POST запроса выполняется передача сложных данных с View страниц и выполнение операций логики. Для механизма определения запроса используется IConfiguration, который определяет по каким критериям парсится запрос, шаблон запроса.

В проекте также реализован WebAPI, который в отличии от MVC – по своей сути отличается тем, что не предоставляет ответ в виде html верстки и страницы представления View Razor5, а напрямую дает ответ пользователю в виде Json, Xml или иного application формата, который пользователь в свою очередь способен корректно распарсить на своем устройстве. Передача модели ответа в сериализованном формате на сегодняшний день является одной из самых удобных и распространенных в угоду разделению работы FrontEnd и Backend разработчиков, а также быстрому доступу к функционалу извне веб-приложения – сайта. Чтобы иметь доступ к данному сервису без страниц у пользователя должны быть AccessKey SecretAccessKey которые генерируются посредством Guid – ключа, совпадение которого 1 к 10 миллиардам, что делает практически невозможным подбор и взлом. Ключи записываются в базу данных за конкретным пользователем, для которого они были сгенерированы на стороне логики сервера.

Произведена реализация алгоритма регрессионного анализа за счет построения коэффициента корреляции и приведения алгоритма к линейной системе уравнений. Прогон алгоритма позволяет вычислить коэффициенты для построения функции по предоставленным пользователем эмпирическим

данным. Для работы алгоритма и построения коэффициентов для нелинейных функций необходимо линеаризовать данные, которые приходят со стороны контроллера, не задевая основной функционал алгоритма линейной регрессии. Таким образом было реализовано приложение для построения функций на основе эмпирических данных.

ЗАКЛЮЧЕНИЕ

В ходе работы было разработано Web-приложение сервис для быстрого регрессионного анализа и расчёта прогнозов, а также с сохраняемыми моделями построенных функций линейного и нелинейного вида. Веб-приложение разработано на основе .Net Framework 4.7, ASP.NET, ADO.NET, Sql как базы данных. Были изучены методы аппроксимации и построения эмпирических функций, а также реализованы алгоритмы регрессионного анализа для эмпирических данных, для анализа зависимости предоставляемых данных и вывода аппроксимирующей формулы.

Была разработана структура классов и интерфейсов на паттерне Трёхслойной архитектуры. В паттерн входят слои - сборки на основе языка C#, такие как: Entities, BLL, BLL.Interfaces, DAO.Interfaces, DAL, Ioc, WebPL.

Приложение действует за счёт контроллера и рендеринга страниц представления по шаблону MVC(Model-View-Controller), а также тесным общением с базой данных. Дополнительно разработан WebAPI для внешних пользователей с открытой клиентской библиотекой.

Основные источники информации:

1. Корреляция [Электронный ресурс] URL - <https://ru.wikipedia.org/wiki> (Дата обращения 01.02.2021) Загл. с экрана. Яз.
2. Формула Бернулли, Пуассона. Коэффициент корреляции. Уравнение регрессии. [Электронный ресурс] URL - <https://topref.ru/referat/94802.html> (Дата обращения 01.02.2021) Загл. с экрана. Яз.
3. Линейный регрессионный анализ [Электронный ресурс]. URL: <https://www.statmethods.ru/statistics-metody/regressionnyj-analiz/#:~:text> (Дата обращения 13.02.2021) Загл. с экрана. Яз. Рус.
4. Регрессионный анализ [Электронный ресурс] URL - <https://ru.wikipedia.org/wiki> (Дата обращения 13.02.2021) Загл. с экрана. Яз.

5. В.П.Боровиков. STATISTICA. Искусство анализа данных на компьютере: для профессионалов (2-е издание), [Электронный ресурс]. URL: <http://window.edu.ru/resource/709/28709/files/ustu316.pdf> (Дата обращения 20.04.2021) Загл. с экрана. Яз. Рус.
6. Свободная энциклопедия. Платформа .NET Framework [Электронный ресурс] – URL: https://ru.wikipedia.org/wiki/.NET_Framework (Дата обращения: 15.04.2021) Загл. С экр. Яз. рус.
7. Приближение и интерполирование функций [Электронный ресурс]. URL:<https://www.booksite.ru/fulltext/1/001/008/092/627.htm> (Дата обращения 20.04.2021) Загл. с экрана. Яз. Рус.
8. Половко А., Бутусов П. Интерполяции. Методы и компьютерные технологии их реализации. [Электронный ресурс]. URL: https://www.studmed.ru/view/polovko-am-butusov-pn-interpolyaciya-metody-i-kompyuternye-tehnologii-ih-realizacii_e938ef58445.html (Дата обращения 23.04.2021)
9. Электронный учебник: Курс “Численные методы” [Электронный ресурс]. URL: http://kappa.cs.petrSU.ru/~sigovtse/study_pr/num_met/index.html (Дата обращения 23.04.2021) Загл. с экрана. Яз.Рус.
10. Е.А.Волков. Численные методы. Москва, “Наука”, Главная редакция физико-математической литературы, 1987 г. [Электронный ресурс]. URL: https://portal.tpu.ru/SHARED/d/DIMMASSIKK/academics/Additional_chapters_of_mathematics (Дата обращения 04.05.2021) Загл. с экрана. Яз. Рус.