МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»
Кафедра математической кибернетики и компьютерных наук

РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ АНАЛИЗА ДАННЫХ С ПОМОЩЬЮ СТАТИСТИЧЕСКИХ МЕТОДОВ И МАШИННОГО ОБУЧЕНИЯ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 451 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Ядчука Никиты Максимовича

Научный руководитель
ст. преподаватель _____ М. И. Сафрончик

Заведующий кафедрой
к. ф.-м. н., доцент _____ А. С. Иванов

Саратов 2020

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Развитие электронно-вычислительной техники привело к росту объемов собираемой и анализируемой информации. Объемы накопленных данных настолько велики, что человек не в состоянии обработать ее самостоятельно. Однако необходимость проведения анализа вполне очевидна, в «сырых» данных могут быть заключены знания полезные при принятии управленческих решений. Для того, чтобы провести автоматизированную процедуру анализа, используются методы Data Mining. [1]

Data Mining – это автоматизированный поиск нетривиальных практически полезных и доступных интерпретации знаний, скрытых в данных, основанный на анализе огромных массивов информации. Для сегментации данных и оценки вероятности последующих событий используются сложные математические алгоритмы. [2]

Целью выпускной квалификационной работы является разработка веб-приложения для анализа данных с помощью статистических методов и машинного обучения Data Mining для кафе и ресторанов.

Необходимо, чтобы система научилась предсказывать вероятное число гостей в интервале до пяти дней, что позволит сэкономить на заготовках, следовательно на закупаемых продуктах, и оплате труда (чем меньше гостей, тем меньше нужно обслуживающего персонала). В последствии предполагается расширить предсказание до количества заказанных блюд, присутствующих в меню, и добавить статистики полезные управленческой команде.

Приложение должно предоставлять пользователю быстрый и удобный доступ к информации по каждому своему заведению, подключенному к данному сервису.

Бакалаврская работа состоит из введения, трех разделов, заключения, списка использованных источников и семи приложений. В первом разделе «Машинное обучение» рассмотрена теоретическая часть использованных в работе методов и способов их обучения. Во втором разделе «Создание и обучение модели» описан процесса разработки прогнозирующего алгоритма и его обучения. Третий раздел «Разработка веб-приложения» содержит обзор средств разработки, описание моделей, создание шаблонов и реализация основных маршрутов приложения.

1 Машинное обучение

Машинное обучение – подраздел искусственного интеллекта, который изучает методы построения алгоритмов, способных обучаться. Различают два типа обучения:

- обучение по прецедентам (индуктивное обучение), основанное на выявлении общих закономерностей в данных;
- дедуктивное обучение, предполагает формализацию знаний экспертов и их перенос в компьютер в виде базы знаний.

Дедуктивное обучение принято относить к области экспертных систем, так что под термином машинное обучение будет подразумеваться обучение по прецедентам. Многие методы машинного обучения тесно связаны с интеллектуальным анализом данных (Data Mining).

Существуют различные способы обучения моделей, в данной работе используется обучение с учителем.

1.1 Понятие модели и алгоритма обучения. Параметры и гиперпараметры модели

Параметры модели – переменная конфигурация, которая является внутренней по отношению к модели и значение которой можно оценить на основе данных.

Гиперпараметр модели – конфигурация, внешняя по отношению к модели, значение которой невозможно оценить по данным.

Модель – это семейство функций $A = \{g(x, \theta) | \theta \in \Theta)\}$. θ называется параметрами модели. При конкретных параметрах модели получается фиксированная функция $g(x)$.

Построение фиксированной функции g по заданной обучающей выборке – это обучающий алгоритм. Обучающий алгоритм также имеет свои параметры (например, скорость обучения) – их называют гиперпараметрами.

1.2 Обучение с учителем

Обучение с учителем – распространенный способ машинного обучения, каждый прецедент которого представляет собой пару «объект, ответ». Задача заключается в нахождении функциональной зависимости ответов от описаний объектов и построении алгоритма, принимающего на вход описание объекта

и возвращающего на выходе ответ. Чаще всего обучение с учителем используется для задач классификации, регрессии и прогнозирования. [3]

1.3 Линейная регрессия

Линейная регрессия – метод аппроксимации зависимостей между входными и выходными переменными на основе линейной модели. [4]

Регрессионная модель имеет следующий вид:

$$y = f(x, b) + \varepsilon, E(\varepsilon) = 0, \tag{1}$$

где b – параметры модели, ε – случайная ошибка модели, а функция $f(x, b)$ имеет вид:

$$f(x, b) = b_0 + b_1 x_1 + ... + b_k x_k = \sum_{i=1}^{k} b_i x_i, \tag{2}$$

где b_i – параметры модели, k – количество параметров, x_i – признаки исследуемых объектов.

1.4 Нейронная сеть

Нейронная сеть – это математическая модель, состоящая из последовательности нейронов, соединенных синапсами. [5]

Нейрон – это вычислительная единица, которая получает информацию на вход, производит над ней простые вычисления и передает на выход. Внутри нейрона содержится функция активации, в данном случае выбрана функция сигмоида:

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \tag{3}$$

Функция сигмоида – гладкая монотонная возрастающая нелинейная функция, которая применяется для «сглаживания» значений некоторой величины. Данная функция преобразовывает поступающие в нее значения в вещественный диапазон $[0, 1]$, что позволяет нейронной сети усиливать слабые сигналы и не насыщаться от сильных сигналов.

Синапс – это связь между двумя нейронами, которая имеет только одну характеристику – вес.

Задача обучения нейронной сети сводится к подбору весов, таким образом чтобы ошибка на выходе была минимальной. Один из методов решения данной задачи является метод обратного распространения ошибки.

Метод обратного распространения ошибки является итеративным алгоритмом. На каждой итерации происходит два прохода сети – прямой и обратный. На прямом вектор распространяется от входов сети к ее выходам и формирует некоторый выходной вектор, соответствующий текущему (фактическому) состоянию весов. Затем вычисляется ошибка нейронной сети как разность между фактическим и целевым значениями. На обратном проходе эта ошибка распространяется от выхода сети к ее входам, и производится коррекция весов по правилу: $\Delta w = E * GRADw$, где E – скорость обучения, $GRADw$ – градиент веса. Таким образом, двигаясь в сторону антиградиента, то есть вычитая градиент (вектор частных производных функции в точке, своим направлением указывающий направление наибольшего возрастания некоторой величины), ошибка нейронной сети уменьшается.

2 Создание и обучение модели

Для выполнения поставленной задачи было принято решение использовать алгоритмы машинного обучения на языке Python, так как на данный момент времени он включает в себя большой выбор библиотек, упрощающих работу в данном направлении.

При построении модели были определены факторы влияющие на посещаемость заведения, такие как: первая или вторая половина дня, день недели, число, месяц, температура, погодное явление, праздничный день, выходной. Данные факторы были использованы в качестве входных данных при обучении модели.

2.1 Подготовка датасета

Так как архивы погоды не бесплатны, был использован инструмент для автоматизации действий веб-браузера Selenium. Был написан следующий скрипт, который отбирал с сайта историю погоды в нужном промежутке времени и записывал в файл.

Таким образом, был собран архив погоды для необходимого временного периода. Для удобства выборки погоды данные были перенесены в SQLite в таблицу «weather».

Также в таблицу «not_working_days» были занесены официальные выходные и праздничные в Российской Федерации.

Следующим шагом было изучение баз данных ресторанов, все они имеют практически одинаковую структуру, так как используют программное обеспечение R-Keeper. Была найдена сущность «Чек», из которой сделана выборка количества гостей за первую и вторую половины дня, исключая лишние списания.

Значения итогового массива датасета нормализованы и принадлежат интервалу [0, 1], а также записаны в файл для дальнейшего использования при обучении модели.

Ожидаемые результаты записаны в файл.

2.2 Линейная регрессия

Данный метод был опробован первым. Для этого выборка была поделена на тренировочную (train) и тестовую (test) в отношении 7:3. Точность результата оценивалась средней квадратической ошибкой. Для упрощения написа-

ния кода использовались такие библиотеки, как «scikit-learn» [6], «pandas» [7], «matplotlib» [8].

В результате средняя квадратическая ошибка составила 0.016.

2.3 Нейронная сеть

Следующим методом была опробована нейронная сеть. Сеть представляет собой класс с объектами входного, выходного слоя и слоя весов. Веса заполняются рандомными значениями в интервале [-1, 1].

Далее был написан алгоритм обратного распространения ошибки, состоящий из прямого прохода по сети и обратного с последующей корректировкой весов. Этот алгоритм выполняется в цикле достаточно большое количество эпох с постепенным уменьшением скорости обучения (коэффициент k). Если прогресса не наблюдается цикл завершает свою работу.

Изначально, была опробована двухслойная нейронная сеть, но ее средняя квадратическая ошибка была примерно равна ошибке линейной регрессии. Поэтому была опробована трехслойная сеть. Сначала на одной эпохе было протестировано оптимальное количество нейронов в каждом слое, лучший результат был показан при 12 нейронах в каждом слое. Далее был подобран коэффициент скорости обучения, равный 0.1. В итоге нейросеть была обучена и каждый слой записан в файл.

Результат превзошел линейную регрессию и средняя квадратическая ошибка составила 0.009.

Исходя из результатов исследования, было принято решение использовать нейронную сеть в качестве модели прогнозирования количества гостей в веб-приложении.

3 Разработка веб-приложения

В данном разделе будет описан процесс разработки приложения, приведены детали реализации и сценарии использования.

Процесс разработки состоял из следующих частей:

- инициализация приложения;
- создание моделей и миграция БД;
- создание шаблонов;
- реализация основных маршрутов.

3.1 Средства разработки

Разработка веб-приложения проходила на языке Python с целью упрощения взаимодействия программных модулей. Для производительности, безопасности и отказоустойчивости приложения был выбран микрофреймворк Flask.

Flask – минималистичный и простой фреймворк, позволяет внедрять различные внешние модули и библиотеки, использовать приложения наподобие SQLAlchemy [9] или чистые SQL-запросы без всяких ограничений. Данные качества полностью соответствуют данному проекту, поэтому выбор остановился именно на нем.

Для хранения данных о пользователях и их заведениях была выбрана база SQLite. Она имеет свободную лицензию, хранится в одном файле, является кроссплатформенной и поддерживает стандарты SQL. [10]

3.2 Описание моделей и взаимосвязей веб-приложения

Модель – это представление таблицы базы данных в код. Она определяет структуру хранимых данных и позволяет Flask получать данные и управлять ими.

При проектировании веб-приложения было создано две сущности:

- пользователь
- заведение

Связь пользователь и заведение – один ко многим.

3.3 Модель «Пользователь»

В данном веб-приложении пользователь может осуществлять вход на сайт, изменять пароль и просматривать статистику и прогноз для каждого

своего заведения, подключенного к системе «MyRestaurant».

По этим критериям была составлена модель в, которой каждому пользователю соответствует:

- уникальный идентификатор;
- имя пользователя на сайте;
- электронная почта;
- хеш пароля.

3.4 Модель «Заведение»

Так как у одного пользователя может быть несколько заведений, следует добавить внешний ключ на идентификатор пользователя.

Таким образом, модели заведения соответствуют следующие поля:

- уникальный идентификатор;
- название заведения;
- ID пользователя (внешний ключ).

3.5 Инициализация приложения

Начальный этап разработки заключался в инициализации приложения Flask и подключения необходимых расширений, а именно:

- flask_sqlalchemy – добавляет поддержку SQLAlchemy в приложение Flask. SQLAlchemy – это библиотека на языке Python для работы с реляционными СУБД с применением технологии ORM, позволяет описывать структуры баз данных и способы взаимодействия с ними на языке Python без использования SQL, есть возможность использовать SQL выражения независимо от ORM; [11]
- flask_login – добавляет поддержку авторизации и обеспечивает управление сессиями пользователей. [12]
- flask_mail – интерфейс для отправки электронных писем из приложения; [13]

Flask использует концепцию blueprint-ов для создания компонентов приложений и поддержки общих шаблонов. Данная концепция позволяет достичь более практичной организации и упрощает повторное использование кода.

3.6 Создание моделей и миграция БД

Создание моделей происходит при помощи объявления класса унаследованного от базового класса `db.Model`.

Для создания столбца таблицы используется функция `db.Column` с указанием дополнительных параметров. Имя столбца определяется наименованием поля класса.

Отношение `db.relationship` позволяет получать доступ к списку всех заведений пользователей. Аргумент `backref` определяет поле, которое будет добавлено объекту класса `Restaurant`, указывающее на объект `User`.

Чтобы описание моделей перешло в базу данных, необходимо выполнить миграцию.

3.7 Создание шаблонов

Flask использует шаблонизатор Jinja2, который позволяет передавать аргументы, использовать условные операторы и циклы в шаблонах, а также наследовать шаблоны.

3.7.1 Базовый шаблон

С помощью шаблонизатора была создана базовая шаблон, включающий в себя общую для всех страниц структуру сайта, а именно подключенные скрипты, файлы стилей и навигационную панель, что позволяет избежать повторение кода и использовать оператор `{% extends %}` для наследования в остальных шаблонах.

Навигационная панель содержит возможности:

- входа/выхода;
- смены пароля;
- переход на главную страницу.

Для небольших экранов реализована складная панель по средством использования классов «navbar-toggler» и «navbar-collapse».

3.7.2 Шаблон главной страницы

Главная страница наследует базовый шаблон, в котором определена навигационная панель сайта, также с помощью условного оператора `{% if %}` происходит проверка авторизации пользователя. Если пользователь авторизован, то на главную страницу выводится список его заведений при помощи

цикла {% for %}. Также используется аргумент restaurant, для получения логотипа и названия заведения.

3.7.3 Шаблон страницы авторизации

Для входа на сайт реализован шаблон авторизации, наследуемый от общего шаблона для входа и восстановления пароля, который содержит реализацию формы.

Шаблон состоит из карточки, содержащей заголовок, задаваемый переменной {{ title }} и основную форму с возможностью вывода сообщений об ошибках.

Тег input задает поля для заполнения пользователем. На тот случай, если пользователь забыл пароль определена ссылка для его восстановления "/forgot_password".

3.7.4 Шаблон страницы статистики

На странице статистики пользователь может посмотреть прогноз количества гостей, а также некоторые статистики, такие как лучшие блюда за последний месяц, среднее пребывание гостя и средний чек. В дальнейшем планируется расширение функционала.

Данный шаблон содержит DatePicker с целью удобства выбора даты для прогноза. Также с помощью тега <img> отображается график, созданный при переходе на страницу конкретного заведения.

При помощи условного оператора {{ if }} производится вывод информации о прогнозе, переменные заключенные в фигурные скобки отображают информацию переданную данному шаблону. Шаблон поделен на колонки и строки при помощи классов "col" и "row", что позволяет структурировать отображаемую информацию.

3.8 Реализация основных маршрутов

Flask использует декоратор route() для связывания функции с URL.

При переходе пользователя на страницу авторизации функция login() возвращает HTML-страницу по шаблону указанном в первом аргументе функции render_template(). При заполнении формы и нажатии кнопки «Войти» осуществляется вызов функции login_post(), в которой проверяется наличие

пользователя, сравнивается хеш паролей, и если все корректно, происходит авторизация и перенаправление на главную страницу. В том случае, если пользователь забыл пароль, есть возможно сбросить его, перейдя по ссылке «Забыли пароль?». После того, как пользователь введет свой почтовый адрес, существующий в базе данных приложения, ему на почту придет письмо, содержащее ссылку с токеном на восстановление пароля. Генерация токена происходит при помощи функции, заданной в классе `User`, `get_reset_password_token()`. Отправка письма выполняется при помощи ранее упомянутого интерфейса `flask_mail`. Создание письма осуществляется встроенной в интерфейс функции `Message`.

Перейдя на страницу статистики, можно посмотреть прогноз количества гостей, выбрав дату. После нажатия на кнопку «`Прогноз гостей`» происходит вызов метода `forecast_guests()` класса `Restaurant`, в который передается дата введенная пользователем. Данный метод запрашивает прогноз погоды при помощи библиотеки «pyowm», использующей веб-API OpenWeatherMap, формирует и нормализует входные данные для нейронной сети, передает массив данных функции `forward()`, которая вычисляет предсказание нейронной сети определенного заведения.

При помощи sql-запроса находятся популярные блюда за последний месяц. Результаты запросы передаются в функцию `diagram`, которая средствами библиотеки «Matplotlib» строит диаграмму, сохраняет в виде картинки формата `.png` и передает в виде параметра шаблону. Также был написан запрос для нахождения среднего чека за последний месяц.

ЗАКЛЮЧЕНИЕ

В ходе выпускной квалификационной работы был разработан алгоритм, прогнозирующий количество гостей, подсчитаны статистики, полезные управленческой команде, реализовано веб-приложение, в которое были интегрированы прогнозирующий алгоритм и статистические измерения. Таким образом, все поставленные задачи были выполнены. Также был изучен фреймворк Flask, использование которого помогло упростить процесс разработки веб-приложения, и несколько фреймворков машинного обучения.

Результатом работы стало готовое к использованию веб-приложение, реализующее все заданные требования.

В последствие предполагается расширить предсказание до количества заказанных блюд, присутствующих в меню, и добавить статистики полезные управленческой команде.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Data Mining — добыча данных. — URL: https://basegroup.ru/community/articles/data-mining (Дата обращения 05.05.2020). Загл. с экр. Яз. рус.

2 *Ian H. Witten, E. F.* Data Mining: Practical Machine Learning Tools and Techniques. / E. F. Ian H. Witten, M. A. Hall. — 3rd, edition. — Morgan Kaufmann, 2011.

3 Машинное обучение: методы и способы. — URL: https://www.osp.ru/cio/2018/05/13054535/ (Дата обращения 25.05.2020). Загл. с экр. Яз. рус.

4 Основы линейной регрессии. — URL: http://statistica.ru/theory/osnovy-lineynoy-regressii/ (Дата обращения 10.05.2020). Загл. с экр. Яз. рус.

5 Нейронные сети. — URL: http://statsoft.ru/home/textbook/modules/stneunet.html (Дата обращения 25.05.2020). Загл. с экр. Яз. рус.

6 scikit-learn Tutorials. — URL: https://scikit-learn.org/stable/tutorial/index.html (Дата обращения 15.05.2020). Загл. с экр. Яз. англ.

7 pandas documentation. — URL: https://pandas.pydata.org/docs/ (Дата обращения 15.05.2020). Загл. с экр. Яз. англ.

8 matplotlib documentation. — URL: https://matplotlib.org/3.2.1/contents.html (Дата обращения 17.05.2020). Загл. с экр. Яз. англ.

9 SQLAlchemy 1.3 Documentation. — URL: https://docs.sqlalchemy.org/en/13/intro.html (Дата обращения 20.05.2020). Загл. с экр. Яз. англ.

10 SQLite Documentation. — URL: https://www.sqlite.org/docs.html (Дата обращения 20.05.2020). Загл. с экр. Яз. англ.

11 Flask-SQLAlchemy Documentation. — URL: https://flask-sqlalchemy.palletsprojects.com/en/2.x/ (Дата обращения 20.05.2020). Загл. с экр. Яз. англ.

12 Flask-Login Documentation. — URL: https://flask-login.readthedocs.io/en/latest/ (Дата обращения 23.05.2020). Загл. с экр. Яз. англ.

13 Flask-Mail Documentation. — URL: https://pythonhosted.org/Flask-Mail/ (Дата обращения 24.05.2020). Загл. с экр. Яз. англ.