МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

РЕАЛИЗАЦИЯ И СРАВНИТЕЛЬНЫЙ АНАЛИЗ КЛАССИФИКАЦИИ ПО ПАРАМЕТРАМ И С ПОМОЩЬЮ ГЛУБОКОГО АНАЛИЗА ТЕКСТА

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы
направления 02.03.03 Математическое обеспечение и администрирование информационных систем
факультета компьютерных наук и информационных технологий
Брускова Олега Дмитриевича

Научный руководитель:
зав.кафедрой, к. ф.-м. н., доцент _____ М.В. Огнева

подпись, дата

Зав. кафедрой:
к.ф.-м.н., доцент _____ М.В. Огнева

подпись, дата

Саратов 2021

ВВЕДЕНИЕ

Актуальность темы.

Глубокий анализ текста позволяет решать весьма нетривиальные задачи. Создавая алгоритм, способный выявлять из текста не только простую фактическую информацию, но и эмоциональный тон, объект обсуждения и другие, более тонкие элементы человеческой речи, программист открывает для себя новый путь решения множества различных задач.

Однако, в связи с комплексностью такого процесса, как глубокий анализ текста, важно понимать, в каких случаях это будет лучшим из возможных решений. Так как, в обратном случае, программист рискует воспользоваться сложным и требовательным алгоритмом в задаче, с решением которой можно справиться и более простым подходом.

Целью бакалаврской работы—является проведение сравнительного анализа двух подходов машинного обучения – глубокий анализ текста обзора и анализ параметров обозреваемого объекта.

Поставленная цель определила **следующие задачи**:

1. Изучение алгоритмов машинного обучения, таких как Наивный Байес, Логистическая регрессия, Деревья решений и Нейронные сети.
2. Предварительная обработка данных для машинного обучения в глубоком анализе текста, таких как «мешок слов», стоп-слова и tf-idf
3. Изучение и реализация «бэггинга», проверка его влияния на результаты других алгоритмов
4. Реализация вышеописанных алгоритмов и их использование на датасете винных изделий
5. Проверка эффективности и скорости обучения для рассмотренных алгоритмов с различными параметрами, сравнение результатов
6. Анализ полученных результатов и выводы о изученных инструментах

Методологические основы машинного обучения и глубокого анализа текста представлены в работах Хенрика Бринка и Джозефа Ридчардса [2], Франсуа Шоле [4] и Джоэла Граса [5].

Теоретическая значимость бакалаврской работы. Значимость данной работы заключается в анализе и сравнении различных методов обработки данных, таких как подбор параметров, а также различные виды предобработки текста,

Структура и объём работы. Бакалаврская работа состоит из введения, 2 разделов, заключения, списка использованных источников и 4 приложений. Общий объем работы – 99 страниц, из них 51 страница – основное содержание, включая 10 рисунков и 5 таблиц, список использованных источников информации – 23 наименования.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Теоретические основы машинного обучения и нейронных сетей» посвящен машинному обучению и различным методам внутри него.

Машинное обучение – это извлечение закономерностей из ограниченного количества примеров. Чтобы лучше понять основные понятия и принципы работы машинного обучения, можно воспользоваться показательной задачей.

Допустим, есть множество обзоров на кинофильмы, каждому из которых соответствует бинарная величина. «0» - если обзор отрицательный, и «1» - если положительный. Неизвестно, кто написал обзор, к какому фильму он относится, все что дано – текст и бинарное число. Поставленная задача – найти закономерность между оценкой фильма и текстом, которому эта оценка соответствует.

Множество текстов в данном случае называется пространством объектов, и обозначается как X. Они являются объектами из-за того, что в итоге на вход будет поступать именно текст обзора, и алгоритм должен будет сопоставить ему определенное значение.

Бинарное число, отвечающее за оценку фильма, которое необходимо получить на выходе – целевая переменная. Множество ее значений –

пространство ответов Y. В данном случае – 0 и 1. Вышеописанная задача – это пример supervised learning или обучения с учителем, а конкретнее – бинарная классификация.

Как и в большинстве задач программирования, не существует одного единственно правильного решения задач машинного обучения. Существует множество алгоритмов, и каждый из них может быть полезен в той или иной ситуации.

Байесовский подход к классификации основан на теореме, утверждающей, что если плотности распределения каждого из классов известны, то искомый алгоритм можно выписать в явном аналитическом виде. Более того, этот алгоритм оптимален, то есть обладает минимальной вероятностью ошибок.

На практике плотности распределения классов, как правило, не известны. Их приходится оценивать по обучающей выборке. В результате байесовский алгоритм перестаёт быть оптимальным, так как восстановить плотность по выборке можно только с некоторой погрешностью. Чем короче выборка, тем выше шансы подогнать распределение под конкретные данные и столкнуться с эффектом переобучения.

Байесовский подход к классификации является одним из старейших, но до сих пор сохраняет прочные позиции в теории распознавания. Он лежит в основе многих достаточно удачных алгоритмов классификации.

Дерево решений — это ориентированный граф, в вершинах которого записаны некоторые необходимые к проверке условия (признаки, которые позволяют отличить одни объекты от других), а в концевых вершинах («листьях») содержатся все классы, на которые можно распределять проходящие по дереву объекты (а в случае решения задачи регрессии — конкретные значения некоторой целевой переменной).

Объект, проходящий по дереву от корня к листу, оказывается однозначным образом отнесён к тому или иному классу или же получает однозначное значение целевой переменной. В случае, если при обучении в

лист попали объекты нескольких классов, всякому новому объекту будет присваиваться метка преобладающего класса.

Деревья решений используются в повседневной жизни в самых разных областях человеческой деятельности, порой и очень далеких от машинного обучения. Деревом решений можно назвать наглядную инструкцию, что делать в какой ситуации.

Логистическая регрессия названа в честь логистической функции, которая лежит в основе этого метода. Логистическая функция представляет собой S-образную, функцию которая переводит любые числовые значения в диапазон от 0 до 1.

Логистическая регрессия использует уравнение в качестве представления. Входные значения x линейно объединяются с использованием весов или значений коэффициентов для прогнозирования выходного значения y. Логистическая регрессия строит вероятность первого класса (класса по умолчанию). Например, если задача заключалась в делении объектов на два класса, то логистическая регрессия построит вероятность принадлежности объекта первому классу.

В некоторых случаях, для достижения наилучшего результата, может возникнуть необходимость создания и обучения нескольких отдельных моделей. Такая совокупность алгоритмов называется ансамблем. Ансамбль считается успешно построенным, если его точность превосходит точность любой из входящих в него моделей. Одним из методов создания ансамблей является бэггинг.

Бэггинг это технология классификации, использующая композиции алгоритмов, каждый из которых обучается независимо. Результат классификации определяется путем голосования. Бэггинг позволяет снизить процент ошибки классификации в случае, когда высока дисперсия ошибки базового метода.

На данный момент глубокие нейронные сети – это один из самых популярных методов машинного обучения. Такой рост популярности не

случаен. Сам метод существует с середины 20-го века, Уоррен Мак-Каллок и Уолтер Питтс разработали компьютерную модель нейронной сети в 1943 году. Однако на пути практического применения данной модели стояло два фактора – недостаточные вычислительные мощности и малое количество доступной информации. Сейчас, благодаря стремительному росту производительности компьютеров и появлению таких отраслей как Big Data и Data Science, практическое применение модели нейронных сетей стало возможным и доступным.

Основное отличие нейронных сетей от других методов машинного обучения заключается в том, что подбор признаков в данной модели выполняется компьютером. В то время как в других методах подбор критериев, по которым компьютер будет, например, разбивать элементы на категории (задача классификации), выполняется человеком, нейронные сети самостоятельно выявляют необходимые признаки в процессе обучения.

Данный фактор позволяет решать гораздо больший спектр задач. Способы классифицировать текст на тонально положительный и тонально отрицательный существуют довольно давно, и для их решения не всегда необходимо прибегать к использованию нейронных сетей. С другой стороны, существуют такие задачи, как, например, классификация изображений животных на изображения «кошек» и изображения «собак». Выявить понятные компьютеру методы для различия таких изображений человек будет не в состоянии, поэтому для решения такой задачи необходимо прибегать к нейронным сетям. Тем не менее, это не делает данный метод объективно лучшим, так как упрощение реализации (особенно при использовании библиотек) и широкий сектор выполнимых задач компенсируется значительным ростом вычислительных требований.

Одной из самых главных задач в машинном обучении является правильный подбор параметров. Зачастую именно выявление и подготовка важной информации для алгоритма может стать решающим фактором в успешности алгоритма.

Так как данная работа во многом сосредоточена на глубоком анализе текста, большая часть рассмотренных и использованных методов будет сосредоточена именно на выявлении признаков из текста обзора. Однако, стоит упомянуть, что в качестве признаков может использоваться практически любая информация, до тех пор, пока она может быть представлена в числовом виде. Так, например, в рамках данной работы для оценки винных изделий в качестве признаков будут использованы такие параметры как цена продукта, страна изготовления, винодельня и.т.д.

Один из самых простых и интуитивно понятных методов выявления параметров из текста является метод «мешок слов». Используя этот метод, мы ставим в приоритет частоту встречаемости слов в тексте. Таким образом алгоритм может выявить закономерности в использовании похожих слов в похожих по тональности обзорах.

Стоит отметить, что данный алгоритм опускает множество важных для понимания текста деталей. Так, например, порядок слов, абзацы, знаки препинания и регистр, в котором были напечатаны слова не учитываются. То есть две английские фразы «this was good, not bad» и «this was good, not bad» являются идентичными в контексте «мешка слов», несмотря на то что являются противоположными по значению.

Однако, несмотря на свои недостатки, «мешок слов» остается весьма популярным способом задания параметров для глубокого анализа текста благодаря своей простоте и эффективности.

«Стоп-слова» не являются самостоятельным методом выявления параметров, а лишь помогают уточнить выявленную из мешка слов информацию.

В большинстве языков есть слова, которые встречаются настолько часто и несут так мало смысловой нагрузки, что практически не влияют на смысл текста. Метод «стоп-слов» позволяет избавить словарь от подобных слов.

Статистическая мера tf-idf (term frequency-inverse document frequency) также дополняет метод «мешка слов», придавая больший «вес» тем словам,

которые чаще встречаются в данном экземпляре текста и реже в других экземплярах.

В машинном обучении и глубоком анализе текста существует множество различных методов и моделей, каждый из которых имеет свои плюсы и минусы. И при решении конкретной задачи, если программист желает добиться максимальной оптимизации, ему необходимо понимать эти различия.

Второй раздел «Эксперименты с алгоритмами машинного обучения для классификации винных изделий» посвящен реализации всех возможных комбинаций алгоритмов и предобработок, описанных в первом разделе.

Целью данной работы является классификация винных изделий. Каждая модель будет определять, является ли вино «хорошим» или «плохим». В качестве данных используется таблица винных изделий с сайта kaggle.com.

Данный датасет был использован для машинного обучения в нескольких работах, однако в этих работах рассматривается только анализ на основе цены и текста, а также обучается лишь одна модель. В контексте данной работы главной задачей является сравнительный анализ подбора параметров и глубокого анализа текста, а также различных моделей машинного обучения. Использованная таблица отлично подходит для сравнения классического машинного обучения и глубокого анализа текста, так как содержит в себе не только обзор, но и дополнительную информацию о продукте.

В качестве языка программирования для этой работы был выбран Python, так как он прост в понимании и имеет ряд библиотек, упрощающих работу с машинным обучением. В ходе работы было использовано множество вспомогательных библиотек, таких как pandas для работы с xls таблицами, datetime и time для подсчета времени обучения и math для некоторых математических операций. Для большей части алгоритмов

преобразования данных и машинного обучения была использована библиотека sklearn.

Первый рассмотренный метод берет в качестве параметров шесть значений – страна изготовления, сад в котором рос виноград, провинция, тип вина и винодельня. Чтобы алгоритмы из библиотеки sklearn смогли воспринять данные, их нужно было перенести в числовую форму. Для этого был создан список уникальных значений каждого поля, после чего каждое значение было заменено на индекс, соответствующий этому значению в списке.

Далее два полученных списка проходят через стандартную для машинного обучения процедуру. Оба списка симметрично перемешиваются, таким образом, чтобы продукты находились в случайном порядке, но индексы между двумя списками по-прежнему совпадали. После этого на основе этих списков создается четыре новых списка – x_train и y_train для обучения модели и y_train и y_val для проверки ее точности. Для проверки используется 1000 элементов, в то время как остальные 149930 используются в обучении. После этого данные готовы к использованию в большинстве моделей машинного обучения. В контексте этой работы были использованы такие модели, как наивный Байес, нейронная сеть, дерево решений, логистическая регрессия а также несколько «беггингов», в частности случайный лес из пяти деревьев и ансамбль из 5 логистических регрессий.

После того, как были получены результаты классического машинного обучения, был проверен метод «мешка слов» на текстах обзоров.

Для этого необходимо снова создать два списка, на этот раз первый будет содержать в себе тексты обзоров, а не параметры. После этого на основе списка обзоров создается «мешок слов» при помощи библиотеки sklearn. После этого мешок слов и список результатов проходит через практически идентичные описанным в выше процессы. После этого были рассмотрены результаты обучения моделей с использованием «мешка слов», из словаря которого были удалены все стоп-слова английского языка. Далее было

рассмотрено влияние масштабирования параметров на качество и скорость обучения моделей при помощи меры tf-idf.

Совместив все результаты в единую таблицу можно получить наглядную демонстрацию проделанной работы

Параметры (точность/ время в сек.) /Алгоритм	Наивный Байес	Дерево Решений	Логистическая Регрессия	Лес Решений	Ансамбль лог. регрессий	Нейронная сеть
Классическое МО	0.667	0.849	0.736	0.805	0.709	0.691
	0:00:00.3244	0:00:00.6642	0:00:02.7604	0:00:00.9358	0:00:00.9437	0:03:02.8065
Мешок слов	N/A	0.882	0.877	0.817	0.855	0.93
	N/A	0:04:07.2864	0:00:15.0703	0:01:10.2414	0:00:07.6797	1:48:59.8636
Мешок слов + стоп слова	N/A	0.876	0.87	0.816	0.846	0.931
	N/A	0:02:35.9722	0:00:11.6210	0:00:54.3281	0:00:07.4646	0:24:14.9930
tf-idf	N/A	0.886	0.859	0.816	0.828	0.923
	N/A	0:02:41.8914	0:00:06.2119	0:01:02.1513	0:00:02.5169	0:58:03.9986

Таким образом, в контексте данной задачи глубокий анализ текста в среднем справляется лучше, чем классическое МО. Нейронная сеть для глубокого анализа текста дала лучший по качеству результат, но обучалась от 20 минут до 2 часов. Это ожидаемо, так как нейронные сети используют для нетривиальных задач, где другие алгоритмы не справляются. Еще одной причиной тому может быть, как четкость и краткость обзоров, так и недостаточное количество информации о винных изделиях.

ЗАКЛЮЧЕНИЕ

В процессе выполнения данной работы были изучены различные плюсы и минусы алгоритмов машинного обучения как в контексте

классического машинного обучения, так и глубокого анализа текста, а также способы формирования параметров для обучения этих моделей.

Было проведено обучение моделей на подобранных из датасета параметрах, таких как страна изготовления, виноградник, цена и другие. На этих данных были обучены такие модели, как наивный Байес, нейронная сеть, дерево решений, логистическая регрессия а также несколько «беггингов», в частности случайный лес из пяти деревьев и ансамбль из 5 логистических регрессий. В результате данного анализа были получены положительные результаты. Лучший результат для классического машинного обучения показало дерево решений, получив в процессе обучения точность в 0.849

После этого было проведено обучение идентичных моделей, таких как наивный Байес, нейронная сеть, дерево решений, логистическая регрессия и другие, но в качестве параметров был использован текст обзоров на изделия. В качестве методов предобработки текста были использованы «мешок слов», «стоп-слова» и «tf-idf». Результат данного обучения оказался лучше, чем обучения на подобранных параметрах. Лучшей моделью оказалась нейронная сеть, получив точность в 0.931.

В конечном итоге лучше всего справилась с глубоким анализом текста нейронная сеть. Немного хуже, но намного быстрее нее справилось дерево решений при использовании меры tf-idf, на третьем месте оказалась логистическая регрессия.

Таким образом, алгоритмы машинного обучения могут давать невероятно точные результаты, но необходимо индивидуально подходить к каждой задаче, подбирая параметры, методы и модели анализа, сравнивая результаты. Без понимания основ работы алгоритмов невозможно построить качественную и эффективную модель.

Основные источники информации:

1 "Введение в Машинное обучение" - Лекции МФТИ // GitHub URL: https://github.com/esokolov/ml-course-hse (дата обращения: 23.05.2019).

2 Хенрик Бринк, Джозеф Ричардс, Марк Феверолф Машинное обучение. СПб.: ПИТЕР, 2017 (дата обращения: 23.05.2019).

3 Онлайн курс "Программирование глубоких нейронных сетей на Python" // asozykin.ru URL: https://www.asozykin.ru/courses/nnpython (дата обращения: 18.05.2019).

4 Франсуа Шолле Глубокое обучение на Python. СПб.: ПИТЕР, 2018 (дата обращения: 18.05.2019).

5 Джоэл Грас Data Science Наука о данных с нуля. СПб.: БХВ-Петербург, 2017 (дата обращения: 23.05.2019).

6 Дж. Вандер Плас Python для сложных задач. СПб.: ПИТЕР, 2018 (дата обращения: 23.05.2019).

7 Дэви Силен, Арно Мейсман, Мохамед Али Основы Data Science и Big Data. СПб.: ПИТЕР, 2017 (дата обращения: 23.05.2019).