**Министерство науки и высшего образования**

**Российской Федерации**

**ФЕДЕРАЛЬНОЕ Государственное Автономное**

**образовательное учреждение высшего образования**

**«новосибирский национальный исследовательский государственный университет»**

**Факультет информационных технологий**

**РЕФЕРАТ**

**На тему: «Сравнение методов решения задачи классификации комментариев в Интернете»**

Студенты:

Заляева Александра Вячеславовна

Юрин Марк Антонович

Преподаватель:

Пименов Иван Сергеевич

Новосибирск 2024

**СОДЕРЖАНИЕ**

[ЦЕЛЬ 3](file:///C:\Users\Dmitry\Downloads\Pract3_Юрин_206.docx#__RefHeading___Toc18443921)

[ЗАДАНИЕ 3](file:///C:\Users\Dmitry\Downloads\Pract3_Юрин_206.docx#__RefHeading___Toc18443921)

[МЕТОДЫ КЛАССИФИКАЦИИ 3](file:///C:\Users\Dmitry\Downloads\Pract3_Юрин_206.docx#__RefHeading___Toc18443922)

Наивный байесовский классификатор3

Стохастический градиентный спуск4

K-ближайших соседей5

Дерево решений5

Случайный лес5

Искусственные нейронные сети6

Метод опорных векторов6

СРАВНЕНИЕ МЕТОДОВ КЛАССИФИКАЦИИ6

ВЫВОД7

[СПИСОК ЛИТЕРАТУРЫ](file:///C:\Users\Dmitry\Downloads\Pract3_Юрин_206.docx#__RefHeading___Toc18443924) 8

[Приложение 1. Матрицы ошибок](file:///C:\Users\Dmitry\Downloads\Pract3_Юрин_206.docx#__RefHeading___Toc18443925) 9

**ЦЕЛЬ**

Целью работы являются изучение методов классификации текстовой информации и получение практических навыков их использования. Классификация является одной из основных задач компьютерной лингвистики. Скорость и качество классификации текстов непосредственно зависят от применяемых методов, поэтому их изучение позволит более эффективно решать задачи анализа данных. Анализ и классификация текстов имеют большое значение в сфере обеспечения общественной безопасности (автоматический поиск противоправной информации) и бизнеса (интеллектуальные помощники, спам-фильтры). Так как компьютерные технологии всё больше и больше проникают в жизнь человека, эта тема будет актуальна ещё долгое время.

**ЗАДАНИЕ**

Для выполнения работы были поставлены следующие задачи:

1. Изучение методов классификации текстовой информации.
2. Написание программы, осуществляющей парсинг комментариев к географическим местам из сервиса Google Maps, а именно к организациям театров. Программа должна получать комментарии из интернета и обрабатывать их определенным образом.
3. Написание программы, осуществляющей классификацию комментариев, используя изученные методы. Для комментариев выделено 2 класса: “позитивные” и “негативные”.
4. Проведение сравнительного анализа результатов классификации и определения лучшего метода классификации для поставленной задачи.

**МЕТОДЫ КЛАССИФИКАЦИИ**

Классификация – один из разделов машинного обучения, посвященный задаче разделения множества объектов на несколько заранее определенных классов. Задача классификации решается с помощью различных методов обучения с учителем.

Существует множество методов решения задачи классификации:

* Наивный байесовский классификатор (NB)
* Стохастический градиентный спуск (SGD)
* K-ближайших соседей (KNN)
* Дерево решений (DT)
* Случайный лес (RF)
* Искусственные нейронные сети (FFBP, DAN2, CNN, MLP)
* Метод опорных векторов (SVM)

**Наивный байесовский классификатор**

Наивный метод Байеса - алгоритм машинного обучения, основанный на применении одноименной теоремы с предположением о том, что признаки попарно условно независимы при фиксированном значении переменной класса. То есть, при “наивном” предположении для Согласно теореме Байеса:

, где у - метка класса и (x1, …, xn) – вектор зависимых признаков.

Используя “наивное” предположение для и то, что есть константа (зависит только от входных данных), получим правило классификации:

Несмотря на предположение об условной независимости входных данных, Байесовский классификатор обеспечивает достаточную точность классификации в прикладных задачах. Более того, для достижения приемлемой точности хватает сравнительно небольшого набора данных для обучения. Основным недостатком Байесовского классификатора является то, что в общем случае качество классификации низкое. Больше всего это проявляется в задачах, где признаки сильно зависимы друг от друга.

В нашей работе наивный байесовский метод представлен классом MultinomialNB из библиотеки sklearn.

**Стохастический градиентный спуск**

Метод градиентного спуска - численный [метод](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D0%B4%D0%B8%D0%B5%D0%BD%D1%82%D0%BD%D1%8B%D0%B5_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D1%8B) нахождения локального минимума или максимума [функции](https://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BB%D0%B5%D0%B2%D0%B0%D1%8F_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F) с помощью движения вдоль [градиента](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D0%B4%D0%B8%D0%B5%D0%BD%D1%82). Он является одним из основных численных методов современной оптимизации.

Алгоритм состоит в следующем:

1. Задать начальное приближение (x0, …, xn) = и точность расчета .
2. Найти .
3. Если , то . Иначе .

Для минимизации значения функции необходимо повторять шаги в направлении, противоположном градиенту функции в текущей точке, то есть в направлении наискорейшего спуска. Цель градиентного спуска - минимизация функции потерь. Достоинствами этого способа обучения являются его эффективность и простота реализуемости.

В нашей работе представлен классом sklearn.linear\_model.SGDClassifier

**K-ближайших соседей**

Метод К-ближайших соседей это алгоритм отложенного обучения, хранящий в n-мерном пространстве экземпляры, соответствующие обучающим данным. Это алгоритм “ленивого” обучения**,** который хранит экземпляры обучающих данных вместо построения общей внутренней модели. Классификация производится простым голосованием k ближайших соседей каждой точки. Чтобы обозначить новую точку, программа просматривает помеченные точки, ближайшие к этой новой точке, они называются её “ближайшими соседями”. Эти соседи голосуют и метка, которая есть у большинства соседей, становится меткой для новой точки. k - количество соседей, которые проверяются. К преимуществам алгоритма можно отнести простоту реализации и устойчивость к выбросам. К недостаткам относится низкая скорость работы при большом размере выборки.

В нашей работе метод представлен классом sklearn.neighbors.KNeighborsClassifier

**Дерево решений**

Дерево решений - непараметрический контролируемый метод обучения, который используется для решения задач [классификации](https://scikit-learn.ru/1-10-decision-trees/#tree-classification) и [регрессии](https://scikit-learn.ru/1-10-decision-trees/#tree-regression). Метод основан на создании модели, предсказывающей значение целевой переменной путём изучения простых правил принятия решений, которые были выведены из характеристик данных. Для этого строится дерево решения, на рёбрах которого записаны признаки, от которых зависит целевая функция. В концевых вершинах графа записаны значения [целевой функции](https://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BB%D0%B5%D0%B2%D0%B0%D1%8F_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F) (выходные значения), а в остальных узлах - признаки, по которым случаи различаются. Чтобы классифицировать новый случай, надо спуститься по дереву до концевой вершины и выдать соответствующее значение. Преимуществами метода являются быстрое обучение, четкие формируемые правила классификации. К недостаткам относятся: неустойчивость к шумам во входных данных, сложный поиск оптимального дерева решений, возможность переобучения.

В нашей работе метод представлен классом sklearn.tree.DecisionTreeClassifier

**Случайный лес**

Метод случайного леса - алгоритм [машинного обучения](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), в котором используется ансамбль деревьев решений. Алгоритм случайного леса применяется преимущественно в задачах регрессии, классификации и кластеризации. Основная идея метода заключается в использовании ансамбля деревьев решений. Каждое из деревьев относит объект к одному из классов. В итоге выбирается класс, за который проголосовало наибольшее количество деревьев. Обычно число деревьев в ансамбле подбирается так, чтобы ошибка на тестовой выборке была минимальной. К преимуществам метода можно отнести низкую вероятность переобучения (в отличие от дерева решений). К недостаткам метода относятся затраты на обучение, так как приходится обучать множество деревьев решений.

В нашей работе этот метод не представлен, но его можно найти в sklearn.tree.DecisionTreeClassifier

**Искусственные нейронные сети**

Нейронная сеть состоит из нейронов, которые расположены слоями, они принимают некоторый входной вектор и преобразуют его в выходной. Процесс заключается в передаче каждому нейрону входных данных. Нейрон применяет к ним нелинейную функцию, а затем передает выходные данные на следующий уровень.

В представленной работе в качестве нейронной сети использовался многослойный перцептрон. Многослойный персептрон (MLP) — это алгоритм обучения с учителем, который применяет функцию к набору данных, где m — число измерений для ввода и n – число измерений для вывода. Между входным и выходным слоями может быть один или несколько нелинейных слоев, называемых скрытыми слоями. Каждый нейрон в скрытом слое преобразует значения из предыдущего слоя, находя скалярное произведение и применяя нелинейную функцию активации. Выходной слой получает значения из последнего скрытого слоя и преобразует их в выходные значения. К преимуществам многослойного перцептрона является его универсальность. К недостаткам можно отнести излишнюю сложность получаемой модели, что может значительно увеличить время обучения.

В нашей работе MLP представлен классом sklearn.neural\_network.MLPClassifier

**Метод опорных векторов**

Метод опорных векторов - метод машинного обучения, применяемый для задачи классификации. Метод основан на построении гиперплоскости, разделяющей входные данные на 2 класса. Для этого обучающая выборка отображается в многомерное пространство, после чего используется регрессия. На выходе работы алгоритма получается гиперплоскость, разделяющая два класса входных данных наилучшим образом. После этапа обучения принадлежность элемента к классу определяется по его положению относительно построенной гиперплоскости. Преимуществом метода является максимизация разделяющей гиперплоскости, так как она позволяет уменьшить количество ошибок классификации. Недостатками являются большое время обучения и неустойчивость к шуму.

В нашей работе метод представлен классом sklearn.svm.LinearSVC

**СРАВНЕНИЕ МЕТОДОВ КЛАССИФИКАЦИИ**

Для проведения исследования была написана программа, которая парсит комментарии к местам из сервиса Google Maps. В ходе выполнения работы было выяснено, что тематика мест должна быть схожа друг с другом. Это связано с тем, что люди используют разные слова и выражения для описания локаций. К примеру, посетители магазина будут хвалить низкие цены, чистоту, тогда как посетители театра будут оценивать качество постановки, декорации, костюмы. Исходя из этих наблюдений, была выбрана одна тематика – театры.

Для выполнения обучения был произведен парсинг 1500 комментариев, 750 из которых позитивные, 750 негативные. Для выполнения тестирования использовалась выборка из 500 комментариев, из которых также половина позитивных, половина негативных. В выборке были повторяющиеся комментарии. Результаты обучения представлены в таблице 1:

Таблица 1. Результаты обучения

|  |  |  |
| --- | --- | --- |
| Название метода | Время обучения, сек | Точность на обучающей выборке, % |
| Наивный байесовский классификатор | 0.006023 | 78.2 |
| Стохастический градиентный спуск | 0.025934 | 79.2 |
| K-ближайших соседей | 0.001042 | 67.8 |
| Дерево решений | 0.071842 | 71 |
| Искусственные нейронные сети (MLP) | 5.207116 | 76.8 |
| Метод опорных векторов | 0.031917 | 80.6 |

Исходя из результатов обучения, можно сказать, что для поставленной задачи наилучшую точность классификации показал метод опорных векторов. Наименьшая точность классификации была получена с помощью метода К-ближайших соседей. Стоит отметить, что метод многослойный перцептрон, несмотря на долгое время обучения, показал “средние” результаты. Скорее всего, это связано с небольшим объемом данных для обучения.

**ВЫВОД**

В результате проведения исследования было выяснено, что для поставленной задачи наиболее эффективным является метод опорных ветров. Однако, этот метод не получится применить при использовании другой классификации комментариев. Например, если разделить комментарии на 3 класса: “Позитивные”, “Нейтральные”, “Отрицательные”. Также, возможны проблемы при обучении методами, которые требуют большого количества данных. Это, в большей степени, связано с небольшим количеством отрицательных комментариев о театрах в Google Maps. Для формирования корпуса комментариев в работе были проанализированы комментарии к 10 крупнейшим театрам Новосибирска, а также к 9 театрам Москвы. Парсинг нескольких тысяч отрицательных комментариев может вызвать затруднения.

###### СПИСОК ЛИТЕРАТУРЫ

1. Naive Bayes // <https://scikit-learn.org/stable/modules/naive_bayes.html> (дата обращения: 03.04.2024).
2. Градиентный спуск // <https://ru.wikipedia.org/wiki/Градиентный_спуск> (дата обращения: 03.04.2024).
3. Stochastic Gradient Descent // <https://scikit-learn.org/stable/modules/sgd.html> (дата обращения: 03.04.2024).
4. Машинное обучение и Data Science (Часть 06): Градиентный спуск // <https://www.mql5.com/ru/articles/11200#:~:text=Согласно.%20Википедии%2C%20градиентный%20спуск%20—,к%20локальному%20максимуму%20этой%20функции> (дата обращения: 04.04.2024).
5. k-nearest neighbors algorithm // <https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm> (дата обращения: 05.04.2024).
6. sklearn.neighbors.KNeighborsClassifier // <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html> (дата обращения: 05.04.2024).
7. Деревья решений // <https://scikit-learn.ru/1-10-decision-trees/> (дата обращения: 05.04.2024).
8. Дерево решений // <https://ru.wikipedia.org/wiki/Дерево_решений> (дата обращения: 05.04.2024).
9. Метод случайного леса // <https://ru.wikipedia.org/wiki/Метод_случайного_леса> (дата обращения: 06.04.2024).
10. Ансамблевые методы // <https://scikit-learn.ru/1-11-ensemble-methods/> (дата обращения: 06.04.2024).
11. Neural network models (supervised) // <https://scikit-learn.org/stable/modules/neural_networks_supervised.html> (дата обращения: 04.04.2024).
12. Support Vector Machines // <https://scikit-learn.org/stable/modules/svm.html> (дата обращения: 04.04.2024).
13. Метод опорных векторов // <https://ru.wikipedia.org/wiki/Метод_опорных_векторов> (дата обращения: 04.04.2024).
14. The Selenium Browser Automation Project // <https://www.selenium.dev/documentation/> (дата обращения: 03.04.2024).
15. XPath Tutorial // [XPath Tutorial (w3schools.com)](https://www.w3schools.com/xml/xpath_intro.asp) (дата обращения: 03.04.2024).

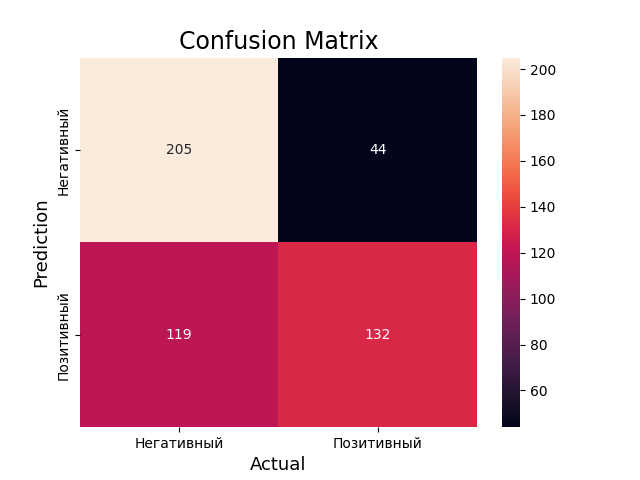
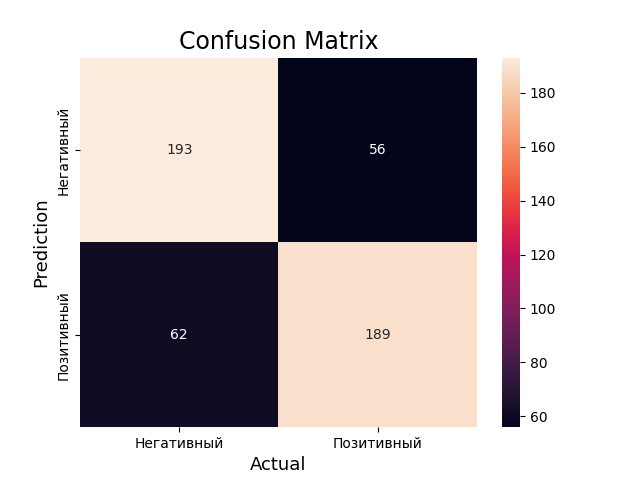
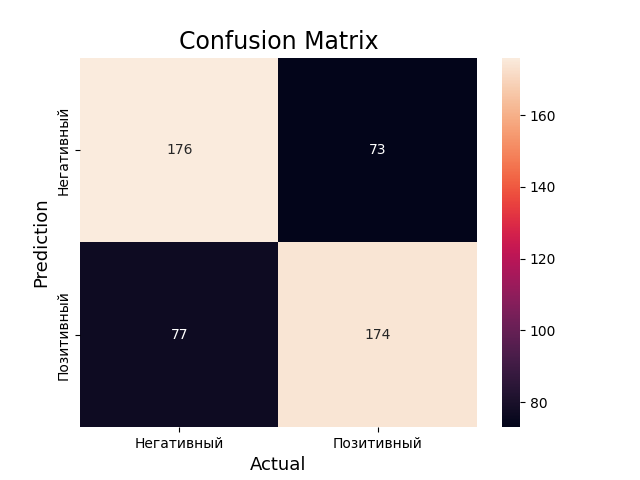
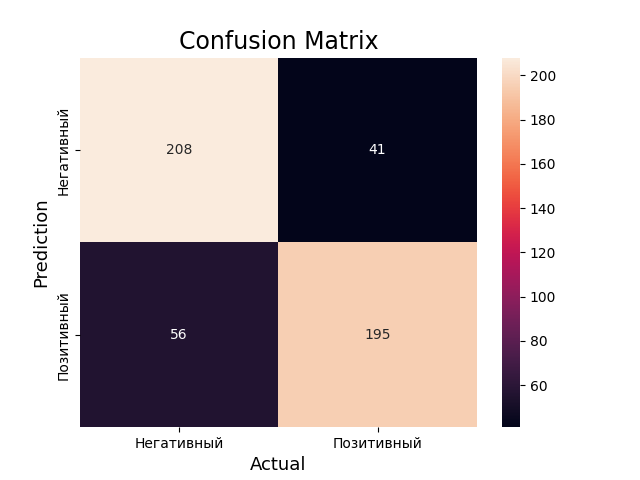
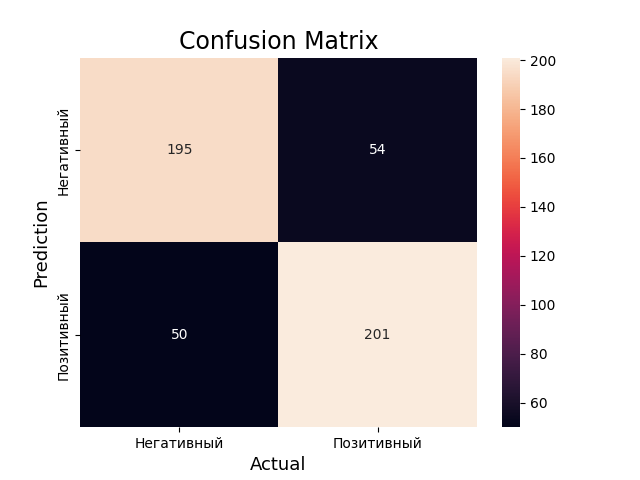
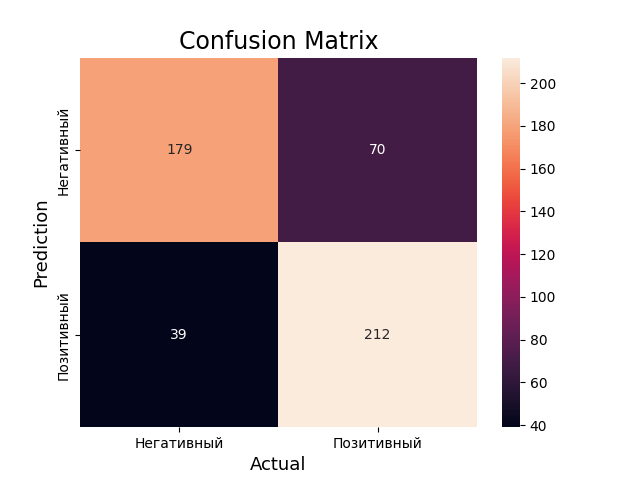
******Приложение 1. Матрицы ошибок**

Рис. 5. Матрица ошибок для метода многослойного перцептрона

Рис. 4. Матрица ошибок для метода дерева решений

Рис. 2. Матрица ошибок для стохастического градиентного спуска

Рис. 1. Матрица ошибок для метода наивного байеса

Рис. 3. Матрица ошибок для метода k-ближайших соседей

Рис. 6. Матрица ошибок для метода опорных векторов