МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра прикладной математики**

**КУРСОВАЯ РАБОТА**

**КЛАССИФИКАЦИЯ ТЕКСТОВ С ПОМОЩЬЮ НАИВНОГО БАЙЕСОВСКОГО КЛАССИФИКАТОРА**

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.Д. Шестак

(подпись)

Направление подготовки 01.03.02 «Прикладная математика и информатика»

Направленность (профиль) «Системный анализ, исследование операций и управление» (Математическое и информационное обеспечение экономической деятельности)

Научный руководитель

канд. физ.-мат. наук, доц. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.М. Кармазин

(подпись)

Нормоконтролер

канд. физ.-мат. наук, доц. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Г.В. Калайдина

(подпись)

Краснодар

2018

# РЕФЕРАТ

Работа содержит 31 страница, 5 рисунков, 2 таблицы, 6 источников.

НАИВНЫЙ БАЙЕСОВСКИЙ КЛАССИФИКАТОР, КЛАССИФИКАЦИЯ, МАШИННОЕ ОБУЧЕНИЕ, АНАЛИЗ ЭМОЦИОНАЛЬНОЙ ОКРАСКИ.

Объектом исследования является наивный байесовский классификатор, и его применение для задачи многоклассовой классификации текстов.

Цель работы: изучение задачи многоклассовой классификации текстов и реализация прикладных средств для её решения. В ходе работы были получены знания о работе наивного байесовского метода классификации и основных принципах машинного обучения. В результате была изучена задача многоклассовой классификации и написана программа, демонстрирующая работу алгоритма решения этой задачи.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc516320551)

[1 Теоретические основы анализа текста 6](#_Toc516320552)

[1.1 Краткие сведения из теории вероятности 6](#_Toc516320553)

[1.2 Краткие сведения из теории обработки естественных языков 7](#_Toc516320553)

[1.3 Краткие сведения из теории машинного обучения](#_Toc516320554) 13

[2 Постановка задачи](#_Toc516320556) 21

[2.1 Задача определения тональности текста 22](#_Toc516320557)

[3 Инструменты программной реализации](#_Toc516320570) 23

[3.1 Язык программирования Python](#_Toc516320571) 23

[3.2 Библиотека Pandas 2](#_Toc516320572)4

[3.3 Библиотека NumPy 2](#_Toc516320572)5

[3.4 Библиотека joblib 2](#_Toc516320572)5

[3.5 Библиотека NLTK 2](#_Toc516320572)6

[3.6 Библиотека sci-kit learn 2](#_Toc516320572)6

[4 Источники данных 2](#_Toc516320570)7

[5 Решение задачи и результаты работы 2](#_Toc516320570)8

[5.1 План решения 2](#_Toc516320571)8

[5.2 Результаты 2](#_Toc516320571)8

[ЗАКЛЮЧЕНИЕ 30](#_Toc516320573)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 31](#_Toc516320574)

# **ВВЕДЕНИЕ**

С развитием интернет-сервисов каждый пользователь получил, в числе прочих, возможность выражать своё мнение относительно товара или услуги, фильма или книги, компании или политического деятеля. Естественным образом появилась возможность обрабатывать такую информацию для определения отношения пользователей к тому или иному объекту. Объёмы информации в интернете делают невозможной процедуру обработки данных человеком, поэтому возникла потребность в программной реализации. Это дало толчок для развития такой области Computer Science, как Sentiment Analysis. Применяя методы Machine Learning. Вместе с этим стало возможным получить мнение интернет-пользователей по интересующей теме значительно быстрее с помощью компьютера, нежели чем при использовании человеческих ресурсов.

Анализ тональности (англ. Sentiment Analysis) – класс методов анализа содержимого, предназначенный для автоматизированного выявления в текстах эмоционально окрашенной лексики и эмоциональной оценки авторов по отношению к объектам, речь о которых идёт в тексте.

Задача анализа тональности или определения тональности текста является задачей классификации текстов в широком смысле, где под классами понимаются мнения, высказанные пользователями.

В данной курсовой работе будет рассмотрена программная реализация многоклассового классификатора на основе наивного байесовского метода. Особенностью такого вида классификатора является разделение мнений не на «хорошо» и «плохо», а на более полную систему, позволяющую иметь широкое представление о том, какого мнения придерживаются пользователи относительно рассматриваемого вопроса.

Таким образом, в работе рассматривается задача определения тональности текста на русском языке и её решение с помощью наивного байесовского классификатора с использованием bag-of-word-модели представления документов\сообщений.

Работа с Machine Learning подразумевает одно из двух направлений: обучение с учителем или самообучение. В данной работе рассматривается первое направление, в связи с чем появляется потребность в размеченных данных для обучения алгоритма. Одной из основных проблем для применения таких ресурсов является поиск или сбор размеченных данных для обучения алгоритмов. Ещё одна проблема связана с тем, что таких данных на русском языке ещё меньше. Но в связи с тем, что актуальность задачи растёт с каждым днём и всё больше и больше компаний заинтересованы в таких ресурсах, такие данные всё же существуют и корпус коротких текстовых сообщений, основанный на постах в социальной сети «Твиттер» предоставляет открытый доступ к ним. Эти данные и будут использоваться в работе.

Цель курсовой работы – изучение проблемы классификации текстов и средств программной реализации поставленной задачи. Итог проделанной работы – были изучены и протестированы средства программной реализации наивного байесовского классификатора.

## **1 Теоретические основы анализа текста**

В данной работе используются аппараты теории вероятности, теории обработки естественных языков и машинного обучения, поэтому в главе 1 даны краткие теоретические сведения из упомянутых дисциплин.

### Краткие сведения из теории вероятности

Вероятностью *P* случайного события *A* называется отношение числа *n* несовместимых равновероятных элементарных событий, составляющих событие *A*, к числу всех возможных элементарных событий *N*:

 (1)

Условной вероятностью события *А* при условии, что произошло событие *В*, называется число

, (2)

где

 – вероятность произведения двух событий.

Полной системой событий называют не более, чем счетное множество событий таких, что:

а) Все события попарно несовместны: ;

б) Объединение этих событий образует пространство элементарных исходов: .

Вероятность события , которое может произойти только вместе с одним из событий , образующих полную систему событий, равна сумме произведений вероятностей гипотез на условные вероятности события, вычисленные соответственно при каждой из гипотез:

 (3)

Формула Байеса

 (4)

где

 – вероятность наступления события *А*;

 – вероятность события *А* при наступлении события *B* (2);

 – вероятность события *В* при наступлении события *А* (2);

 – полная вероятность наступления события *B* (3).

* 1. **Краткие сведения из теории обработки естественных языков**

Обработка естественного языка (англ. Natural Language Processing, NLP) – общее направление искусственного интеллекта и математической лингвистики. Оно изучает проблемы компьютерного анализа и синтеза естественных языков.

Анализ тональности текста (сентимент-анализ, англ. Sentiment analysis, англ. Opinion mining) – класс методов контент-анализа в компьютерной лингвистике, предназначенный для автоматизированного выявления в текстах эмоционально окрашенной лексики и эмоциональной оценки авторов (мнений) по отношению к объектам, речь о которых идёт в тексте.

Тональность – это эмоциональное отношение автора высказывания к некоторому объекту (объекту реального мира, событию, процессу или их свойствам/атрибутам), выраженное в тексте. Эмоциональная составляющая, выраженная на уровне лексемы или коммуникативного фрагмента, называется лексической тональностью (или лексическим сентиментом). Тональность всего текста в целом можно определить, как функцию (в простейшем случае сумму) лексических тональностей составляющих его единиц (предложений) и правил их сочетания.

Тональность документа можно определять по бинарной шкале. В этом случае для определения тональности документа используется два класса оценок: позитивная или негативная. Одним из минусов данного подхода является то, что эмоциональную составляющую документа не всегда можно однозначно определить, т.е. документ может содержать признаки как позитивной, так и негативной оценки. Здесь на помощь приходит многоклассовая классификация. Но она может также и порождать новые проблемы. Применяются различные методы распознавания полярности обзоров товара и отзывов о фильмах соответственно. Это пример работы на уровне документа.

Выделяют следующие методы классификации тональности текста:

- методы, основанные на правилах и словарях;

- машинное обучение с учителем;

- машинное обучение без учителя;

- метод, основанный на теоретико-графовых моделях.

В данной работе рассматривается метод машинного обучения с учителем. Сутью данного метода является то, что на первом этапе обучается машинный классификатор (например, байесовский) на заранее размеченных текстах, а затем используют полученную модель при анализе новых документов.

Краткий алгоритм:

1) Собирается коллекция документов, на основе которой обучается машинный классификатор;

2) Каждый документ раскладывается в виде вектора признаков (аспектов), по которым он будет исследоваться;

3) Указывается правильный тип тональности для каждого документа;

4) Производится выбор алгоритма классификации и метод для обучения классификатора;

5) Полученную модель используем для определения тональности документов новой коллекции.

Как правило, построение коллекций документов представляет собой весьма затратную по памяти задачу. В связи с этим, очень часто используют методы уменьшения размерности задачи. Данные методы применяются для уменьшения размерности самого документа, что влечет за собой уменьшение и всей хранимой коллекции.

Также, может случится ситуация, когда слова, наиболее часто встречающиеся во множестве слов одного из классов словаря, синтаксически различаются. Для задачи анализа тональности текста разница в синтаксисе не влияет на принятие решений, поскольку важна лишь оценочная семантика слова. Такими словами, могут быть, например, слова *плохо, плохое, плохая* и т.д. Эти слова семантически одинаковые, а синтаксически разные. Такое различие в синтаксисе приводит к тому, что подсчет веса для этих слов будет производится раздельно, что будет оказывать влияние на точность модели. Чтобы исправить данную ситуацию, применяются методы обработки естественных языков для приведения словоформ в каноническую форму. Для рассматриваемой задачи такими методами могут быть:

1) Стемминг – это процесс нахождения основы слова для заданного исходного слова. Основа слова не обязательно совпадает с морфологическим корнем слова. Например: *кровать* *крова*;

2) Лемматизация – процесс приведения словоформы к лемме – её нормальной (словарной) форме. Например: *стояли* *стоять*, *раскидистые* *раскидистый*;

3) Удаление стоп-слов – удаление из текста документа слов, которые самостоятельно не несут никакой смысловой нагрузки. Например: местоимения *я, ты, мы, вы*;

4) Удаление пунктуационных символов.

Текстовое содержание документов нужно как-то представлять для удобства обработки. Для этого в данной работе используется модель «Мешок слов» (англ. Bag-of-words model) и ее обобщение на комбинации термов.

Модель «Мешок слов» – это упрощенное представления текстовой информации, используемое в задачах обработки естественных языков и поиска информации. В этой модели текст представляется в виде мешка (мультимножества) его слов или словосочетаний в случае комбинаций термов, игнорируя грамматику и в некоторых случаях даже порядок слов, но сохраняя множественность. Каждому такому элементу (слову или словосочетанию) ставится в соответствие некоторый вес . В этом случае документ определяется вектором:

, (5)

где  – множество всех уникальных термов (слов\словосочетаний) из обучающей выборки;

 – вес i-го терма.

Возможны следующие варианты весов [2]:

1. Булевский вес:

=

1. Количество вхождений i-го терма в документе:

;

1. Частота терма (*TF – term frequency*):

, (6)

где

– число вхождений i-го терма;

 – общее количество термов документа.

1. TF-IDF (*IDF – inverse document frequency – обратная частота документа*):

, (7)

, (8)

где

 – количество документов в корпусе;

 – количество документов, в которых встречается ;

 – частота терма.

Модель «Мешок слов», как правило, используется в методах классификации документов, где частота появления каждого слова используется как функция для обучения классификатора.

Например, пусть дан следующий текст:

*«Петр любит решать квадратные уравнения. Марк тоже любит решать квадратные уравнения. Но никто из них не воспринимает кубические уравнения.»*

Тогда вид мешка слов для этого текста в случае, когда элемент – слово, а вес – количество вхождений слова в документ, будет выглядеть следующим образом (таблица 1):

Таблица 1 – Пример модели мешка слов для текста

|  |  |
| --- | --- |
| Слово | Количество |
| *«Петр»* | 1 |
| *«Марк»* | 1 |
| *«любит»* | 2 |
| *«решать»* | 2 |
| *«квадратные»* | 2 |
| *«уравнения»* | 3 |
| *«тоже»* | 1 |
| *«но»* | 1 |
| *«никто»* | 1 |
| *«из»* | 1 |
| *«них»* | 1 |
| *«не»* | 1 |
| *«воспринимает»* | 1 |
| *«кубические»* | 1 |

Эта модель легко обобщается на комбинации термов.

N-грамма – последовательность из  элементов. Вообще говоря, с семантической точки зрения, это может быть последовательность звуков, слогов, слов или букв, однако на практике чаще встречается N-грамма как ряд слов. Например, текст, предложенный выше в качестве примера, в случае использования 3-грамм, разбивается на следующую последовательность:

*«Петр любит решать», «любит решать квадратные», «решать квадратные уравнения», «квадратные уравнения Марк», …, «воспринимает кубические уравнения».*

В этом случае мешок слов в случае использования количества вхождений 3-граммы в документ в качестве веса, будет выглядеть следующим образом (таблица 2):

Таблица 2 – Пример модели мешка слов для текста, представленного с помощью 3-грамм

|  |  |
| --- | --- |
| 3-грамма | Количество |
| *«Петр любит решать»* | 1 |
| *«любит решать квадратные»* | 2 |
| *«решать квадратные уравнения»* | 2 |
| *«квадратные уравнения Марк»* | 1 |
| *«уравнения Марк тоже»* | 1 |
| *«Марк тоже любит»* | 1 |
| *«тоже любит решать»* | 1 |
| *«квадратные уравнения Но»* | 1 |
| *«уравнения Но никто»* | 1 |
| *«Но никто из»* | 1 |
| *«никто из них»* | 1 |
| *«из них не»* | 1 |
| *«них не воспринимает»* | 1 |
| *«не воспринимает кубические»* | 1 |
| *«воспринимает кубические уравнения»* | 1 |

Таким образом, документ можно определить как вектор d, определяемый равенством (5).

* 1. **Краткие сведения из теории машинного обучения**

Классификация документов – одна из задач информационного поиска (раздел машинного обучения), заключающаяся в отнесении документа к одной из нескольких категорий на основании содержания документа.

Бинарная классификация документов – это разновидность задачи классификации документов, у которой число категорий равно 2.

Многоклассовая классификация документов в общем случае не реализуется. Она сводится к последовательности бинарных решений. Это означает, что методы, которые, на первый взгляд, работают только для бинарных данных, можно без особых усилий обобщить на многоклассовые данные.

Классификатор – это алгоритм, соотносящий некие входные данные с одним или несколькими классами. В отличие от алгоритмов кластеризации эти классы должны быть определены заранее.

Выборка – конечный набор прецедентов (объектов, случаев, событий, испытуемых, образцов, и т.п.), некоторым способом выбранных из множества всех возможных прецедентов, называемого генеральной совокупностью.

Выборка называется случайной, если вероятностная модель порождения данных предполагает, что выборка из генеральной совокупности формируется случайным образом. Объем (длина) выборки , считается произвольной, но фиксированной, неслучайной величиной. Формально это означает, что с генеральной совокупностью , связывается вероятностное пространство , где  – множество всех выборок длины ,  – заданная на этом множестве сигма-алгебра событий,  – вероятностная мера. Тогда, случайная выборка  – это последовательность из прецедентов, выбранная из множества  согласно вероятностной мере .

Выборка называется однородной, если все ее прецеденты  одинаково распределены, то есть выбраны из одно и того же распределения .

Выборка называется независимой, если вероятностная мера на  представима в виде произведения  вероятностных мер на , то есть для любой системы подмножеств . Если на  существует плотность распределения , то независимость означает, что -мерная плотность распределения на  представима в виде произведения  одномерных плотностей, то есть:

 (9)

Выборка называется простой, если данная выборка является случайной, однородной и независимой.

Как правило, для задачи классификации, генеральная совокупность  разбивается на несколько выборок: обучающая выборка, тестовая выборка и проверочная.

Обучающая выборка – это выборка, по которой производится настройка (оптимизация параметров) модели зависимости.

Если модель зависимости построена по обучающей выборке , то оценка качества этой модели, сделанная на той же выборке , оказывается, как правило, оптимистически смещенной. Это нежелательное явление называется переобучением. Поэтому возникает необходимость в построении тестовой выборки.

Тестовая выборка – это выборка, по которой оценивается качество построенной модели.

Если обучающая и тестовая выборки независимы, то оценка, сделанная по тестовой выборке, является несмещенной. Оценку качества, сделанную по тестовой выборке, можно применить для выбора наилучшей модели. Однако тогда она снова окажется оптимистически смещённой. Для получения несмещённой оценки выбранной модели приходится выделять третью выборку – проверочную.

Проверочная выборка – это выборка, по которой осуществляется выбор наилучшей модели из множества моделей, построенных по обучающей выборке.

В общем виде, формально, задача классификации документов, являясь задачей классификации текстовой информации, определяется следующим образом [2]:

Пусть существует описание документа , где  – векторное пространство документов, и фиксированный набор классов . Из обучающей выборки  с помощью метода обучения  необходимо получить классифицирующую функцию (или классификатор) , которая отображает документы в классы .

Для задачи, рассматриваемой в данной работе множество  состоит из трёх элементов .

Для определения качества работы классификатора введем меру оценки этого качества. Пусть:

-  – количество истинно-положительных результатов;

-  – количество истинно-отрицательных результатов;

-  – количество ложно-положительных результатов;

-  – количество ложно-отрицательных результатов.

Тогда на основе этих значений определим меры точности (*precision*)и полноты (*recall*):

(*Precision*)  (10)

(*Recall*)  (11)

Смысл этих мер заключается в следующем: точность – это доля результатов, которая действительно принадлежит данному классу, а полнота – процент найденных результатов от их общего числа.

Если  или , то ценность классификатора падает. Поэтому для усреднения обоих значений определяется -мера, как гармоническое среднее точности и полноты:

 (12)

При этом, согласно исследованию, эксперты обычно соглашаются в оценках тональности конкретного текста в 79 % случаев [6] . Следовательно, программа, которая определяет тональность текста с точностью 70 %, делает это почти так же хорошо, как и человек.

В рамках данной работы был реализован наивный байесовский классификатор, основанный на (4). В качестве результирующего класса выбирается класс с максимальной апостериорной вероятностью:

, (13)

по формуле (4)

, (13.1)

так как плотности распределения чаще всего неизвестны, производится их оценка по обучающей выборке, при этом оценка вероятности документа в обучающей выборке , потому что ищется аргумент, максимизирующий функцию правдоподобия, от которого полная вероятность (3)  не зависит, то

 (13.2)

в силу особенности представления в памяти ЭВМ чисел с плавающей точкой при работе с числами, близкими к нулю, которые возникнут при вычислении , появляется сильная потеря точности. Во избежание этого, так как логарифм монотонно возрастает , и максимум функции  будет идентичен максимуму функции  получаем

 (13.3)

поскольку наивный байесовский классификатор принимает допущения, что порядок следования признаков объекта не имеет значения и что вероятности признаков не зависят друг от друга при данном классе, а значит , где , то есть

, (13.4)

, (13.5)

где вероятностные оценки определяются следующим образом:

, (14)

, (15)

где

 – количество вхождений i-го элемента в документах класса *с*;

 – количество документов  класса *с.*

Графическая схема наивного байесовского классификатора представлена на рисунке 1.

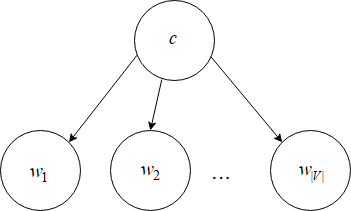


Рисунок 1 – Графическая схема наивного байесовского классификатора

У данной модели классификатора есть одна проблема: пусть на вход алгоритма подается новый документ , в котором есть элемент, который еще не содержится в мешке слов. Пусть вес этого элемента равен . Тогда очевидно, что , а значит , что влечет за собой неизбежные проблемы при программной реализации алгоритма. Избежать этой ситуации позволяет размытие по Лапласу (*k-additive smoothing*). Данный метод позволяет делать размытие при вычислении вероятностей категориальных данных. Для предложенной модели формула размытия по Лапласу выглядит следующим образом:

, (16)

где

 – количество вхождений i-го элемента в документах класса *с*;

 – коэффициент размытия.

С байесовской точки зрения, данный метод соответствует математическому ожиданию апостериорного распределения, используя в качестве априорного распределения распределение Дирихле, параметризируемое параметром .

Достоинствами наивного байесовского классификатора являются:

- простота реализации;

- быстрый процесс обучения. Сложность обучения ;

- высокие результаты при текстовой классификации [21].

Недостатки:

- значения, возвращаемые при классификации, нельзя трактовать, как вероятности, то есть нельзя ответить на вопрос, с какой долей уверенности получился результирующий класс.

# **2 Постановка задачи**

Задача анализа или определения тональности текста, являясь задачей текстовой классификации, является задачей фильтрации спама в широком смысле, где под спамом понимается негативно окрашенную тональность текста, а под неспамом – позитивно окрашенную тональность документа. В узком смысле под задачей фильтрации спама понимают задачу фильтрации почтового спама. В связи с этим в дальнейшем задачу фильтрации спама в широком смысле будем называть задачей определения тональности текста, а задачу фильтрации в узком смысле – задачей фильтрации почтового спама.

Задача анализа тональности или определения тональности текста является задачей классификации текстов в широком смысле, где под классами понимаются разбитое на подмножества множество мнений, высказанных пользователями. Разбиение может производиться самыми разнообразными способами и ограничено лишь возможностью генерации новых разбиений или составлении обучающей выборки по выбранному разбиению. Так, например, для задачи фильтрации спама используется бинарное разбиение на spam и span. Которые являются следствием разбиения на pos и neg. Но программная реализация от этого не пострадает, так как её можно свести к последовательности бинарных решений (рис. 2).

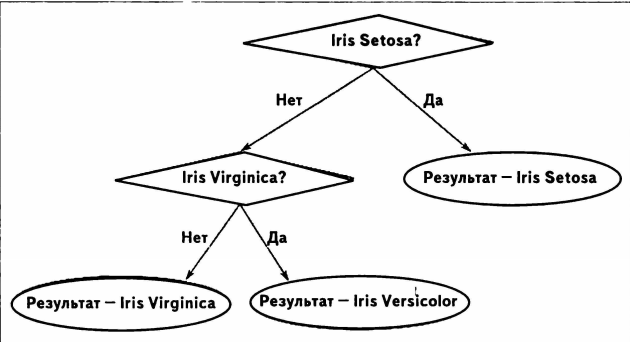


Рисунок 2 – Наглядная демонстрация приведения задачи многоклассовой классификации к последовательности бинарных решений

### 2.1 Задача определения тональности текста

Выше было показано, что решение задачи многоклассовой классификации может быть сведено к последовательному решению бинарных задач, поэтому, ввиду невозможности рассмотрения всевозможных вариантов разбиения, в работе будет показан способ решения задачи классификации на 3 класса с помощью последовательности бинарных решений.

Пусть существует описание документа , где  – векторное пространство документов, и фиксированный набор классов . Из обучающей выборки  с помощью метода наивного Байеса необходимо построить классификатор , отображающий документы в классы .

1. **Инструменты программной реализации**

В данной главе даны краткие сведения об инструментах программной реализации, используемых для решения поставленной задачи.

* 1. **Язык программирования Python**

Язык программирования Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Данный язык поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и так далее. Эталонной реализацией Python является интерпретатор CPython, поддерживающий большинство активно используемых платформ. Он распространяется под свободной лицензией Python Software Foundation License, позволяющей использовать его без ограничений в любых приложениях, включая проприетарные.

Одной из причин, почему Python прекрасно подходит для задач машинного обучения, математических задач и прочих научных расчетов, является то, что Python – это удобное и выразительное средство связи между читабельным кодом и быстрыми библиотеками, написанными на С\C++.

Другая причина заключается в том, что на данный момент существует много реализаций различных методов машинного обучения, соответствующих библиотек и прочих вспомогательных инструментов таких как, например, пакет для научных вычислений NumPy, пакет для работы с данными Pandas и так далее.

* 1. **Библиотека Pandas**

Pandas – программная библиотека на языке Python для обработки и анализа данных. Работа Pandas с данными строится поверх библиотеки NumPy, являющейся инструментом более низкого уровня. pandas предоставляет специальные структуры данных и операции для манипулирования числовыми таблицами и временными рядами. Название библиотеки происходит от эконометрического термина «панельные данные», используемого для описания многомерных структурированных наборов информации. Pandas распространяется под новой лицензией BSD.

Несмотря на то, что Python в течение долгого времени успешно применяется для очистки и подготовки данных, его использование при проведении анализа и моделирования затруднено. Зачастую требуется выполнять часть задач при помощи инструментов, отражающих специфику предметной области, таких как язык R. Возможности Pandas восполняют описанный недостаток, позволяя полностью проводить все этапы анализа информации на Python.

Возможности библиотеки, используемые в данной работе:

- объект DataFrame для манипулирования индексированными массивами двумерных данных;

- инструменты для обмена данными между структурами в памяти и файлами различных форматов;

- встроенные средства совмещения данных и способы обработки отсутствующей информации;

- слияние и объединение наборов данных.

Библиотека оптимизирована для высокой производительности, наиболее важные части кода написаны на Cython и C.

3.3 **Библиотека NumPy**

NumPy - библиотека с открытым исходным кодом для Python, предоставляющая поддержку многомерных массивов и высокоуровневых математических функций, предназначенных для работы с многомерными массивами.

Математические алгоритмы, реализованные на чистом Python, часто работают гораздо медленнее тех же алгоритмов, реализованных на компилируемых языках. NumPy предоставляет реализации вычислительных алгоритмов (в виде функций и операторов), оптимизированные для работы с многомерными массивами. В результате любой алгоритм, который может быть выражен в виде последовательности операций над массивами (матрицами) и реализованный с использованием NumPy, работает так же быстро, как эквивалентный код, выполняемый в MATLAB [20].

3.4 **Библиотека joblib**

Библиотека joblib – набор инструментов для облегчения конвейерной обработки с помощью языка Python.

Библиотека предлагает:

- прозрачное кэширование выходных значений и ленивую переоценку (memorize pattern);

- простоту параллельных вычислений;

- протоколирование и отслеживание выполнения.

Библиотека оптимизирована для быстроты и надежности работы, в частности, при работе с большими данными, имея специальную оптимизацию для массивов из библиотеки NumPy. Распространяется по BSD лицензии.

**3.5 Библиотека NLTK**

Библиотека NLTK, или NLTK (англ. Natural Language Toolkit*)*, – пакет библиотек и программ для символьной и статистической обработки естественного языка, написанных на языке программирования Python. Содержит графические представления и примеры данных. Сопровождается обширной документацией, включая книгу с объяснением основных концепций, стоящих за теми задачами обработки естественного языка, которые можно выполнять с помощью данного пакета.

NLTK хорошо подходит для задач компьютерной и эмпирической лингвистики, когнитивистики, искусственного интеллекта, информационного поиска и машинного обучения.

В данной работе рассмотрены средства этой библиотеки для предобработки исходного текста: стемминг, лемматизация, корпус стоп-слов и так далее.

**3.6 Библиотека sci-kit learn**

Библиотека sci-kit learn – это свободно распространяемая библиотека для машинного обучения для языка Python. Она содержит в себе множество алгоритмов классификации, регрессии и кластеризации, такие как машины опорных векторов, случайный лес, градиентный бустинг, метод k-средних, метод наивной байесовской классификации и т.д.

Библиотека специально разработана для использования вычислительных и научных библиотек NumPy и SciPy.

**4 Источники данных**

Область применения моделей, разрабатываемых для решения задач фильтрации спама в широком и узком смыслах, напрямую зависят от данных, на которых данные модели разрабатываются.

Для задачи анализа тональности текста для текстов на русском языке был взят корпус, подготовленный с помощью программного интерфейса API Twitter [24]. Распределение количества твиттов по классам в данном корпусе представлено на рисунке 3. Данные, содержащиеся в этих корпусах, предоставляются в сыром виде и подлежат дальнейшей предобработке для возможного улучшения качества модели.

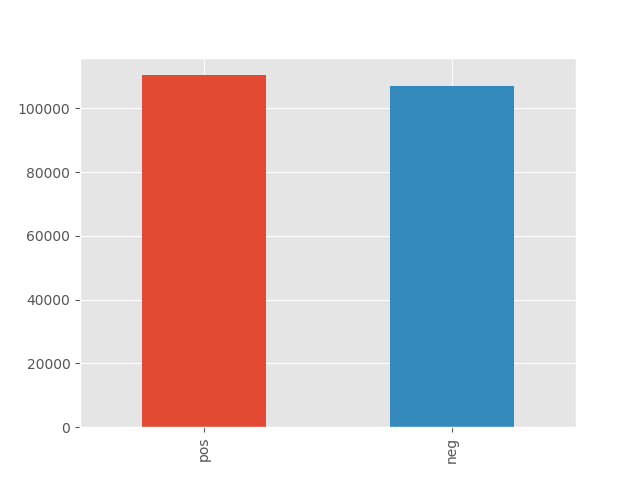


Рисунок 3 – Распределение количества твиттов по классам в наборе данных для русского языка

# **6. Решение задачи и результаты работы**

# **6.1 План решения**

На основе всех вышеизложенных выкладок получим краткий алгоритм действий:

а) Загружаем размеченные данные

б) Проводим обработку полученных данных

1) Удаление стоп-слов и знаков пунктуации

2) Лемматизация

3) Стемминг

в) Векторизуем полученные данные для дальнейшей обработки

г) Разбиваем данные на обучающие и тестовые

д) Обучаем классификатор

е) Применяем процедуру классификации для тестовых и обучающих данных, при этом проверяя результаты по заранее проставленным меткам

ж) Выводим результаты

Рассмотрим пункт е) более детально, а также продемонстрируем работу программы, задавая различные способы разбиения множества при задании последовательности бинарных решений классификации. Таким образом рассмотрим 3 подхода для задания разбиений:

А) На первом этапе классификации будем делить данные на pos и other, а на втором на neu и neg.

Б) Сначала делим на neg и other, а затем на neu и pos.

**6.2 Результаты работы**

Ниже представлены результаты работы классификатора, реализованного с помощью библиотек, описанных в тексте курсовой работы на размеченных данных, взятых из корпуса коротких ответов на русском языке[2].

На рисунке 4 первый коэффициент – это промежуточные вычисления точности разбиения при первом бинарном распределении данных. Общая точность байесовского метода это среднее арифметическое между точностью разбиения на первом и втором шагах.



Рисунок 4 – Результаты работы классификатора с предположением А)

На рисунке 5 первый коэффициент не изменился, что не удивительно, ведь входные данные на первом этапе не изменяются. А вот второй коэффициент указывает на то, что точность при таком подходе уменьшилась по сравнению с предыдущим вариантом.



Рисунок 5 – Результаты работы классификатора с предположением Б)

Исходя из того, что различие в общих точностях не велико, можно сделать вывод, что особых различий в работе классификатора при разных способах постановки бинарных разбиений не наблюдается.

# **ЗАКЛЮЧЕНИЕ**

В курсовой работе были рассмотрены основные теоретические основы машинного обучения, обработки естественных языков и эмоционального анализа текстов в целом. В результате получена программная реализация инструмента для решения поставленной задачи о многоклассовой классификации. Было показано, что реализованный алгоритм можно обобщить на случай большего числа классов. Тем самым была достигнута поставленная цель курсовой работы, реализован инструмент для решения прикладной задачи многоклассовой классификации текстов.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Луис Педро Коэльо, Вилли Ричарт Построение систем машинного обучения нa языке Python. 2-е издание/ пер. с англ. Слинкин А. А. - М.: ДМК Пресс, 2016. - 302 с.: ил.
2. Manning D. C., Raghavan R., Schütze H. An Introduction to Information Retrieval Draft. Online edition. Cambridge University Press. – 2009. – 544 pp.
3. Корпус коротких текстов на русском языке на основе постов Twitter. URL: <http://study.mokoron.com/> [14 Ноября 2018]
4. Анализ тональности высказываний в twitter. URL: <http://habr.com/company/infopulse/blog/261589/> [15 Октября 2018]
5. Making Sentimental Analysis Easy With Scikit-Learn. URL: [https://www.twilio.com/blog/2017/12/sentiment-analysis-scikit-learn.html [10](https://www.twilio.com/blog/2017/12/sentiment-analysis-scikit-learn.html%20%5b10) Декабря 2018]
6. How Companies Can Use Sentiment Analysis to Improve Their Business. URL: https://mashable.com/2010/04/19/sentiment-analysis/#NA7bCHMEw5qq (дата обращения 14.04.2018).