Московский государственный университет имени М. В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра алгоритмических языков

Васильева Надежда Андреевна

Модификация текста с целью изменения его тональности

Курсовая работа

Научный руководитель:

доцент, к.ф.-м.н. Волкова И.А

Москва 2023

Аннотация

Данная работа посвящена модификации текста с целью изменения его тональности. В работе изучаются различные методы машинного обучения, используемые для анализа тональности текста, их преимущества и недостатки. Для определения тональности текстов из датасета выбран наиболее оптимальный метод.

Затем реализовано определение тональности для отзыва введенного пользователем при помощи оптимального методы, а также функция, позволяющая изменять тональность текста, заменяя положительные тональные слова на их антонимы, что позволяет изменять тональность текста с положительной на отрицательную и наоборот.

Содержание

1. Постановка задачи

1. Изучить существующие подходы к анализу тональности, их преимущества и недостатки, выбрать подход для определения тональности текстов из датасета.
2. Выбрать наиболее оптимальный метод для определения тональности текстов. При помощи этого метода определять тональность введенного пользователем отзыва.
3. Изменить тональность введенного пользователем отзыва при помощи замены тональных слов на их антонимы, используя библиотеку nltk, а именно словаря WordNet, а также словаря положительных и отрицательных слов, собранного по датасету.
4. Провести эксперименты. Оценить результаты и точность работы алгоритма.

2. Введение

Анализ тональности текста — это подраздел обработки естественного языка (NLP), целью которого является классификация текста по тональности.

Тональность — это мнение, отношение и эмоции автора по отношению к объекту, о котором говорится в тексте. В качестве объектов могут выступать объекты реального мира, люди, события или процессы. Обычно используется бинарная классификация, выявление в тексте положительных и отрицательных оттенков. Но также может добавляться нейтральный класс или стоять более сложная задача, допустим выявление оценок или промежуточных оценок тональности, которые поставит пользователь: «Отлично», «Хорошо», «Не очень», «Плохо» и другие.

Анализ тональности текста имеет широкий спектр применений в современном мире. С его помощью можно выявлять отношение пользователей к продукту, применять данный анализ для политических, социологических, экономических, маркетинговых исследований, строить рекомендательные и обучающие системы.

В этой работе будет произведен не только анализ тональности текста при помощи методов машинного обучения, но и изменение степени его тональности.

Изменение степени тональности текста - перспективное направление для работы, так как в современном мире в сфере NLP уже сделано многое по анализу тональности, но пока не выявлено каким образом можно изменять яркость окрашенности текста.

3. Обзор существующих решений

Для решения данной задачи NLP предлагается использовать элементы машинного обучения. Ввиду этого здесь рассматриваются решения различных авторов с соответствующими алгоритмами.

В работе [1] рассмотрены различные подходы к анализу тональности текстов: анализ на словарях, методы машинного обучения, нейронные сети и метод на основе трансформеров. Также проанализированы достоинства и недостатки каждого из этих подходов и их точность на одних и тех же датасетах. Также рассмотрены методы аугментации и автоматической разметки данных в задаче анализа тональности, например модель RuBERT. Также для каждого из подходов дана статья с более подробной информацией о нем. В работе рассказано о факторах, которые усложняют анализ тональности, такие как многозначность лексики и зависимость значения слова от его тональности.

В работе [2] рассматривается задача определения тональности и то, насколько она сильна. Берется аннотированный корпус высказываний, в котором каждому высказыванию по тональности сопоставлено число из диапазона от -3 до 3, определяющее его тональность. Строятся 4 модели регрессии: одна является основной - предсказывает это число, а 3 другие дополнительные:

- основная задача (определение конкретного числа) + определение тональности как дополнительная
- основная + определение интенсивности
- основная + определение тональности + определение интенсивности.

Для основной задачи активационной функцией является гиперболический тангенс, для определения тональности – sigmoid - для определения интенсивности - softmax. Так как обучается сразу несколько моделей то получаются различные варианты как комбинировать признаки.

В работе [3] рассказана интересная теория о соотношении звука и смысла слов, например, почему слова «Ярмарка», «наряд», «иллюминация» ассоциируются у нас с чем-то положительным, а вот слова «келья», «кнут», «прыщ» с чем-то негативным. Понятно, что мы ассоциируем то, что означают эти слова с чем-то хорошим/плохим, а потому и слова кажутся нам эмоционально окрашенными. Язык многогранен и неясен, особенно если говорить о семантике слов. Было бы интересно смотреть на то, как часто встречаются слова с такой положительной/отрицательной оценкой и как это влияет на определение тональности.

В работе [4] проводится исследование совместной встречаемости слов в политических твитах. Авторы выявили, что позитивные и негативные слова часто вместе встречаются с нейтральными, и наоборот - позитивные не часто с другими позитивными (как и негативные не часто с другими негативными). А негативные с позитивными встречаются часто. Затем из дерева убирают малочастотные слова и все связи между словами, которые редко встречаются, чтобы убрать влияние нейтральных слов. В итоге авторы получают кластеры с темами за или против кандидата на выборах. Средняя тональность кластера совпадает с тональностью его темы. Кластеры за кандидата получились более позитивные, чем против. При этом в негативных кластерах есть позитивные слова.

В работе [5] авторы рассматривают проблему автоматического анализа тональности текста и предлагают методику, основанную на использовании машинного обучения. Они описывают процесс сбора и разметки данных, необходимых для построения модели. Далее, авторы предлагают алгоритмы и методы для обработки текста, включая токенизацию, лемматизацию и удаление стоп-слов. Затем, в статье описывается процесс построения модели с использованием различных алгоритмов машинного обучения, таких как SVM (Support Vector Machines) и ансамбли деревьев решений. Авторы также обсуждают особенности работы с оценочной лексикой в различных предметных областях и предлагают подходы к адаптации модели для конкретных сфер. Результаты экспериментов, проведенных авторами, демонстрируют эффективность предложенной модели в извлечении оценочной лексики и определении тональности текста в различных предметных областях. Они также проводят сравнение с другими существующими методами и анализируют преимущества и ограничения своего подхода.

Были рассмотрены различные подходы к анализу тональности, определению степени тональных слов и их вклада в текст, также в работах были рассмотрены подходы к составлению словарей тональных слове. Кроме рассмотренных работ, описывающих методы машинного обучения для анализа текстов, в работе [3] был рассмотрен иной подход с точки зрения лингвистики к определению тональности при помощи методов машинного обучения.

4. Теоретическая часть

4.1 Определение тональности при помощи методов машинного обучения

4.1.1 Логистическая регрессия

Логистическая регрессия является линейным методом классификации. Этот метод используется для прогнозирования вероятности некоторого события (тональности) по значениям множества признаков .

Чтобы понять основную идею данного метода, нужно рассмотреть его для бинарной классификации.

Пусть дана выборка - $(x_i, y_i), i = \overline{1, m}$ состоящая из m объектов (отзывов), каждый из которых описывается n признаками (токенами) $x_i \in \mathbb{R}^n$ и принадлежит одному из двух классов $y_i \in \{0, 1\}$.

Цель: построить функцию такую, чтобы прогнозируемый ответ $\hat{y}_i \in [0, 1]$ был как можно ближе к фактическому ответу y_i .
Как раз структуру такой функции задает логистическая регрессия:

$$\hat{y}_i = f(x_i, w, b) = \sigma(w^T \cdot x_i + b)$$

где $w \in \mathbb{R}^n, b \in \mathbb{R}-$ параметры модели, оцениваемые в ходе обучения, которые также называются просто коэффициентами.

То есть логистическая регрессия для объекта x_i получает предсказание в 2 шага:
1) преобразование из вектора в число ($z \in \mathbb{R}$):

$$z = \sum_{j=1}^{n} w_j \cdot x_j + b$$

2) теперь число z преобразуется в меру вероятности с помощью функции активации - сигмоида :

$$a = \sigma(z) = \frac{1}{1 + e^{-z}}$$

Для подбора двух неизвестных параметров w и b нужно воспользоваться стандартным, для машинного обучения, способом – путем минимизации функции потерь.

Введем функцию потерь. Для логистической регрессии потеря на объекте y_i, если мы предсказываем $\hat{y}_i$, определяется как:

$$L_i(\hat{y}_i) = -y_i log(\hat{y}_i) - (1 - y_i) log(1 - \hat{y}_i)$$

Когда $y_i = 1$, L_i для соответствующего предсказания $\hat{y}_i$ равна $-log(\hat{y}_i)$. И если предсказывается вероятность близкая к 1, то это тот результат, который и нужен, т.е. небольшая потеря. Если иначе - предсказывается вероятность близкая к 0, то потери будут большими, так как $-log(\hat{y}_i)$ значительно падает.

Теперь вспоминая про выборку из m объектов, можно определить функцию потерь на этой выборке:

$$L = \frac{1}{m} \sum_{i=1}^{m} L_i(\hat{y}_i) \to \min_{w,b}$$

То есть задача свелась к задаче минимизации функции потерь. Минимизация данной функции нужна для того, чтобы ошибка на обучающей выборке была минимальна.

Минимизация происходит с помощью градиентного спуска. Надо идти в направлении наискорейшего спуска, а это направление задаётся антиградиентом. Шаг будет следующим (для b аналогично):

$$w_{j+1} := w_j - \alpha \frac{\partial L}{\partial w}(w_j, b_j)$$

где α – скорость градиентного спуска, производная функции потерь для

одного объекта по параметру w: $\frac{\partial L_i}{\partial w} = \frac{\partial L_i}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w} = (a - y_i) \cdot x_i$ а по

параметру b: $\frac{\partial L_i}{\partial b} = (a - y_i)$, учитывая следующее:

$$L_i(w, b) = L_i(a(z(w, b))) = -y_i log(a) - (1 - y_i) log(1 - a)$$

$$\frac{\partial L_i}{\partial a} = -\frac{y_i}{a} + \frac{1 - y_i}{1 - a} = \frac{-y_i + a}{a(1 - a)} \quad \frac{\partial a}{\partial z} = a(1 - a) \quad \frac{\partial z}{\partial w} = x_i \in \mathbb{R}^n$$

4.1.2 Метод опорных векторов

Метод опорных векторов (SVM - Support Vector Machines) - это метод машинного обучения, который может использоваться для классификации и регрессии. Он был разработан для решения задач, в которых данные не могут быть линейно разделимыми. Основная идея метода заключается в поиске гиперплоскости в многомерном пространстве, которая разделяет два класса. Гиперплоскость выбирается таким образом, чтобы максимизировать расстояние между гиперплоскостью и ближайшими точками обучающей выборки, называемыми опорными векторами.

Ключевая формула, используемая в методе опорных векторов (SVM), связана с поиском гиперплоскости разделения классов. Для случая линейно разделимых данных, формула может быть представлена следующим образом: $w^T x + b = 0$.

Одним из главных преимуществ SVM является то, что он работает эффективно даже в случае большой размерности пространства признаков. Это возможно благодаря тому, что метод основан на работе с отдельными опорными векторами, а не со всеми точками обучающей выборки. Это позволяет SVM быстро и эффективно работать с большими объемами данных.

Существует несколько видов SVM, но наиболее распространенными являются линейный SVM и нелинейный SVM. Линейный SVM используется в случаях, когда данные линейно разделимы. Нелинейный SVM применяется, когда данные не могут быть линейно разделены. Для этого метод использует функцию ядра (kernel function), которая переводит данные в более высокую размерность и позволяет разделить классы гиперплоскостью в этом новом пространстве. В данной программе используется радиальное базисное функциональное ядро (Radial Basis Function Kernel, RBF Kernel): $K(x_i, x) = exp(-\gamma x_i - x^2)$, где γ - параметр ядра.

Применение метода опорных векторов в задаче анализа тональности заключается в обучении модели на наборе данных, состоящем из комментариев или текстовых сообщений с различными тональностями. Каждый текст представляется в виде вектора признаков, таких как количество вхождений слов с позитивной или негативной окраской. Затем SVM строит гиперплоскость, которая разделяет позитивные и негативные тексты. После этого, при поступлении нового текста, модель может предсказать его тональность на основе его вектора признаков и положения этого вектора относительно гиперплоскости.

В заключение, метод опорных векторов является мощным инструментом для классификации и регрессии, который может быть использован для решения различных задач, в том числе и для анализа тональности текстов. Он показывает хорошую производительность на больших объемах данных и может использоваться как для линейно разделимых, так и для нелинейно разделимых данных.

Одним из недостатков метода опорных векторов является то, что он требует большого количества ресурсов для обучения модели, особенно для нелинейных SVM. Кроме того, выбор оптимальных параметров для SVM может быть нетривиальной задачей и требовать дополнительных усилий.

В целом, метод опорных векторов является одним из наиболее эффективных и мощных методов машинного обучения для анализа тональности текстов. Он может быть использован в сочетании с другими методами, такими как логистическая регрессия и случайный лес, для повышения точности и качества анализа.

4.1.3 Наивный байесовский классификатор

Метод Байеса относится к вероятностным методам классификации. В его основе лежит теорема Байеса (пусть $c \in C$):

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

где $P(c|d)$ – вероятность того, что отзыв d соответствует классу c, $P(d|c)$ – вероятность найти отзыв d среди всех отзывов класса c, $P(c)$ — априорная вероятность найти отзыв класса c в корпусе отзывов; $P(d)$ — априорная вероятность отзыва d в корпусе отзывов.

Задача данного классификатора - найти более вероятностный класс. Для этого считаются вероятности всех классов и выбирается тот класс, который имеет наибольшую вероятность. Также $P(d)$ не зависит ни от какого класса из C, то есть это константа, которую можно игнорировать. Учитывая это, можно записать следующую формулу:

$$c_{max} = \underset{c \in C}{\operatorname{argmax}}[P(d|c)P(c)]$$

При представлении отзыва в виде вектора признаков ($d = (w_1, , w_n)$), делается “наивное” предположение о том, что все координаты независимы, то есть:

$$P(d|c) = P(w_1, w_2, ..., w_n|c) = P(w_1|c)P(w_2|c)...P(w_n|c) = \prod_{i=1}^{n} P(w_i|c)$$

Иначе говоря, опускается тот факт, что в тексте появление двух разных токенов часто взаимосвязано.

Тогда данное вычисление можно подставить в формулу для c_{max}:

$$c_{max} = \underset{c \in C}{\operatorname{argmax}} \left[P(c) \prod_{i=1}^{n} P(w_i|c) \right]$$

Теперь стоит поговорить о том, как посчитать вероятность $P(c)$ и $P(w_i|c)$. Их значение вычисляется на обучающей выборке. А вероятность определяется как:

$$P(c) = \frac{D_c}{D} \qquad\qquad P(w_i|c) = \frac{N_{ic}}{N_c}$$

где D_c – количество отзывов, содержащихся в классе c, D – корпус отзывов в обучающей выборке, N_{ic} – количество раз, сколько признак i встречается в отзывах класса c, N_c – общее количество признаков для класса c.

Формулу для оценки вероятности слова можно изменить - добавить сглаживание (smoothing) (эта оценка используется в полиномиальной наивной байесовской модели):

$$P(w_i|c) = \frac{N_{ic} + \alpha}{N_c + n\alpha}$$

где α - множитель сглаживания (если $\alpha = 1$ метод называется сглаживанием Лапласа, а если множитель $\alpha < 1$, то это сглаживание Лидстоуна), n – число признаков.

Данное сглаживание является устранением проблемы неизвестных слов – если в проверочном наборе встретилось слово, которое ранее не появлялось в тренировочном наборе данных, то оценка вероятности (без сглаживания) данного признака будет равна 0. Это значит, что отзыв не получится классифицировать с этим словом. А при использовании сглаживания, слова, которые не были при обучении, получают вероятность уже не равную 0.

Подставив описанные выше записи для вычисления вероятностей, получается окончательная формула, по которой работает метод Байеса:

$$c_{max} = \underset{c \in C}{\operatorname{argmax}} \left[\frac{D_c}{D} \prod_{i=1}^{n} \frac{N_{ic}+\alpha}{N_c+n\alpha} \right]$$

Для данного метода был рассмотрен подход полиномиальной моли, которая имеет полиномиальное распределение данных, но существуют и другие варианты наивного байесовского классификатора, использующие соответствующие меры. Например, модель Бернулли, которая реализует классификацию для данных, использующих многомерное распределение Бернулли. Отличие рассмотренного и Бернулли методов в том, что во второй – выборки должны быть представлены в виде векторов признаков с двоичным значением и при вычислении вероятности принадлежности d к c явно учитывается отсутствие признака i в классе c, а первый подход игнорирует это.

4.1.4 Метод случайного леса

Метод случайного леса (Random Forest) - это алгоритм машинного обучения, который может использоваться для решения задач классификации и регрессии. Он основан на использовании нескольких деревьев решений, которые работают вместе для принятия окончательного решения.

В задачах анализа тональности текста, метод случайного леса используется для определения тональности текстового отзыва. Для этого сначала нужно подготовить данные, то есть разделить набор текстов на две группы - положительные и отрицательные. Затем создается словарь слов, которые будут использоваться в качестве признаков для обучения алгоритма. В этом словаре должны быть только те слова, которые часто встречаются в положительных и отрицательных текстах.

Далее создается обучающая выборка, которая состоит из набора текстов, для которых уже известна тональность. Каждый текст представляется в виде набора признаков - в данном случае, это список слов из словаря. На основе этой выборки обучается модель методом случайного леса. Обучение происходит путем построения нескольких деревьев решений и использования метода бутстрэпа (выбор случайного подмножества данных с повторениями) для улучшения обобщающей способности модели.

Теперь стоит более подробно рассмотреть алгоритм создания обучающего дерева решений.

- Дерево начинается со своего корня. В нем изначально хранится весь набор данных (в данной задаче - это множество отзывов, каждый из которых задан числовым вектором размерности n). Для корня выбирается наиболее подходящий признак и наилучшее пограничное значение, задающее бинарное условие. Вследствие чего будет два потомка, в которые данные отправляются по следующим правилу: все

отзывы, для которых условие истинно, переходят в левый узел, а отзывы, для которых условие ложно, - в правый узел.

- Далее продолжается процесс разделения - последующие узлы рекурсивно разделяются на более мелкие подмножества в соответствии с пограничным значением выбранного признака. Условия разделения автоматически выбираются на каждом шаге с учетом того, какое условие наиболее подходящим способом разделяет текущий набор отзывов. Для того, чтобы измерить качество разбиения, существует несколько функций. Первая – основана на информационной энтропии, вторая --неопределенность Джини. Далее речь пойдет про вторую метрику. Данный показатель определяет, как часто случайно выбранный пример будет распознан неверно. Коэффициент Джини задается следующей формулой :

$$Gini(Q) = 1 - \sum_{i=1}^{n} p_i^2$$

- где p_i- результирующее множество, n - число классов в нем,Q- вероятность i-го класса. Коэффициент Джини равен 0, если все примеры набора относятся к одному классу. Значит, чем меньше коэффициент Джини, тем меньше вероятность того, что выбранный пример в множестве будет классифицирован неправильно.
- Следующий шаг - определение остановки алгоритма. Существует несколько способов (в том числе тех, которые помогают при переобучении):

    - Построение полного дерева – когда все листья дерева являются однородными, то есть когда каждый лист содержит примеры, принадлежащие одному и тому же классу.
    - Ограничение глубин дерева – когда заранее задается максимальное число разбиений в ветвях, по достижении которого построение дерева завершается.
    - Определение минимального числа примеров в узле – когда любой узел имеет число примеров меньше заданного, он дальше не может создавать узлы.

Так, в процессе классификации осуществляются переходы сверху вниз между внутренними узлами дерева решений на основе условий. Если алгоритм дошел до конечного узла, классификация считается законченной. Таким образом, путь от корня до конечного узла - конъюнктивное правило, а все дерево - группа правил дизъюнктивного выражения.

После обучения модель проверяется на наборе тестовых данных. Для этого выбираются некоторые тексты, для которых уже известна тональность, и пытаются определить ее с помощью обученной модели. Если точность работы модели высока, то ее можно использовать для определения тональности любого нового текста. Для этого текст нужно представить в виде списка признаков (слов из словаря), и передать его в модель для классификации.

Один из основных преимуществ метода случайного леса заключается в том, что он способен обрабатывать большие объемы данных и работать с текстами, содержащими множество слов. Также этот метод позволяет учитывать взаимодействия между признаками, что может быть полезно при работе с сложными текстами.

В целом, метод случайного леса является эффективным и точным способом определения тональности текста. Он может использоваться в различных областях, где необходимо анализировать большие объемы текста.

4.2 Инвертирование тональности отзыва

Повышение тональности осуществляется на отзывах, которые метод опорных векторов обозначил как “Негативные”. Модификация текста производится благодаря лексической замене слов в тексте на их антонимы с использованием библиотеки wordnet из nltk.

Наибольшее влияние на тональность отзывов оказывают отрицания перед глаголами и прилагательными, а также сами прилагательные. Для определения какие именно прилагательные из датасета являются негативно окрашенными, то есть требуют замены через подбор антонимов, будет использоваться библиотека SentiWordNet, а именно:

- senti_synsets - это метод в библиотеке SentiWordNet, который возвращает список senti_synset-объектов для заданного слова. Каждый объект представляет собой синтетический набор значений (synset) с эмоциональными значениями, выраженными в положительной, нейтральной и отрицательной оценке. Эти значения отражают относительную силу соответствующих эмоциональных оценок в контексте, в котором используется слово. Функция senti_synsets() полезна для определения тональности текста и контекстного определения субъективных значений слов в задачах обработки естественного языка, так как позволяет получить эмоциональную окраску каждого значения слова в контексте.
- neg_score представляет собой значение эмоциональной окраски (score) для отрицательной тональности слова в данном контексте. Чем выше neg_score, тем

сильнее отрицательная окраска слова в данном контексте. Атрибут neg_score используется для определения эмоциональной окраски текста или его части. Если для данного слова в заданном контексте значение neg_score выше, то можно предположить, что эта часть текста имеет более негативную тональность

Для поиска антонимов к найденным негативным словам используется библиотека WordNet, которая содержит лексическую базу данных английского языка. В ней каждое слово представлено в виде синсета, то есть группы синонимов, которые представляют одно и то же понятие или означают одинаковый объект. Каждый синсет содержит определение слова, примеры использования, а также указание на связанные слова и синсеты.

Для подбора антонимов в данной функции используется WordNet, который является лексической базой данных английского языка. Антонимы в WordNet представлены с помощью отношения "антонимии" между лексическими единицами (леммами), которые относятся к разным синсетам. Синсеты - это наборы лемм, которые относятся к одному и тому же понятию.

Кроме того, WordNet содержит информацию о связанных словах, таких как гиперонимы (слова, которые являются более общими понятиями), гипонимы (слова, которые являются более конкретными примерами) и многие другие. Эти связи могут быть использованы для более точной замены слова на его антоним, исходя из контекста предложения.

Автоматически учитывать контекст при замене слов на антонимы - это очень сложная задача, и для её решения требуются более продвинутые методы машинного обучения и обработки естественного языка, наиболее вероятным для этого кажется использование моделей сверточных или рекуррентных нейронных сетей.

5. Практическая часть

5.1 Подготовка данных для обучения

Изначальный набор данных содержит 413778 отзывов о мобильных телефонах с Amazon, каждый из которых включает в себя текст с впечатлением об устройстве и его оценкой по пятибалльной шкале.

Распределение оценок, интерпретированное графически изображено на Рис 1.

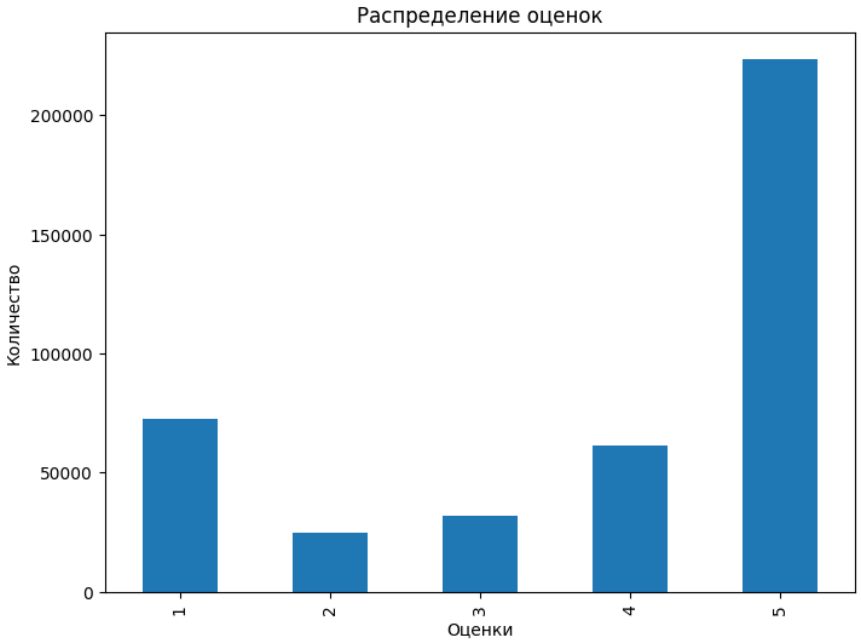

Рис. 1

Можно заметить, что датасет получается не сбалансированным, то есть могут возникнуть неточности при обучении модели, а значит необходимо провести балансировку датасета, сделав равным количество отрицательных и положительных отзывов. Также перед этим необходимо удалить отзывы, содержащие оценку без текста или же текст без оценки.

В рамках задачи, будем считать, что отзыв положителен, если за товар выставлена оценка «4» или «5», и отрицателен во всех остальных случаях. Обозначим флагом «1» положительные отзывы, а флагом «0» отрицательные.

После проведения этих этапов получаем измененный датасет, в котором оставлено 50000 отрицательных и 50000 положительных отзывов и на котором удобно проводить обучение модели. Датасет сбалансирован по количеству отзывов двух категорий.

Далее нужно создать набор данных в удобном формате, чтобы можно было работать с данными напрямую. Потом, как и в любой задаче NLP, необходимым процессом является преобразование текста в вид, который методы машинного обучения могут считывать. А также формирование данных в удобный для человека формат, чтобы можно было легко работать с ними.

Основные этапы подготовки данных для машинного обучения включают в себя:

- Обработка текста
    - Приведение к одному регистру
    - Удаление ненужной информации и раскрытие сокращений
    - Удаление стоп-слов
- Векторное представление данных

Для начала приведем все слова к маленькому регистру и удалим знаки препинания и эмоджи. Затем удалим возможное наличие ссылок в отзывах при помощи регулярного выражения. Также при помощи регулярных выражений удалим сокращения, допустимые в английском языке [X].

После этого проведем удаление стоп-слов (слова, которые часто встречаются в языке и не несут особой смысловой нагрузки) в английском языке, поиск которых осуществлен при помощи функции из библиотеки nltk - stopwords.words.

Затем проводим лемматизацию (лемматизация - это процесс приведения слова к его базовой форме (лемме), что позволяет уменьшить размерность пространства признаков и повысить точность модели) также при помощи функции из библиотеки nltk - WordNetLemmatizer. После этих действий получили датасет, с которым будет удобно работать, представление датасета изображено на Рис.2, представленном ниже.

	Reviews	Rating	label	pre_process
0	Cheap had to buy another one	1	0	cheap buy another one
1	Bought this phone a couple months ago. Yesterd...	1	0	bought phone couple month ago yesterday went a...
2	My phone not working nowApple say there is wir...	3	0	phone working nowapple say wire side cut isman...
3	Was not please the phone was not unlocked so I...	1	0	please phone unlocked send back made phone eve...
4	I sent the phone to Colombia, and They had to ...	3	0	sent phone colombia change battery one phone c...
...	...	...	...	...
99995	Ok watch. It's good for making or taking phone...	4	1	ok watch itis good making taking phone call dr...
99996	Perfect condition. Completely jail broke, popp...	5	1	perfect condition completely jail broke popped...
99997	excellent, very good	5	1	excellent good
99998	I already bought this same phone from here. It...	5	1	already bought phone itis fine quality
99999	Producto de buena calidad, se recomienda para ...	5	1	producto de buena calidad se recomienda para f...

100000 rows × 4 columns

Рис. 2

5.2 Подготовка к обучению модели

Перед подбором гиперпараметров и началом обучения модели необходимо совершить еще одно действие: указать процентное соотношение обучающей и тестовой выборок, а также преобразовать текстовые данные в векторные признаки при помощи TF-IDF.

TF-IDF (Term Frequency-Inverse Document Frequency) - это статистическая мера, используемая для оценки важности слова в отзыве, который является частью коллекции отзывов или корпуса. TF-IDF состоит из двух частей: TF (частота термина) и IDF (обратная частота документа(отзыва)).

Класс TfidfVectorizer автоматически выполняет преобразование текста в вектор признаков на основе метода TF-IDF, что понадобится в дальнейшем.

5.3 Подбор гиперпараметров

Подбор гиперпараметров в процессе обучения модели имеет большое значение, поскольку это позволяет найти оптимальный набор параметров для модели, что может улучшить ее качество и точность прогнозирования.

Гиперпараметры являются параметрами модели, которые задаются в процессе обучения и настраиваются вручную перед началом обучения. Эти параметры, такие как скорость

обучения, коэффициент регуляризации и т.д., определяют архитектуру модели и способ ее обучения.

Подбор оптимальных значений гиперпараметров является задачей оптимизации, которая может быть выполнена различными способами, такими как решетчатый поиск, случайный поиск, оптимизация на основе градиентного спуска и т.д. Эти методы позволяют определить оптимальные значения гиперпараметров, которые максимизируют качество модели и минимизируют ошибку.

В своем коде я использую функцию GridSearchCV из библиотеки sklearn для выполнения поиска по сетке (grid search) оптимальных гиперпараметров для логистической регрессии и метода опорных векторов. Для модели случайного леса гиперпараметры передаются в виде словаря params, в частности я задаю количество деревьев (200), максимальную глубину каждого дерева (15) и функцию, которая будет использоваться для измерения качества разделения на каждом узле (gini). В ходе экспериментов такие значения параметров показали наилучшие результаты. Байесовский классификатор не имеет гиперпараметров, которые необходимо настраивать на основе данных обучения. Это происходит потому, что в байесовском классификаторе распределения признаков задаются заранее, и они остаются неизменными во время обучения и классификации.

5.4 Сравнение результатов работы выбранных методов машинного обучения

В работе произведен анализ тональности при помощи четырех методов машинного обучения: логистической регрессии, метода опорных векторов, метода случайного леса и полиномиального байесовского классификатора.

Для оценки качества моделей использовались различные метрики: Accuracy_score, Precision_score, Recall_score и F1-score.

Accuracy_score (точность) - это метрика, которая показывает, как часто модель правильно предсказывает класс. Она вычисляется как отношение числа правильно классифицированных примеров к общему числу примеров в тестовой выборке.

Precision_score (точность) - это метрика, которая показывает, как часто модель правильно предсказывает положительный класс. Она вычисляется как отношение числа правильно предсказанных положительных примеров к общему числу примеров, которые модель предсказала как положительные.

Recall_score (полнота) - это метрика, которая показывает, как много положительных примеров модель нашла относительно общего числа положительных примеров в тестовой выборке. Она вычисляется как отношение числа правильно предсказанных положительных примеров к общему числу положительных примеров в тестовой выборке.

F1-score (F-мера) - это гармоническое среднее между precision и recall. Она является более устойчивой метрикой, чем Accuracy_score, особенно в случае несбалансированных классов.

Подробные вычисления метрик и отчеты о классификации можно увидеть в ноутбуке, приведем для каждой модели значения Accuracy_score, Precision_score, Recall_score, так как на сбалансированном датасете по этим метрикам можно достаточно хорошо оценить точность модели. Представим результаты в виде таблицы (смотрите таблицу 1)

Таблица 1

Логистическая регрессия	Accuracy_score: 0.893 Precision_score: 0.8970432516086992 Recall_score: 0.8864294913071474
Метод опорных векторов	Accuracy_score: 0.91604 Precision_score: 0.9148252310164725 Recall_score: 0.9163715389568577
Полиномиальный наивный байесовский классификатор	Accuracy_score: 0.8664 Precision_score: 0.8800836820083682 Recall_score: 0.8465067611075338
Метод случайного леса	Accuracy_score: 0.79112 Precision_score: 0.7525245441795232 Recall_score: 0.8637314874436575

Можно заметить, что наилучшее значение метрик имеет метод опорных векторов, это может быть связано с тем, что SVM пытается найти гиперплоскость, которая максимально разделяет два класса данных. Данные в задаче определения тональности линейно разделимы, а значит метод опорных векторов отлично подходит для ее решения.

Этот метод показал наилучший результат, а значит в дальнейшей задаче определения тональности введенного отзыва будем использовать этот метод.

5.5 Подготовка текста, введенного пользователем для определения тональности

Для начала приводим исходный, введенный пользователем отзыв о мобильных телефонах на английском языке в вид, который может быть обработан моделью, а именно произведем: удаление всех символов, кроме буквенных символов английского алфавита

- приведение текста к нижнему регистру
- разбиение текста на слова (токенизация)
- замену сокращений (если таковые есть) на полные формы слов
- приведение каждого слова к его начальной форме (лемматизация)
- удаление стоп-слов (часто встречающихся, но не несущих смысловой нагрузки слов) из текста

Далее произведем предсказание положительности/отрицательности тональности отзыва при помощи метода опорных векторов, обученного на тестовых данных датасета, описанного выше. На различных тестах предсказание получается корректным.

5.6 Изменение тональности отзыва, введенного пользователем

После определения тональности введенного отзыва производится изменение его тональности, а именно тональность негативного отзыва становится положительной и наоборот.

Для начала, составляются словари отрицательно и положительно окрашенных прилагательных из датасета отзывов, так как прилагательные привносят наибольшую смысловую окраску в предложение. Составление словарей осуществлялось с помощью функций find_negative_words и find_positive_words. Рассмотрим детальнее функцию find_negative_words: она сначала пытается получить синтетические наборы эмоциональных значений для этого слова с помощью функциии senti_synsets из SentiWordNet. Если наборы найдены, то функция вычисляет среднюю оценку отрицательности для всех синтетических наборов. Если эта оценка больше 0.5, то слово добавляется в список отрицательных слов - negative_words. Если для текущего слова синтетические наборы не найдены, функция ищет все синсеты для этого слова с помощью функции synsets из WordNet. Затем функция проходит по всем леммам (синонимам) в каждом из найденных синсетов и проверяет, есть ли у них антонимы. Если у леммы есть антоним и он соответствует словам "good" или "positive", то текущее слово также добавляется в список отрицательных слов. Аналогично осуществляется

работа функции find_positive_words, только там в случае отсутствия синтетических наборов слова проверяются на антомичность словам "bad" или "negative".

После составления словарей выполняется функция invert_tone принимает на вход текст, множество слов wordset и множество антонимов antonym_set. Она проходит по каждому слову в тексте и проверяет, является ли это слово отрицательным прилагательным. Если слово является отрицательным прилагательным, то функция ищет все его синсеты в WordNet и собирает все антонимы каждого синсета в множество antonyms. Затем функция проверяет, есть ли хотя бы один из антонимов в множестве antonym_set. Если есть, то функция заменяет это слово на найденный антоним в новом тексте new_text. Если нет, то функция заменяет это слово на первый антоним в множестве antonyms, если такой есть, в новом тексте new_text. Если ни один антоним не найден, функция просто добавляет это слово в новый текст. Функция также проверяет, следует ли пропустить следующее прилагательное (если перед этим стояло слово "not") и не заменять его антонимом.

На вход функции в качестве словарей wordset и antonym_set подаются положительные и отрицательные слова из датасета.

5.7 Эксперименты и выводы

Для удобства изображения полученных выводов в ходе экспериментов составим таблицу 2. В ней я буду указывать введенный пользователем отзыв, его тональность, измененный отзыв и тональность измененного отзыва.

Таблица 2

Исходный отзыв	Тональность исходного отзыва	Измененный отзыв	Тональность измененного отзыва	Оценка
Phone is absolutely useless and security police is not safe.	Отрицательная	Phone is absolutely useful and security police is safe .	Положительная	Корректно

Device is better, than anything I've seen in my life! Colours are beautiful and favorable.	Положительная	Device is worse , than anything I 've seen in my life ! Colours are ugly and unfavorable .	Отрицательная	Корректно
This bad device has the ugliest screen, that I has ever seen in my life!	Отрицательная	This goodness device has the beautiful screen , that I has ever seen in my life !	Положительная	Корректно
This phone is bad! I am absolutely unpleasant! The main thing is impractical screen.	Отрицательная	This phone is goodness ! I am absolutely pleasant ! The main thing is practical screen .	Положительная	Корректно
This phone is good ! I am absolutely pleasant ! The main thing is practical screen .	Положительная	This phone is bad ! I am absolutely unpleasant ! The main thing is practical screen .	Положительная	Некорректно, связано с тем, что слово practical не было найдено в словаре положительных слов из отзывов.
I am absolutely sad about buying this phone!	Отрицательная	I am absolutely glad about buying this phone .	Положительная	Корректно

Таким образом, можно заметить, что в большинстве случаев инвертировать тональность отзыва получается корректно (при проведенных экспериментах в 80 процентах примеров инвертирование тональности выполняется корректно), однако при замене прилагательных не всегда после изменения тональности отзыв звучит хорошо, так как подход к изменению тональности, описанный выше не учитывает контекст отзыва. Я попыталась учесть сферу, о который написаны отзывы (мобильные телефоны) проверкой на наличие антонимов в словаре положительных слов, но это далеко не всегда может выполняться

6. Заключение

Основные результаты выполненной работы:

1. Изучены методы машинного обучения, являющиеся наиболее эффективными в задаче анализа тональности: логистическая регрессия, метод опорных векторов, метод случайного леса и байесовский классификатор.
2. Адаптированы датасет для обучения моделей. В результате проведенных экспериментов выбран наилучший метод для определения тональности текста - метод опорных векторов.
3. Реализовано и протестировано экспериментальное инвертирование тональности введенного текста на основе словарного подхода.

В данной программе замена слов на антонимы осуществляется на основе словарного подхода, который не учитывает контекст отзыва. Антонимы подбираются из словарей положительных и отрицательных слов, составленных по датасету. Однако, такой подход имеет ограничения и может не гарантировать корректное инвертирование тональности отзыва с сохранением контекста.

Проблема заключается в том, что значение слова зависит от контекста, в котором оно используется. Простая замена слова на его антоним может привести к искажению смысла предложения или отзыва.

Для того, чтобы при модификации тональности текста учитывался контекст, однако для этого требуется использование нейронных сетей, а также языковой модели BERT. При помощи методов глубокого обучения можно точнее оценивать не просто положительность, отрицательность или нейтральность текста, но и менее сильные тональные окраски текста.

Планы предусматривают введение не только двух основных тональностей, положительной и отрицательной, но и включение третьей, нейтральной тональности, которая придаст тексту баланс и объективность.

Благодаря нейронным сетям и языковым моделям, вроде BERT, мы сможем точно определить контекст и анализировать семантику, чтобы инвертировать тональность отрицательных и положительных отзывов. При этом мы обеспечим сохранение смысловой нагрузки и подходящего контекста, чтобы не искажать исходный смысл текста.

Далее, приступим к добавлению тональности для нейтральных отзывов. Наша цель будет состоять в том, чтобы привнести определенную эмоциональную оттеночность в нейтральные тексты, делая их более интересными и привлекательными для читателей.

Кроме того, будем стремиться уменьшить силу тональности в положительных и отрицательных отзывах. Мы хотим создать более сбалансированные и нюансированные тексты, где сильная эмоциональная окраска будет заменена на более умеренную. Это поможет предотвратить чрезмерные перегибы в оценке и обеспечит более объективную информацию для читателей.

Для реализации модификации текста с изменением степени его тональности будут использоваться методы глубокого обучения и анализа текстовых данных. С помощью нейронных сетей, можно достичь высокой точности и эффективности, при этом уделив особое внимание сохранению контекста и смысла текста, чтобы изменения в тональности были сбалансированными и не искажали исходный смысл.

7. Список литературы

[1] Н. В. Лукашевич. Автоматический анализ тональности текстов: проблемы и методы. URL: http://intsysjournal.org/pdfs/26-1/1-5-Lukashevich.pdf (дата обращения: 16.12.22)

[2] Steinberger J., Lenkova P., Ebrahim M., Ehrmann M., Hurriyetoglu A., Kabadjov M., Steinberger R., Tanev H., Zavarella V., Vazquez S. Creating sentiment dictionaries via triangulation. URL: https://aclanthology.org/W11-1704/ (дата обращения: 19.12.22)

[3] А.П. Журавлев. Звук и смысл. 1991.

[4] Tan∗ S., Na J., Positional Attention-based Frame Identification with BERT A Deep Learning Approach to Target Disambiguation and Semantic Frame Selection URL: https://arxiv.org/pdf/1910.14549.pdf (дата обращения: 23.12.22)

[5] Н. В. Лукашевич, И. И. Четвёркин, Построение модели для извлечения оценочной лексики в различных предметных областях. URL: https://www.mathnet.ru/links/fb9ca4aac2ee2f120573985cafd23b9d/mais298.pdf (дата обращения: 30.01.23)

[6] Yassine Al Amrania, Mohamed Lazaarb, Kamal Eddine ELlkadiri. Random Forest and Support Vector Machine based Hybrid Approach to Sentiment Analysis URL: https://www.sciencedirect.com/science/article/pii/S1877050918301625?via%3Dihb (дата обращения: 09.04.23)

[7] Daniel Jurafsky, James H.. Martin Speech and Language Processing. Chapter 4. URL: https://web.stanford.edu/~jurafsky/slp3/4.pdf (дата обращения: 21.03.23)

[8] Федотов Станислав, Синицин Филипп. Учебник по машинному обучению от Школы Анализа Данных. URL: https://academy.yandex.ru/handbook/ml (дата обращения: 16.04.23)

[9] George B. Aliman, Tanya Faye S. Nivera, Jensine Charmille A. Olazo, Daisy Jane P. Ramos, Chris Danielle B. Sanchez, Timothy M. Amado, Nilo M. Arago, Romeo L. Jorda Jr., Glenn C. Virrey, Ira C. Valenzuela. Sentiment Analysis using Logistic Regression. URL: https://www.dlsu.edu.ph/wp-content/uploads/pdf/research/journals/jciea/vol-7-1/4aliman.pdf (дата обращения: 05 04.23)

[10] Raksha Sharma, Mohit Gupta, Astha Agarwal, Pushpak Bhattacharyya. Adjective Intensity and Sentiment Analysis. URL: https://aclanthology.org/D15-1300.pdf (дата обращения: 16.04)

[11] Stefano Baccianella, Andrea Esuli, Fabrizio Sebastiani. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. URL: https://aclanthology.org/L10-1531/ (дата обращения: 20.04)