Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет
«Высшая школа экономики»

Факультет математики

Удалова Маргарита Сергеевна

Интерпретируемые методы машинного обучения для автоматического оценивания задач.

Выпускная квалификационная работа — бакалаврская работа
по направлению подготовки 01.03.01 — Математика,
образовательная программа «Совместный бакалавриат НИУ ВШЭ и ЦПМ»

Рецензент:
ведущий исследователь
лаборатории искусственного интеллекта
ПАО "Сбербанк"
Цвигун Аким Олегович

Научный руководитель:
Старший преподаватель
Соколов Евгений Андреевич

Москва 2021

Аннотация

Многоклассовая классификация — это задача классификации данных более чем на два класса. Существует очень много реальных задач, в которых встречается многоклассовая классификация. Одна из них — задача проверки рукописных ответов учеников, которые оцениваются по N-бальной шкале. Многоклассовая классификация предполагает, что каждому образцу присваивается только одна метка.

Данная работа является продолжением моей курсовой работы "Машинное обучение для анализа образовательных материалов" за 3-й год обучения. В дополнение к ранее изученным методам мы пытаемся улучшить восприятие алгоритмов и повысить доверие людей к используемым моделям путем интерпретации результатов.

Мы изучаем подмножество алгоритмов, таких как логистическая регрессия и дерево решений. Они обычно используются в качестве интерпретируемых моделей для задач классификации. А также изучаем методы интерпретации, опирающиеся на отдельные объекты из выборки, взятой из набора данных конкурса «The Hewlett Foundation: Automated Essay Scoring» на онлайн-платформе Kaggle.

1 Введение

Основной задачей системы образования является подготовка человека к жизни в быстроменяющемся мире, где новая информация поступает изо дня в день. И очень важна точность в анализе результатов успеваимости обучающихся и их успехов в различных областях, так как непосредственно исследование достижений учащихся является одним из основных инструментов для индивидуализации процесса обучения. Кроме того анализ решает множество задач, к которым можно отнести кластеризацию обучающихся и педагогов для нахождения зависимостей и связей между ними, оценку качества образования, оценку риска получения обучающимися неудовлетворительной оценки при выполнении определенной работы и др.

В настоящее время машинное обучение является одним из наиболее перспективных направлений, которое применяется в области информационных технологий. Применение машинного обучения в образовании в настоящее время очень широко интересно исследователям и ученым, чтобы решить проблему высокой стоимости и медленного оборота ручного подсчета тысяч письменных ответов. Кроме того многие школы постепенно исключают письменные ответы в пользу вопросов с множественным выбором ответов, которые в меньшей степени способны оценить критические рассуждения и навыки письма учащихся.

Точность очень важна при анализе результатов успеваемости студентов. А модели машинного обучения могут быть отлажены и проверены только тогда, когда они могут быть интерпретированы. Поэтому возможность корректно объяснять входные характеристики модели крайне важна. Хорошая интерпретируемость дает возможность для поиска полезных связей в данных, чтобы использовать их для поиска новых улучшений в работе модели. Кроме того хорошая интерпретация данных корректирует понимание моделируемого процесса, что необходимо для людей, использующих алгоритм.

Бывают случаи, когда применяются более простые модели засчет их легкой интерпретируемости, несмотря на то, что их качество может оказаться ниже, чем у сложных моделей. На сегодняший день, работа с большими данными становится доступной, поэтому использование сложных моделей выходит на передний план, тем самым создавая вектор для баланса между сложностью модели и ее интерпретируемости.

2 Постановка задачи

В данной работе мы решаем задачу многоклассовой классификации с непересекающимися классами. *Классификация* — один из разделов машинного обучения, посвященный решению следующей задачи. Имеется множество объектов (ситуаций), разделённых некоторым образом на классы. Задано конечное множество объектов, для которых известно, к каким классам они относятся. Это множество называется *обучающей выборкой*. Классовая принадлежность остальных объектов не известна. Требуется построить алгоритм, способный классифицировать произвольный объект из исходного множества. Рассмотрим формальную постановку задачи:

Пусть X — множество описаний объектов, $Y = \{y_1, \ldots, y_k\}$ — конечное множество меток классов. Существует такое неизвестное отображение $y^* : X \to Y$, что значения известны лишь на конечном множестве обучающей выборки $X^\ell =$

$\{(x_1, y_\ell), \cdots, (x_\ell, y_\ell)\}$. Задача заключается в построении такой модели $a(x) : X \to Y$, которая будет способна хорошо классифицировать произвольный объект $x \in X$.

Признак — это такое отображение $f : X \to D_f$, где D_f — множество допустимых значений признака. Если заданы признаки $f_1, \cdots, f_n$, то вектор $x = (f_1(x), \cdots, f_n(x))$ называется признаковым описанием объекта $x \in X$. Признаковые описания допустимо отождествлять с самими объектами. При этом множество $D_{f_1} \times \cdots \times D_{f_n}$ называют *признаковым пространством.*

3 Методы обработки текста

3.1 Токенизация и пунктуация

Токенизация − это процесс разделения некоторых участков или всего текста на отдельные его части − токены. Токенами могут служить абзацы, предложения, слова или символы. Также в данный метод преобразования текста входит избавление от знаков препинания, и понижение регистра, т.к. например, “отлично“ и “ОТЛИЧНО“ для нас одинаковые слова имеющие лишь разный регистр букв, но чтобы алгоритм воспринимал их как одно и тоже слово необходимо одинаковое написание.
Рассмотрим небольшой пример токенизации предложения:
«В 2000-х гг. я впервые прочитала книгу о Шерлоке Холмсе, которая поразила меня своим тиражом в 500,000 экземпляров в месяц!»
После процесса токенизации хотим увидеть следующее:
[“В”, “2000-х”, “гг.”, “я”, “впервые”, “прочитала”, “книгу”, “о”, “Шерлоке”, “Холмсе”, “,”, “которая”, “поразила”, “меня”, “своим”, “тиражом”, “в”, “500,000”, “экземпляров”, “в”, “месяц”, “!”].
Однако метод простого разбиения по пробелам не подойдет, потому что как правило знаки пунктуации выделяются вместе со словами: “Холмсе,”, “месяц!”. Можно убрать все не буквенно-цифровые символы из текста, но тогда некоторые значимые для нас слова и выражения потеряют тот смысл, который они несли в тексте: в нашем предложении “2000-х” станет “2000х”, а “500,000” − “500000”. Кроме того в реальных текстах как правило много шума, всречающегося в виде лишних знаков пунктуации, ссылок, html-разметки и просто опечаток.
Разрешить подобные трудности для токенизации поможет использование морфологических правил, основанных на регулярных выражениях.

3.2 Стемминг и лемматизация

Рассмотрим теперь применение стемминга и/или лемматизации к тексту. [1, ч. 2.2.4]
Стемминг − это один из методов, в процессе которого удаляются суффиксы и окончания у каждого слова, чтобы оставшаяся часть, называемая стемом, была одинаковой для всех грамматических форм слова. Данная процедура может значительно улучшить качество модели, однако есть ряд неопределенностей, которые могут возникнуть при применении стемминга. Например, слова “просмотрел” и “смотровая” перейдут в разные стемы “просмотр” и “смотр”.
Лемматизация − процесс, при котором выполняется поиск каждой лексемы в словаре и происходит преобразование каждого слова в его каноническую (словарную) форму, которая называется леммой. Также как и в стемминге тут есть некоторые

неопределенности. Существуют такие слова, которые при одном и том же написании имеют разный смысл в контексте и соответственно образованы от разных начальных форм слов. Например, слово “гладь” в одном контексте “гладь кошку!” имеет значение глагола в повелительном наклонении, образованного от глагола “гладить”, а в другом контексте “гладь озера” — существительное, обозначающее ровную поверхность. Данную неопределенность можно решить с помощью вероятностной модели, которая будет учитывать контекст, т.е слова, стоящие рядом, и выявлять наиболее подходящий тег для извлечения существующей леммы слова.
Процесс лемматизации основан на поиске эталона и помогает справляться с необычными случаями и верно обрабатывать лексемы, являющиеся разными частями речи. Например, слова “копоть”, “копия” и “копать” будут переведены в разные леммы, тогда как стемминг превратил бы их в один и тот же стем — “коп”.

3.3 TF-IDF

Для задач анализа текстов и информационного поиска можно использовать статическую меру TF-IDF, которая определяет ключевые слова и словосочетания для документа d из некоторой коллекции документов (текстов) D.
Для начала рассмотрим TF-меру (Term Frequency), которая рассчитывает важность каждого слова из документа d, т.е. относительную частоту встречаемости в нем:

$$\mathrm{TF}(t,d) = \frac{C(t)}{\sum_{k \in d} C(k)},$$

где $C(t)$ − число вхождений слова t в документ d.
Теперь рассчитаем IDF-меру ((inverse document frequency) − обратную частоту встречаемости для каждого слова из документа d:

$$\mathrm{IDF}(t,D) = \log\left(\frac{|D|}{|\{d_i \in D | t \in d_i\}|}\right),$$

Логарифмирование необходимо для уменьшения масштаба весов, так как правило в коллекции достаточно много документов.
Наконец, для каждого слова t из документа d рассчитаем вес:

$$TF - IDF(t,d,D) = \mathrm{TF}(t,d) \times \mathrm{IDF}(t,D) \tag{1}$$

Интерпретировать данную формулу можно так: чем чаще слово встречается в документах корпуса, тем ближе к нулю будет значение меры (1), т.е. не будет ключевым для конкретного документа.

3.4 Векторное представление документов

Если снова возвратиться к задачам анализа текстов, то в них применяется метод представления документов в виде векторов.
Пусть D − коллекция текстов, а V − это словарь, содержащий все рассматриваемые слова. Каждому документу $d \in D$ ставят в соответствие вектор

$$\vec{d_j} = \left(w_{1j}, \ldots, w_{|V|j}\right),$$

где w_{ij} — вес $i-$ого слова в документе d_j, например это могут быть веса TF-IDF (для TfidfVectorizer) или просто частота появления слова в документе (для CountVectorizer). Представив документы в таком виде, теперь можно находить похожие по смыслу тексты, сравнивая их векторы.

4 Модели

4.1 Логистическая регрессия

Одним из распространенных способов решения задачи классификации является метод логистической регрессии. В базовой постановке решается задача бинарной классификации.

Пусть объекты описываются n числовыми признаками $f_j : X \to \mathbb{R}, j = 1, \ldots, n$. Тогда пространство признаковых описаний объектов есть $X = \mathbb{R}^n$. Пусть $Y-$ конечное множество номеров (имён, меток) классов.

Положим $Y = \{-1, +1\}$. В логистической регрессии строится линейный алгоритм классификации $a : X \to Y$ вида:

$$a(x, w) = \text{sign}\left(\sum_{j=1}^{n} w_j f_j(x) - w_0\right) = \text{sign}\langle x, w\rangle,$$

где w_j — вес j-ого признака, w_0 — порог принятия решения, $w = (w_0, w_1, \ldots, w_n)$ — вектор весов, а $\langle x, w\rangle$ — скалярное произведение признакового описания объекта на вектор весов. Предполагается, что искусственно введён «константный» нулевой признак: $f_0(x) = -1$.

Задача обучения линейного классификатора заключается в том, чтобы по выборке X^m настроить вектор весов w. В логистической регрессии для этого решается задача минимизации эмпирического риска с функцией потерь данного вида:

$$Q(w) = \sum_{i=1}^{m} \ln\left(1 + \exp\left(-y_i \langle x_i, w\rangle\right)\right) \to \min_w \tag{2}$$

Теперь после нахождения весов можно вычислять классификацию для любого объекта x. Кроме того логистическая регрессия дает возможность прогнозирования не просто отнесения объекта к определенному классу, но и вероятности его принадлежности классам.

Для многоклассовой классификации есть несколько способов сведения к ней бинарного случая. Разберем наиболее часто применяемый из них — метод *all-versus-all*:

Обучим C_K^2 классификаторов $a_{ij}(x)$, $i, j = 1, \ldots, K$, $i \neq j$.

Классификатор $a_{ij}(x)$ будем настраивать по подвыборке $X_{ij} \subset X$, содержащей только объекты классов i и j:

$$X_{ij} = \{(x_n, y_n) \in X [y_n = i] = 1 \text{ или } [y_n = j] = 1\}.$$

Соответственно, классификатор $a_{ij}(x)$ будет выдавать для любого объекта либо класс i, либо класс j.

Чтобы классифицировать новый объект, подадим его на вход каждого из построенных бинарных классификаторов. Каждый из них проголосует за своей класс; в качестве ответа выберем тот класс, за который наберется больше всего голосов:

$$a(x) =_{k\in\{1,\dots,K\}} \sum_{i=1}^{K}\sum_{j\neq i}[a_{ij}(x) = k].$$

4.2 Многоклассовая логистическая регрессия

В логистической регрессии для двух классов мы строили линейную модель $b(x) = \langle w, x\rangle + w_0$, а затем переводили её прогноз в вероятность с помощью сигмоидной функции $\sigma(z) = \dfrac{1}{1+\exp(-z)}$. Для многоклассовой задачи построим K линейных моделей $b_k(x) = \langle w_k, x\rangle + w_{0k}$, каждая из которых даёт оценку принадлежности объекта одному из классов. Преобразуем вектор оценок $(b_1(x), \dots, b_K(x))$ в вероятности, воспользовавшись оператором $\mathrm{SoftMax}(z_1, \dots, z_K)$, который производит «нормировку» вектора:

$$\mathrm{SoftMax}(z_1, \dots, z_K) = \left(\frac{\exp(z_1)}{\sum_{k=1}^{K}\exp(z_k)}, \dots, \frac{\exp(z_K)}{\sum_{k=1}^{K}\exp(z_k)}\right).$$

В этом случае вероятность k-го класса будет выражаться как

$$P(y = kx, w) = \frac{\exp\left(\langle w_k, x\rangle + w_{0k}\right)}{\sum_{j=1}^{K}\exp\left(\langle w_j, x\rangle + w_{0j}\right)}.$$

Обучим эти веса с помощью метода максимального правдоподобия — так же, как и в случае с двухклассовой логистической регрессией:

$$\sum_{i=1}^{\ell}\log P(y = y_i x_i, w) \to \max_{w_1,\dots,w_K}.$$

4.3 Решающие деревья

Модели логистической регрессии терпят неудачу в ситуациях, когда связь между признаками и результатом нелинейна или когда признаки взаимодействуют друг с другом.[2, p.165-192] Давайте определим бинарное дерево решений с помощью жадного алгоритма:

Дерево решений — это классификатор, выраженный в виде рекурсивного разбиения признакового пространства.

Рассмотрим бинарное дерево, в котором:

- каждой внутренней вершине v приписана функция (или предикат) $\beta_v :\to \{0, 1\}$;
- каждой листовой вершине v приписан прогноз $c_v \in Y$ (в случае с классификацией листу также может быть приписан вектор вероятностей).

Рассмотрим теперь алгоритм $a(x)$, который стартует из корневой вершины v_0 и вычисляет значение функции β_{v_0}. Если оно равно нулю, то алгоритм переходит в левую дочернюю вершину, иначе в правую, вычисляет значение предиката в новой вершине

и делает переход или влево, или вправо. Процесс продолжается, пока не будет достигнута листовая вершина; алгоритм возвращает тот класс, который приписан этой вершине. Такой алгоритм называется *бинарным решающим деревом.*

На практике в большинстве случаев используются одномерные предикаты β_v, которые сравнивают значение одного из признаков с порогом:

$$\beta_v(x; j, t) = [x_j < t].$$

Существуют и многомерные предикаты, например:

- линейные $\beta_v(x) = [\langle w, x \rangle < t]$;
- метрические $\beta_v(x) = [\rho(x, x_v) < t]$, где точка x_v является одним из объектов выборки любой точкой признакового пространства.

Решающее дерево $a(x)$ разбивает всё признаковое пространство на некоторое количество непересекающихся подмножеств $\{J_1, \ldots, J_n\}$, и в каждом подмножестве J_j выдаёт константный прогноз w_j. Значит, соответствующий алгоритм можно записать аналитически:

$$a(x) = \sum_{j=1}^{n} w_j [x \in J_j].$$

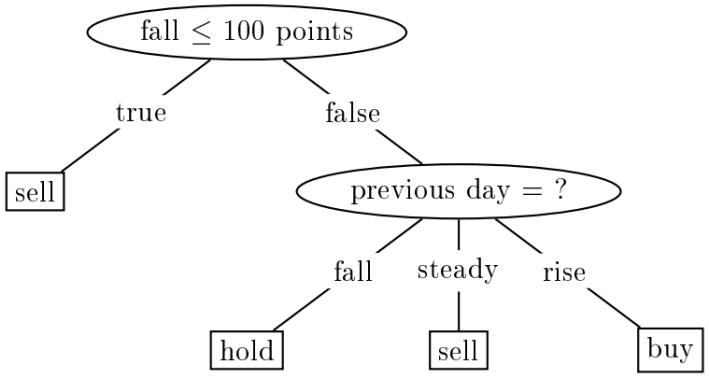


Рис. 1. Decision tree with artificial data

5 Интерпретируемые методы

Рассмотренные выше модели довольно просты в интерпретации взаимосвязи между признаками и целевой переменной. Мы оцениваем изменение целевой переменной, смотря на изменение конкретного признака на единицу.

У модели логистической регрессии целевая переменная принимает значения от 0 до 1. Взвешенная сумма $\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$ переводится в вероятность с помощью сигмоидной функции, поэтому веса не влияют на вероятность линейно:

$$\log\left(\frac{P(y=1)}{1-P(y=1)}\right) = \log\left(\frac{P(y=1)}{P(y=0)}\right) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

Чтобы оценить, как изменится прогноз, когда один из признаков x_j изменится на 1 единицу, мы можем сначала применить функцию $exp()$ к обеим сторонам уравнения:

$$\frac{P(y=1)}{1-P(y=1)} = \text{odds} = \exp\left(\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p\right).$$

Затем мы сравниваем, что произойдет, когда мы увеличим значение признака на единицу. Для этого посмотрим на соотношение двух прогнозов:

$$\frac{odds_{x_j+1}}{odds} = \frac{\exp\left(\beta_0 + \beta_1 x_1 + \ldots + \beta_j\left(x_j+1\right) + \ldots + \beta_p x_p\right)}{\exp\left(\beta_0 + \beta_1 x_1 + \ldots + \beta_j x_j + \ldots + \beta_p x_p\right)}.$$

Затем применим свойство:

$$\frac{\exp(a)}{\exp(b)} = \exp(a-b)$$

Тогда,

$$\frac{odds_{x_j+1}}{odds} = \exp\left(\beta_j\left(x_j+1\right) - \beta_j x_j\right) = \exp\left(\beta_j\right).$$

Получаем, что увеличение признака x_j на одну единицу изменяет отношение $\frac{odds_{x_j+1}}{odds}$ в $exp(\beta_j)$ раз.

Модель дерева решений в интерпретации очень проста. У нас есть корневая вершина и мы переходим от края к следующей вершине смотря на значения предиката, по которому происходит разбиение и т.д. Ребро говорит нам, на какие подмножества мы смотрим. В конце мы достигаем листовой вершины, которая предсказывает результат. Все ребра соединены союзом "И".

Мы можем оценить важность признака, если пройдем через все разбиения, где использовался этот признак, и измерим, насколько он уменьшил дисперсию или индекс Джини по сравнению с родительским узлом. Индекс Джини говорит нам, насколько "нечист" узел, например, если все классы имеют одинаковую частоту, узел "нечист" , если присутствует только один класс, он максимально чист. Дисперсия и индекс Джини минимизируются, когда точки данных в узлах имеют очень похожие значения для y.

6 Model-Agnostic Methods

В качестве альтернативы использования только интерпретируемых моделей существуют методы интерпретации, которые не зависят от используемой модели машинного обучения. Большим преимуществом таких методов является их гибкость. Для решения той или иной задачи мы можем использовать любую модель машинного обучения, которая нам нравится, а методы интерпретации могут быть применены к любой модели. Все, что основано на интерпретации модели машинного обучения, например графический или пользовательский интерфейс, также становится независимым от базовой модели машинного обучения. В данной работе мы рассмотрим некоторые основные методы.

6.1 Local interpretable model-agnostic explanations (LIME)

Идея данного метода заключается в том, что мы обучаем модель черного ящика, из которой для дальнейшей интерпретации нам необходимы лишь ее прогнозы для интересующего нас объекта. Наша цель — понять, почему модель машинного обучения сделала определенный прогноз. LIME проверяет, что происходит с прогнозами, когда мы создаем различные вариации данных в модель машинного обучения, изменяя признаковое описание объекта. LIME генерирует новый набор данных и соответствующие им прогнозы модели черного ящика. На этом новом наборе данных LIME обучает интерпретируемую модель, а также определяет вес отклонения от признакового описания исходного объекта. Интерпретируемая модель может быть абсолютно любой, например та же логистическая регрессия или дерево решений. Изученная модель должна быть хорошим приближением к предсказаниям модели машинного обучения локально, но не обязательно хорошим глобальным приближением. Такого рода точность также называется локальной верностью.

Математически локальные суррогатные модели с ограничением интерпретируемости могут быть выражены следующим образом:

$$\text{explanation}(x) = \arg\min_{g \in G} L\left(f, g, \pi_x\right) + \Omega(g),$$

где explanation(x) — это такая модель g (например модель логистической регрессии), которая минимизирует функцию потерь L (например ROC-AUC), которая оценивает насколько близко объяснение к предсказанию исходной модели f (например, модель xgboost), в то время как сложность модели $\Omega(g)$ остается низкой. G — это семейство возможных объяснений. Мера близости π_x определяет, насколько велика окрестность вокруг объекта x, которую мы рассматриваем для объяснения.

Рассмотрим на примере текстовых данных работу LIME.

Выбирая исходный текст из набора данных, новые тексты создаются путем случайного удаления слов. Текст представлен в виде двоичного вектора, где каждое значение — это значение двоичного признака для каждого слова. Признак равен 1, если соответствующее слово включено в текст, и 0, если оно было удалено.

Рассмотрим алгоритм на очень простых примерах классификации спама:

	Текст	Класс
267	PSY is a good guy	0
173	For Christmas Song visit my channel! ;)	1

Следующим шагом создаем другие варианты набора данных, например, измененные варианты одного из текстов:

	For	Christmas	Song	visit	my	channel!	;)	prob	weight
2	1	0	1	1	0	0	1	0.17	0.57
3	0	1	1	1	1	0	1	0.17	0.71
4	1	0	0	1	1	1	1	0.99	0.71
5	1	0	1	1	1	1	1	0.99	0.86
6	0	1	1	1	0	0	1	0.17	0.57

Каждый столбец соответствует одному слову из предложения. Каждая строка является вариацией исходного текста, 1 означает, что слово входит в данный вариант, а 0 означает, что слово было удалено. В столбце "prob"показана прогнозируемая вероятность спама для каждого из вариантов предложения. Столбец "вес"показывает близость изменения к исходному предложению, рассчитанному как 1 минус доля слов, которые были удалены, например, если было удалено одно из семи слов, близость будет 1 - 1/7 = 0,86.

6.2 SHapley Additive exPlanations (SHAP)

Данный метод объяснения предсказаний предполагает, что каждое значение признака для объекта является своего рода "игроком" в командной игре, где предсказание алгоритма является "выигрышем". Shapley Values — метод из теории коалиционных игр, который говорит нам, как справедливо распределить "выигрыш" между всеми признаками, участвовавшими в игре. На самом деле "Игра" — это задача прогнозирования для одного экземпляра набора данных. "Выигрыш" — это прогноз, полученный моделью для этого экземпляра за вычетом среднего прогноза для всех экземпляров. "Игроки" — это значения характеристик экземпляра, которые сотрудничают, чтобы спрогнозировать определенное значение.

6.2.1 Shapley Values

Shapley Values для значения конкретной характеристики объекта — это средний предельный вклад этого значения признака, взвешенный и суммированный по всем возможным комбинациям значений признаков. Интерпретация Shapley Value для значения признака j такова: значение j-го признака внесло вклад ϕ_j в прогноз для конкретного объекта по сравнению со средним прогнозом для набора данных. Определим значение данной характеристики через $value\ function$ (val) — функцию выплат значений признаков в S:

$$\phi_j(val) = \sum_{S \subseteq \{x_1,\ldots,x_p\} \setminus \{x_j\}} \frac{|S|!(p-|S|-1)!}{p!} \left(\mathrm{val}\left(S \cup \{x_j\}\right) - \mathrm{val}(S)\right),$$

где S — подмножество признаков, используемых в модели, x — вектор значений признака описываемого экземпляра, p — количество объектов, $val_x(S)$ — это прогноз для значений признака в наборе S, которые изолированы по сравнению с объектами, которые не включены в набор S:

$$val_x(S) = \int \hat{f}\left(x_1, \ldots, x_p\right) d\mathbb{P}_{x \notin S} - E_X(\hat{f}(X)).$$

Фактически мы выполняем несколько интеграций для каждого признака, который не содержится в S. Например, модель машинного обучения работает с 4 признаками x_1, x_2, x_3 и x_4, и мы оцениваем прогноз для подмножества S, состоящего из значений признаков x_1 и x_3:

$$val_x(S) = \mathrm{val}_x\left(\{x_1, x_3\}\right) = \int_{\mathbb{R}} \int_{\mathbb{R}} \hat{f}\left(x_1, X_2, x_3, X_4\right) d\mathbb{P}_{X_2 X_4} - E_X(\hat{f}(X)).$$

6.2.2 SHAP method

Цель SHAP состоит в том, чтобы объяснить предсказание на определенном объекте x путем вычисления вклада каждого признака в предсказание. В этом методе вычисляются Shapley Values, которые говорят нам, как справедливо распределить полученное предсказание между признаками, учавствовавшими в нем. В качестве значения признака может быть как индивидуальное значение, так и группа значений. Например, для объяснения изображения пиксели могут быть сгруппированы в суперпиксели, а предсказание распределено между ними. Одним из нововведений, которые предлагает SHAP, является то, что объяснение Shapley Value представлено как метод атрибуции аддитивных признаков, линейная модель. Такое представление объединяет LIME и Shapley Values. SHAP определяет объяснение следующим образом:

$$g\left(z'\right) = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j,$$

где g — модель объяснения , $z' \in \{0,1\}^M$ вектор коалиции, M максимальный размер коалиции и $\phi_j \in \mathbb{R}$ — the Shapley values для признака j. В векторе коалиции запись 1 означает, что соответствующее значение признака "присутствует" а 0 — что оно "отсутствует".

7 Практика

7.1 Описание задачи

Для получения пракических навыков применения методов интерпретации моделей машинного обучения в обработке ответов учеников и их оценивания было принятно участие в соревновании «The Hewlett Foundation: Short Answer Scoring» по оцениванию ответов учеников, которое проходило на Kaggle − онлайн-платформе для проведения конкурсов по машинному обучению.

Мы будем учиться применять методы для объяснения работы используемых алгоритмов, чтобы улучшить восприятие того, как они работают и повысить доверие людей к используемым моделям путем интерпретации результатов.
Обучающая выборка состоит из 17207 объектов (тексты ответов в формате ASCII) и 5 колонок (id − идентификатор ответа, EssaySet − идентификатор вопроса, на который дается ответ (всего их 10), EssayText — ответы на вопросы, Score1 и Score2 − оценки двух независимых проверяющих, одну из которых возьмем в качестве целевой переменной). Тестовая выборка состоит из 5732 объектов и 3 признаков (id − идентификатор ответа, EssaySet − идентификатор вопроса, на который дается ответ, EssayText — ответы на вопросы). Метрика качества — Area Under ROC Curve, AUC-ROC:

7.2 План экспериментов

7.2.1 Предобработка текстов

В качестве первого шага для очистки данных удаляем все неалфавитные символы, за исключением цифр, так как в ответах на вопросы числа могут играть ключевую

роль в оценивании. При замене удаленные символы заменяем пробелами. Далее все заглавные буквы заменяем строчными буквами, убираем лишние пробелы и стоп-слова. Стоп-слова − это слова, которые широко распространены в употреблении и не несут никакой смысловой нагрузки в тексте. Например, «a», «this», «that», «and», «so», «on» и т.д.

После очистки текста применяем стемминг, а потом векторизуем каждый текст с помощью tf-idf в формат, который может использовать наша модель для обучения.

7.2.2 Обучение базовых моделей

В качестве базовых моделей были взяты логистическая регрессия и решающее дерево. Каждую модель обучаем на наборе данных текстового формата, перебираем параметры с помощью *GridSearchCV*, мощного инструмента для автоматического определения наилучшего набора параметров для моделей машинного обучения.

Качество работы алгоритма на тестовой выборке измеряем с помощью интегральной метрики качества семейства — *площадь под ROC-кривой* (Area Under ROC Curve, AUC-ROC). Чтобы изобразить ROC-кривую в двумерном пространстве, возьмем в качестве одной из координат долю неверно принятых объектов (False Positive Rate, FPR), а другой — долю верно принятых объектов (True Positive Rate, TPR):

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}};$$
$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}.$$

где матрица ошибок выглядит так

	$y = 1$	$y = 0$
$\hat{y} = 1$	True Positive (TP)	False Positive (FP)
$\hat{y} = 0$	False Negative (FN)	True Negative (TN)

Каждый возможный выбор порога t соответствует точке в этом пространстве. Всего различных порогов имеется $\ell+1$. Максимальный порог $t_{\max} = \max_i b(x_i)$ даст классификатор с TPR = 0, FPR = 0. Минимальный порог $t_{\min} = \min_i b(x_i) -$ даст TPR = 1 и FPR = 1. ROC-кривая — это кривая с концами в точках $(0, 0)$ и $(1, 1)$, которая последовательно соединяет точки, соответствующие порогам $b(x_{(1)}) -, b(x_{(1)}), b(x_{(2)}), \ldots, b(x_{(\ell)})$. Площадь под данной кривой называется AUC-ROC, и принимает значения от 0 до 1.

7.2.3 Интерпретация результатов

Для более детального понимания и оценивания полученных результатов мы используем библиотеки *lime* и *shap*. Модель черного ящика — это xgboost классификатор, обученный на матрице данных с подбором оптимальных гиперпараметров. Каждый текст — это один документ (= одна строка в матрице), а каждый столбец — количество вхождений данного слова.

7.3 Эксперименты и результаты

Вычисления проводились на компьютере MacBook Pro 13. Использовался процессор 1,4 GHz 4-ядерный процессор Intel Core i5, 16 Гб оперативной памяти, операционная система macOs Big Sur. Эксперименты были реализованы на языке программирования Python 3.7.10 с использованием библиотек scikit-learn, pandas, numpy, nltk, pymorphy2, xgboost, lime.

Обучающая выборка была случайным образом поделена на две части в отношении 70:30 (обозначим их для удобства, как X_{train} и X_{test} соответственно). На X_{train} проводилось обучение, а на X_{test} тестирование моделей.

С предобработанными данными я обучила логистическую регрессию и решающее дерево с параметрами {max_depth: 90, max_features: 'auto', min_samples leaf: 8} и получила следующие результаты:

Качество каждой из моделей распределились следующим образом:

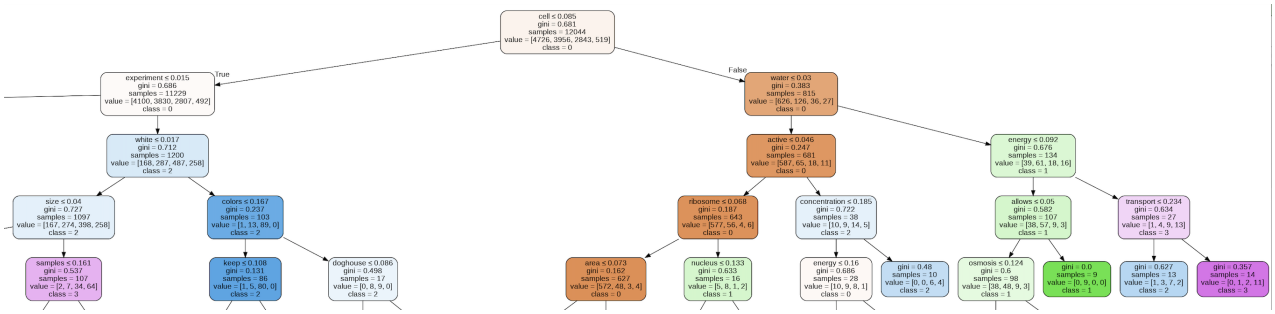
Модель	ROC-AUC
Логистическая регрессия	0.8697
Решающее дерево	0.7859
Xgboost	0.8871

Веса признаков в линейной модели в случае, если признаки отмасштабированы, характеризуют степень их влияния на значение целевой переменной. В задаче классификации текстов, кроме того, признаки являются хорошо интерпретируемыми, поскольку каждый из них соответствует конкретному слову. Изучим влияние конкретных слов на значение целевой переменной. Так словарь достаточно большой, то выведем топ-10 наиболее весомых признаков (слов) для этой модели:

body	1.32
rna	1.31
cells	1.26
mass	1.23
do	1.19
hypothesis	1.17
dna	1.17
wall	1.13
limestone	1.12
prophase	1.09

В нашем наборе данных вопросы, на которые дают ответ ученики, затрагивают различные учебные сферы. Поэтому замечаем, что модель к более важным признакам относит слова, характерные для узкой области науки, например, "rna "dna "limestone"и тд.

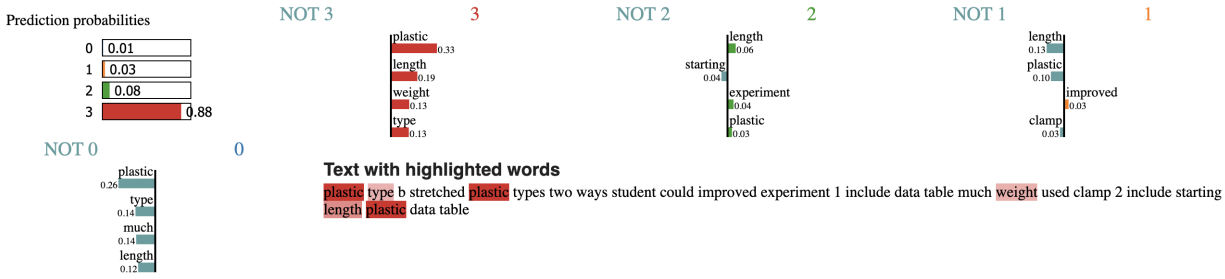
Теперь посмотрим на результат обучения решающего дерева на наших данных. С помощью встроенных методов визуализировали бинарное дерево, с помощью которого интерпретировать результат проще всего:



Рассмотрим только некоторое поддерево, в которое попала корневая вершина. В самом первом разбиении оказался признак 'cell', по которому объекты далее распределяются на два подмножества. В правую дочернюю вершину попадают объекты, у которых значение 'cell' $>$ 0,085, а в левую дочернюю вершину все объекты, у которых 'cell' $\leq$ 0,085. Кроме того в вершине графа при каждом разбиении указывается индекс Джини, количество объектов, попавших в текущую вершину и преимущественный класс.

Далее переходим к применению LIME на отдельных примерах. Во время экспериментов была применена функция LimeTextExplainer для генерации локальных объяснений для прогнозов. В качестве параметров функции требуется индекс ответа для объяснения, прогнозируемая оценка, созданная на основе модели черного ящика, и количество признаков, используемые для объяснения. Было рассмотрено восемь ответов учеников, принадлежащих к разным темам вопросов (см. таблицу 1).

Рассмотрим более подробно объяснение для текста "The plastic type B stretched the most out of all the other plastic types.Two ways the student could have improved his experiment was,1) to include in his data table how much weight he used on the clamp.2) To include what the starting length of the plastic was in his data table". С помощью визуализации мы можем получить такие данные:



В этом документе слово «plastic» имеет наивысший положительный вклад для оценки 3.

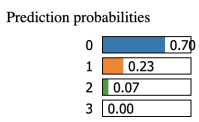
Наша модель предсказывает, что этот ответ должен быть оценен как наивысший балл 3 с вероятностью 88%. Если мы удалим слово «plastic» из документа, мы ожидаем, что модель будет предсказывать оценку 3 с вероятностью 88% - 33% = 55%.

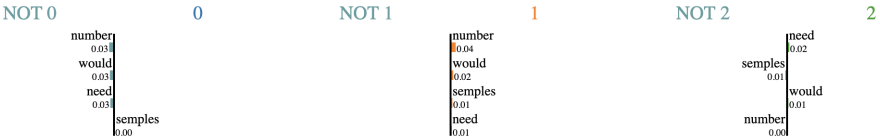
С другой стороны, слово «plastic» имеет отрицательный вклад для выставления оценок 1 или 0 и наша модель определила, что для выставления этих оценок слово «length» и «improved» соответсвенно имеют небольшой положительный вклад.

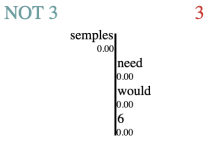
Аналогичный процесс интерпретации остальных примеров можно сделать, основываясь на визуализациях ниже:

Таблица 1. Примеры ответов.

	Текст	Оценка	Тема вопроса
123	You would need what samples there, what you do after number 6.	1	1
712	Additional information needed to replicate this experiment would be to give the samples and how much of the samples were used in the procedure. Also the amount of vinegar and the size of the container used should be given.	3	1
2805	Based on the students data, I can conclude that plastic B has the most stretchability.	2	2
2048	The plastic type B stretched the most out of all the other plastic types. Two ways the student could have improved his experiment was, 1) to include in his data table how much weight he used on the clamp. 2) To include what the starting length of the plastic was in his data table.	3	2
11359	Rose is a very posistive individual. The way that she keeps up a happy appearance when she is talking to her little sister proves this statement. 'He had to go. The job in Los Angeles paid three times what he was making here.	2	7
10359	Rose is caring. We know this because in paragraph 4-6, Rose is very concerned about the well-being of her younger sister. She asks her what's wrong and if she is feeling okay.	2	7
6291	All of the articles have something to do with invasive animals. The python, panda, koala. They can't live here, yet some people still try to make it work. But sometimes thats not always a good thing.	0	4
4872	In this article, the word "invasive"means animals that have taken over area, unchecked by natural predators, that are major threats to biodiversity. For example, the article says that the pythons in Florida are imperiling five endangered species. This could lead to an unbalance in the enviroment. Invasive animals could also stretch their habitat to a larger area.	2	4

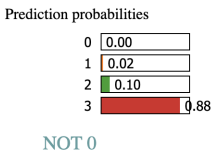


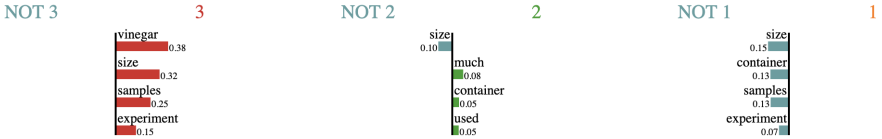


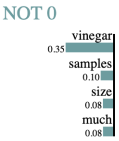


Text with highlighted words

would need semples number 6

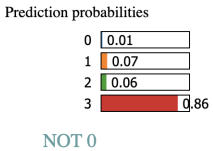


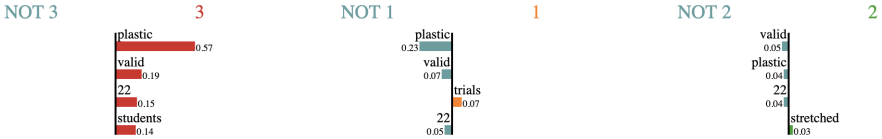


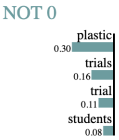


Text with highlighted words

additional information needed replicate experiment would give samples much samples used procedure amount vinegar size container used given

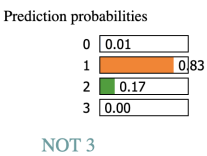


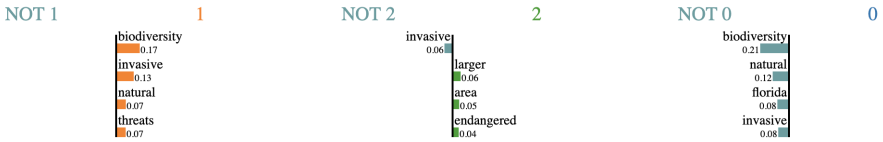


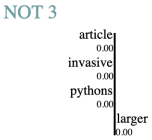


Text with highlighted words

based students data conclude plastic b stretchability data shows trials t1 t2 plastic b stretched 22 23 ^p b students could done another trial make data valid could let weights hang five minutes

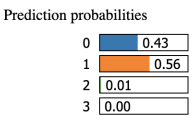


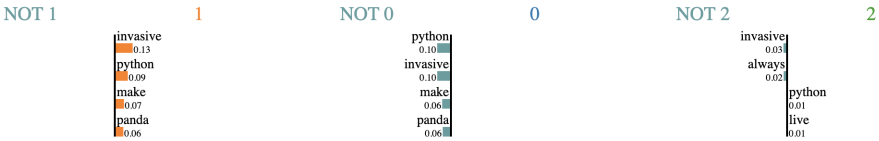


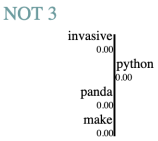


Text with highlighted words

article word invasive means animals taken area unchecked natural predators major threats biodiversity example article says pythons florida imperiling five endangered species could lead unbalance enviroment invasive animals could stretch habitat larger area

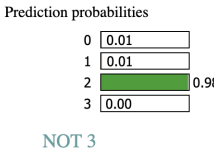


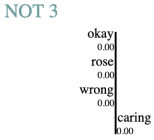




Text with highlighted words

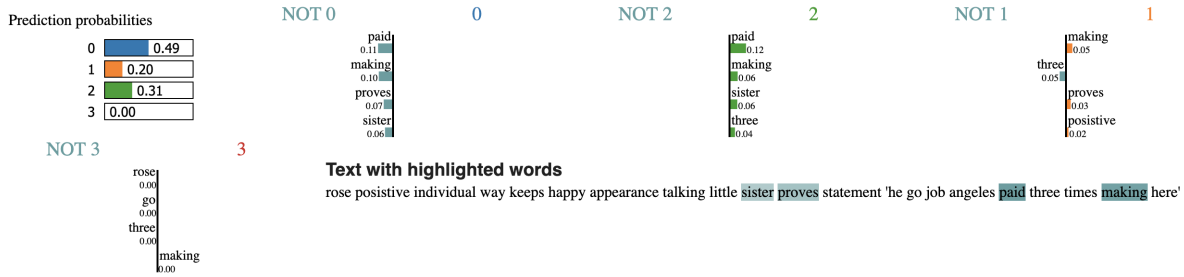
articles something invasive animals python panda koala can't live yet people still try make work sometimes thats always good thing





Text with highlighted words

rose caring know paragraph 4-6 rose concerned well-being younger sister asks what's wrong feeling okay

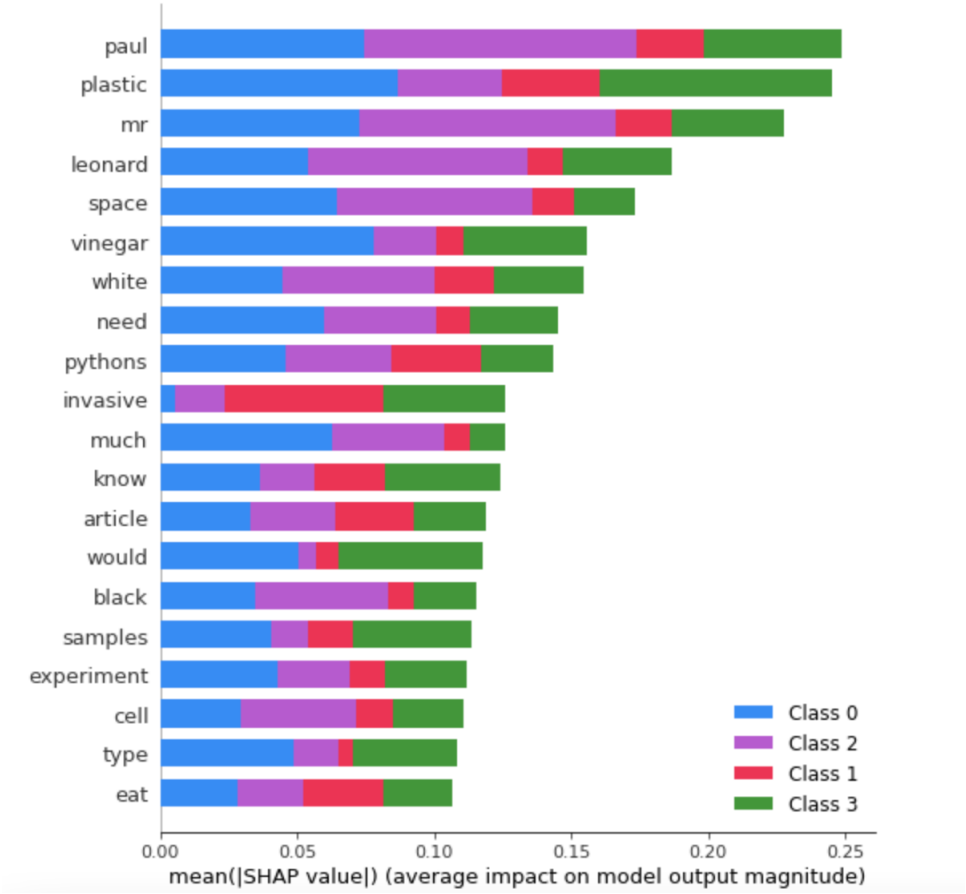


Для применения метода SHAP я обучила многоклассовую логистическую регрессию с 4 классами. После чего с помощью встроенных методов библиотеки shap вычислила Shapley values. Данные значения можно использовать для интерпретации как отдельных прогнозов, так и влиянии каждого признака на прогноз модели в целом. Если мы запустим SHAP для каждого экземпляра, мы получим матрицу Shapley values. Эта матрица имеет одну строку на каждый экземпляр данных и один столбец на каждый признак. Мы можем интерпретировать всю модель, анализируя Shapley values в этой матрице.

Идея важности функции SHAP проста: важны те признаки, которые имеют большое абсолютное Shapley values. Поскольку нам нужна глобальная важность, мы суммируем абсолютные значения Шепли для каждой характеристики по всем данным:

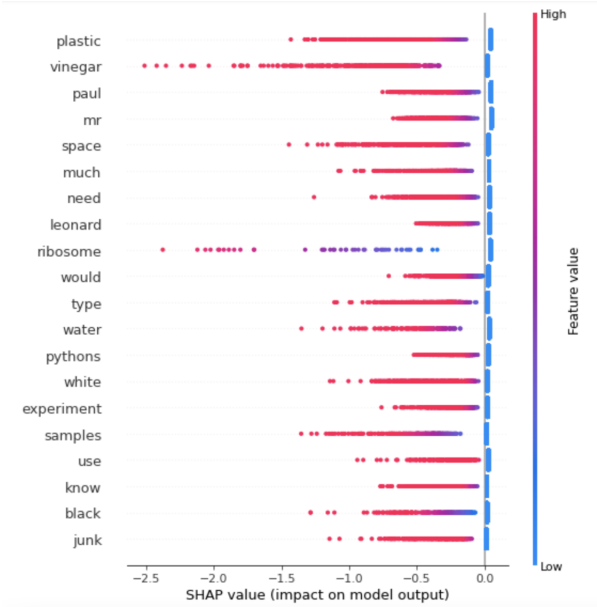
$$I_j = \sum_{i=1}^{n} \left| \phi_j^{(i)} \right|$$

Затем мы сортируем значения по убыванию важности и строим их график. На следующем рисунке показана оценка важности методом SHAP для прогнозирования оценки ученика по написанному мини-эссе:

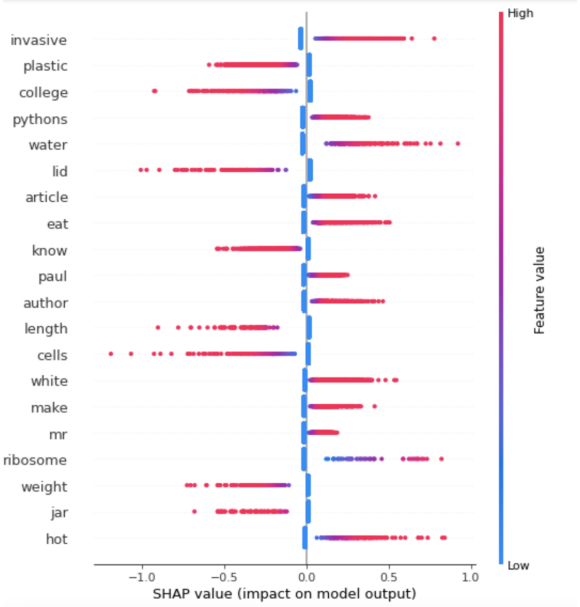


Слова "paul" и "plastic" , были наиболее важными характеристиками, изменившими прогнозируемую абсолютную вероятность на 2,5 процента по оси x. Также мы можем заметить диапазон влияния конкретных признаков на предсказание отдельного класса. Например, "paul" "mr" "leonard" являеются более важными словами для предсказания класса 2 и меньше всего оказывают влияние для прогноза класса 1. А "plastic" практически одинаковый вклад делает как в класс 0, так и класс 3. И это вполне объяснимо, так как оценивание работ должно проверять наличие определенных понятий, входящих в ответ ученика. Следовательно те слова, которые влияют только на один класс — для высокой оценки обязательно должны использоваться в ответе, нежели те, которые влияют на предсказания несколько противоположных классов.

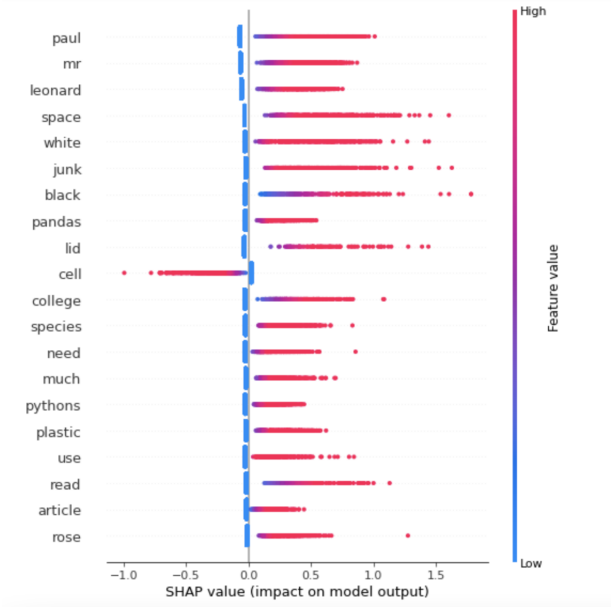
Чтобы получить общее представление о том, какие слова наиболее важны для модели, мы можем построить значения SHAP для каждого признака для каждого текста. График ниже сочетает в себе важность слов с Shapley values. Каждая точка на графике представляет собой Shapley value для признака и объекта. Положение по оси Y определяется значением признака, а по оси X — Shapley Values. Цвет представляет силу влияния признака от низкого до высокого. Перекрывающиеся точки колеблются в направлении оси Y, поэтому мы получаем представление о распределении значений Шепли для каждого объекта. Слова упорядочены по степени важности. Тем самым на графике, построенном для класса 0, мы можем заметить, что все слова с высоким значением SHAP влияют отрицательно на прогноз. Сильнее всего влияют слова "vinegar" и "ribisome". Подобные графики были также построены и для остальных трех классов.



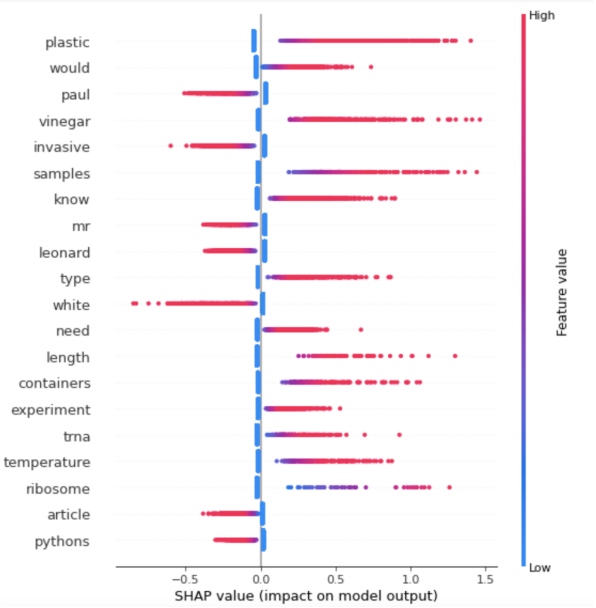
class 0



class 1



class 2



class 3

8 Заключение

В рамках поставленных задач были проведены разбор и анализ методов интерпретации предсказанных результатов модели, которые проводились на наборе данных с объектами в текстовом формате, размером около 50 слов, которые для начала были отчищены и предобработаны с помощью регулярных выражений и стемминга.

Были обучены такие базовые модели, как логистическая регрессия и решающие деревья, которые показали достаточно высокое качество. Эти модели сами являются хорошо интепретируемыми, поддтверждением чему были выведены данные о важности слов в моделе логистической регрессии, а также построенное бинарное дерево. Кроме того на данных обучался xgboost классификатор, показавший самое лучшее качество — 0,88. Именно его мы использовали в качестве модели, предсказывающей вероятность оценивания конкретной работы ученика, которая необходима для ин-

терпретации с помощью метода LIME. Анализ конкретных экземпляров с помощью данного метода дали возможность заметить некоторые важные признаки (слова) в тексте, по которым объект относился к определенному классу засчет создания разных вариаций исходного текста.

Использование различные методов интерпретации как для простых, так и для сложных моделей, дает возможность прощупывать алгоритм предсказаний как в глобальном масштабе всего набора данных, так и в окрестности отдельного объекта. Корректное объяснение входных характеристик модели крайне важна, тк это позволяет смелее работать с более сложными моделями и корректировать понимание моделируемого процесса, что необходимо для людей, использующих алгоритм.

Из всех рассмотренных методов преимущественное использование Model-Agnostic Methods заключается в удобстве применения независимо от того, какая модель используется для экспериментов. SHAP объединяет значения LIME и Shapley value. Это очень полезно для лучшего понимания обоих методов. Это также помогает объединить область интерпретируемого машинного обучения. В отличие от LIME SHAP имеет прочную теоретическую основу. Также в данном методе прогноз справедливо распределен между значениями признаков. Мы получаем контрастные интерпретации, которые сравнивают прогноз со средним прогнозом.

Список литературы

[1] Christoph Molnar. Interpretable Machine Learning A Guide for Making Black Box Models Explainable. 2021-04-08

[2] Rokach, Lior & Maimon, Oded. (2005). Decision Trees. 10.1007/0-387-25465-X_9.

[3] Alain, Guillaume, et al. "Understanding intermediate layers using linear classifier probes."arXiv preprint arXiv:1610.01644 (2018).

[4] Wachter, Sandra, Brent Mittelstadt, and Chris Russell. "Counterfactual explanations without opening the black box: Automated decisions and the GDPR."(2017)

[5] Dandl, Susanne, Christoph Molnar, Martin Binder, Bernd Bischl. "Multi-Objective Counterfactual Explanations". In: Bäck T. et al. (eds) Parallel Problem Solving from Nature – PPSN XVI. PPSN 2020. Lecture Notes in Computer Science, vol 12269. Springer, Cham (2020)

[6] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze. An Introduction to Information Retrieval. // *Cambridge University Press, 2009.*

[7] Danijel Kučak. Machine Learning in Education − a Survey of Current Research Trends

[8] Воронцов К.В. Математические методы обучения по прецедентам. // *МФТИ (2004)*

[9] Бенгфорт Бенджамин. Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка. // *Питер. 2019. Стр. 101−104*